USER’S MANUAL

MSC.Dytran

Version 4.7

MSC SOFTWARE
SIMULATING REALITY

Corporate Headquarters
MSC.Software Corporation
815 Colorado Boulevard
Los Angeles, CA 90041-1777 U.S.A.
Tel: (323) 258-9111 or (800) 336-4858
FAX: (323) 259-3838

Headquarters, European Operations
MacNeal-Schwendler GmbH
Innsbrucker Ring 15
Postfach 80 12 40
81612 München, GERMANY
Tel: (89) 431 9870
FAX: (89) 436 1716

Headquarters, Far East Operations
MSC Japan Ltd.
Entsuji-Gadelius Building
2-39, Akasaka 5-chome
Minato-ku, Tokyo 107-0052, JAPAN
Tel: (03) 3505-0266
FAX: (03) 3505-0914

DISCLAIMER

The concepts, methods, and examples presented in this text are for educational purposes only and are not intended to be exhaustive or to apply to any particular engineering problem or design. MSC.Software Corporation assumes no liability or responsibility to any person or company for direct or indirect damages resulting from the use of any information contained herein.

DT∗V4.7∗Z∗Z∗Z∗DC-USR

Preface

“MSC provides quality engineering software and related support services
for the long term.”

MSC.Software Corporation (MSC) has provided sophisticated computer-aided engineering (CAE) software to its clients since 1963. MSC’s products cover a wide range of engineering disciplines including structural analysis, heat transfer, crash dynamics, electromagnetic field analysis, and graphics pre- and postprocessing. MSC’s products run on approximately 30 different workstations, mainframes, and supercomputers.

MSC’s business activities are truly international with almost one-half of its revenues coming from outside North America. To support its clients, MSC maintains a network of company owned regional and international offices. MSC.Dytran is developed and maintained by staff located in MSC’s offices in Gouda, The Netherlands.

MSC’s main product, MSC.Nastran, is the world’s most comprehensive and widely used finite element structural program. The original development of NASTRAN began in 1966 under the sponsorship of the National Aeronautics and Space Administration. MSC was involved from the start, and has developed and marketed an enhanced proprietary version of NASTRAN called MSC.Nastran since 1972. After hundreds of man-years of development, this program is now capable of solving very large (10^6 DOFs) structural problems in a wide range of applications.

Building on the success of MSC.Nastran, MSC has produced a range of compatible products that cover most aspects of engineering analysis:

Structural and Heat Transfer Analysis

MSC.Nastran Finite element program for the analysis of structures. It includes static, dynamic, linear, nonlinear, aeroelasticity, and heat transfer capabilities.

Explicit Transient Dynamics

MSC.Dytran Three-dimensional analysis of transient fluid-structure interaction and the extreme deformation of materials using coupled Euler-Lagrange techniques.

Graphics Pre- and Postprocessors

MSC.Patran Interactive graphics pre- and postprocessor for MSC’s analysis products.

September 1999

Notes to the Reader

The MSC.Dytran User's Manual, Version 4.7 has undergone a significant revision from the previous MSC.Dytran User's Manual, Version 4.0. All new functionality in MSC.Dytran since Version 4.0 is included in this User's Manual.

One of the most eye-catching changes you will see is the company and product name changes appearing in this User's Manual. You were accustomed to the names The MacNeal-Schwendler Corporation and MSC/DYTRAN, but you will now see the new names MSC.Software Corporation and MSC.Dytran.

Some of the features of this manual listed by chapter include:

Chapter 1, Introduction. The introductory material gives an overview of the features of MSC.Dytran. It introduces the reader in the techniques available in the fluid-structure analyses and the explicit solution techniques.

Chapter 2, Modeling. Includes all theory associated with the capabilities implemented in the code. This chapter gives a simple theoretical introduction of the subject to provide modeling guidelines. Each section in this chapter begins with the essentials required to get started and is followed by a discussion and usage guidelines where applicable. Although the chapter is written in a rather tutorial fashion, it can continue to serve as a reference as the user's proficiency increases.

Chapter 3, Running the Analysis. The chapter gives an introduction about the actual usage of MSC.Dytran. It provides some detail about applicability of material models, boundary conditions and loads. The definition of contact, coupling and ALE surfaces is discussed. An overview of all output data including the element and grid point variables that can be requested for output is given here. Finally, the use of user defined FORTRAN user subroutines is discussed and some examples are included.

Chapter 4, Input Data. Covers all File Management System, Executive Control statements, Case Control commands, Bulk Data entries, and parameters for explicit transient analyses. A short description of each is included as well as cross-references and guidelines describing their use. Advanced readers or experienced MSC.Dytran users may find it possible to start directly with this chapter. It is less tutorial than either Chapters 2 or 3 and is probably one of the most useful reference chapters.

Chapter 5, Diagnostic Messages. Explains the way the diagnostic messages are defined and how they are to be interpreted.

Chapter 6, References. Lists literature references for more detailed information about certain items.

Appendix A, Using XDYTRAN. Describes how the OSF/Motif-based user interface XDYTRAN can be used to start-up and control MSC.Dytran jobs, or to create customized versions of MSC.Dytran.

Appendix B, Using XDEXTR. Describes how the OSF/Motif-based user interface XDEXTR can be used to translate information stored in MSC.Dytran archives and time history files to various import file formats.

Appendix C, MSC.Dytran and Parallel Processing. Describes how to use MSC.Dytran in a parallel computing environment.

Appendix D, Using ATB. Describes how the included occupant modeling code ATB can be used.

Appendix E, Using MADYMO. Describes how the coupling with the occupant modeling code MADYMO is performed.

Appendix F, Example Input Data. Describes the translation of a physical problem into an MSC.Dytran input data.

Appendix G, Using USA. Describes how the interface to the Underwater Shock Analysis program USA can be used.

New for Version 4.7

The Version 4.7 release of MSC.Dytran offers several new features and functionalities since the release of MSC.Dytran Version 4.5 on UNIX and MSC.Dytran Version 4.6 on Microsoft Windows NT 4.0. These new features and functionalities enhance the capabilities in areas such as structural crashworthiness and occupant safety, underwater explosions and gasdynamics.

1. MSC.Dytran on UNIX Workstations and Microsoft Windows NT 4.0 is Year 2000 Compliant

   To ensure the viability of MSC.Dytran Version 4.7 operational integrity, security, and authorization systems on UNIX workstations and Microsoft Windows NT 4.0 computer systems, as they relate to Year 2000 compliance, several tests were conducted.

   The test results indicated that:

   - The license for MSC.Dytran is valid to operate over a supported date range that extends from before January 1, 2000 and encompasses a time period reasonably suitable for the intended use of the product. This includes the critical moment of February 29, 2000. The license authorization is applicable to both nodelock and network licensing.
   - MSC.Dytran does not perform date calculations.
   - MSC.Dytran only obtains a date from the computer OS and transfers it to print and results files.
   - Date stamping data, although not four digits, will not lead to ambiguity in interpretation of data.

2. FLEXlm Licensing

   MSC.Dytran Version 4.5 introduced the FLEXlm license manager as the license system for network and nodelock licensing.

   Existing licensing problems on some UNIX platforms have been solved, and with Version 4.7 campus licensing is also supported.

   On Microsoft Windows NT 4.0 platforms, campus licensing is supported.

   Please refer to the *MSC.Dytran Installation and Operations Guide*, Version 4.7 for detailed instructions on installing MSC.Dytran Version 4.7 and the FLEXlm licensing system.

3. Initial Metric Method for Airbag Analysis

   The Initial Metric Method is typically useful for airbag modeling. When using out-of-plane folding technique, the membrane elements can deform quite significantly. The final shape of the deformed bag can be negatively influenced. In order to overcome this problem, MSC.Dytran Version 4.7 offers a way to initialize strains inside elements such that the final shape is preserved. It is called the Initial Metric Method. Elements can be initialized smaller than the original state, but also can be initialized larger.

- For elements that are initialized smaller, stresses will only start to build up after the original state has been reached.
- Elements that are larger will have a positive Initial Metric Method strain. When growing larger, their Young’s modulus is assumed to be twice as large during one time step. When shrinking there will be no stresses applied until the original state is reached.
- The Initial Metric Method can also be applied when scaling the model of the airbag, such that the model will fit inside the inflator housing.

4. Coupling to MADYMO

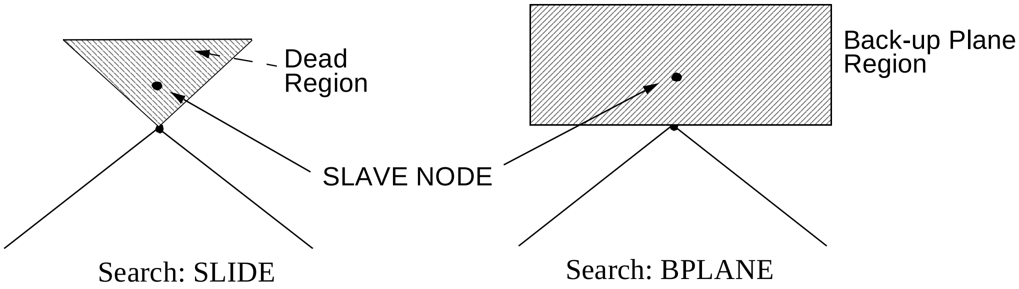
   - The coupling to MADYMO has been upgraded to MADYMO V5.3.
   - The coupling between MSC.Dytran and MADYMO in MSC.Dytran Version 4.7 uses a direct coupling instead of the pvm message passing system. The MSC.Dytran libraries and MADYMO libraries are linked into one executable.
   - Rigid planes in MADYMO can be visualized with MSC.Dytran Version 4.7. By requesting the geometry of the planes to be written out to the MSC.Dytran archive file, the MADYMO planes are covered with dummy shell elements at each requested output step and can be visualized during postprocessing.

5. Multiple Detonation Sources for Explosives

   In MSC.Dytran Version 4.5 only a single detonation point for high explosive materials could be used. In MSC.Dytran Version 4.7 the use of multiple detonation points for explosive materials has been implemented.

6. Contact Improvements

   A new search option called BPLANE has been implemented in the contact algorithm. The existing search option SLIDE searches for the closest master face under the assumption that a slave node will only slide from the current master face to its neighbors during one time step. This search algorithm is much faster than the FULL option. However, this search option can cause problems for slave nodes that have entered the “dead-region” on the “penetrated side” between neigboring master faces, where it can not be projected on either face, and the contact is lost because of that.



The BPLANE search option alleviates the problems for slave nodes that have entered the “dead-region” by automatically creating a plane perpendicular to the folding line between two master faces. Contact for slave nodes that resides in the back-up plane area is preserved and the nodes remain in the contact search algorithm. A force perpendicular to the BPLANE is applied to those nodes, since they are on the penetrated side of the master surface.

Because the BPLANE search algorithm is very fast and robust, it is the recommended contact search algorithm for airbag analysis.

7. Eulerian Boundary Condition Output

In analysis involving Eulerian elements, the material state at the boundaries of the mesh can be written to an archive file for interpretation during postprocessing. Available are the mass flow, transported impulse and energy through the boundary, the velocity at the boundary and forces and impulse acting on the Eulerian boundary.

8. Eulerian Body Loading

For Eulerian materials, a body force loading has been implemented. The body force acceleration field acts on top of the gravitational acceleration field.

9. Rigids

- The moments of inertia of a rigid material (MATRIG) can be defined in a local coordinate system, which is attached to the center of gravity of the MATRIG.
- A hinge connection between a rigid and a deformable Lagrangian structure (RBHINGE) has been implemented.
- An RBC3 constraint for a rigid body can be used together with a FORCE or MOMENT constraint. The FORCE or MOMENT entry can not constrain a degree of freedom that is already constrained by the RBC3 entry.

10. Rigid Lagrangian Wall

    Based upon the penalty method, a new rigid Lagrangian wall with friction has been implemented.

11. Rigid-body Rotation Correction for the BLT Shell

    For the Belytschko-Lin-Tsay shell element, a rigid-body rotation correction is now available. The rigid-body rotation correction on the hourglass forces will improve the accuracy of the analysis in problems where shell elements undergo a large rigid-body rotation.

12. USA Interface

    - The USA interface has been extended with a cavitation model.
    - A new bubble explosion algorithm has been implemented in MSC.Dytran Version 4.7, providing more accurate results in underwater explosion analysis.

13. Composites

    Aside from a rectangular coordinate system, the fiber direction for a composite laminate can also be defined in a cylindrical or spherical coordinate system. The fiber direction will be in the radial direction through the center of the shell element.

Trademarks

- MSC® and MSC.® are registered trademarks of MSC.Software Corporation.
- MSC.Dytran™ is a trademark of MSC.Software Corporation.
- NASTRAN® is a registered trademark of the National Aeronautics and Space Administration.
    - MSC.Nastran is an enhanced proprietary version developed and maintained by MSC.Software Corporation.
- Patran® is a registered trademark of MSC.Software Corporation.
- MADYMO® is a registered trademark of TNO-Industry.
- FEMB™ is a trademark of Engineering Technology Associates, Inc.
- UNIX® is a registered trademark through X/Open Company Limited.
- I-DEAS is a registered trademark of Structural Dynamics Research Corporation.
- OSF/MOTIF™ is a Trademark of the Open Software Foundation.
- Other product names and trademarks are the property of their respective owners.

Contents

CHAPTER 1

Introduction

1.1 Overview

MSC.Dytran is a three-dimensional analysis code for analyzing the dynamic, nonlinear behavior of solid components, structures, and fluids. It uses explicit time integration and incorporates features that simulate a wide range of material and geometric nonlinearity.

It is particularly suitable for analyzing short, transient dynamic events that involve large deformations, a high degree of nonlinearity, and interactions between fluids and structures. Typical applications include:

- Air bag inflation
- Air bag-occupant interaction
- Sheet metal forming analysis
- Weapons design calculations, such as self-forging fragments
- Birdstrike on aerospace structures
- Hydroplaning
- Response of structures to explosive and blast loading
- High-velocity penetration
- Ship collision

Lagrangian and Eulerian solvers are available to enable modeling of both structures and fluids. Meshes within each solver can be coupled together to analyze fluid-structure interactions. Solid, shell, beam, membrane, spring, and rigid elements can be used within the Lagrangian solver to model the structure; three-dimensional Eulerian elements can be used to create Eulerian meshes. Both the Lagrangian and the Eulerian solvers can handle hydrodynamic materials and materials with shear strength.

A general material facility can be used to define a wide range of material models including linear elasticity, yield criteria, equations of state, failure and spall models, and explosive burn models. Specific material properties can also be used for elastoplastic, orthotropic composite materials.

Transient loading can be applied to the Lagrangian elements as concentrated loads and surface pressures or indirectly as enforced motion or initial conditions. Loads can be applied to material in the Eulerian mesh by pressure or flow boundaries, and initial conditions of element variables can be prescribed.

Single-point constraints can be applied to Lagrangian grid points. Rigid walls can also be created that act as barriers to either prevent the motion of Lagrangian grid points or the flow of Eulerian material.

Contact surfaces allow parts of Lagrangian meshes to interact with each other or with rigid geometric structures. This interaction may include contact, sliding with frictional effects, and separation. Single-surface contact can be used to model buckling of structures where material may fold onto itself.

Interaction between Eulerian and Lagrangian meshes is achieved by coupling. This is based on the creation of coupling surfaces on Lagrangian structures. The coupling surface, which must form a closed volume, calculates the forces arising from the interaction and then applies the forces to the material within the Eulerian mesh and the material of the Lagrangian structure.

An alternative of constituting a fluid-structure interaction is by means of Arbitrary Lagrange Euler (ALE). This is based on the interaction at a coupling surface between the structure and the Eulerian region. The Eulerian mesh is capable of following the structure by means of an ALE moving grid algorithm. A typical application where ALE is especially efficient is the birdstrike analysis.

A simple but flexible prestress facility allows structures to be initialized with an MSC.Nastran computed prestate analysis.

A restart facility allows analyses to be run in stages.

MSC.Dytran is efficient and extensively vectorized. It provides cost-effective solutions on the latest generation of computers ranging in size from engineering workstations to the largest supercomputers. In addition, some applications can exploit the parallel processing facility for distributed memory systems that is available for simple element processing and rigid body-deformable body contact. For shared memory parallel systems, the Hughes-Liu, BLT, and Keyhoff shell formulations can use parallel processing.

This document is a user’s manual for MSC.Dytran that describes the facilities available within the code and how they can be used to model the behavior of structures. This manual explains how to run the analysis and includes advice on modeling techniques, checking the data, executing the analysis, a description of the files that are produced, and methods of postprocessing. It also gives a detailed description of all available input commands.

If you need assistance in using the code, understanding the manual, obtaining additional information about a particular feature, or selecting the best way to analyze a particular problem, contact your local MSC representative. MSC offices are located throughout the world.

We also particularly welcome your suggestions for improvements to the program and documentation so that we can keep MSC.Dytran relevant to your requirements.

1.2 Features of MSC.Dytran

The main features of MSC.Dytran include:

Elements

- Euler solid elements with four, six, and eight grid points.
- Lagrange solid elements with four, six, and eight grid points.
- Shell and membrane elements with three and four grid points.
- Beam, spring, and damper elements with two grid points.
- Spotweld elements with failure.
- Seatbelt elements.

Materials

- General material model with the definition of elastic properties, yield criterion, equation of state, spall and failure models, and explosive burn logic.
- Constitutive models for elastic, elastoplastic, and orthotropic materials.
- Constitutive models for multilayered composite materials.
- Constitutive model for sheet metal forming applications.
- Strain rate dependent material models for shells and beams.
- Constitutive models for foams, honeycombs, and rubbers.

Rigid Bodies

- Rigid ellipsoids.
- Externally defined rigid ellipsoids.
- Multifaceted rigid surfaces.
- MATRIG and RBE2-FULLRIG rigid body definition.

Constraints

- Single-point constraints.
- Kinematic joints (shell/solid connections).

- Local coordinate systems.
- Rigid body joints.
- Drawbead model in contact.

Tied Connections

- Connections to rigid ellipsoids.
- Two surfaces tied together.
- Grid points and surfaces tied together.
- Shell edges to shell surface connections.

Rigid Walls

- Rigid walls for Lagrangian elements.
- Rigid barriers to Eulerian material transport.

Contact and Coupling

- Master-slave contact between Lagrangian domains.
- Efficient single surface contact for shell structures.
- Arbitrary Lagrange-Euler (ALE) coupling.
- General Euler-Lagrange coupling for fluid-structure interactions.
- Contact with rigid ellipsoids.
- Coupling with external programs.
- Drawbead model embedded in contact.
- Contact algorithm for seat belt elements.

Loading

- Concentrated loads and moments.
- Pressure loading.
- Enforced motion.
- Eulerian flow boundaries.
- Body forces.

Initial Conditions

- Initialize any grid-point and/or element variable.
- Initialize by MSC.Nastran pre-state.
- Initialize contact.

Solution

- Structural subcycling.
- General coupling subcycling.
- Highly efficient, explicit transient solution.
- Almost completely vectorized.
- Dynamic relaxation for quasi-static solutions.
- Simple and flexible restart procedure.
- External user subroutines for advanced features.
- Application sensitive default setting.

Pre- and Postprocessing

- MSC.Nastran style input.
- Pre- and postprocessing by MSC.Patran.
- Input compatible with most modeling packages.
- Free or fixed format input.
- Translator for I-DEAS Version 6.
- Readers for The Data Visualizer from WaveFront Technologies.
- ATB output in MSC.Dytran format.

1.3 Structure of This Manual

This manual is a complete user’s guide to MSC.Dytran. It is assumed throughout the manual that you are familiar with the basic techniques of finite element analysis and that you have experience in running linear, static, and dynamic analyses. The complexities of MSC.Dytran make it difficult to use without experience in the application of finite element technology.

Chapter 1 (Sections 1.4 through 1.7) describes some aspects of MSC.Dytran that you should be familiar with before running an MSC.Dytran analysis.

Chapter 2 describes the capabilities of the program and how it can be used to model various aspects of dynamic material behavior. You should be acquainted with this section before you begin to model your problem since the techniques required to build an MSC.Dytran model are different from those for other codes.

Chapter 3 describes how to carry out the analysis. In addition to describing how to run and restart your analysis, it offers advice on modeling techniques, checking your data, and postprocessing the results.

Chapter 4 provides a detailed description of the input file and the commands necessary to define the various features of the code. You will need to refer to this when creating your input file.

Chapter 5 lists the format of the diagnostic messages that can be produced by MSC.Dytran.

Chapter 6 provides a list of references that give further information about particular aspects or facilities in the program.

Finally, Appendices A through G describe how to use the OSF/Motif based user interface XDYTRAN, the procedure for running the OSF/Motif based translator XDEXTR, how to use the parallel processing facility, how to use MSC.Dytran in conjunction with ATB, MADYMO or USA, and provide an example input file for a small problem.

1.4 Learning to Use MSC.Dytran

The simplest and quickest way to learn to use MSC.Dytran is to attend the training courses held regularly throughout the world by MSC.Software Corporation. The courses are designed to enable you to use the code quickly and reliably, to give you an in-depth understanding about how MSC.Dytran works, and how to solve problems in the most efficient way with the minimum use of computer resources. For details on when and where the courses are being held, contact your local MSC representative listed at the back of this manual.

If you are unable to attend a course and have to learn to use MSC.Dytran by reading this manual, then continue reading Chapter 1 for an overview of MSC.Dytran and how it differs from general finite element programs. Then, read those parts of Chapter 2 that describe the features you need to use to solve your first problem, concentrating particularly on the Case Control commands and Bulk Data entries that you will use to define the input data. Chapter 3 is essential reading since it describes the entire process of running an MSC.Dytran analysis from the initial modeling to postprocessing the results. Finally, while you are creating the input file, use Chapter 4 as a reference section to quickly locate the information needed to define the individual entries. If you are familiar with MSC.Nastran, read Section 4.2 on page 4-3, which describes the main differences between MSC.Nastran and MSC.Dytran.

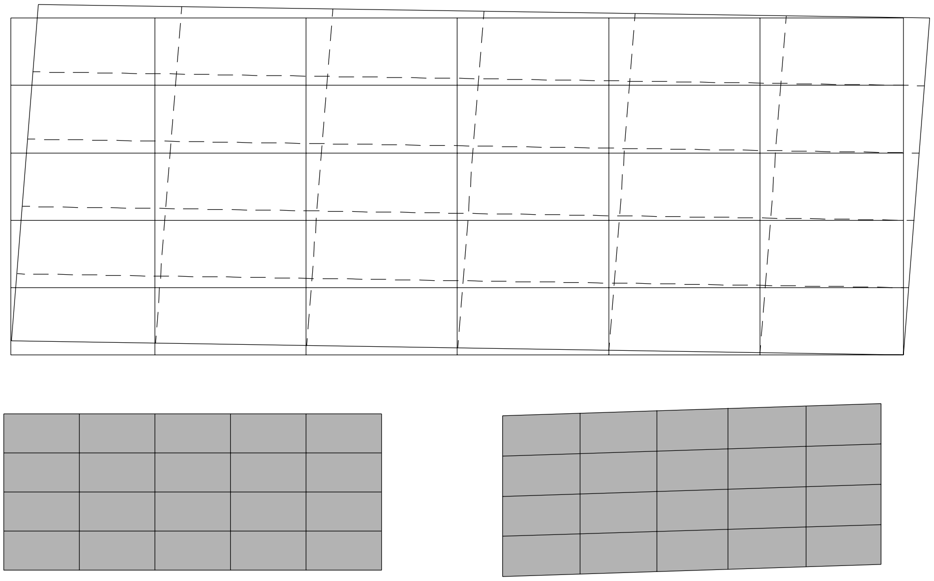
Make your first problems as simple as possible and gradually increase their complexity as you build experience in using MSC.Dytran. Remember that you can always contact your local MSC representative if you need clarification on any information provided in this manual, or if you encounter problems. Your MSC representative is there to help you!

1.5 Principles of the Eulerian and Lagrangian Processors

MSC.Dytran features two solving techniques, Lagrangian and Eulerian. The code can use either one, or both, and can couple the two types to produce interaction.

The Lagrangian method is the most common finite element solution technique for engineering applications.

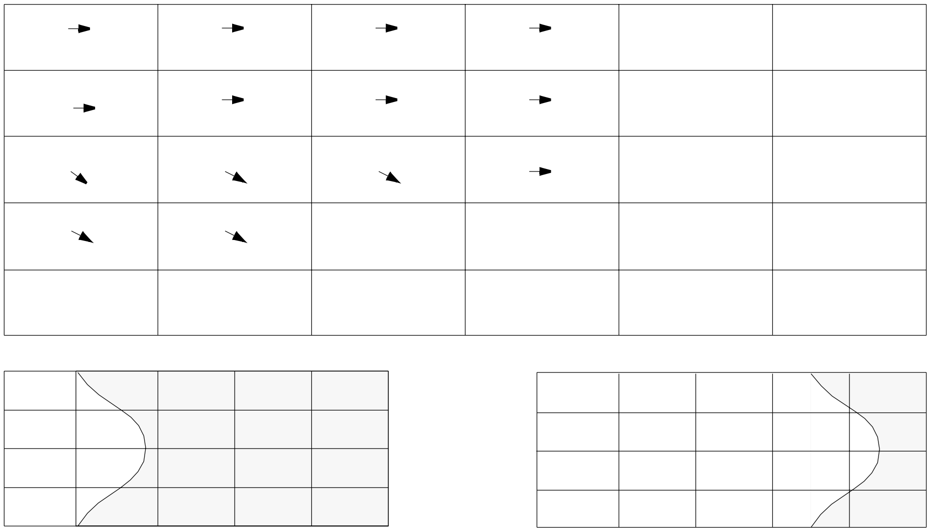
When the Lagrangian solver is used, grid points are defined that are fixed to locations on the body being analyzed. Elements of material are created by connecting the grid points together, and the collection of elements produces a mesh. As the body deforms, the grid points move with the material and the elements distort. The Lagrangian solver is, therefore, calculating the motion of elements of constant mass.



The Eulerian solver is most frequently used for analyses of fluids or materials that undergo very large deformations.

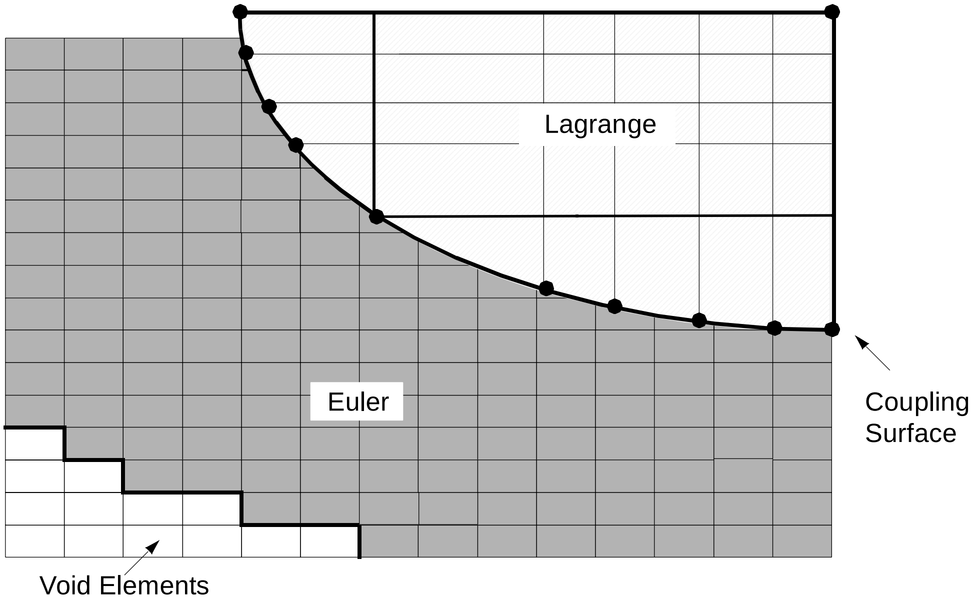
In the Eulerian solver, the grid points are fixed in space and the elements are simply partitions of the space defined by connected grid points. The Eulerian mesh is a “fixed frame of reference.” The material of a body under analysis moves through the Eulerian mesh; the mass, momentum, and energy of the material are transported from element to element. The Eulerian solver therefore calculates the motion of material through elements of constant volume.

It is important to note that the Eulerian mesh is defined in exactly the same manner as a Lagrangian mesh. General connectivity is used so the Eulerian mesh can be of an arbitrary shape and have an arbitrary numbering system. This offers considerably more flexibility than the logical rectangular meshes used in other Eulerian codes.

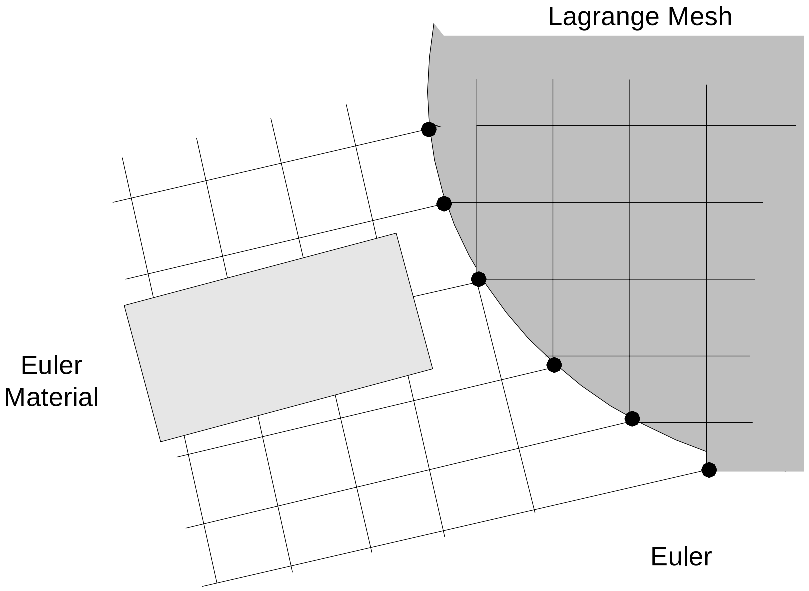


However, you should remember that the use of an Eulerian mesh is different from that of the Lagrangian type. The most important aspect of modeling with the Eulerian technique is that the mesh must be large enough to contain the material after deformation. A basic Eulerian mesh acts like a container and, unless specifically defined, the material cannot leave the mesh. Stress wave reflections and pressure buildup can develop from an Eulerian mesh that is too small for the analysis.

Eulerian and Lagrangian meshes can be used in the same calculation and can be coupled using a coupling surface. The surface acts as a boundary to the flow of material in the Eulerian mesh, while the stresses in the Eulerian material exerts forces on the surface causing the Lagrangian mesh to distort.



The alternative way of constituting a fluid-structure interaction, the Arbitrary Lagrange Euler coupling, allows Eulerian meshes to move. The structure and the Eulerian region are coupled by means of ALE coupling surfaces. The structure serves as a boundary condition for the Eulerian region at the interfaces. The Eulerian material exerts a pressure loading on the structure at the interface. The Eulerian region moves according to an ALE motion prescription in order to follow the motion of the structure. The Eulerian material flows through the Eulerian mesh while the mesh grid points can also have an arbitrary velocity.



1.6 Description of the Explicit Solution Method

The detailed theory of MSC.Dytran is outside the scope of this manual. However, it is important to understand the basics of the solution technique, since it is critical to many aspects of the code and is completely different from the usual finite element programs with which you may be familiar. If you are already familiar with explicit methods and how they differ from implicit methods, then you may disregard this section.

Implicit Methods

The majority of finite element programs use implicit methods to carry out a transient solution. Normally, they use Newmark schemes to integrate with respect to time. If the current time step is step n, then a good estimate of the acceleration at the end of step n + 1 will satisfy the following equation of motion:

$$Ma'_{n+1} + Cv'_{n+1} + Kd'_{n+1} = F^{ext}{}_{n+1}$$

where M = mass matrix of the structure

C = damping matrix of the structure

K = stiffness matrix of the structure

F^{ext} = vector of externally applied loads

a'_{n+1} = estimate of acceleration at step n + 1

v'_{n+1} = estimate of velocity at step n + 1

d'_{n+1} = estimate of displacement at step n + 1

and the prime denotes an estimated value.

The estimates of displacement and velocity are given by:

$$d'_{n+1} = d_n + v_n \Delta t + ((1-2\beta)a_n \Delta t^2)/2 + \beta a'_{n+1} \Delta t^2$$

or

$$d'_{n+1} = d^*_n + \beta a'_{n+1} \Delta t^2$$

$$v'_{n+1} = v^*_n + \gamma a'_{n+1} \Delta t$$

or

$$v'_{n+1} = v_n + (1-\gamma)a_n\Delta t + \gamma a'_{n+1}\Delta t$$

where Δt is the time step and β and γ are constants.
The terms $d*_n$ and $v*_n$ are predictive and are based on values that are already calculated.

Substituting these values in the equation of motion results in

$$Ma'_{n+1} + C(v*_n + \gamma a'_{n+1}\Delta t) + K(d*_n + \beta a'_{n+1}\Delta t^2) = F^{ext}{}_{n+1}$$

or

$$[M + C\gamma\Delta t + K\beta\Delta t^2]a'_{n+1} = F^{ext}{}_{n+1} - Cv*_n - Kd*_n$$

The equation of motion may then be defined as

$$M*a'_{n+1} = F^{residual}{}_{n+1}$$

The accelerations are obtained by inverting the $M*$ matrix as follows:

$$a'_{n+1} = M*^{-1}F^{residual}{}_{n+1}$$

This is analogous to decomposing the stiffness matrix in a linear static analysis. However, the dynamics mean that mass and damping terms are also present.

Explicit Methods

The equation of motion

$$Ma_n + Cv_n + Kd_n = F^{ext}{}_n$$

can be rewritten as

$$Ma_n = F^{ext}{}_n - F^{int}{}_n$$

$$a_n = M^{-1}\ F^{residual}{}_n$$

where F^{ext} = vector of externally applied loads

F^{int} = vector of internal loads (e.g., forces generated by the elements and hourglass forces)

$F^{int} = Cv_n + Kd_n$

M = mass matrix

The acceleration can be found by inverting the mass matrix and multiplying it by the residual load vector.

If M is diagonal, its inversion is trivial, and the matrix equation is the set of independent equations for each degree of freedom is as follows:

$$a_{ni} = F^{residual}{}_{ni} / M_i$$

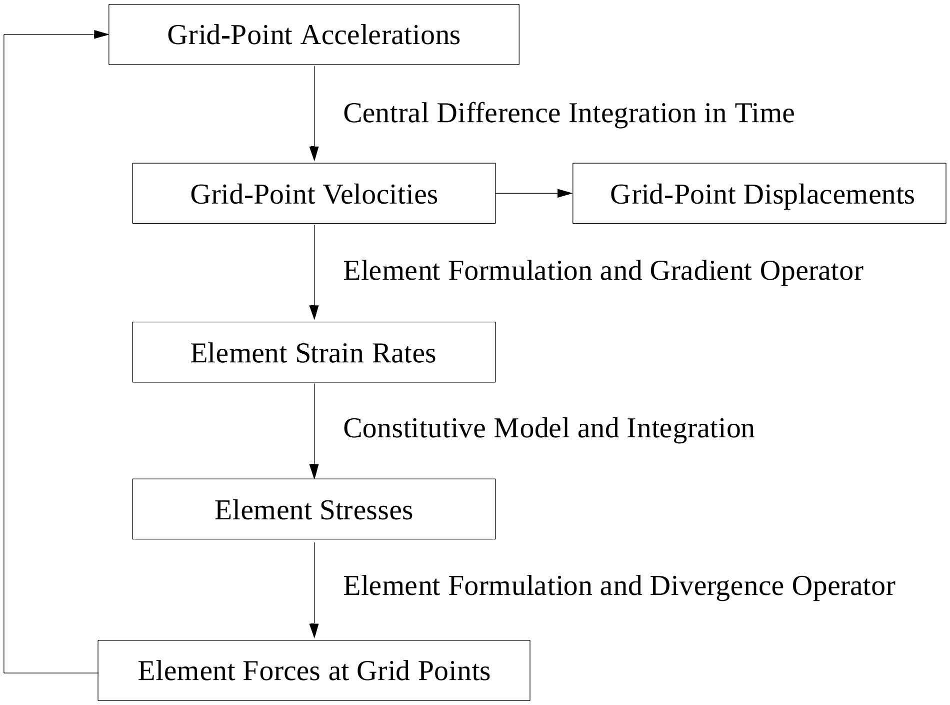
The central difference scheme is used to advance in time:

$$v_{n+1/2} = v_{n-1/2} + a_n(\Delta t_{n+1/2} + \Delta t_{n-1/2})/2$$

$$d_{n+1} = d_n + v_{n+1/2}\Delta t_{n+1/2}$$

This assumes that the acceleration is constant over the time step.

Explicit methods do not require matrix decompositions or matrix solutions. Instead, the loop is carried out for each time step as shown in the diagram that follows.


Grid-Point Accelerations
Central Difference Integration in Time
Grid-Point Velocities
Grid-Point Displacements
Element Formulation and Gradient Operator
Element Strain Rates
Constitutive Model and Integration
Element Stresses
Element Formulation and Divergence Operator
Element Forces at Grid Points

Implicit methods can be made unconditionally stable regardless of the size of the time step. However, for explicit codes to remain stable, the time step must subdivide the shortest natural period in the mesh. This means that the time step must be smaller than the time taken for a stress wave to cross the smallest element in the mesh. Typically, explicit time steps are 100 to 1000 times smaller than those used with implicit codes. However, since each iteration does not involve the costly formulation and decomposition of matrices, explicit techniques are still very competitive with implicit methods.

1.7 When to Use MSC.Dytran

The time step for implicit solutions can be much larger than is possible for explicit solutions. This makes implicit methods more attractive for transient events that occur over a long time period and are dominated by low frequency structural dynamics. Explicit solutions are better for short, transient events where the effects of stress waves are important. There is, of course, an area where either method is equally advantageous and may be used.

Explicit solutions have a greater advantage over implicit solutions if the time step of the implicit solution has to be small for some reason. This may be necessary for problems that include:

- Material nonlinearity. A high degree of material nonlinearity may require a small time step for accuracy.
- Large geometric nonlinearity. Contact and friction algorithms can introduce potential instabilities, and a small time step may be needed for accuracy and stability.
- Those analyses where the physics of the problem demands a small time step (e.g. stress wave effects).
- Material and geometric nonlinearity in combination with large displacements. Convergence in implicit methods becomes more difficult to achieve as the amount of nonlinearity for all types increases.

Explicit methods have increasing advantages over implicit methods as the model gets bigger. For models containing several thousand elements and including significant nonlinearity, MSC.Dytran may provide the cheapest solution even for problems dominated by low-frequency structural dynamics.

Once MSC.Dytran is selected to analyze a particular problem, you can use the Lagrangian solver, the Eulerian solver, or Euler-Lagrange coupling.

The benefit of the Lagrangian solver is that the displacements, deformation, and stresses in structures can be monitored with a high degree of precision. However, extreme deformations may lead to drastically reduced time steps and extended run times. The Lagrangian solver should be used for structural components that may undergo large deformation and for which the dimensions, deformed geometry, and residual stress state are of major importance. Try to use the Lagrangian solver whenever possible.

The benefit of the Eulerian solver is that complex material flow can be modeled with no limit to the amount of deformation. With increasing deformation, however, the boundaries between the materials may become less precise. The Eulerian solver should be used for bodies of material, such as fluids or solids, which may experience extremely large deformations, shock wave propagation, and even changes of state.

With the coupling feature, the advantages of both solvers can be used in one analysis. This allows you to model the interaction of precisely defined structural components with fluids and highly deformable materials.

CHAPTER 2

Modeling

2.1 Introduction

MSC.Dytran contains two finite element solvers, Lagrangian (finite element) and Eulerian (finite volume). In the Lagrangian solver, the grid points are fixed to locations on the body under analysis. Elements of material are created by connecting the grid points together, and the collection of elements produces a mesh. As the body deforms, the grid points move in space and the elements distort. The Lagrangian solver is therefore calculating the motion of elements of constant mass.

In the Eulerian solver, the grid points are fixed in space and the elements are simply partitions of the space defined by connected grid points. The Eulerian mesh is then a fixed frame of reference. The material of a body under analysis moves through the Eulerian mesh, and the mass, momentum, and energy of the material is transported from element to element. In ALE applications, the Eulerian grid points may move in space, whereby the material flows through a moving and deforming Eulerian mesh. It is important to realize that the Eulerian grid-point motion is decoupled from the material motion.

The input for the two solvers is essentially the same. The only choice you must make is what type of property the element is to have. For example, when a solid element is to be part of a Lagrangian mesh, it is assigned a PSOLID property; however, where it is to be part of an Eulerian mesh it is assigned the PEULER property. The actual definition of the grid points and element connectivity is exactly the same for both types of solvers.

2.1.1 Units

MSC.Dytran does not require the model to be defined in any particular set of units. Any set of units may be used as long as it is consistent. It is advisable to use SI units whenever possible. Some examples of consistent sets of units include:

Quantity	SI	Imperial	mm/kg/s/K	mm/tonne/s/K	mm/kg/ms/K
Length	m	in (2.54 10^{-2} m)	mm (10^{-3} m)	mm (10^{-3} m)	mm (10^{-3} m)
Time	s	s	s	s	ms (10^{-3} s)
Mass	kg	lbf-s^{2}/in (1.7513 10^{2} kg)	kg	tonne (10^{3} kg)	kg
Angle[a]	radian	radian	radian	radian	radian
Force	kg-m/s^{2}	lbf (4.4482 kg-m/s^{2})	kg-mm/s^{2} (10^{-3} kg-m/s^{2})	tonne-mm/s^{2} (1 kg-m/s^{2})	kg-mm/ms^{2} (10^{3} kg-m/s^{2})
Density	kg/m^{3}	lbf-s^{2}/in^{4} (1.0687 10^{7} kg/m^{3})	kg/mm^{3} (10^{9} kg/m^{3})	tonne/mm^{3} (10^{12} kg/m^{3})	kg/mm^{3} (10^{9} kg/m^{3})
Stress	kg/m/s^{2}	lbf/in^{2} (6.8948 10^{3} kg/m/s^{2})	kg/mm/s^{2} (10^{3} kg/m/s^{2})	tonne/mm/s^{2} (10^{6} kg/m/s^{2})	kg/mm/ms^{2} (10^{9} kg/m/s^{2})
Energy	kg-m^{2}/s^{2}	lbf-in (1.1298 10^{-1} kg-m^{2}/s^{2})	kg-mm^{2}/s^{2} (10^{-6} kg-m^{2}/s^{2})	tonne-mm^{2}/s^{2} (10^{-3} kg-m^{2}/s^{2})	kg-mm^{2}/ms^{2} (1 kg-m^{2}/s^{2})
Temperature	°K	°R (5/9 °K)	°K	°K	°K
Spec. Heat Capacity	m^{2}/s^{2}/°K	in^{2}/s^{2}/°R (1.1613 10^{-3} m^{2}/s^{2}/°K)	mm^{2}/s^{2}/°K (10^{-6} m^{2}/s^{2}/°K)	mm^{2}/s^{2}/°K (10^{-6} m^{2}/s^{2}/°K)	mm^{2}/ms^{2}/°K (1 m^{2}/s^{2}/°K)
Heat Convection	kg/s^{3}/°K	lbf/in/s/°R (3.1523 10^{2} kg/s^{3}/°K)	kg/s^{3}/°K	tonne/s^{3}/°K (10^{3} kg/s^{3}/°K)	kg/ms^{3}/°K (10^{9} kg/s^{3}/°K)
Thermal Conductivity	kg-m/s^{3}/°K	lbf/s/°R (8.0068 kg-m/s^{3}/°K)	kg-mm/s^{3}/°K (10^{-3} kg-m/s^{3}/°K)	tonne-mm/s^{3}/°K (1 kg-m/s^{3}/°K)	kg-mm/ms^{3}/°K (10^{6} kg-m/s^{3}/°K)
Thermal Expansion	m/m/°K	in/in/°R (9/5 m/m/°K)	mm/mm/°K	mm/mm/°K	mm/mm/°K

[a] MSC.Dytran uses radians as the unit for angles, and radians/time-unit for the angular velocity. For convenience, the user can often input angles in degrees and angular velocity in revolutions/time-unit. The unit required is specified on the input cards.

Sometimes the standard units are not convenient to work with. For example, Young's modulus is frequently specified in terms of MegaPascals (MPa or equivalently, N/mm^2) where 1 Pascal is 1 N/m^2. As shown in the table below, SI units are fundamental units with only conversion factors for stress and temperature.

Quantity	Common Units to	SI Units	Multiplication Factor
Length	meter (m)	meter (m)	1.0
Time	second (s)	second (s)	1.0
Mass	kilogram (kg)	kilogram (kg)	1.0
Angle	degree (^{o})	radian (rad)	1.745329 10^{-2}
Density	kg/m^3	kg/m^3	1.0
Force	Newton (N)	kg-m/s^2	1.0
Stress	MegaPascal (MPa)	kg/m/s^2	1.0 10^6
Temperature	Celcius (^{o}C)	Kelvin (^{o}K)	^{o}K = ^{o}C + 273.15
Spec. Heat Capacity	J/kg/^{o}C	m^2/s^2/^{o}K	1.0
Heat Convection	W/m^2/^{o}C	kg/s^3/^{o}K	1.0
Thermal Conductivity	W/m/^{o}C	kg-m/s^3/^{o}K	1.0
Thermal Expansion	m/m/^{o}C	m/m/^{o}K	1.0

Imperial or American units can cause confusion, however, since the naming conventions are not as clear as in the SI system. Below you can find a conversion table which will help you to derive Imperial Units from common US units:

Quantity	US Common Units to	Imperial Units	Multiplication Factor
Length	inch (in)	inch (in)	1.0
Time	second (s)	second (s)	1.0
Mass (1)	pound (lb)	lbf-s^2/in	2.590076 10^{-3}
Mass (2)	slug (lbf-s^2/ft)	lbf-s^2/in	8.333333 10^{-2}
Density	lb/in^3	lbf-s^2/in^4	2.590076 10^{-3}
Force	pound force (lbf)	pound force (lbf)	1.0
Stress	lbf/in^2	lbf/in^2	1.0
Temperature	Fahrenheit (°F)	Rankine (°R)	°R = 459.67 + °F
Spec. Heat Capacity	Btu/lb/°F	in^2/s^2/°R	3.605299 10^6
Heat Convection	Btu/in^2/sec/°F	lbf/in/s/°R	9.336076
Thermal Conductivity	Btu/in/s/°F	lbf/s/°R	9.338018 10^3
Thermal Expansion	in/in/°F	in/in/°R	1.0

Unit systems cannot be mixed. All the input to MSC.Dytran must be defined in the appropriate units for the chosen consistent set.

2.1.2 Input Format

A detailed description of the format of the input file is given in Chapter, but a brief overview is necessary here if the rest of this section is to make sense. The input data is stored in a text file with up to 80 characters on each line. This input is divided into the following sections: File Management Section, Executive Control Section, Case Control Section, Bulk Data Section, and parameter options.

File Management Section (FMS)

This section contains information concerning the file names to be used in the analysis. The section is optional and must be the first section in the input file. Each line of the file in this section is called a File Management statement.

Executive Control Section

The Executive Control Section comes between the FMS and Case Control. This section is little used in MSC.Dytran since there is no Executive System. Each line of the file in this section is called an Executive Control statement.

Case Control Section

The Case Control Section precedes the Bulk Data Section and contains information relating to the extent of the analysis and what output is required in printed form and what should be stored in files for subsequent postprocessing. Each line of the file in this section is called a Case Control command.

Bulk Data Section

The Bulk Data Section contains all the information necessary to define the finite element model—its geometry, properties, loading, and constraints. The section consists of a number of Bulk Data entries, each of which defines a particular part of the model. A single entry may occupy several lines of the input file, and it contains several fields, each of which is comprised of a single piece of data. The Bulk Data Section is usually by far the largest section in the input file.

Parameter Options

PARAM entries are defined in the Bulk Data Section. These entries are used to define various options that control aspects of the analysis. Each parameter option has a default value that is used if the option does not appear in the input file.

2.2 Grid Points

The grid points define the geometry of the analysis model. A grid point is defined on a GRID Bulk Data entry by specifying the grid-point coordinates in the basic coordinate system or in the coordinate system referred to from the GRID entry.

2.2.1 Coordinate Systems

The basic coordinate system is a rectangular one with its origin at (0.0, 0.0, 0.0) and its axes aligned with the x, y, and z axes of the model. This is implicitly defined within MSC.Dytran and is obtained by setting the coordinate system number to blank or zero. Local coordinate systems can be either rectangular, cylindrical, or spherical, and must be related directly or indirectly to the basic system. The CORD1R, CORD1C, and CORD1S entries are used to define rectangular, cylindrical, and spherical coordinate systems in terms of three grid points. The CORD2R, CORD2C, and CORD2S entries define the coordinate system in terms of the coordinates of three points in a previously defined coordinate system. Any number of local coordinate systems can be defined to ease the task of defining the geometry of the model. On input, the geometry of all the grid points is transformed to the basic system, and the sorted output gives the grid points positions in this system.

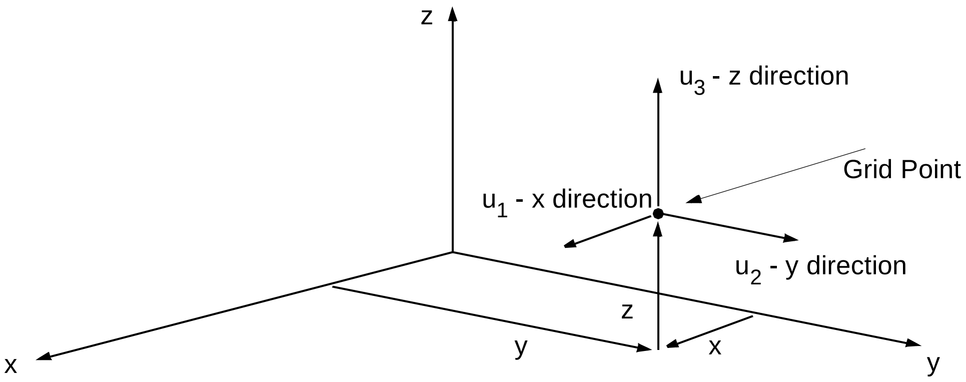


Figure 2-1. Rectangular.

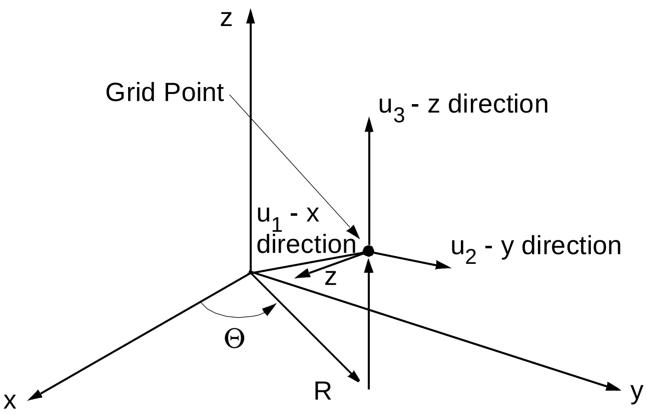


Figure 2-2. Cylindrical.

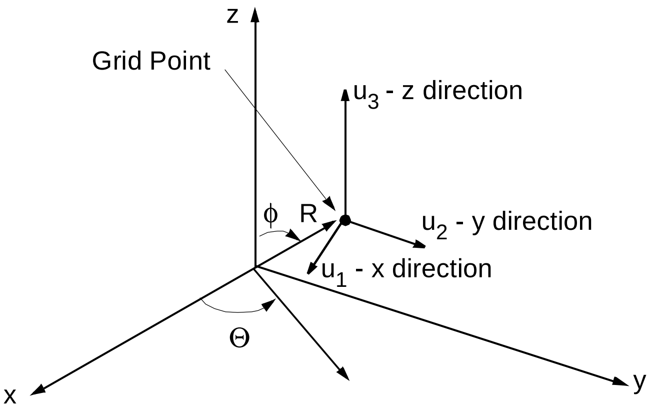


Figure 2-3. Spherical.

2.2.2 Degrees of Freedom

Each grid point can have up to six displacement components—or degrees of freedom (DOF)—depending on the elements connected to it. The degrees of freedom are three translations and three rotations in a rectangular system at an individual grid point. By default, this system will be aligned with the basic coordinate system. The coordinate system that is used to define the location of the grid point and the coordinate system to define the directions of its degrees of freedom need not be the same. The constraints acting on the grid point are in the direction of the displacement coordinate system. The displacement coordinate system is the basic system.

2.2.3 Constraints

Permanent single-point constraints can be applied on the GRID entry and are used automatically for all solutions. Note that single-point constraints can also be applied using the SPC and SPC1 entries.

The GRDSET entry allows you to specify default values for the definition coordinate system and the single-point constraints. If a zero or blank value is encountered on a GRID entry, the default value from the GRDSET entry is used. This facility saves you entering large amounts of data, for example, in the case of plane structures where all of the out-of-plane motion is prevented.

2.2.4 Grid-Point Properties

Generally, the properties of the model are associated with the structural elements, rather than the grid points. There is one exception to this, however. Mass properties are input at grid points using the CONM2 entry. These masses are in addition to those arising from the density of the structural elements.

2.2.5 Lagrangian Solver

Grid points are the fundamental definition of the geometry of the model. The spatial coordinates of grid points are defined on GRID Bulk Data entries. Each grid point can have up to six displacement components or degrees of freedom, depending on the element to which the grid point is connected. These degrees of freedom are the three translational components and three rotational components in the basic coordinate system. Permanent single-point constraints can be applied to Lagrangian grid points using a field on the GRID entry or by using one of the SPCn entries. The grid points can be constrained in any combination of the three translational components (1,2,3) and the three rotational components (4,5,6).

Solid, plate, and beam elements can be joined together by being attached to common grid points. This connection acts as a hinge where three DOF elements (solids) are connected to six DOF elements (plates/beams). If a connection of the rotational degrees of freedom is desired, you can use the KJOIN entry.

2.2.6 Eulerian Solver

The definition of a grid point is common to both the Eulerian and Lagrangian solver. Grid points are the fundamental definition of the geometry of the model. The spatial coordinates of grid points are defined on GRID Bulk Data entries.

While Lagrangian grid points can have up to six displacement components, grid points used for the definition of Eulerian elements have either zero or three degrees of freedom. They are an entirely geometric device used to define the spatial position of the Eulerian mesh.

Lagrangian and Eulerian elements cannot have common grid points. If you want to connect Lagrangian and Eulerian elements, you must create separate grid points for the two element types and then use the ALE and SURFACE Bulk Data entries.

2.2.7 Grid-Point Sequencing

The order of grid-point numbering has no effect on the solution; therefore, you are free to choose any numbering system that is convenient for data generation or postprocessing. Gaps in the grid-point numbering are allowed, and you are encouraged to use a numbering system that allows you to easily identify the location of a grid point in the model from its assigned number.

2.2.8 Mesh Generation and Manipulation

A rectangular mesh with an equidistant grid containing CHEXA elements aligned with the basic coordinate system axes can be created using the MESH Bulk Data entry.

If you want to move certain grid points you can apply an offset to the grid-point coordinates with the GROFFS Bulk Data entry.

2.3 Lagrangian Elements

2.3.1 Element Definition

There are many types of Lagrangian elements available within MSC.Dytran: solid elements (CHEXA, CPENTA, CTETRA), shell elements (CQUAD4 or CTRIA3), membrane elements (CTRIA3), beam elements (CBAR, CROD, CBEAM) and spring elements (CSPR, CVISC, CELAS, CDAMP). Most of the elements have a large strain formulation and can be used to model nonlinear effects.

The topology of an element is defined in terms of the grid points to which the element is connected. These connectivity entries are identified by a “C” prefixed to the element name, such as CHEXA or CQUAD4. The order of the grid points in this connectivity entry is important since it defines a local coordinate system within the element and therefore the position of the top and bottom surfaces of shell and membrane elements.

The connectivity entry references a property definition entry that may define some other geometric properties of the element, such as thickness. These entries are identified by a “P” prefixed to the type of element (e.g., PSOLID, PSHELL). The property entry also references a material entry.

The material entries are used to define the properties of the materials used in the model. The material models are covered in detail in Section 2.5.3 on page 2-27.

The elements can all be used with each other within the limits of good modeling practice. Care is needed when using solid and shell elements in a model since the solid elements only have translational degrees of freedom, while the shells have both translational and rotational degrees of freedom.

All the Lagrangian elements in MSC.Dytran are simple in their formulation; the solid and shell elements are based on trilinear and bilinear displacement interpolation, respectively. The elements are integrated at a single point at the centroid of the element.

Parabolic and other higher-order elements are not available to ensure maximum efficiency in the solution. The explicit formulation of MSC.Dytran requires many time steps in an analysis, perhaps in excess of 100000. It is vital, therefore, that each step is as efficient as possible. It has been shown that a larger number of simple elements produces a cheaper solution than a smaller number of more complex elements.

Users of MSC.Nastran should note that although the MSC.Dytran elements have the same names as those in MSC.Nastran, they are different in their formulation and behavior.

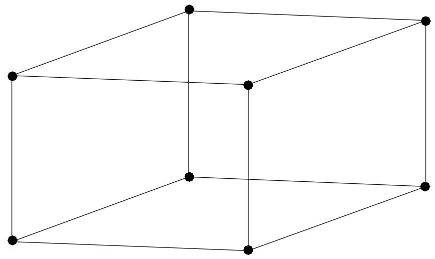
Explicit models tend to have fine meshes in regions of high plasticity or internal contacts since simple, constant force or moment elements are used.

2.3.2 Solid Elements

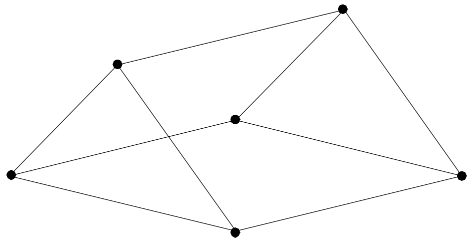
MSC.Dytran has three different forms of solid elements, which are shown below:

CHEXA Six-sided solid element with eight grid points.
CPENTA Five-sided solid element with six grid points.
CTETRA Four-sided solid elements with four grid points.

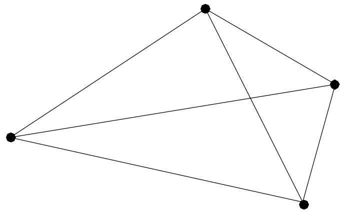
The PSOLID entry is used to assign material properties to the element.



CHEXA



CPENTA



CTETRA

The elements use one-point Gaussian quadrature to integrate the gradient/divergence operator. The Gauss point is located at the element centroid.

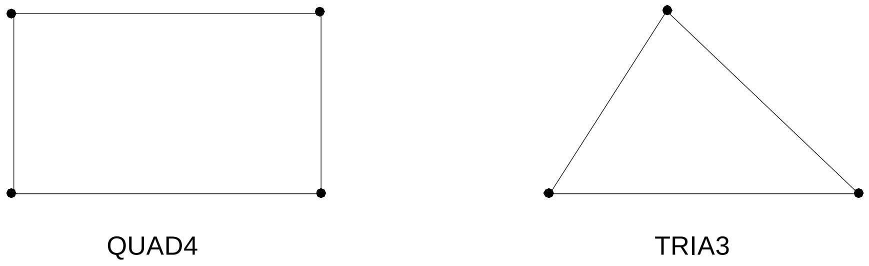
The CPENTA and CTETRA elements are degenerate forms of the CHEXA element where the grid points of the element are coincident. These elements have significantly reduced performance compared to the CHEXA element and should only be used when absolutely necessary and then should be placed well away from any areas of interest. The CTETRA element in particular tends to be too stiff and should be avoided if possible. With practice, it is possible to mesh solid regions with very complex geometry using CHEXA elements only.

The elements can be distorted to virtually any shape, although their performance is best when they are close to cuboidal. Elements inevitably become distorted during the analysis, but the code does not perform any checks on element shape, which ensures that the analysis does not abort due to one or two badly distorted elements. Therefore, the burden is on the user to ensure that the elements have sensible shapes both before and during the analysis.

2.3.3 Shell Elements

Two shell elements are available in MSC.Dytran: CQUAD4, which is a quadrilateral shell element with four grid points, and CTRIA3, which is a triangular shell element with three grid points. The CQUAD4 element uses the Belytschko-Tsay, Hughes-Liu, or Key-Hoff formulation, while the CTRIA3 uses the CO-triangle formulation.

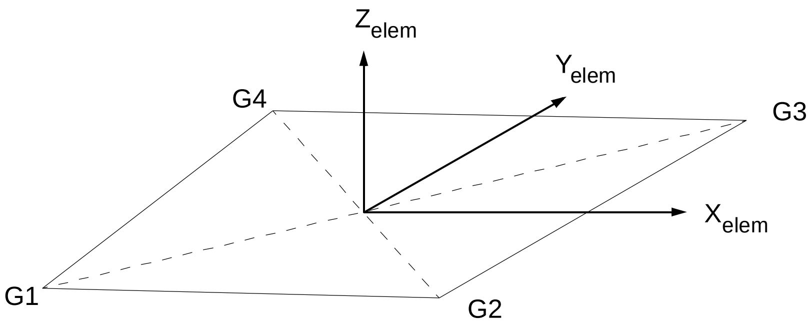
Of the various shell formulations, the Belytschko-Tsay is the most efficient and should be used in most situations. The Key-Hoff is more expensive, but performs better at large strains (over 5%). When a part of the structure suffers very large straining, you should consider using Key-Hoff shells in that area and Belytschko-Tsay shells elsewhere. The Hughes-Liu shell is substantially more expensive than the previous ones and offers an advantage only if the thickness varies within the element.



The PSHELLn or PCOMP entry is used to assign properties to the element.

Element Coordinate System

The connectivity of the Belytschko-Tsay and Hughes-Liu element, as input on the CQUAD4 or CTRIA3 entry, defines the element coordinate system. It is a rectangular coordinate system, and the direction of axes depends on the order of the grid points in the connectivity entry. The z-axis is perpendicular to the two diagonals of the element, which are given by the vectors from grid point 1 to grid point 3 and from grid point 2 to grid point 4. The x-axis is the vector from grid point 1 to grid point 2. The x-axis is always forced to be orthogonal with the z-axis. The y-axis is perpendicular to both the x-axis and the z-axis and is in the direction defined by the right-hand rule.



Each element has its own coordinate system. The top surface of a shell element is defined in the positive z-direction and the bottom surface is in the negative z-direction. The element coordinate system for the Key-Hoff and the shared-memory parallel version of Belytschko-Tsay element defines the x-axis as the line connecting the midpoints of sides G1-G4 and G2-G3.

2.3.4 Membrane Elements

The CTRIA3 element can be specified as a membrane element rather than a normal shell element. This membrane element uses a different formulation that allows the element to carry in-plane loads but no bending stiffness.

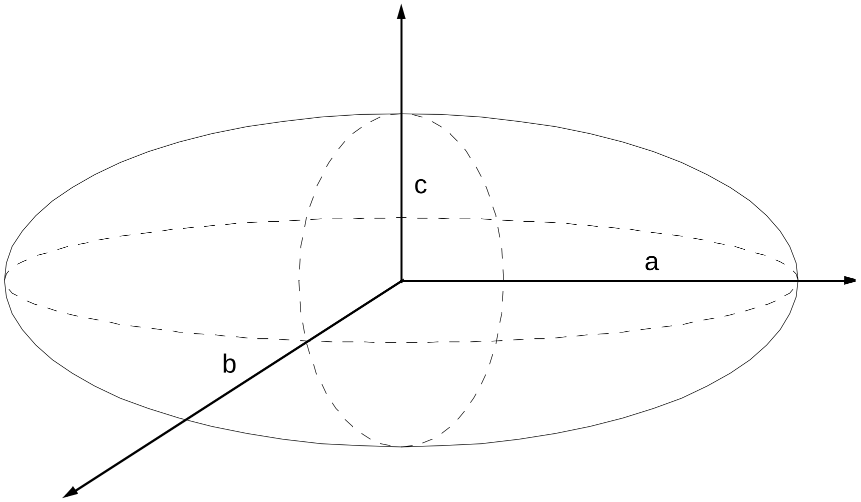
Triangular membrane elements are not large strain elements, and therefore the in-plane deformations should be small. Membrane elements can only be elastic.

2.3.5 Rigid Bodies

Rigid Ellipsoids

A rigid ellipsoid is defined on the RELLIPS Bulk Data entry. The definition consists of the ellipsoid name, mass, orientation in space, and the shape. The ellipsoid orientation is determined by the longest and the shortest axis direction. The shape is defined by three numbers (a, b, and c where $a \geq b \geq c$) that define the length of the axes. In addition, the rotational and/or translational motion of the rigid ellipsoid can be specified. The moments of inertia of the ellipsoid are calculated under the assumption that the mass is evenly distributed over the body.

The initial velocities can be specified in either the basic coordinate system or the body’s own coordinate system defined by the vectors of the major and minor axes.



The RELEX entry allows the body to be defined in an external program. Only the name of the body is required on the input entry. These are normally used for modeling of anthropomorphic dummies. This can be done by coupling MSC.Dytran with the MADYMO computer code or by using ATB, which is included in MSC.Dytran. For ATB see Appendix D, for MADYMO see Appendix E.

Specific grid points or rigid bodies can be connected to rigid ellipsoids using the RCONREL entry. Contact with rigid ellipsoids can be defined through the use of the CONTREL entry.

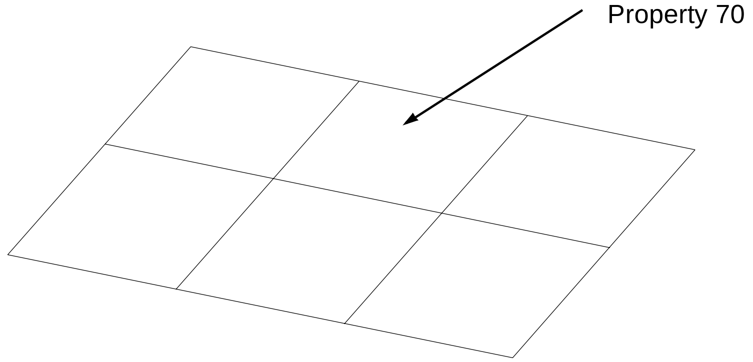
Rigid Bodies

While rigid ellipsoids are geometric entities of a fixed form, rigid bodies are user-defined surfaces that are specified as rigid. A rigid body can have almost any shape as determined by the surface from which it is made.

The RIGID entry defines the mass, center of gravity, and inertia tensor of the body and references a surface that describes the body’s shape.

The surface is defined on the SURFACE entry.

For example, the following data defines a rigid plate.

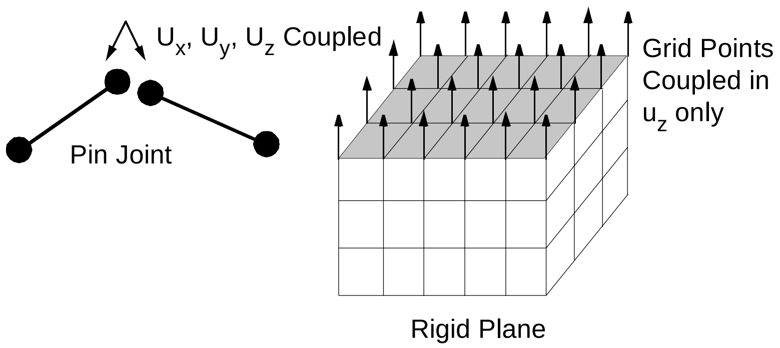


```
RIGID, 1, 100, 359, ,5, 2.5, 0.0, , +
+, , , , , , , , , +
+, 4495., , , 4495., , 4495.
SURFACE, 100, , PROP, 100
SET1, 100, 70
PSHELL1, 70, , DUMMY
CQUAD4, 1, 70, 1, 2, 12, 11
```

When a CONTACT entry references the same surface number as the RIGID entry, the body is also included in the contact surface and may interact with the other defined surfaces. Similarly, when the surface is referenced in a COUPLE or ALE entry, the rigid body is coupled to an Eulerian mesh.

Rigid Elements

Particular degrees of freedom on grid points can be specified to have the same displacement using the RBE2 entry. The degrees of freedom attached to the RBE2 move the same amount throughout the analysis. This facility can be used, for example, to model pin joints and rigid planes:


U_x, U_y, U_z Coupled
Pin Joint
Grid Points Coupled in u_z only
Rigid Plane

For rigid elements, the motion of all the degrees of freedom that are coupled is obtained by averaging their unconstrained motion. The rigid element constraints act in the basic coordinate system.

The location of the grid points is irrelevant, but you must be careful not to overconstrain the model. In the rigid plane shown above, all the grid points in the plane must have the same displacement so the plane itself cannot rotate. When rotation is required, you must use rigid elements.

There are a number of restrictions needed when using rigid elements. No grid point connected to an RBE2 can be

- Subjected to enforced motion.
- Attached to a rigid body.
- Attached to a tied connection.
- A slave point for a rigid wall.

Also, if a degree of freedom on one grid point in an RBE2 is constrained, that degree of freedom on all of the other grid points in the RBE2 should also be constrained. The RBE2 does not automatically constrain the other grid points in that RBE2 since it averages the motion of all grid points.

Translational and rotational degrees of freedom can be coupled.

An RBE2 definition using the FULLRIG option couples all degrees of freedom. All grid points defined on the RBE2 entry together behave as a rigid body.

The PARAM,CFULLRIG entry automatically converts all 123456 constraints on a normal RBE2 to the FULLRIG option.

An RBE2-FULLRIG entry can be merged with other RBE2-FULLRIG entries and with MATRIG entries into one rigid assembly by using PARAM,MATRMERG or PARAM,MATRMRG1. (See the explanation for MATRIG below).

RBE2-FULLRIG basically behaves in the same way as MATRIG. The only difference is that the grid points of an RBE2-FULLRIG are attached to elements which have deformable materials. Therefore, RBE2-FULLRIG is more expensive to use than MATRIG which can skip the whole material solver. In addition, for an element with a deformable material whose grid points belong to one RBE2-FULLRIG, the stresses and strains should vanish. In practice however, there can be spurious noise due to the discretization of the nodal rotations from one cycle to the next. It is advised, therefore, to use MATRIG instead of RBE2-FULLRIG, when possible.

MATRIG

Parts of the mesh can be made rigid by replacing the material definition with a MATRIG entry. All elements referred to by the MATRIG material number will behave as a rigid body. This can be convenient in situations where large rigid body motions arise, which are expensive to simulate with deformable elements.

MATRIG definitions can also be merged. In this case, the set of MATRIG entries behaves as one rigid body. In addition, MATRIG entries can also be merged with RBE2 entries which have the FULLRIG option. Merging can be achieved with PARAM,MATRMERG or PARAM,MATRMRG1. The PARAM,MATRMERG merges all MATRIG and RBE2-FULLRIG definitions which are mentioned on the entry in a new rigid assembly. The properties (mass, center of gravity and moments of inertia) are computed from the properties of each of the individual merged definitions. The PARAM,MATRMRG1 entry performs the same merging but there can be pre-defined properties for the new rigid assembly.

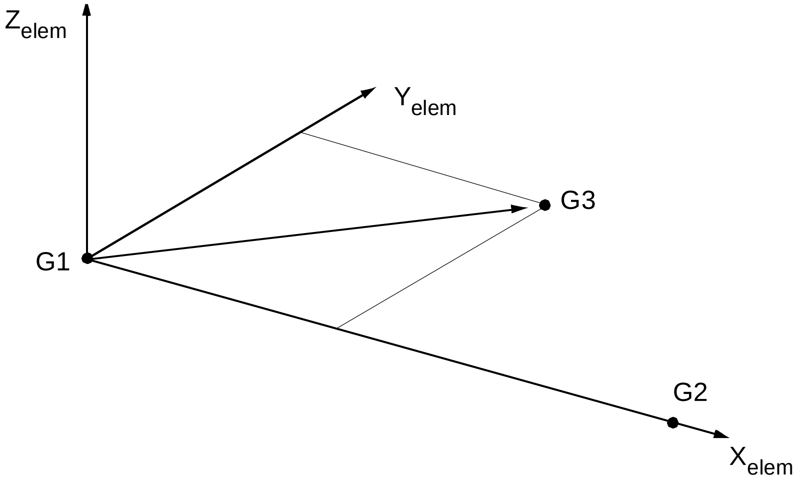
2.3.6 Beam Elements

The beam element is defined using either the CBAR or CBEAM entry. Both have the same effect and define the same element. CBAR is easier to use and is recommended for this reason. The CBEAM entry allows compatibility with modeling packages that do not use the CBAR entry. The properties of the beam can be defined using the PBAR, PBEAM, or PBEAM1 entry. Only the basic data used for the PBAR entry is extracted from PBEAM; the additional features of PBEAM available in MSC.Nastran are not used in MSC.Dytran.

Element Coordinate System

The beam element connects two grid points, but you must define the orientation of the beam and its element coordinate system. The definition can be done in two ways:

- Using a third grid point in the xy-plane.
- Using a vector in the xy-plane.



The element x-axis is aligned with the direction of G1 to G2. A vector with its origin at G1 is either defined explicitly or by defining a third grid point, in which case the vector is from G1 to G3. This vector defines the xy-plane with the element y-axis perpendicular to the element x-axis. The element z-axis is perpendicular to both the element's x- and y-axis.

The element coordinate system is defined at the start of the calculation. It is automatically updated depending on the distortion of the beam during the analysis.

Formulations

There are two types of beam formulations:

- Belytschko-Schwer
- Hughes-Liu

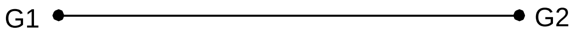
The element material can either be defined as elastic by referencing a MAT1 entry, or as elastoplastic by referencing a DMATEP entry.

If an elastoplastic material is specified for Belytschko-Schwer beams, a resultant plasticity model is used, whereby the entire cross-section yields at once. It is not possible to choose a strain rate dependent yield model for elastoplastic Belytschko-Schwer beams.

2.3.7 Rod Elements

A rod element can be defined using a CROD entry. A rod connects two grid points and can carry only axial tension and compression. It cannot carry any torsion or bending; for torsion or bending, the CBAR or CBEAM element should be used.

The only required property is the cross-sectional area of the rod that is specified using the PROD entry.



2.3.8 Spring Elements

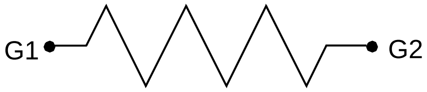
There are two types of spring elements available in MSC.Dytran: the CSPR and CELASn spring elements.

CPSR spring elements only connect translational degrees of freedom. The CELASn spring elements can connect both translational and rotational degrees of freedom. For rotational springs, you should define the moment/angle characteristic. In the remainder of this section, force and displacement are described for simplicity. You should substitute these terms by moment and angle for rotational springs.

The spring properties are defined using PSPRn or PELAS entries. There are three types of spring elements available: linear, nonlinear, and user-defined spring elements.

CSPR Elements

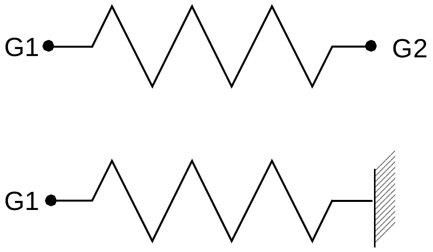
The CSPR element always connects two grid points and defines the force/deflection characteristic between the two points. The force always acts in the direction of the line connecting the grid points. As the position of the grid points changes during the analysis, the line of action of the force will change as well. The CSPR element is similar to the CROD element except that the force/deflection characteristic is defined directly rather than defining the area and material properties.



The spring properties are defined using PSPRn entries. There are three types of springs: one linear, one nonlinear, and one that is defined via a user subroutine.

CELAS1 and CELAS2 Elements

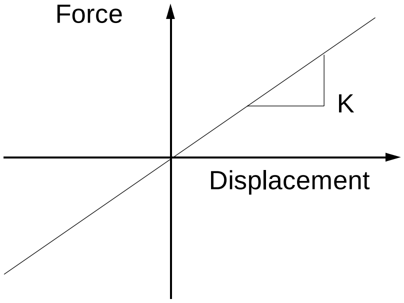
The CELASn elements connect either one or two grid points. If only one grid point is specified, the spring is grounded. In addition, you must specify the direction of the spring. The force in the spring always acts in this direction regardless of the motion of the grid points during the analysis.



The CELAS1 and CELAS2 elements are linear springs. The spring characteristic from a CELAS1 spring element is defined by referring to a PELAS entry. The spring characteristic for a CDAMP2 spring element is defined on the CDAMP2 entry directly.

Linear Elastic Springs (PSPR and PELAS)

The force is proportional to the displacement of the spring.



You must define the stiffness K of the spring.

Nonlinear Elastic Springs (PSPR1, PELAS1)

The force is not proportional to the displacement, but no permanent deformation of the spring occurs.

The force/deflection characteristic can be of any shape and is defined by specifying a table of force/deflection values using a TABLED1 entry. Loading and unloading occurs corresponding to the curve. You must define the entire curve in both tension and compression. The force associated with a particular displacement is determined by linear interpolation within the table range or by using the end point values outside the table range.

User-Defined Springs (PSPREX and PELASEX)

In this case, the force/displacement characteristic is defined in an external FORTRAN subroutine. The PSPREX and PELASEX entries let you define property data that is passed to the subroutine by MSC.Dytran. The subroutine is included in an external file that is referenced by the USERCODE statement in the File Management Section. For details on how to use user subroutines, see Section 3.15 on page 3-74.

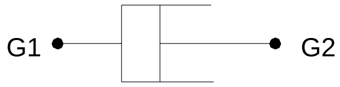
User-defined springs can, of course, have any characteristic that you want based on the displacement, velocity, or acceleration of the end points. They are, however, less efficient to use than the linear and nonlinear elastic springs.

2.3.9 Damper Elements

There are two types of damper elements available in MSC.Dytran: the CVISC and the CDAMPn damper elements.

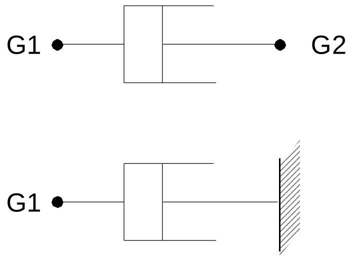
The CVISC damper elements connect translational degrees of freedom only. The PELASn damper elements can connect both translational and rotational degrees of freedom. For translational dampers, you should define the force/velocity characteristic. For rotational dampers, you should define the moment/angular velocity characteristic. In the remainder of the section, the force and velocity are described for simplicity. You should substitute these terms with moment and angular velocity for rotational dampers.

The damper properties are defined using PVISCn or PDAMP entries. There are three types of dampers available: linear, nonlinear, and user-defined dampers.



CDAMP1 and CDAMP2 Element

The CDAMPn elements connect either one or two grid points and are the equivalent of the CELASn spring elements. If only one grid point is specified, the damper is grounded. In addition, you must specify the direction of the damper. The damping force always acts in this direction regardless of the motion of the grid points during the analysis.



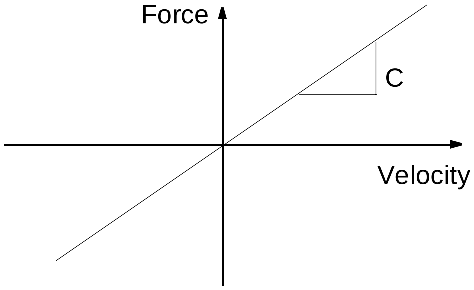
The CDAMP1 and CDAMP2 elements are linear dampers. The damper characteristic for CDAMP1 element is defined by referring to a PDAMP entry. For a CDAMP2 element, the damper characteristic is defined on the CDAMP2 entry directly.

The damper properties are defined using PVISCn and PDAMPn entries. There are three types of dampers: linear, nonlinear, and one that is defined using a user subroutine.

Linear Dampers (PVISC and PDAMP)

The force is proportional to the relative velocity of the end points.

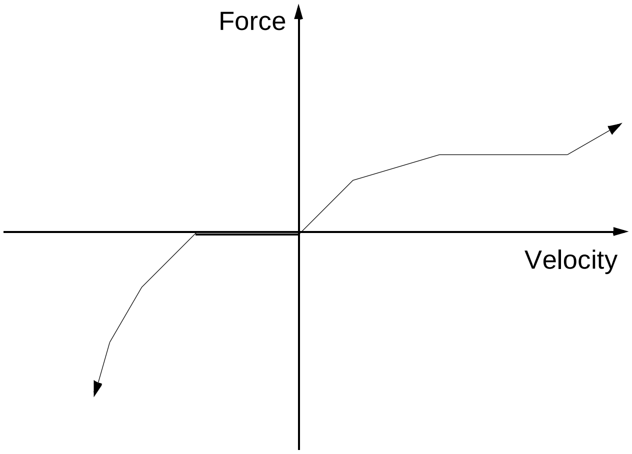
You must define the damping constant C.



Nonlinear Dampers (PVISC1)

The force/velocity characteristic is nonlinear.

The force/velocity characteristic can be of any shape and is defined by specifying a table of force/velocity values using a TABLED1 entry. You must specify the entire curve in both tension and compression. The force associated with a particular velocity is determined by linear interpolation within the table range or by using the end point values outside the table range.



User-Defined Dampers (PVISCEX)

In this case the force/velocity characteristic is defined in an external FORTRAN subroutine. The PVISCEX entry lets you define property data that is passed to the subroutine by MSC.Dytran. The subroutine is included in an external file that is referenced by the USERCODE statement in the File Management Section. For details on how to use user subroutines, see Section 3.15 on page 3-74.

The user-defined dampers can, of course, have any characteristic that you want, based on the displacement, velocity, or acceleration of the end points. They are, however, less efficient to use than the linear and nonlinear dampers.

2.3.10 Lumped Masses

Additional mass and inertia can be applied to a grid point using the CONM2 entry.

All grid points in the model have mass, either by the properties of the structural elements attached to the grid points or by using a CONM2 entry. If, for example, a spring is connected at a grid point and there is no other element attached to the grid point, a CONM2 entry is used to define the mass at that grid point.

2.4 Eulerian Elements

2.4.1 Element Definition

In the Eulerian solver, the mesh is defined by grid points and solid elements. The elements are specified as being (partially) filled with certain materials or with nothing (VOID), and initial conditions are defined.

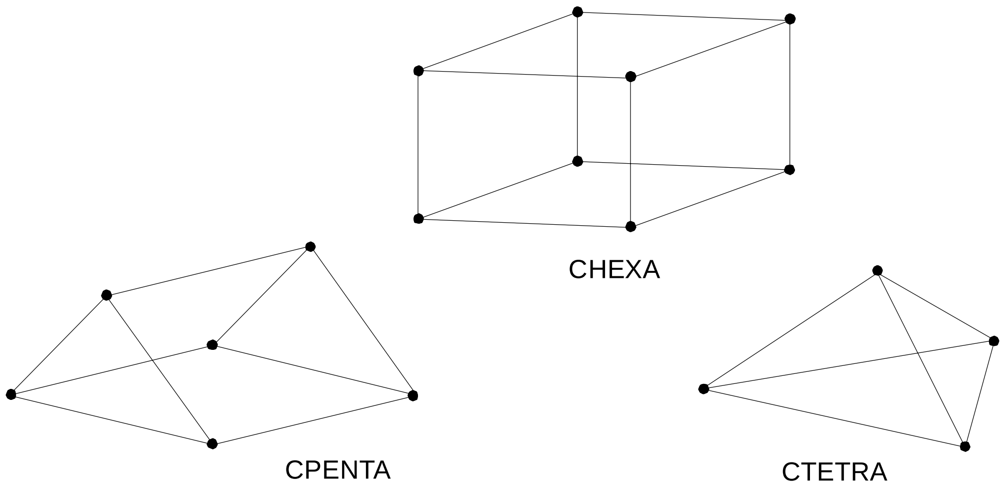
As the calculation proceeds, the material moves relative to the Eulerian mesh. The mass, momentum, and energy of the material is transported from element to element depending on the direction and velocity of the material flow. MSC.Dytran then calculates the impulse and work done on each of the faces of every Eulerian element.

Eulerian elements can only be solid but have a general connectivity and therefore are defined in exactly the same way as Lagrangian elements.

2.4.2 Solid Elements

There are three types of Euler elements, a six-sided CHEXA with eight grid points defining the corners, a CPENTA with six grid points, and a CTETRA with four grid points. The connectivity of the element is defined in exactly the same manner as a Lagrangian element, that is, with a CHEXA, CPENTA, or CTETRA entry. However, in order to differentiate between Lagrangian and Eulerian solid elements, the property entry for Euler is PEULERn rather than PSOLID.

Unlike Lagrangian solid elements, the CPENTA and CTETRA elements perform just as well as the CHEXA element. They can be used, therefore, wherever meshing demands such use.



The PEULERn entry references a DMAT material entry that is used to define the material filling the elements at the start of the calculation. When no material entry is referenced (the field contains a zero), the element is initially void.

2.5 Constitutive Models

2.5.1 Definition

Most elements reference a property entry, which in turn, references a material entry. There are several material models available that allow a wide range of linear and nonlinear material behavior. Material properties are defined using the material entries listed below:

DMATEL	Elastic material
DMATEP	Elastoplastic material with failure
DMATOR	Orthotropic elastic material
MAT1	Elastic material
MAT8	Orthotropic elastic material with failure
DYMAT14	Soil and crushable foam
DYMAT24	Piecewise linear plasticity
DYMAT26	Crushable orthotropic material
FOAM1	Foam material (polypropylene)
FOAM2	Foam material (expanded polypropylene)
RUBBER1	Mooney-Rivlin rubber material
SHEETMAT	Sheet metal material (anisotropic plastic material)

2.5.2 Choice of Constitutive Model

With so many different material models available, knowing which one to use is not always easy, particularly since a number of the models offer similar behaviors. The following sections describe the behavior of each material model, suggest typical applications for it, and mention other material models that offer similar but slightly different features. No attempt is made to explain the mathematical theory of the models.

The main rule to follow when selecting a material model is to keep it as simple as possible. Simple models are much more efficient since they require fewer calculations, and it is often easier to understand their behavior. You should also consider how accurate your knowledge of material properties is. No matter how sophisticated the material model and the formulation of the elements, the results can only be as accurate as your input data. The large strain properties of materials under dynamic cyclic loading at high strain rates is an area where little information is available, and often requires special testing. Such tests are difficult to carry out and may have a large margin of error associated with them. If you do not have a high level of confidence in your material properties, use a relatively simple material model and consider running several analyses with different models and assumptions to see how sensitive the results are to the input data.

The list below indicates which material entries can be used with the various types of elements:

- DMAT

  Lagrangian solids and membranes.
  Eulerian solids.

- DMATEL

  Lagrangian solids and membranes.

- DMATEP

  Lagrangian shells and beams.

- DMATOR

  Lagrangian solids.

- MAT1

  Lagrangian shells and beams.

- MAT8

  Lagrangian shells.

- SHEETMAT

  Lagrangian shells.

- DYMAT14

  Lagrangian solids.

- DYMAT24

  Lagrangian solids, shells, and beams.

- DYMAT26

  Lagrangian solids.

- RUBBER1

  Lagrangian solids.

- FOAM1

  Lagrangian solids.

- FOAM2

  Lagrangian solids.

The MSC/DYNA Version 3 materials listed in Section 2.5.3 on page 2-27 can be addressed in MSC.Dytran using the exact same format as used in MSC/DYNA. Materials listed in parentheses are implemented in MSC.Dytran by the name shown in parentheses. For details about the exact definitions, see the Bulk Data Section of Chapter 4. The other material definitions are mapped to equivalent MSC.Dytran materials.

MSC/DYNA Version 3	Material Description	MSC.Dytran Version 3
MAT1	Isotropic, linear, elastic material.	(MAT1)
DYMAT1	Isotropic, elastic material.	DMATEP
DYMAT2	Orthotropic, elastic material. (Solid Lagrangian elements.)	DMATOR
DYMAT3	Elastoplastic, nonlinear material with isotropic hardening.	DMATEP
DYMAT5	Nonlinear, elastic perfectly plastic soil and crushable foam. Crushing under hydrostatic loading, elastoplastic under deviatoric loading.	DYMAT14
DYMAT6	Viscoelastic material.	DMAT + SHRLVE
DYMAT12	Elastoplastic, nonlinear material with isotropic hardening.	DMATEP
DYMAT12A	Like DYMAT12, but the shear and bulk modulus define the material behavior.	DMATEP
DYMAT13	Nonlinear, isotropic, elastotropic material with failure.	DMATEP
DYMAT13A	Like DYMAT13, but the shear and bulk modulus define the material behavior.	DMATEP
DYMAT14	Nonlinear, elastic perfectly plastic, compressible soil and crushable foam, with failure. Crushing under hydrostatic loading, elastoplastic under deviatoric loading. (Solid Lagrangian elements).	(DYMAT14)
DYMAT24	Elastoplastic, nonlinear, plastic material with isotropic hardening. Stress-strain curve is piecewise linear.	(DYMAT24)
DYMAT26	Orthotropic crushable material. (Solid Lagrangian elements.)	(DYMAT26)

2.5.3 Materials

2.5.3.1 DMAT – General Material

The DMAT material entry is a general material definition and provides a high degree of flexibility in defining material behavior. The basis of the DMAT entry is the reference of a combination of material descriptions: equation of state, yield model, shear model, failure model, and spall model. Each of these functions is defined by its own entry and is described further in Sections 2.5.4 on page 2-60 through 2.5.8 on page 2-77. The only material parameter defined on the DMAT entry is the reference density.

The DMAT entry can be used to define all types of material behavior from materials with very simple linear equations of state to materials with complex yielding and shearing behavior and different failure criteria.

The required input is the reference density, the number of an EOSxxx entry defining the equation of state, and the number of an SHRxxx entry defining the shear properties of the material. The equation of state defines the bulk behavior of the material. It may be a polynomial equation, a gamma law gas equation, or an explosive equation. A single-term polynomial equation produces a linear elastic behavior.

Further material property definitions are optional. A referenced YLDxxx entry selects one of the following: a hydrodynamic response (zero yield stress), a von Mises criterion that gives a bilinear elastoplastic behavior, or a Johnson-Cook yield model where the yield stress is a function of plastic strain, strain rate, and temperature. If no YLDxxx model is referenced, the material is assumed to be fully elastic.

A FAILxxx entry can be referenced to define a failure model for the material. This failure model can be based on a maximum plastic strain limit, a maximum stress limit, or a user-defined failure criterion included in an external subroutine. If no FAILxxx entry is referenced, the material has no failure criterion.

A PMINxxx entry can be referenced to define the spall characteristics of the material. Currently, only the PMINC entry is available. The entry provides a constant spall limit for the material. When no PMINxxx entry is referenced, the material has no spall limit for Lagrangian elements and a zero spall limit for Eulerian elements.

2.5.3.2 DMATEL – Elastic Material

The DMATEL entry provides a convenient way of defining the properties of isotropic elastic materials. The reference density is defined along with any two of the four elastic material constants: Young’s modulus E, Poisson’s ratio ν, bulk modulus K, and shear modulus G.

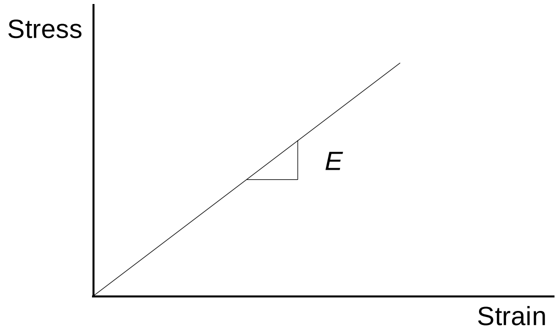


Figure 2-4. Elastic Stress-Strain Curve.

The elastic constants are related by the following equations:

$$G = \frac{E}{2(1+\nu)}, \quad K = \frac{E}{3(1-2\nu)}$$

2.5.3.3 DMATEP – Elastoplastic Material

The DMATEP entry defines the properties of an isotropic, elastoplastic material with failure.

The reference density is required, together with any two of the four elastic material constants: Young's modulus *E*, Poisson's ratio ν, bulk modulus *K*, and shear modulus *G*. When only these elastic properties are defined, the material behavior is linear, isotropic, and elastic. A YLDVM entry can also be referenced, in which case a bilinear or piecewise linear elastoplastic material model is obtained. For CQUADy and CTRIAz elements a YLDJC entry can be referenced to define a Johnson-Cook yield model. A FAILxxx entry can be referenced to define a failure model for the material. This failure model can be based on a maximum plastic strain limit or a user-defined failure criterion included in an external user subroutine. When no FAILxxx entry is referenced, the material has no failure criterion at all.

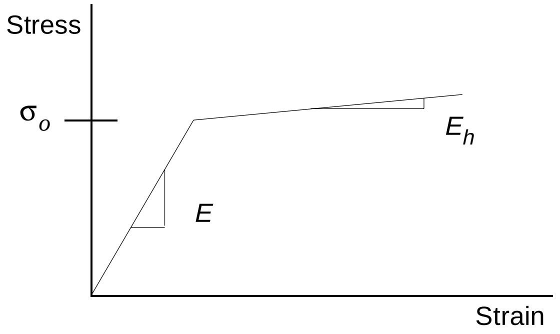


Figure 2-5. Elastic-Plastic, Stress-Strain Curve.

2.5.3.4 DMATOR – Orthotropic Material

The DMATOR entry defines the properties of an orthotropic elastic material. The material model can only be used with Lagrangian solid elements.

The model is for orthotropic linear elastic materials. You must define the material properties in a material coordinate system (a, b, c). The relationship between stress σ and strain ε is as follows:

$$\sigma = [C]\varepsilon$$

where $[C]$ = $[T]^t[C_L][T]$

$[T]$ = the transformation matrix between the material coordinate system (a, b, c) and the basic coordinate system, and

$[C_L]$ = the local constitutive matrix defined in the material coordinate system

$$[C_L]^{-1} = \begin{bmatrix} 1/E_a & -\nu_{ba}/E_b & -\nu_{ca}/E_c & 0 & 0 & 0 \\ -\nu_{ab}/E_a & 1/E_b & -\nu_{cb}/E_c & 0 & 0 & 0 \\ -\nu_{ac}/E_a & -\nu_{bc}/E_b & 1/E_c & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/G_{ab} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/G_{bc} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/G_{ca} \end{bmatrix}$$

Since $\nu_{ab}/E_a = \nu_{ba}/E_b$, $\nu_{ca}/E_c = \nu_{ac}/E_a$, and $\nu_{cb}/E_c = \nu_{bc}/E_b$, the matrix is symmetrical.

You must define the following properties:

E_a, E_b, E_c Young’s moduli in the principal material directions.

ν_{ab}, ν_{ca}, ν_{cb} Poisson ratios between the b- and a-axis, the c- and a-axis, and the c- and b-axis.

G_{ab}, G_{bc}, G_{ca} Shear moduli in the ab, bc, and ca planes.

The material coordinate system is defined by specifying two vectors, V1 and V2.

The first vector defines the direction of the a-axis. The c-axis is perpendicular to both vectors. The b-axis is perpendicular to the a- and c-axis. The material coordinate system is independent of the element’s shape and position. A FAILxxx entry can be referenced to define a failure model for the material. The failure model can be based on a maximum stress limit, a maximum pressure limit, or a user-defined criterion included in an external user subroutine.

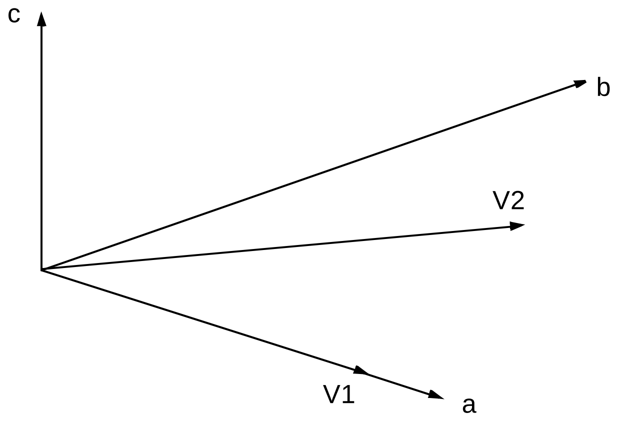


Figure 2-6. Material Coordinate System.

2.5.3.5 MAT8 – Fiber-Composite Material with Failure

The orthotropic material model is used in shell elements to build a multilayered composite element. The material describes the elastic behavior of brittle material with failure based on the interactive stress criteria of failure per mode. The elastic stress-strain relation between the fiber and matrix stresses and strains is formulated as

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \end{bmatrix} = \frac{1}{(1-\nu_{12}\nu_{21})} \begin{bmatrix} E_{11} & \nu_{21}E_{11} \\ \nu_{21}E_{11} & E_{22} \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \end{bmatrix}$$

evaluated at $t + 1/2\Delta t$.

The shear stress-strain relation is defined as

$$\gamma_{12} = \frac{1}{G_{12}}\sigma_{12} + 3\alpha\bar{\sigma}_{12}^{2}\sigma_{12}$$

where α is an experimentally derived value. Setting α to zero reduces the elastic behavior in relation to orthotropic Hooke's Law.

For the prediction of failure, MSC.Dytran has a variety of models available. The first class of models contains the interactive models that predict the onset of failure, but not the failure mode. This class contains the Tsai-Hill, and Tsai-Wu failure theories. The second class not only predicts the onset of failure, but provides the fiber compression (fiber buckling), matrix tension (matrix cracking), matrix compression, or in-plane shear failure. Theories that fall in the latter class are the Chang-Chang, maximum stress, modified Tsai-Wu, and Hashin failure theory.

In addition to the closed-form theories mentioned above, MSC.Dytran has the option to combine several theories in a combination model to define the failure for each separate mode. If this is not sufficient, it is possible to supply a user model, which can accommodate up to ten user history variables.

A summary of failure theories is given below.

Tsai-Hill

$$\frac{\sigma_{11}^2}{X^2} - \frac{\sigma_{11}\sigma_{22}}{X^2} + \frac{\sigma_{22}^2}{Y^2} + \frac{\sigma_{12}^2}{S^2} \geq 1$$

Tsai-Wu

$$F_1\sigma_{11} + F_2\sigma_{22} + F_{11}\sigma_{11}^2 + F_{22}\sigma_{22}^2 + 2F_{12}\sigma_{11}\sigma_{22} + F_{66}\sigma_{12}^2 \geq 1$$

$$F_1 = \frac{1}{X_T} - \frac{1}{X_C} \quad F_2 = \frac{1}{Y_T} - \frac{1}{Y_C}$$

$$F_{11} = \frac{1}{X_T X_C} \quad F_{22} = \frac{1}{Y_T Y_C} \quad F_{66} = \frac{1}{S^2}; \quad F_{12} \text{ by biaxial test}$$

Modified Tsai-Wu

Matrix failure	$F_2\sigma_{22} + F_{22}\sigma_{22}^2 + F_{66}\sigma_{12}^2 \geq 1$

Maximum Stress

Fiber tension	$\sigma_{11} \geq X_T \; (\sigma_{11} > 0)$
Fiber compression	$\lvert\sigma_{11}\rvert \geq X_C \; (\sigma_{11} < 0)$
Matrix tension	$\sigma_{22} \geq Y_T \; (\sigma_{22} > 0)$
Matrix compression	$\lvert\sigma_{22}\rvert \geq Y_C \; (\sigma_{22} < 0)$
Matrix shear	$\lvert\sigma_{12}\rvert \geq S$

Hashin

Fiber tension $$\left(\frac{\sigma_{11}}{X_T}\right)^2 + \left(\frac{\sigma_{12}}{S}\right)^2 \geq 1 (\sigma_{11} > 0)$$

Fiber compression $$|\sigma_{11}| \geq X_C \ (\sigma_{11} < 0)$$

Matrix tension $$\left(\frac{\sigma_{22}}{Y_T}\right)^2 + \left(\frac{\sigma_{12}}{S}\right)^2 \geq 1 \quad (\sigma_{22} > 0)$$

Matrix compression $$\left(\frac{\sigma_{22}}{2S_T}\right)^2 + \left[\left(\frac{Y_C}{2S_T}\right)^2 - 1\right]\frac{\sigma_{22}}{Y_C} + \left(\frac{\sigma_{12}}{S}\right)^2 \geq 1 \quad (\sigma_{22} < 0)$$

Chang

Fiber breakage $$\left(\frac{\sigma_{11}}{X_T}\right)^2 + T \geq 1 \quad (\sigma_{11} > 0)$$

Matrix cracking $$\left(\frac{\sigma_{22}}{Y_T}\right)^2 + T \geq 1 \quad (\sigma_{22} > 0)$$

Matrix compression $$\left(\frac{\sigma_{22}}{2S}\right)^2 + \left[\left(\frac{Y_C}{2S}\right)^2 - 1\right]\frac{\sigma_{22}}{Y_C} + T \geq 1 \quad (\sigma_{22} < 0)$$

$$T = \left(\frac{\sigma_{12}}{S}\right)^2 \frac{1 + \frac{3}{2}\alpha G_{12}\sigma_{12}^2}{1 + \frac{3}{2}\alpha G_{12}S^2}$$

When a failure criterion is satisfied, the next stage is to define how the remaining modes are affected by the failed mode. A standard model is available, which is an average of the various theories provided in the literature. However, the property degradation rules are not fixed and can be easily redefined by the user. The property degradation rules describe how stress increments are related to strain increments in the various directions after failure in a particular mode has occurred.

Material Constant	Failure Mode Fiber Tens	 Fiber Comp	 Matrix Tens	 Matrix Comp	 Shear
E1	X	X			
E2	X	X	X	X	
ν12	X	X	X	X	
G12	X		X		X

For example, in matrix compression failure, the material constants E2 (lateral Young’s modulus), and ν12 (Poisson’s ratio) are set to zero.

Finally, the model describes how the stresses are relaxed to zero after failure has occurred. The relaxation can start either when a particular mode has failed or when all material properties (E1, E2, ν12, G12) are degraded to zero according to the property degradation rule. The relaxation always occurs in time, either in problem time units by a propagation velocity, or simply by time steps. This model is referred to as the post-failure degradation rule.

2.5.3.6 SHEETMAT – Anisotropic Plastic Material Model

The SHEETMAT entry defines the Krieg constitutive material model. This model is primarily intended to describe the anisotropic plastic behavior of thin-rolled metal sheets. It can only be used with Lagrangian shell element formulations (BLT, BELY, CO-TRIA and KEYHOFF) because the model is based on a plane stress formulation.

The main input parameters of SHEETMAT can be categorized into three groups: elasticity, criterion of yielding and rule of hardening. These input parameters (see the following table) reference keywords that will be described in the following sections. Furthermore, strain-rate dependence is considered and finally, the use of the forming limit diagram is treated in view of postprocessing purposes.

TYPE	ELASTICITY	YIELDING	HARDENING

ISOTROPIC*	ELASTIC=ISO:
	Exx
	NUxy (or Gxy)
		TYPEYLD=ISO:
		RO=R45=R90=1.0
			TYPEHRD=ISO

NORMAL ANISOTROPIC	ELASTIC=PLANISO:
	Exx (or Eyy)
	Ezz
	Nuxy (or Gxy)
	NUxz (or NUyz)
	Gxz (or Gyz)
		TYPEYLD=NORMANI:
		RO=R45=R90
			TYPEHRD=NORMANI

PLANAR ANISOTROPIC	not available	TYPEYLD=PLANANI: R0≠R45≠R90	not available

*Default

Elasticity

SHEETMAT includes two models of elastic behavior: fully isotropic and planar isotropic elasticity. Both forms of elasticity are most easily defined by giving the strain-stress relation expressed in so-called engineering constants for orthotropic materials:

$$\begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{Bmatrix} = \begin{bmatrix} 1/\mathrm{E}_{xx} & -\nu_{xy}/\mathrm{E}_{xx} & -\nu_{xz}/\mathrm{E}_{xx} & 0 & 0 & 0 \\ -\nu_{xy}/\mathrm{E}_{xx} & 1/\mathrm{E}_{yy} & -\nu_{yz}/\mathrm{E}_{yy} & 0 & 0 & 0 \\ -\nu_{xz}/\mathrm{E}_{xx} & -\nu_{yz}/\mathrm{E}_{yy} & 1/\mathrm{E}_{zz} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/\mathrm{G}_{xy} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/\mathrm{G}_{yz} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/\mathrm{G}_{xz} \end{bmatrix} \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xz} \\ \sigma_{yz} \\ \sigma_{xz} \end{Bmatrix}$$

The isotropic case is the simplest form of linear elasticity for which only the Young's modulus $(E_{xx} = E_{yy} = E_{zz})$ and Poisson's ratio $(\nu_{xy} = \nu_{yz} = \nu_{xz})$ or shear modulus $(G_{xy} = G_{yz} = G_{xz})$ must be defined. Planar isotropic material behavior is equivalent to transversely isotropic material behavior, which means that the through-the-thickness (elastic) properties may differ from the in-plane isotropic (elastic) properties. The values of E_{xx} (or E_{yy}), E_{zz}, ν_{xy} (or G_{xy}), ν_{xz} (or ν_{yz}) and G_{xz} (or G_{yz}) are required to define a planar isotropic material.

The engineering constants must be specified with respect to the rolling direction of the material which is defined by a local material coordinate system. This coordinate system may differ from the local element coordinate system and may be defined via XMAT, YMAT, and ZMAT on the SHEETMAT entry (or by specifying THETA on the CQUAD4/CTRIA3 entry).

As a result of the rolling process, the plastic properties normal to the sheet are likely to be different from the in-plane properties, i.e., normal anisotropy. In addition, the properties may depend on the in-plane orientation with respect to the rolling direction, i.e., planar anisotropy. The Krieg material model can represent normal anisotropy in both yielding and hardening. Planar anisotropy is confined to yielding.

Yielding Criteria

The plasticity model of Krieg uses a standard Hill yield surface model. Three possibilities are provided: isotropic yielding, normal anisotropic, and planar anisotropic yielding. Isotropic yielding is equivalent to von Mises yielding. It is defined by giving the value of uniaxial yield stress as a function of uniaxial (effective) plastic strain (and effective plastic strain rate). The yield stress can be expressed as:

$$\sigma_y = [a + b(\bar{\varepsilon}^p + c)^n][1 + k(\bar{d}^p)^m]$$

where a = stress constant

b = hardening parameter

c = strain offset

n = strain-hardening exponent

k = strain-rate sensitivity constant

m = strain-rate exponent

$\bar{\varepsilon}^p$ = effective plastic strain

$\bar{d}^p$ = effective plastic strain rate

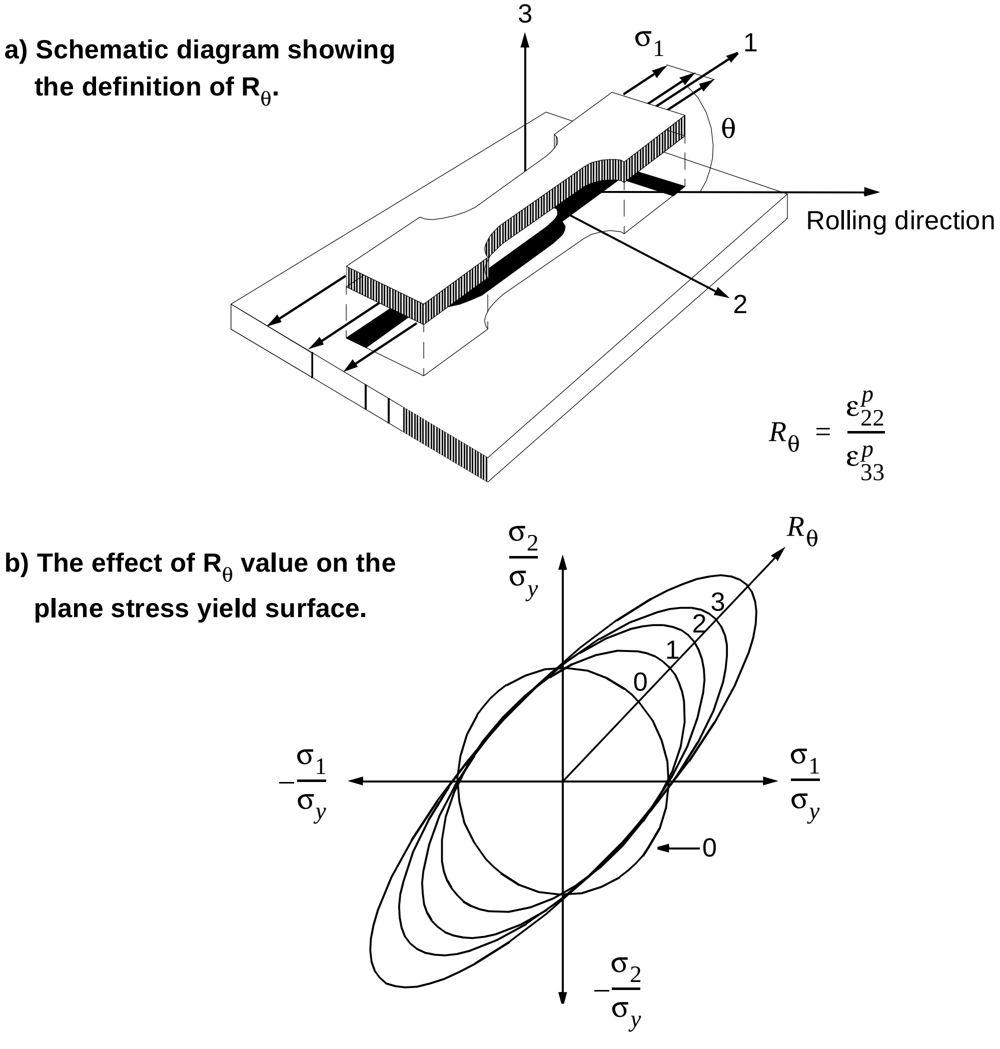
The power-law coefficients (*a*, *b*, *c*, *n*, *k*, *m*) are usually determined by a least squares fit of experimental true stress-strain data, obtained from uniaxial tensile tests.

For anisotropic materials, the coefficients can be different for the (uniaxial) out-of-plane direction, the rolling, and transverse rolling direction, as well as at $45°$ to the rolling direction.

The representation of normal or planar anisotropy is achieved by defining a single power-law yield function. The different stress-plastic strain curves are recovered from the power-law yield function by means of multiplication by constants. The yielding directionality is controlled via the yield matrix Q_{ij} in the yield function ϕ:

$$\phi = \sigma_i Q_{ij} \sigma_j - \sigma_y^{\;2}$$

The coefficients of the yield matrix σ_{ij} are governed by the anisotropic yield parameters R_0, R_{45}, and R_{90} which are the so-called Lankford coefficients. R_0 represents the width-to-thickness plastic strain ratio measured from a uniaxial test in rolling direction. R_{90} represents the ratio measured from a uniaxial test in transverse rolling direction. R_{45} represents the ratio measured from a test at 45 degrees to the rolling direction (see Figure 2-7a). The *R* values can be entered on the SHEETMAT entry.



a) Schematic diagram showing the definition of R$_\theta$.

3
σ_1
1
θ
Rolling direction
2
$R_\theta = \frac{\varepsilon_{22}^p}{\varepsilon_{33}^p}$

b) The effect of R$_\theta$ value on the plane stress yield surface.

$\frac{\sigma_2}{\sigma_y}$
R_θ
3
2
1
0
$-\frac{\sigma_1}{\sigma_y}$
$\frac{\sigma_1}{\sigma_y}$
0
$-\frac{\sigma_2}{\sigma_y}$

Figure 2-7. Anisotropic Plasticity.

For fully isotropic material, the in-plane and out-of-plane (i.e., normal) material properties are the same which means that the width plastic strain must be equal to the through-the-thickness plastic strain, implying $R_0 = R_{45} = R_{90} = 1$. These values are the defaults on the SHEETMAT entry.

A material is called normal anisotropic when the material is in-plane isotropic, but has different out-of-plane properties compared to the in-plane properties. The R value ($R_0 = R_{45} = R_{90}$) is not equal to one. Consequently, only the R_0 value is required on the SHEETMAT entry.

The SHEETMAT definition also allows (planar) anisotropic yielding behavior to be modeled. This implies that the R value depends on the in-plane orientation with respect to the rolling direction. Therefore, you must specify all of the values for R_0, R_{45}, and R_{90} individually.

The effect of the R value on the yield surface is schematically shown in Figure 2-7b.

Hardening Rules

The work-hardening rule defines the way the yield surface changes with plastic straining. Besides perfect plasticity—where yield stress does not change with plastic strain—two possibilities are provided with SHEETMAT: isotropic hardening and normal anisotropic hardening.

Isotropic hardening (default for SHEETMAT) means that the yield surface changes uniformly in all directions so that the yield stress increases in all stress directions as plastic straining occurs.

SHEETMAT also allows normal anisotropic hardening, which means the growth of the yield surface may require more plastic strain in thickness direction than in other directions. This distinct hardening in thickness direction can be controlled by a hardening matrix in which the coefficients are also given by the Lankford coefficients.

Strain-Rate Dependence

In some metals, the rate of stretching affects the mechanical properties; the material yields at a higher effective stress state for higher imposed strain rates. The yield stress for a plastic process is also higher. This effect can be accounted for in the power-law yield function by defining the strain-rate sensitivity constant k, and the strain-rate exponent m. By default, strain-rate dependence is not taken into account.

Forming Limit Diagram

A forming limit diagram (FLD) can be input on the SHEETMAT entry to evaluate actual and potential problems in sheet-metal forming processes. The diagram forms the lower bound of experimental strains corresponding to regions affected by necking. This implies strains below the limit curve are acceptable.

The forming limit diagram is defined on SHEETMAT to be composed of two polynomial functions (see Figure 2-8.). You can supply the coefficients representing these functions for the material under consideration.

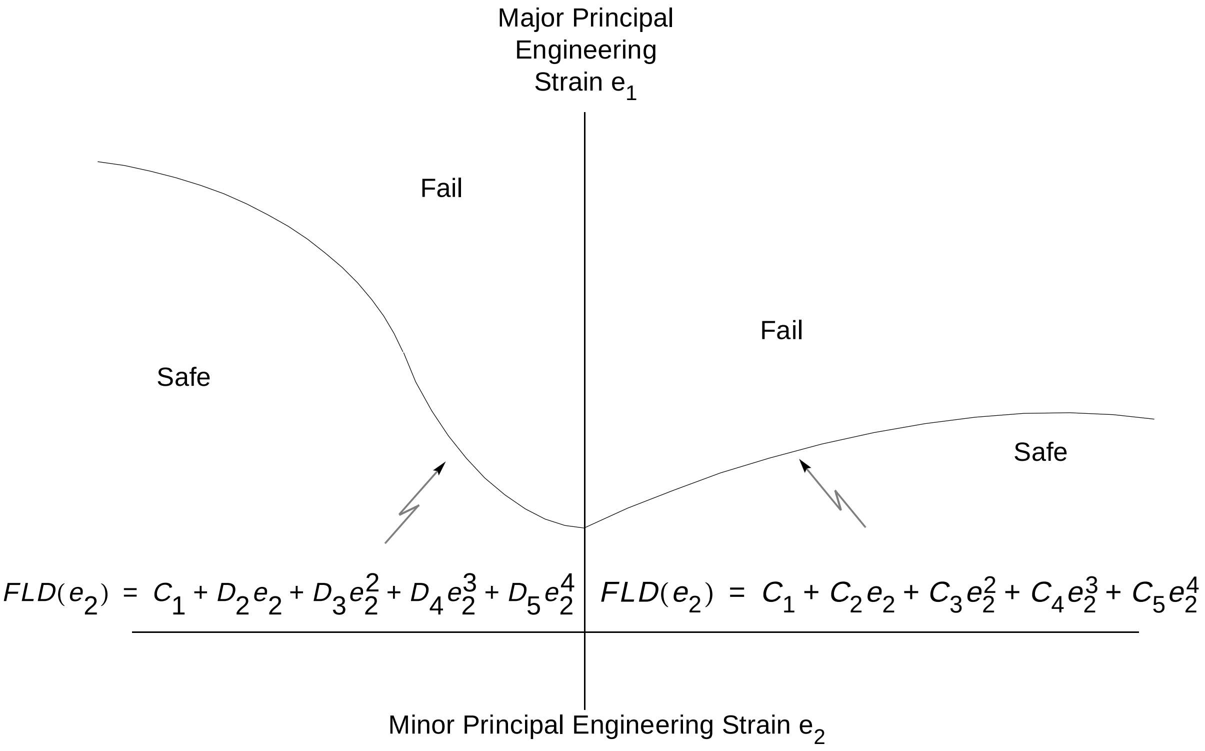

Major Principal Engineering Strain e_1
Fail
Fail
Safe
Safe
$FLD(e_2) = C_1 + D_2 e_2 + D_3 e_2^2 + D_4 e_2^3 + D_5 e_2^4$
$FLD(e_2) = C_1 + C_2 e_2 + C_3 e_2^2 + C_4 e_2^3 + C_5 e_2^4$
Minor Principal Engineering Strain e_2

Figure 2-8. Forming Limit Diagram Represented by Two Polynomials.

Two different ways of postprocessing are possible. First, a contour plot of the Forming Limit Parameter (FLP) can be made. The FLP denotes the ratio of predicted strain and allowable strain. In equation form:

$$FLP = \frac{e_1}{FLD(e_2)}$$

where e_1 and e_2 are respectively major and minor principal engineering strain at the integration point.

The parameter is accessible via the output variable FLP# (where # equals the integration layer number). The FLP contour plot shows an overall view of regions where necking (followed by failure) possibly occurs. Failure is indicated when FLP is greater than or equal to one.

The second method of visualization is to use the minor and major principal strains (output variables EPSMN# and EPSMX#) and plotting these strains for any particular element versus the experimental forming limit diagram. By convention, these strains are output as true strain. The forming limit diagram is usually plotted against engineering strains. As a result, the output variables EPSMN# and EPSMX# must be converted to engineering strains.

2.5.3.7 DYMAT14 – Soil and Crushable Foam

This model is for materials exhibiting compressible plasticity; that is, their behavior is pressure dependent. It can be used to model aspects of the behavior of a wide range of materials that contain voids and crush or compact under pressure. Examples include soils, foams, concrete, metallic honeycombs, and wood.

The material model is based on that developed by Krieg and Key. It uses isotropic plasticity theory and the response of the material to deviatoric (shear) loading and hydrostatic (pressure) loading is completely uncoupled.

Deviatoric Behavior

When the YSURF option is used on the DYMAT14 entry, the yield surface in principal stress space is a surface of revolution centered about the hydrostatic pressure line. It is defined by $\Phi_s(J_2, p) = 0$, where

$$\Phi_S = (J_2, p) = J_2 - (B_0 + B_1\, p + B_2\, p^2)$$

where p is the pressure J_2 is the second invariant of the stress deviation tensor:

$$J_2 = \frac{1}{2} s_{ij}\, s_{ij}$$

where s_{ij} are the deviatoric stresses. J_2 can also be defined in terms of the principal stresses σ_{ij}:

$$J_2 = \frac{1}{6}\left[(\sigma_{11} - \sigma_{22})^2 + (\sigma_{22} - \sigma_{33})^2 + (\sigma_{33} - \sigma_{11})^2\right] + \sigma_{12}^2 + \sigma_{22}^2 + \sigma_{31}^2$$

The coefficients B_0, B_1 and B_2 can be related to the user-defined constants A_0, A_1 and A_2. This relation depends on the YSTYP field on the DYMAT14 entry. If the YSTYP field is DYTRAN, then

$B_0 = A_0$

$B_1 = A_1$

$B_2 = A_2$

Thus, if A_1 and A_2 are zero, the yield surface is cylindrical. If only A_2 is zero, the surface will be conical and otherwise the surface will have a shape as shown in Figure 2-9.

If the YSTYP-field is DYNA, then

$$B_0 = \frac{1}{3}A_0^2$$

$$B_1 = \frac{2}{3}A_0A_1$$

$$B_2 = \frac{1}{3}A_1^2$$

and A_2 is ignored.

In this case, the yield surface is cylindrical when A_1 is zero and it has a shape as shown in Figure 2-9 when A_1 is nonzero.

For both options of YSTYP the yield stress σ_y can be expressed in terms of the coefficients A_0, A_1, and A_2. The yield stress is defined as

$$\sigma_y = \sqrt{\frac{3}{2}J_2}\text{, where } J_2 = \{J_2 | \Phi_S(J_2, p) = 0\}$$

Thus,

$$\sigma_Y = \sqrt{3(B_0 + B_1p + B_2p^2)}$$

$$= \begin{cases} \sqrt{3(A_0 + A_1p + A_2p^2)} & \text{if YSTYP = DYTRAN} \\ A_0 + A_1p, & \text{if YSTYP = DYNA} \end{cases}$$

The cut-off pressure can be supplied by the user but should not have a positive value. When the cut-off pressure is left blank, MSC.Dytran calculates this value as the intersection point of the yield surface with the hydrostat. When only B_0 is nonzero (and therefore only A_0 is nonzero), the cut-off pressure is calculated as -100 times the bulk modulus defined on the DYMAT14 entry.

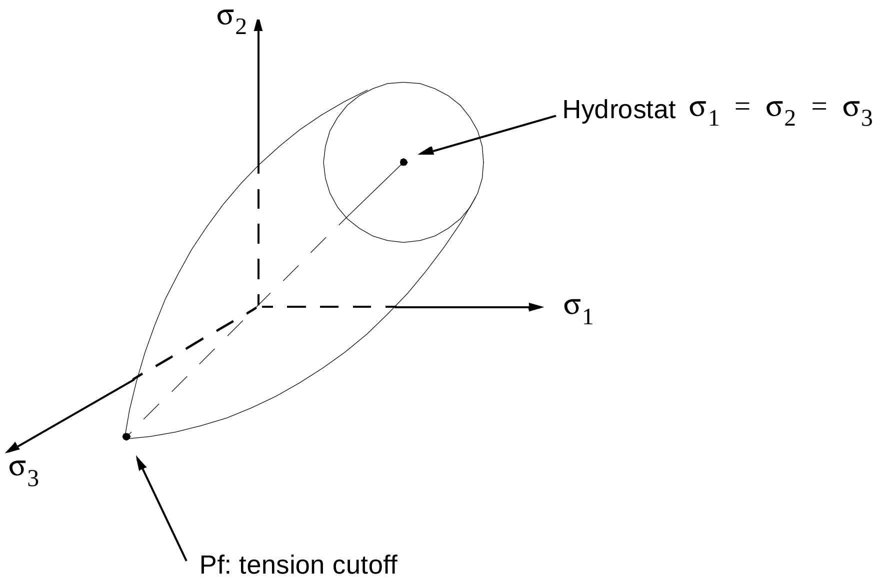


Figure 2-9. Yield Surface with Hydrostat.

The open end of the cylinder, cone, or paraboloid points into compression and is capped by a plane that is normal to the hydrostat. There is no strain hardening on the yield surface, so the relationship between deviatoric stress σ' and deviatoric strain ε' is elastic perfectly plastic as shown in Figure 2-10.

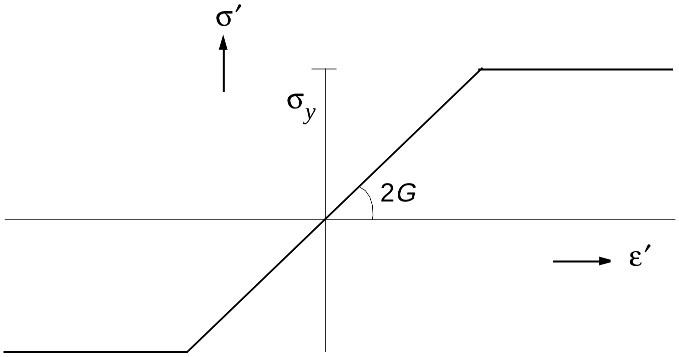


Figure 2-10. Stress-Strain Curve.

In other words, in case of yielding, the yield surface remains stationary as yielding occurs. The elastic behavior is governed by the shear modulus *G*.

Hydrostatic Behavior

The hydrostatic component of the loading causes volumetric yielding. This means that the cap on the open end of the yield surface moves along the hydrostat as volumetric yielding occurs. The relationship between hydrostatic pressure and volumetric strain is defined using a TABLED1 entry and can be of any shape.

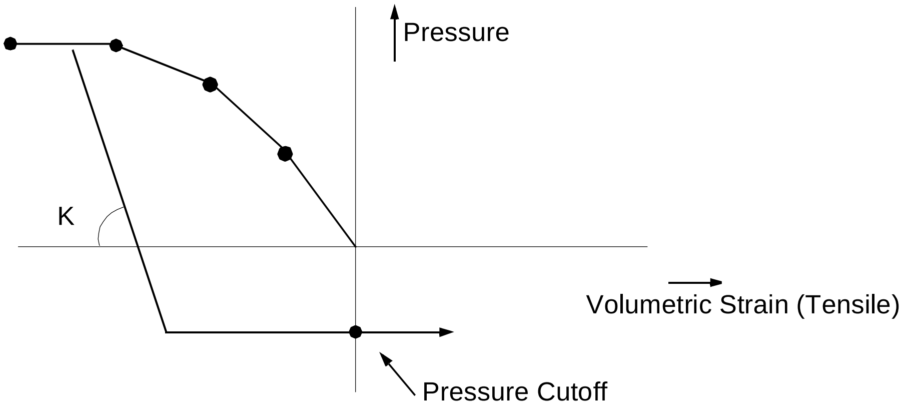


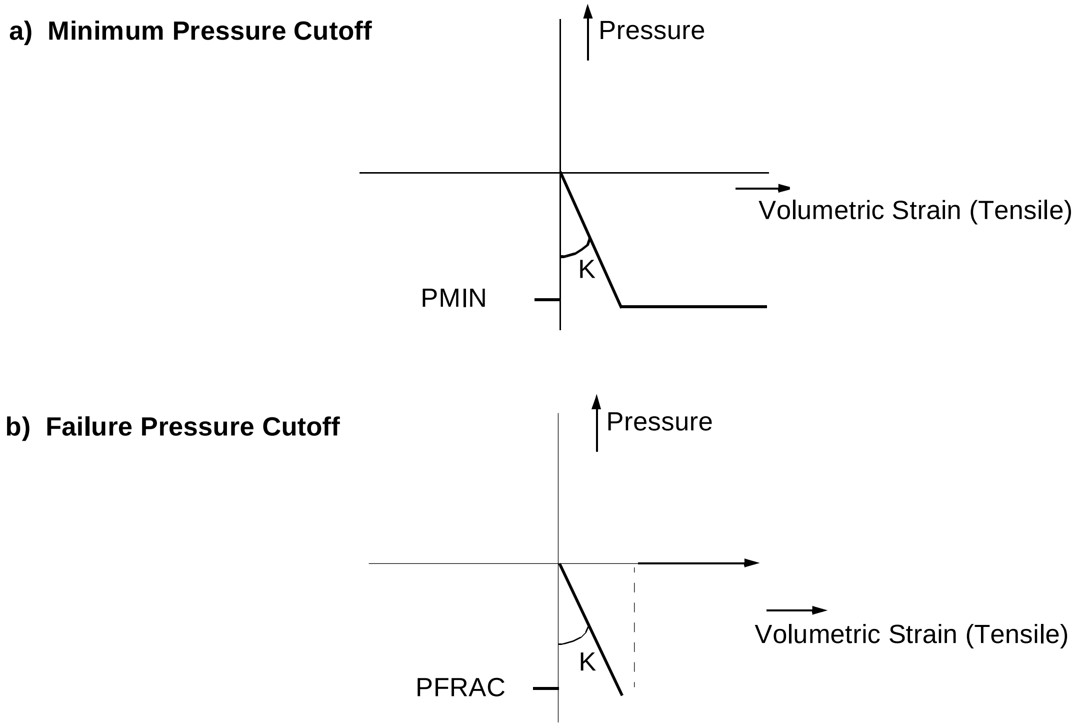
Figure 2-11. Volumetric Yielding.

The curve can be defined in terms of the crush factor or volumetric strain. The crush factor is defined as 1-*V/V*0 where *V* is the current volume and *V*0 the initial volume. It is a number between 0 and 1 where 0 indicates no crush and 1 indicates that the material is completely crushed and has zero volume. The crush factor, in fact, is minus the engineering strain. The volumetric strain is defined as

$$\int_{to}^{t} \frac{dV}{V} \text{ or } ln(V/V0)$$

The volumetric strain must always be negative.

The material unloads elastically from any point on the curve with a user-defined bulk modulus *K*. You can also specify a minimum pressure *(PMIN)* or a failure pressure *(PFRAC)*. In the first case, since pressure is positive in compression, this corresponds to a tensile cutoff for the material. The pressure cannot fall below the minimum value. If the initial loading is tensile, the material will behave elastically with a bulk modulus *K* until the minimum pressure is reached. Further tensile straining produces no increase in pressure. In the second case, you specify a failure pressure rather than a minimum pressure. If the pressure falls below the failure pressure, the element fails and cannot carry tensile loading for the remainder of the analysis. It can still carry compressive loading.


a) Minimum Pressure Cutoff

b) Failure Pressure Cutoff

Figure 2-12. Pressure as Function of Volumetric Strain.

Under compressive loading the material follows the strain-pressure curve:

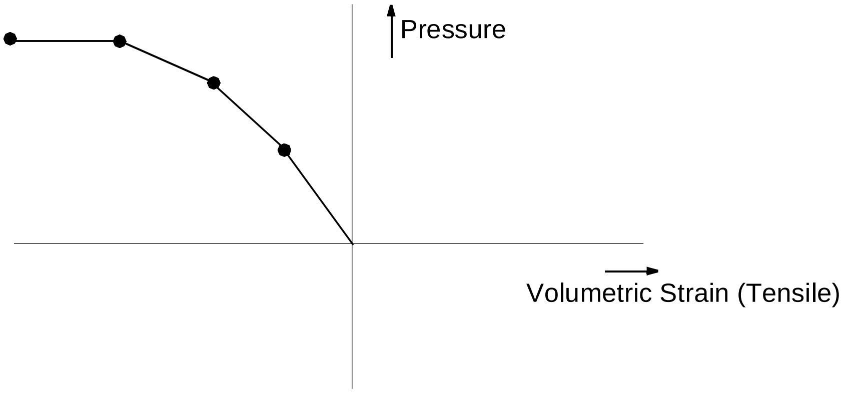


Figure 2-13. Pressure as Function of Volumetric Strain in Compression.

If the material then unloads, it does so elastically until the minimum (or failure) pressure is reached, after which further tensile straining does not produce any increase in pressure.

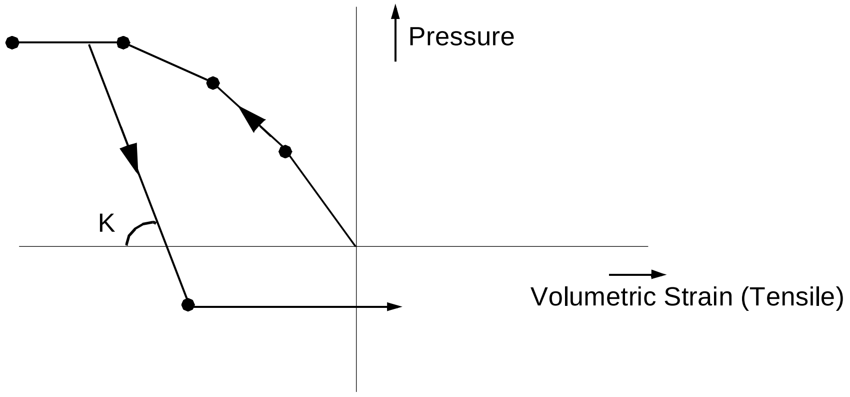


Figure 2-14. Pressure as Function of Volumetric Strain in Compression and Expansion.

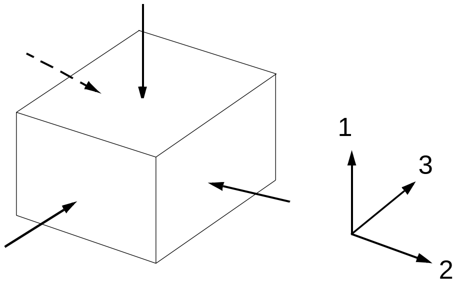
Determination of Yield Curve

The remainder of this section describes the experiments that can be performed to obtain the pressure-strain curve and values for A_0, A_1, and A_2 for the YSURF option.

The most accurate way is to perform a volumetric test and a uniaxial compression test. If a volumetric test is not available, a uniaxial compression test can give a good approximation.

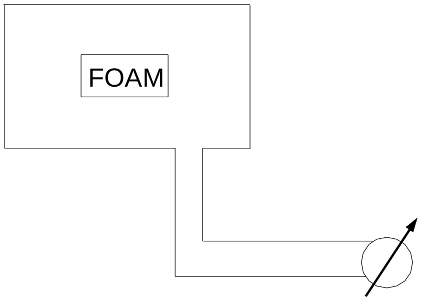
1. Volumetric test.

   All sides are equally compressed.



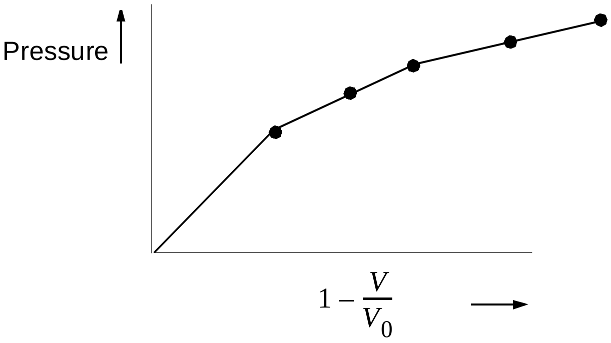
$$\lambda_1 = \lambda_2 = \lambda_3 = \lambda_v$$

The volumetric test can be performed by exerting pressure on the foam via a fluid.

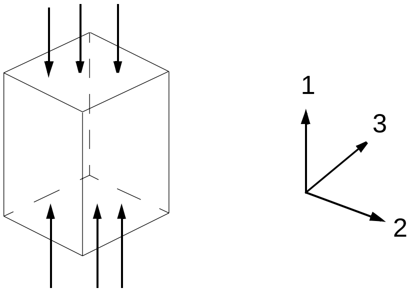


The volumetric change is equal to additional fluid entering the chamber.

The test results directly in a pressure-crush curve:



2. Uniaxial compression test



The stress in the 1-direction t_{11} can be measured as a function of $\frac{V}{V_0}$. Note that the engineering stress is equivalent to the true stress since Poisson effects are typically small for crushable foams. As for the strains holds:

$$e_{11} \approx ln\frac{V}{V_0}, e_{22} \approx 0, e_{33} \approx 0$$

During crushing, the stresses are computed by the following equations:

MSC/DYNA Method

$$t_{11} = -\frac{2}{3}A_0 + p\left(-\frac{2}{3}A_1 - 1\right)$$

$$t_{22} = \frac{1}{3}A_0 + p\left(\frac{1}{3}A_1 - 1\right)$$

$$t_{33} = \frac{1}{3}A_0 + p\left(\frac{1}{3}A_1 - 1\right)$$

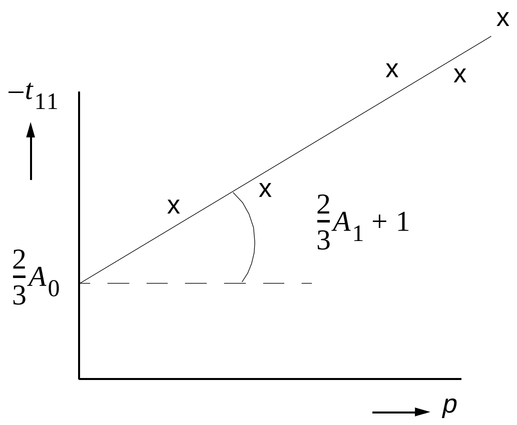
MSC.Dytran Method

$$t_{11} = -\frac{2}{3}\sqrt{3(A_0 + A_1 p + A_2 p^2)} - p$$

$$t_{22} = \frac{1}{3}\sqrt{3(A_0 + A_1 p + A_2 p^2)} - p$$

$$t_{33} = \frac{1}{3}\sqrt{3(A_0 + A_1 p + A_2 p^2)} - p$$

Therefore, when the volumetric test can be carried out, you obtain the $p\left(\frac{V}{V_0}\right)$ relation. From the uniaxial test, we find $t_{11}\left(\frac{V}{V_0}\right)$. For the DYNA option the constants A_0 and A_1 can then be fitted from the $p\left(\frac{V}{V_0}\right) - t_{11}\left(\frac{V}{V_0}\right)$ curve:



For the DYTRAN option, the constants A_0, A_1, and A_2 must be fitted from a $p\left(\frac{V}{V_0}\right) - t_{11}\left(\frac{V}{V_0}\right)$ curve, which is not a straight line.

When the volumetric test is not available, the following approximation can be made:

$$t_{22} = t_{33} = 0$$

So that the pressure becomes:

$$p = -\frac{1}{3}(t_{11} + t_{22} + t_{33}) = -\frac{1}{3}t_{11}$$

When t_{11} can be measured from a uniaxial test the pressure curve is determined. The constants A_0, A_1, and A_2 are determined such that the above equations hold.

MSC/DYNA:	$A_0 = 0.0$
	$A_1 = 3.0$
MSC.Dytran:	$A_0 = 0.0$
	$A_1 = 0.0$
	$A_2 = 3.0$

2.5.3.8 DYMAT24 – Piecewise Linear Plasticity

This model can be used for isotropic, elastoplastic materials where the stress-strain characteristic is too complex to be modeled by a bilinear representation. You can specify a table containing a piecewise linear approximation of the stress-strain curve for the material.

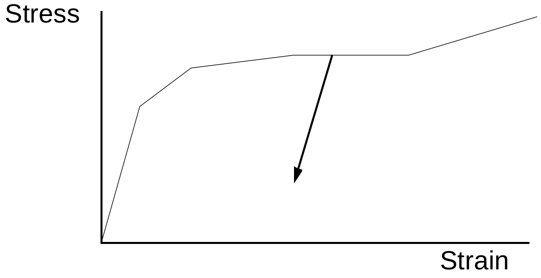


Figure 2-15. Stress-Strain Curve.

Every iteration the stress σ is determined from the current equivalent strain ε by interpolating from the stress-strain table:

$$\sigma = [(\sigma_i - \sigma_{i-1})(\varepsilon - \varepsilon_{i-1})/(\varepsilon_i - \varepsilon_{i-1})] + \sigma_{i-1}$$

where σ_i and ε_i are the points in the table.

The stress-strain characteristic used internally in MSC.Dytran is defined in terms of true stress and equivalent plastic strain. However, for convenience, the stress-strain characteristic can be input in any of the following ways:

- True stress/true strain.
- Engineering stress/engineering strain.
- True stress/plastic strain.
- True stress/plastic modulus.

Alternatively, you can specify the hardening modulus and yield stress, in which case a bilinear representation is used:

$$\sigma = \sigma_y + E_l \varepsilon_P$$

where ε_P is the equivalent plastic strain. Hardening is assumed to be isotropic, the yield surface expands as the material yields.

This material can be used with all solid, shell (except for membranes), and Hughes-Liu beam elements. Strain-rate sensitivity and failure can be included for all of these elements. Strain-rate sensitivity can be defined in two ways:

1. You can specify a table giving the variation of a scale factor S with strain-rate $\dot{\varepsilon}$. The scale factor is multiplied by the stress found from the stress-strain characteristic to give the actual stress. The failure criterion is based on plastic strain. When the plastic strain exceeds the specified value, the element fails. All stresses are set to zero, and the element can carry no load.
2. You can specify the constants D and P in Cowper-Symonds rate enhancement formula:

$$\frac{\sigma_d}{\sigma_y} = 1 + \left(\frac{\dot{\varepsilon}}{D}\right)^{1/P}$$

where σ_d is the dynamic stress, σ_y is the static yield stress, and $\dot{\varepsilon}$ is the equivalent strain rate.

2.5.3.9 DYMAT26 – Crushable Orthotropic Material

The DYMAT26 entry defines the properties of an orthotropic, crushable material model. It can only be used with Lagrangian solid elements.

The input required for the material consists of two parts: data for the fully compacted state and data for the crushing behavior. For the fully compacted material, the input consists of the density, the elastic modulus for the fully compacted material, Poisson’s ratio for the fully compacted material, the yield stress for the fully compacted material, and the relative volume at which the material is fully compacted.

The behavior during crushing is orthotropic and is characterized by uncoupled strain behavior when the initial Poisson’s ratios are not supplied. During crushing, the elastic moduli (and the Poisson’s ratios only if they are supplied) vary from their initial values to the fully compacted values. This variation is linear with relative volume.

When the material is fully compacted, the behavior is elastic perfectly plastic with isotropic plasticity.

The load tables define the magnitude of the average stress in a given direction as the material’s relative volume changes. At defining the curves, care should be taken that the extrapolated values do not lead to negative yield stresses.

2.5.3.10 RUBBER1 – Mooney-Rivlin Rubber Model

The RUBBER1 entry defines the properties of a Mooney-Rivlin rubber model. It can only be used with Lagrangian solid elements.

The constitutive behavior of this material is defined as a total stress-total strain relationship. Rather than by Hooke’s law, the nonlinear elastic material response is formulated by a strain energy density function accounting for large strain components. The strain energy density function is defined according to the Mooney-Rivlin model:

$$W(I_1, I_2, I_3) = A(I_1 - 3) + B(I_2 - 3) + C\left(\frac{1}{I_3{}^2} - 1\right) + D(I_3 - 1)^2$$

The constants *A* and *B*, and Poisson's ratio ν are the input parameters for the model. The constants *C* and *D* are related to the input parameters as:

$$C = \frac{1}{2}A + B$$

$$D = \frac{A(5\nu - 2) + B(11\nu - 5)}{2(1 - 2\nu)}$$

I_1, I_2, and I_3 are strain invariants in terms of stretches. Stretches are defined as:

$$\frac{\delta x_i}{\delta X_j} = \lambda_{ij}$$

where x_i and X_j are the coordinates of the deformed and original geometry, respectively.

For rubber-like materials the shear modulus *G* is much less than the bulk modulus *K*. In this case, $G = 2(A + B)$.

The stresses are computed as:

$$\tau = (\det F)^{-1} \cdot F \cdot \sigma \cdot F^T$$

where σ is the second Piola-Kirchhoff stress tensor:

$$\sigma = 2\frac{\partial W}{\partial C}$$

The Cauchy-Green stretch tensor *C* is defined as:

$$C = F^T F$$

where *F* is the deformation gradient tensor

$$F = \frac{\delta x}{\delta X}$$

In terms of principal stretches $(\lambda_1, \lambda_2, \lambda_3)$, i.e., the stretches in the coordinate system where all shear strains and shear stresses vanish, the expressions for the deformation gradient tensor *F*, and the Cauchy-Green stretch tensor *C* simplify to

$$F = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}, C = \begin{bmatrix} \lambda_1^2 & 0 & 0 \\ 0 & \lambda_2^2 & 0 \\ 0 & 0 & \lambda_3^2 \end{bmatrix}$$

The strain invariants I_1, I_2, and I_3 read

$$I_1 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2$$

$$I_2 = \lambda_1^2\lambda_2^2 + \lambda_2^2\lambda_3^2 + \lambda_3^2\lambda_1^2$$

$$I_3 = \lambda_1^2\lambda_2^2\lambda_3^2$$

The stresses can be written as

$$J\tau_{ii} = \lambda_i\frac{\partial W}{\partial \lambda_i}$$

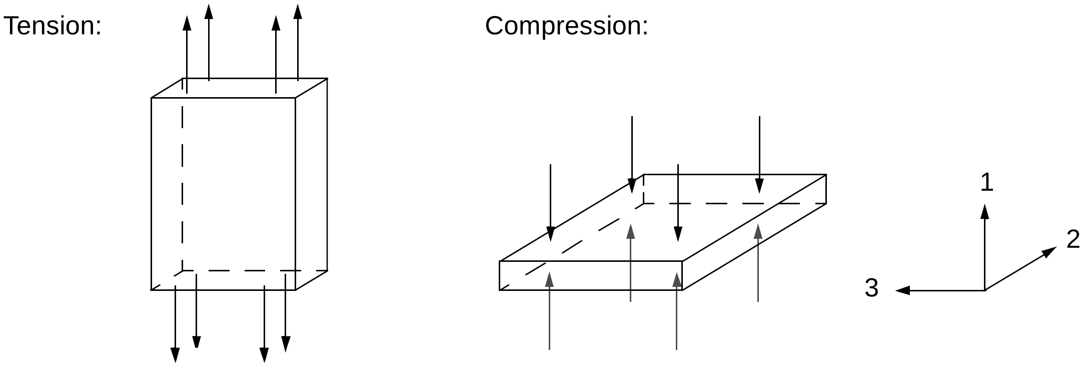
where $J = \lambda_1\lambda_2\lambda_3 = \dfrac{dV}{dV_0}$

Determination of Rubber Material Parameters

The remainder of this section describes the experiments that can be performed to obtain the material parameters as they appear in the strain-energy density function. The most commonly performed tests are uniaxial, planar (shear), and volumetric tests.

A planar or shear test can be used to determine the shear modulus $G(= 2(A + B))$. Tensile or compression tests provide the same information. Since rubber is a nearly incompressible material, the volume is assumed to be constant. Therefore, the principal stretches λ_1, λ_2, and λ_3 can be written as

$$\lambda_1 = \lambda_s;\ \lambda_2 = 1;\ \lambda_3 = \frac{1}{\lambda_s}$$



The stresses in the 1- and 3-direction are given by

$$\tau_{11} = \lambda_1 \frac{\partial W}{\partial \lambda_1} = \lambda_1 \left[\frac{\partial W}{\partial I_1} \frac{\partial I_1}{\partial \lambda_1} + \frac{\partial W}{\partial I_2} \frac{\partial I_2}{\partial \lambda_1} + \frac{\partial W}{\partial I_3} \frac{\partial I_3}{\partial \lambda_1} \right]$$

$$= 2(A + B)(\lambda_s^{\;2} - 1)$$

$$\tau_{33} = \lambda_3 \frac{\partial W}{\partial \lambda_3} = \lambda_3 \left[\frac{\partial W}{\partial I_1} \frac{\partial I_1}{\partial \lambda_3} + \frac{\partial W}{\partial I_2} \frac{\partial I_2}{\partial \lambda_3} + \frac{\partial W}{\partial I_3} \frac{\partial I_3}{\partial \lambda_3} \right]$$

$$= 2(A + B)\left(\frac{1}{\lambda_s^{\;2}} - 1 \right)$$

The corresponding forces per unit cross-sectional area then become

$$F_1 = \tau_{11} \frac{A_1}{A_{0_1}} = \frac{\tau_{11}}{\lambda_s} = 2(A + B)\left(\lambda_s - \frac{1}{\lambda_s} \right) = G\left(\lambda_s - \frac{1}{\lambda_s} \right)$$

$$F_3 = \tau_{33} \frac{A_3}{A_{0_3}} = \tau_{33} \lambda_s = 2(A + B)\left(\lambda_s - \frac{1}{\lambda_s} \right) = G\left(\lambda_s - \frac{1}{\lambda_s} \right)$$

where A_{0_1} and A_{0_3} are the original areas. A_1 and A_3 are given as

$$A_1 = \frac{1}{\lambda_s} A_{0_1}, A_3 = \lambda_s A_{0_3}$$

Fitting the measured force versus stretch curve with curve from the model, $F = G\left(\lambda_s - \frac{1}{\lambda_s}\right)$, the shear modulus can be estimated.

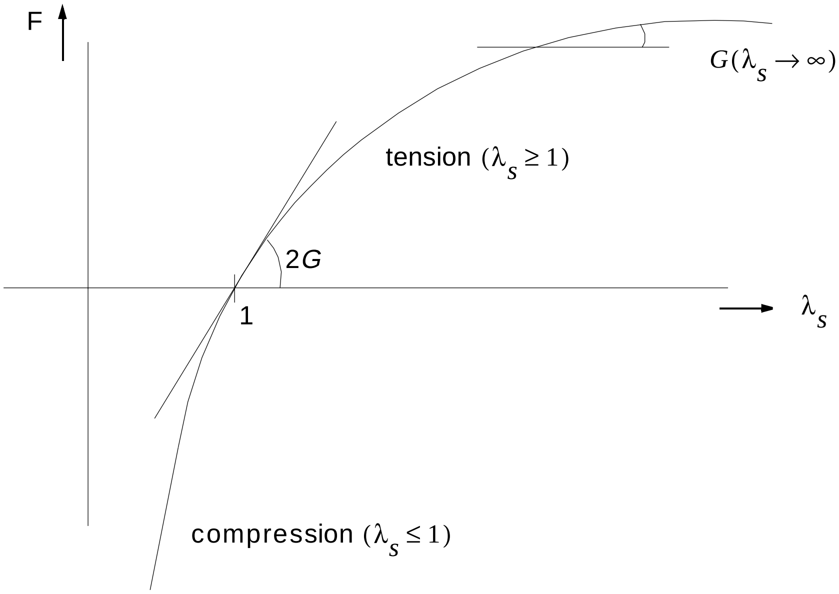
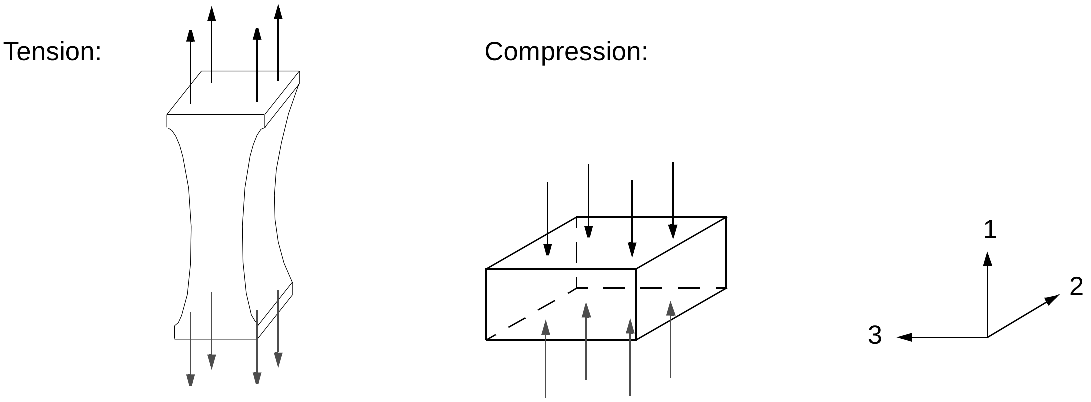

F
$G(\lambda_s \to \infty)$
tension $(\lambda_s \geq 1)$
$2G$
1
λ_s
compression $(\lambda_s \leq 1)$

Figure 2-16. Force Versus Stretch Diagram.

The experiment is usually performed with a thin, short, and wide rectangular strip of material fixed at its wide edges to rigid loading clamps that are moved apart.

The above test does not show how the constants *A* and *B* can be determined. For this purpose, a uniaxial test (elongation or compression) is recommended. No sides are clamped and one side (the 1-direction, see figure below) is either elongated or compressed. Since the material is nearly incompressible, the principal stretches are then given by

$$\lambda_1 = \lambda_\mu, \lambda_2 = \lambda_3 = \frac{1}{\sqrt{\lambda_\mu}}$$



The stress per unit deformed cross-sectional area in uniaxial direction is given by

$$\tau_{11} = \lambda_1 \frac{\partial W}{\partial \lambda_1} = 2A(\lambda_\mu{}^2 - 1) + 4B(\lambda_\mu - 1)$$

The corresponding force applied to a unit original cross-sectional area *F* then becomes

$$F = \tau_{11} \frac{A_1}{A_{0_1}} = \frac{\tau_{11}}{\lambda_\mu} = 2A\left(\lambda_\mu - \frac{1}{\lambda_\mu}\right) + 4B\left(1 - \frac{1}{\lambda_\mu}\right)$$

where A_{0_1} is the original area at time zero, and A_1 is given as

$$A_1 = \frac{1}{\lambda_\mu} A_{0_1}$$

Furthermore, since

$$\frac{dF}{d\lambda_\mu} = 2A + \frac{2A + 4B}{\lambda_\mu{}^2}, \frac{d^2F}{d\lambda_\mu{}^2} = -\frac{4A + 8B}{\lambda_\mu{}^3}$$

it follows that *F* is an increasing convex function due to the only relevant physical conditions

$$A > 0$$

$$A + 2B > 0$$

The analytical function is schematically shown as follows:

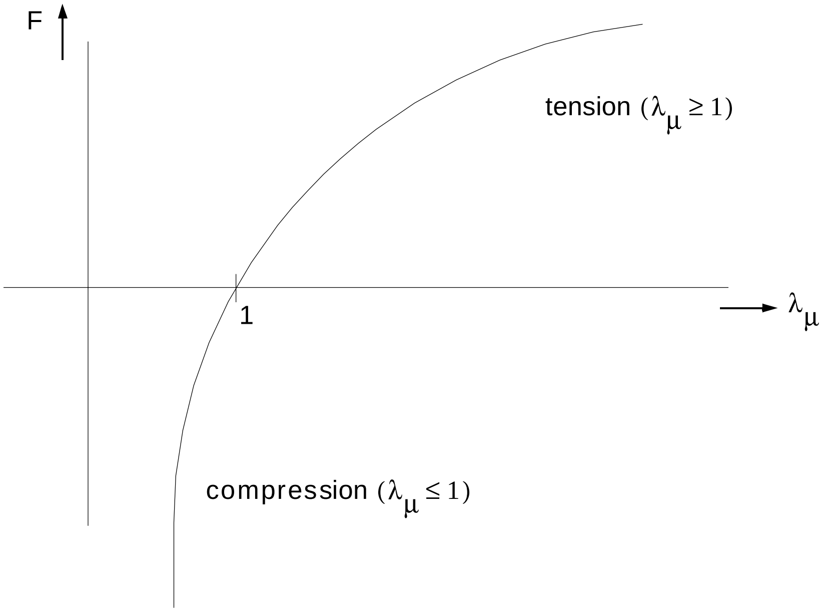


Figure 2-17. Force Versus Stretch Diagram.

Linear fitting can easily be achieved by applying the transformation

$$\tilde{F} = \frac{\lambda_\mu}{\lambda_\mu - 1}F$$

For the Mooney-Rivlin approach, the force $\tilde{F}$ then becomes

$$\tilde{F} = 2A\lambda_\mu + 2(A + 2B)$$

which is a straight line with slope 2*A* and the intersection point with $\lambda_\mu = 0$ axis equal to $2(A + 2B)$. It must be noted, however, that the transformation can only be applied to the measured force for intervals of λ_μ, where the measured force is an increasing convex function of the principal stretch λ_μ. A reasonable estimation interval for compression (λ_1, λ_2), and for tension (λ_3, λ_4) is indicated in the following figure.

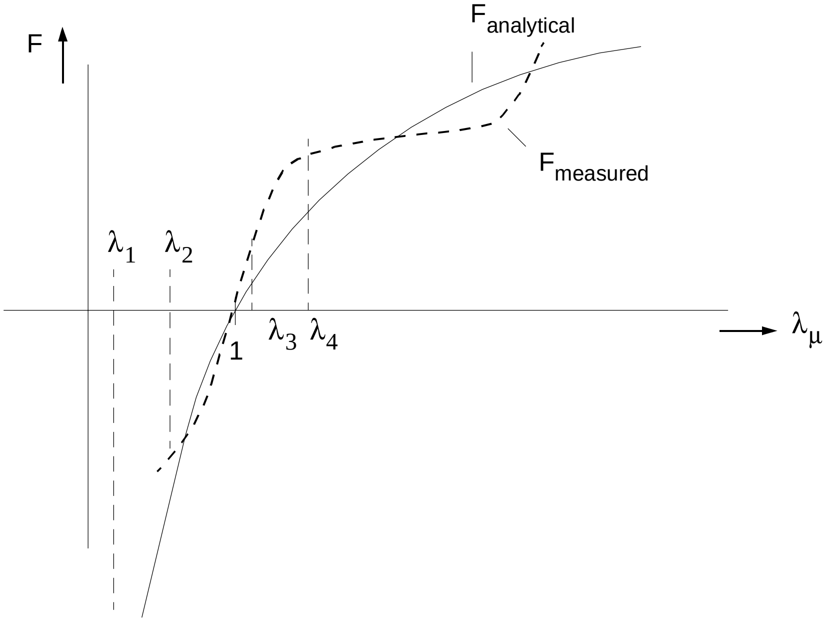
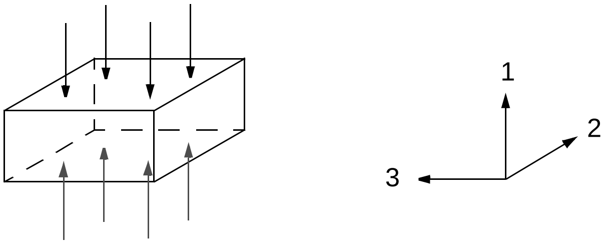


Figure 2-18. Force Versus Stretch Diagram.

The final test to be discussed is a volumetric compression test. It can be used to determine the bulk modulus *K*. The test can be performed in two ways.

1. Two sides clamped (the 2- and 3-directions), one side compressed (the 1-direction):

$$\lambda_1 = \lambda_v, \lambda_2 = \lambda_3 = 1.$$



Since the area A_1 does not change shape, the force applied to a unit cross-sectional area is equal to the stress

$$F = \tau_{11} = 2(A + 2B)\left(\lambda_v^2 - \frac{1}{\lambda_v^4}\right) + 4D(\lambda_v^4 - \lambda_v^2)$$

The constant D was defined as

$$D = \frac{A(5\nu - 2) + B(11\nu - 5)}{2(1 - 2\nu)}$$

and

$$\nu = \frac{3K - 2G}{6K + 2G}$$

the force can be written as

$$F = \frac{1}{2}K\lambda_v{}^2\,(\lambda_v{}^2 - 1) - \left(\frac{14}{3}A + \frac{32}{3}B\right)\lambda_v{}^4$$

$$+\left(\frac{20}{3}A + \frac{44}{3}B\right) + \lambda_v{}^2 - \frac{2(A + 2B)}{\lambda_v{}^4}$$

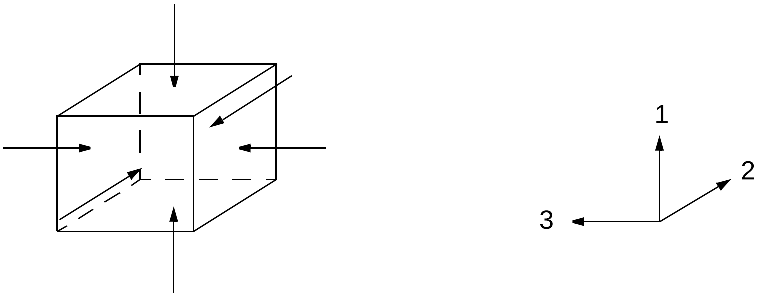
The material is assumed to be nearly incompressible; therefore, $\lambda_v = 1 - \varepsilon$ with $\varepsilon \ll 1$. Applying this assumption to the above equation and neglecting higher-order terms yields

$$F \approx -\left(K + \frac{4}{3}G\right)\varepsilon$$

As a result, the slope of the measured force curve around $\lambda_v = 1$ gives an estimate for $K + \frac{4}{3}G$. When G is known, using the expression for Poisson's ratio ν will result in a value for the input parameter ν.

2. All sides equally compressed:

$$\lambda_1 = \lambda_2 = \lambda_3 = \lambda_\nu$$



For this test, the pressure P can be measured. An analytical expression for the pressure according to the Mooney-Rivlin approach is

$$P = -\frac{1}{3}(\tau_{11} + \tau_{22} + \tau_{33}) = 2A\left(\frac{1}{\lambda_v^{15}} - \frac{1}{\lambda_v}\right) + 4B\left(\frac{1}{\lambda_v^{15}} - \lambda_v\right) - 4D\lambda_v^{3}(\lambda_v^{6} - 1)$$

Again, substitution of $\lambda_v = 1 - \varepsilon$ and neglecting higher-order terms of ε yields

$$P = 2(14A + 32B + 12D)\varepsilon = 3K\varepsilon$$

Therefore, the slope of the pressure curve at $\lambda_v = 1$ determines the bulk modulus K and Poisson's ratio ν.

2.5.3.11 FOAM1 – Foam Material (Polypropylene)

This model is used for an isotropic, crushable material model where Poisson's ratio is effectively zero.

The yield behavior is assumed to be completely determined by one stress-strain curve. In effect, this means that a uniaxial compression or tension test, a shear test, or a volumetric compression test all yield the same curves when stress (or pressure) is plotted versus strain (or relative volume $\frac{V}{V_0}$). The yield surface in three-dimensional space is a sphere in principal stresses

$$\tau_{11}^{2} + \tau_{22}^{2} + \tau_{33}^{2} = R_s^{2}$$

where the radius of the sphere R_s depends on the strains as follows

$$R_s = f(R_e)$$

with

$$\varepsilon_{11}^{2} + \varepsilon_{22}^{2} + \varepsilon_{33}^{2} = R_e^{2}$$

and *f* is the function supplied in the stress-strain table.

2.5.3.12 FOAM2 – Foam Material with Hysteresis

This model is used for an isotropic, crushable material model where Poisson's ratio is effectively zero and the unloading curve is a user-specified nonlinear hysteresis response stress-strain curve. The yield stress can also be made strain rate dependent.

The yield behavior is assumed to be completely determined by one stress-strain curve and a scale factor depending on the strain rate. In effect, this means that a uniaxial compression or tension test, a shear test, or a volumetric compression test all yield the same curves when stress (or pressure) is plotted versus strain (or relative volume $\frac{V}{V_0}$). The yield surface in three-dimensional space is a sphere in principal stresses

$$\tau_{11}{}^2 + \tau_{22}{}^2 + \tau_{33}{}^2 = R_s{}^2$$

where the radius of the sphere R_s depends on the strains and strain rates as follows

$$R_s = f_1(R_e)f_2(R_r)$$

with

$$\varepsilon_{11}{}^2 + \varepsilon_{22}{}^2 + \varepsilon_{33}{}^2 = R_e{}^2$$

and

$$\dot{\varepsilon}_{11}{}^2 + \dot{\varepsilon}_{22}{}^2 + \dot{\varepsilon}_{33}{}^2 = R_r{}^2$$

and f_1 is the function supplied in the stress-strain table and f_2 (if defined) is the function supplied in the factor-strain rate table.

The unloading curve is a nonlinear hysteresis response curve which is constructed such that the ratio of the dissipated energy (area between compressive loading and unloading curve) to total energy (area under the loading curve) is equal to the energy dissipation factor alpha.

In the case of linear unloading, MSC.Dytran automatically constructs a piecewise linear unloading curve, whose segments are parallel to the corresponding segments of the loading curve, except for the first and last segment which pass through the origin and point P (the point on the compression curve where the unloading starts), respectively. In the case of quadratic and exponential unloading the curves are respectively constructed from a parabolic function $a_0\varepsilon^2 + a_1\varepsilon$ and an exponential function $a_0(e^{-a_1\varepsilon} - 1)$. The coefficients are computed such that the unloading curve starts in point P, and the area between the loading and unloading curves satisfies the energy dissipation condition.

When the unloading reaches the origin, further unloading follows a straight line with a slope equal to the Young’s modulus until the tensile stress is reached. Either a minimum or a failure cut-off stress can be specified. In the first case the stress cannot fall below the minimum value, in the second case the stress is set to zero when the minimum is reached.

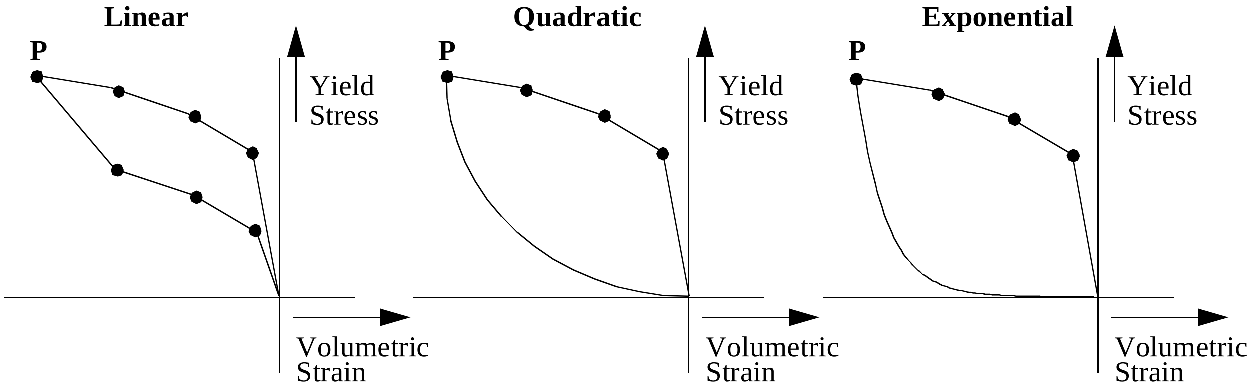


Figure 2-19. FOAM2 Unloading Curves.

2.5.4 Shear Models

The shear model is referenced from a DMAT entry. It defines the shear behavior of the material. At present, an elastic shear model is available with a constant shear modulus. For Lagrangian solids, a linear viscoelastic shear model is also available.

2.5.4.1 SHREL – Constant Modulus Shear Model

The SHREL entry defines a shear model with a constant shear modulus *G*. The model is referenced from a DMAT entry that defines the general material properties.

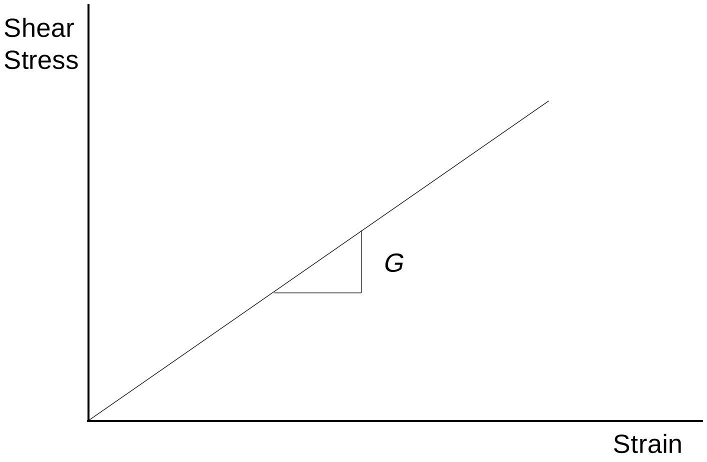


Figure 2-20. Elastic Shear as Function of Strain.

2.5.4.2 SHRLVE – Linear Viscoelastic Shear Model

The deviatoric stress components are given by

$$\sigma'_{ij}(t) = 2\int_0^t G(t-\tau)\frac{\partial\varepsilon'_{ij}(\tau)}{\partial\tau}\mathrm{d}\tau + 2G_\infty\varepsilon'_{ij}(t) + 2\eta_0\frac{\partial\varepsilon'_{ij}(t)}{\partial t} \tag{2-1}$$

where $G(t-\tau) = (G_0 - G_\infty)e^{-\beta(t-\tau)}$.

The variables in the above equations are as follows:

$\sigma'_{ij}(t)$ = deviatoric stress component

$\varepsilon'_{ij}(\tau)$ = deviatoric strain component

$G(t-\tau)$ = shear relaxation modulus

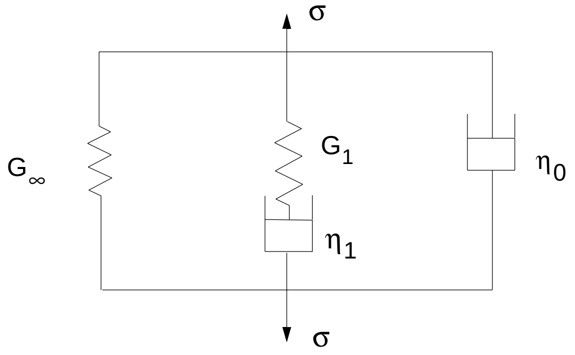
G_∞ = long term shear modulus

G_0 = short term shear modulus

η_0 = shear viscosity constant

β = decay coefficient

To understand the behavior of this material, it is instructive to look at a mechanical spring-damper model with a force/deflection behavior that is identical to the linear viscoelastic stress-strain behavior.



$$G_1 = G_0 - G_\infty$$
$$\beta = G_1 \S \eta_1$$

Figure 2-21. Generalized Maxwell Model.

The mechanical model is a Maxwell element in parallel with a single spring and a single damper. The stress-strain relation for this mechanical model is derived first. The strain $\varepsilon(t)$ is equal for all elementary parts in the generalized Maxwell model. The stress in each of the elementary bodies is given by

spring: $$\sigma_\infty(t) = 2G_\infty\varepsilon(t) \qquad \text{(a)}$$

dashpot: $$\sigma_0(t) = 2\eta_0\frac{d\varepsilon(t)}{dt} \qquad \text{(b)} \qquad \text{(2-2)}$$

Maxwell element: $$\frac{d\sigma_1(t)}{dt} + \frac{G_1}{\eta_1}\sigma_1(t) = 2G_1\frac{d\varepsilon(t)}{dt} \qquad \text{(c)}$$

Equation (2-2)(c) is easily derived by noting that for a Maxwell element the strain rate is the sum of the strain rates of the spring and the damper

$$\left(\frac{d\varepsilon(t)}{dt}\right)_{Maxwell} = \left(\frac{d\varepsilon(t)}{dt}\right)_{spring} + \left(\frac{d\varepsilon(t)}{dt}\right)_{damper}$$

$$\text{(2-3)}$$

$$= \frac{1}{2G_1}\left(\frac{d\sigma(t)}{dt}\right)_{spring} + \frac{1}{2\eta_1}(\sigma(t))_{damper}$$

Since the stresses in the spring and the damper are equal, Eq. (2-2)(c) can be found by reordering Eq. (2-3)

$$\text{Maxwell element: } \sigma_1(t) = 2\int_0^t G_1 e - \beta_1(t-\tau)\frac{d\varepsilon(\tau)}{d(\tau)}d\tau \tag{2-4}$$

where $\beta_1 = G_1/\eta_1$

Since the elementary parts are linked in parallel, the stress in the generalized Maxwell model can be found by adding the stresses as given by Eqs. (2-2)(a), (2-2)(b), and (2-2)(c)

$$\sigma(t) = 2\int_0^t G_1 e^{-\beta(t-\tau)}\frac{d\varepsilon^{(\tau)}}{d\tau}d\tau + G_\infty\varepsilon(t) + 2\eta_0\frac{d\varepsilon(\tau)}{d(t)} \tag{2-5}$$

Equation (2-5) is completely analogous to Eq. (2-1).

Based on Figure 2-21, two types of behavior can immediately be distinguished

I = Solid behavior: $G_\infty > 0$

II = Liquid behavior: $G_\infty = 0$

Fluid behavior occurs when the additional spring G_∞ is removed from the generalized Maxwell model.

By means of some examples, the material response is demonstrated. The examples show the stress response to enforced strain.

Example 1: Constant Strain Rate

$$\frac{d\varepsilon(t)}{dt} = \dot{\varepsilon}_0 ;\ \varepsilon(t) = \dot{\varepsilon}_0 t \tag{2-6}$$

Substituting Eq. (2-6) into Eq. (2-5) and solving for the integral gives

$$\sigma(t) = 2\left\{(\mu_1(1-e^{-\beta t})) + G_\infty + \mu_0\right\}\dot{\varepsilon}_0 \tag{2-7}$$

and

$$\frac{d\sigma(t)}{dt} = 2\left\{(G_1 e^{-\beta t}) + G_\infty\right\}\dot{\varepsilon}_0 \tag{2-8}$$

The above relations are sketched in Figure 2-22.

Due to the additional dashpot μ_0, an instantaneous response occurs for both the solid and the fluid.

The stress in the solid rises more strongly towards a constant stress rate. The fluid reaches a maximum value for its stress.

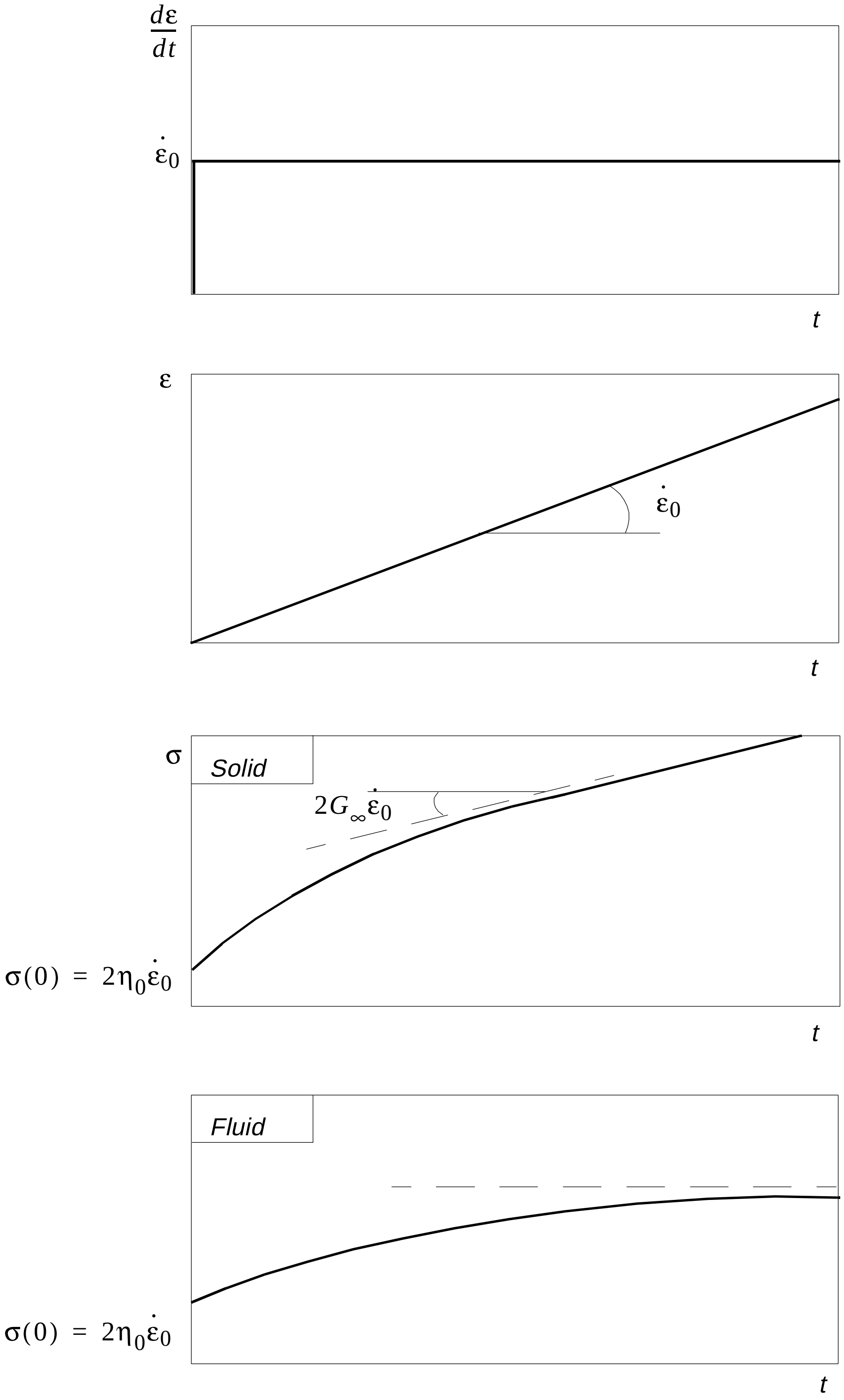


Figure 2-22. Response of Solid and Fluid to Constant Strain (Example 1).

Example 2:

Constraint strain rate for $0 \geq t < t_0$.

Zero strain rate for $t \geq t_0$.

This example demonstrates the stress relaxation behavior of a linear viscoelastic material. It shows that although the strain is not increasing, the stress relaxes until it reaches a constant value. For a fluid, the stress relaxes completely to zero.

$$\begin{cases} 0 \leq t \leq t_0 \dfrac{d\varepsilon(t)}{dt} = \dot{\varepsilon}_0 \;;\quad \varepsilon(t) = \dot{\varepsilon}_0 t \\ t \geq t_0 \dfrac{d\varepsilon(t)}{dt} = 0 \quad ;\; \varepsilon(t) = \dot{\varepsilon}_0 t_0 \end{cases} \tag{2-9}$$

Su*b*stitutin*g* Eq. (2-9) into Eq. (2-5) and solving the integral gives

$$0 \leq t \leq t_0 \colon \sigma(t) \text{ and } \frac{d\sigma(t)}{dt}$$

as given by Eq. (2-7) and Eq. (2-8)

$$t \geq t_0 \colon \sigma(t) = 2\left\{\left(\mu_1(e^{-\beta(t-t_0)} - e^{-\beta t})\right) + G_\infty t_0\right\}\dot{\varepsilon}_0 \tag{2-10}$$

$$\frac{d\sigma(t)}{dt} = -2G_1(e^{-\beta(t-t_0)} - e^{-\beta t}) \tag{2-11}$$

The response is sketched in Figure 2-23. Until $t = t_0$ the response is equal to that shown in Figure 2-22.

The instantaneous relaxation at $t = t_0$ is again due to the additional dashpot η_0.

A solid relaxes to a finite value, equal to the stress in the G_∞ spring of the generalized Maxwell model. A fluid relaxes completely to zero.

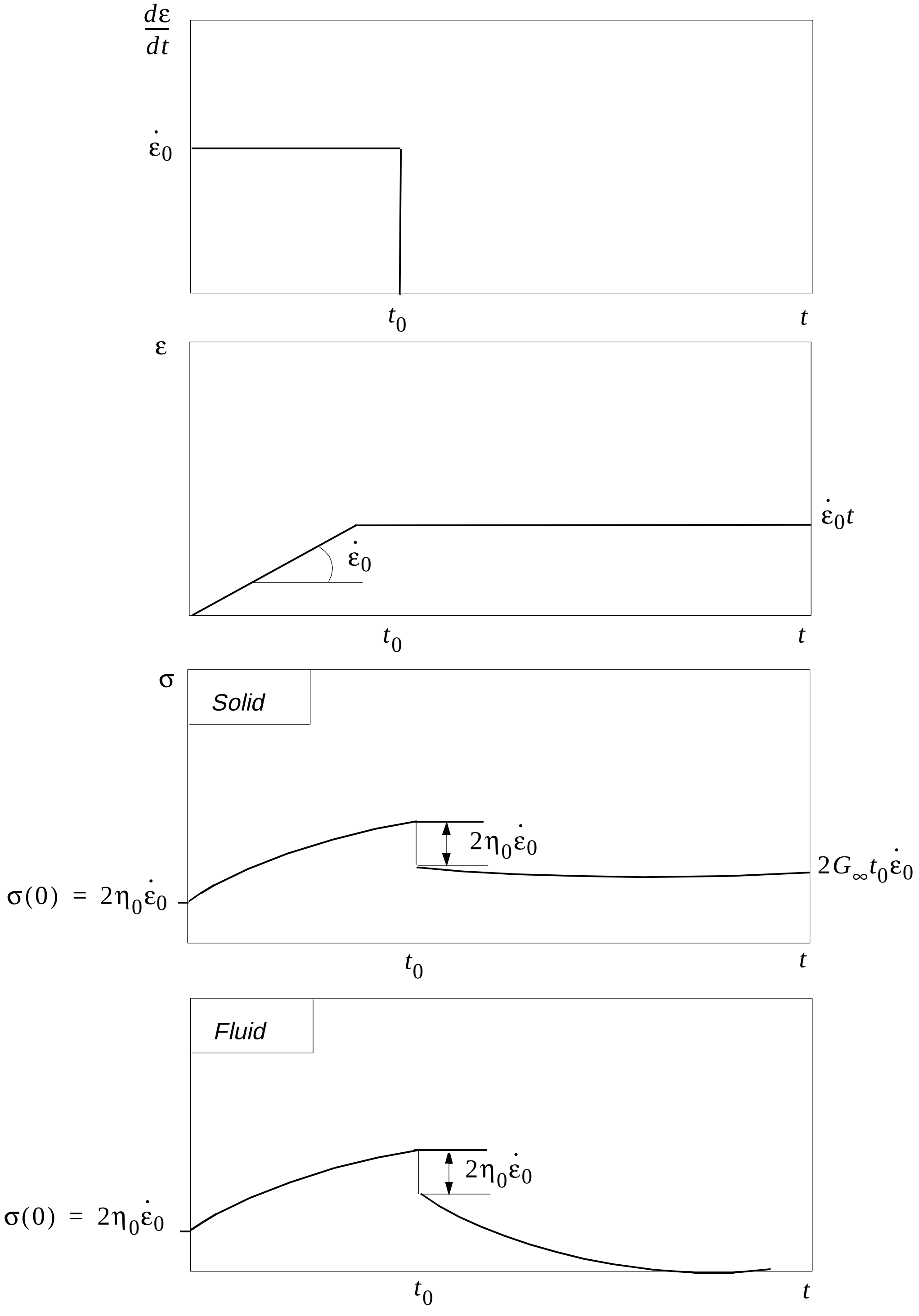


Figure 2-23. Stress Relaxation of Linear Viscoelastic Material After a Period of Constant Strain Rate (Example 2).

2.5.5 Yield Models

Yield models may be referenced by DMAT, DMATEP, or DYMAT24 entries. The yield models can be used to model elastic perfectly plastic behavior, bilinear elastoplastic behavior, piecewise linear behavior, or hydrodynamic behavior (zero yield stress).

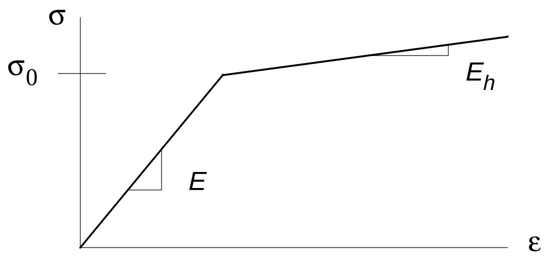
2.5.5.1 YLDHY – Hydrodynamic Yield Model

The YLDHY entry defines a yield model with constant zero yield stress. This model should be used for fluids that have no shear strength and are, therefore, hydrodynamic.

2.5.5.2 YLDVM – von Mises Yield Model

The YLDVM entry defines a von Mises yield model. The yield stress and hardening modulus are defined by giving either a bilinear or piecewise linear stress-strain curve. With Lagrangian and Eulerian solid elements, only an elastic perfectly plastic yield model can be used. The hardening modulus is not used.

Bilinear Representation



where the yield stress σ_y is given by

$$\sigma_y = \sigma_0 + \frac{EE_h}{E - E_h}\varepsilon_p$$

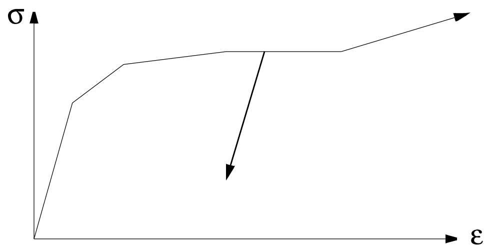
where σ_0 = yield stress

E = Youngs modulus

E_h = hardening modulus

ε_p = equivalent plastic strain

Piecewise Linear Representation



During every iteration, the stress *s* is determined from the current equivalent strain ε by interpolating from the stress-strain table

$$\sigma = [(\sigma_i - \sigma_{i-1})(\varepsilon - \varepsilon_{i-1})/(\varepsilon_i - \varepsilon_{i-1})] + \sigma_{i-1}$$

where σ_j and ε_j are the points in the table. The stress-strain characteristic used internally in MSC.Dytran is in terms of true stress and equivalent plastic strain. However, for convenience, the stress-strain characteristic can be input in any of the following ways:

- True stress/true strain.
- Engineering stress/engineering strain.
- True stress/plastic strain.
- True stress/plastic modulus.

True stress is defined as

$$\sigma_{true} = \frac{F}{A}$$

where *F* = current force, *A* = current area.

Plastic strain ε_{pl} is

$$\varepsilon_{pl} = \varepsilon_{true} - \varepsilon_{el}$$

where ε_{true} = true strain, ε_{el} = elastic strain.

True strain is defined as

$$\varepsilon_{true} = \int \frac{dl}{l}$$

where *dl* = incremental change in length, *l* = current length.

By comparison, engineering stress σ_{eng} and strain ε_{eng} are given by

$$\sigma_{eng} = \frac{F}{A_0} \text{ where } A_0 \text{ = original area}$$

$$\varepsilon_{eng} = \frac{(I - I_0)}{I_0} \text{ where } I_0 \text{ = original length}$$

True stress/true strain and engineering strain are related by the following formulas:

$$\sigma_{true} = \sigma_{eng}(1 + \varepsilon_{eng})$$

$$\varepsilon_{true} = ln(1 + \varepsilon_{eng})$$

At small strains, there is little difference between true stress-strain and engineering stress-strain. However, at moderate and large strains there can be very large differences, and it is important that the correct stress-strain characteristic is input.

When defining the material properties using Young’s modulus, yield stress, and hardening modulus, the hardening modulus must be estimated from a plot of true stress versus true strain. This estimate may well require a measured material characteristic to be replotted.

Some simple examples follow:

True Stress Versus True Strain

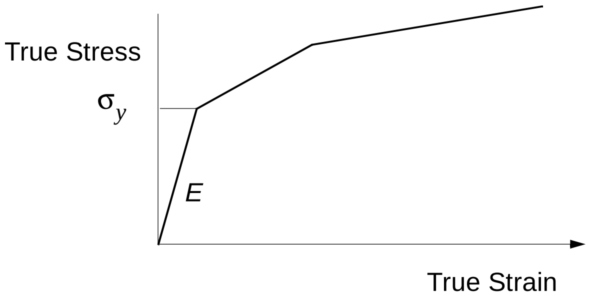


Figure 2-24. True Stress Versus True Strain Curve.

The slope of the first segment of the curve gives the Young’s modulus for the material (when it is not defined explicitly) and the first nonzero stress point gives the yield stress σ_y. The point corresponding to the origin can be omitted.

Engineering Stress Versus Engineering Strain

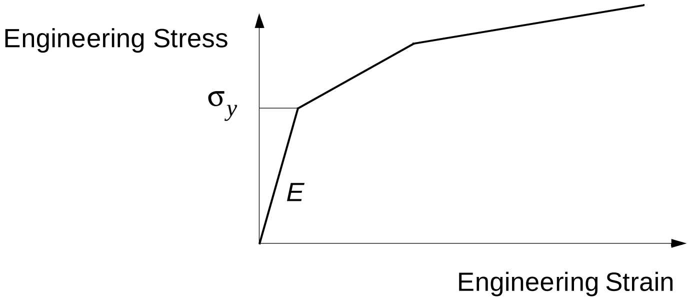


Figure 2-25. Engineering Stress Versus Engineering Strain Curve.

True Stress Versus Plastic Strain

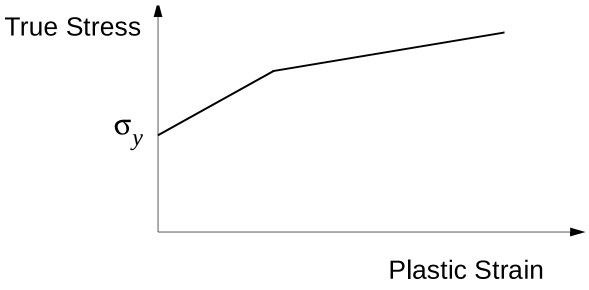


Figure 2-26. True Stress Versus Plastic Strain Curve.

Since the curve is defined in terms of the equivalent plastic strain, there is no elastic part in the curve. The first point must be the yield stress of the material at zero plastic strain. Young's modulus is defined separately.

True Stress Versus Plastic Modulus

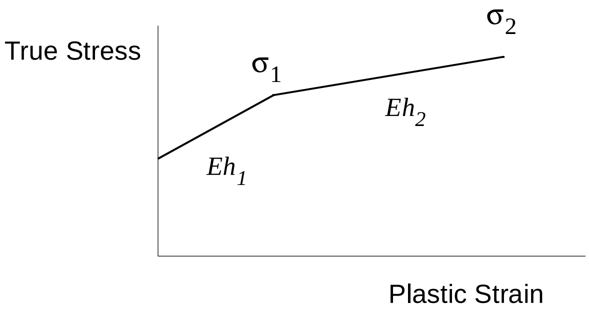


Figure 2-27. True Stress Versus Plastic Strain Curve.

This option is slightly different since the curve is specified as a series of pairs of stress and hardening moduli, rather than as a series of pairs of stress and strain. Young's modulus and yield stress are defined explicitly so that the table consists of pairs of values with the hardening modulus (x-axis) and the true stress (y-axis) at the end of the segment.

Yielding occurs when the von Mises stress

$$\sigma_{vm} = \sqrt{[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2]/2}$$

exceeds the yield stress σ_y. The principal stresses are σ_1, σ_2, and σ_3.

Isotropic hardening is assumed, which means that the yield surface increases in diameter as yielding occurs, but its center does not move.

This yield model can be used with beam, shell, and solid elements. When used with shell or solid elements, strain-rate sensitivity and failure can be included. Strain-rate sensitivity can be defined in two ways:

1. You can specify a table giving the variation of a scale factor *S* with strain-rate *dε/dt*. The scale factor is multiplied by the stress found from the stress-strain characteristic to give the actual stress. The failure criterion is based on plastic strain. When the plastic strain exceeds the specified value, the element fails. All the stresses are set to zero, and the element can carry no load. (This failure criterion is referred to from the DMATEP or the DYMAT24 entry.)
2. You can specify the constants *D* and *P* in Cowper-Symonds rate enhancement formula

$$\frac{\sigma_d}{\sigma_y} = 1 + \left\{\frac{\dot{\varepsilon}}{D}\right\}1/\mathrm{P}$$

   where σ_d is the dynamic yield stress, σ_y is the static yield stress and is $\dot{\varepsilon}$ the equivalent strain rate.

2.5.5.3 YLDJC – Johnson-Cook Yield Model

The YLDJC entry defines a Johnson-Cook yield model in which the yield stress is a function of the plastic strain, strain rate, and temperature

$$\sigma_y = (A + B\,\varepsilon_p^n)(1 + C\,ln(\dot{\varepsilon}/\dot{\varepsilon}_o))(1 - T^{*m})$$

where T^* = $\dfrac{(T - T_r)}{(T_m - T_r)}$

ε_p = effective plastic strain

$\dot{\varepsilon}$ = effective strain rate

$\dot{\varepsilon}_0$ = reference strain rate

T = temperature

Tr = room temperature

T_m = melt temperature

A, B, n, C, and *m* are constants.

2.5.6 Equations of State

Equations of state are referenced from the DMAT entry. The equation of state for a material is of the basic form

Pressure = *f* (density, specific internal energy)

The simplest equation of state is the gamma law equation of state, defined by the EOSGAM entry. The only input required is the ratio of specific heats for an ideal gas.

The EOSPOL entry defines a polynomial equation of state.

The EOSTAIT entry defines an equation of state based on the Tait model in combination with a cavitation model.

The EOSJWL entry defines an equation of state based on the JWL explosive model. It is used to calculate the pressure of the detonation products of high explosives. The JWL model is empirically based and requires the input of five constants.

2.5.6.1 EOSGAM – Gamma Law Equation of State

The EOSGAM model defines a gamma law equation of state for gases where the pressure is a function of the density, the specific internal energy, and the ideal gas ratio of specific heats γ of an ideal gas

$$p = (\gamma - 1)\rho\ e$$

where e = specific internal energy unit mass

ρ = overall material density

γ = ratio of specific heats (C_p/C_v)

2.5.6.2 EOSPOL – Polynomial Equation of State

The EOSPOL model defines a polynomial equation of state where the pressure is related to the relative volume and specific internal energy by a cubic equation.

In compression $(\mu > 0)$

$$p = a_1\mu + a_2\mu^2 + a_3\mu^3 + (b_0 + b_1\mu + b_2\mu^2 + b_3\mu^3)\rho_0 e$$

In tension $(\mu \le 0)$

$$p = a_1\mu + (b_0 + b_1\mu)\rho_0 e$$

where μ = $\eta - 1$

η = ρ/ρ_0

ρ = overall material density

ρ_0 = reference density

e = specific internal energy per unit mass

The EOSPOL equation of state can also be used to model viscous fluids, see also Section 2.5.7 on page 2-76.

2.5.6.3 EOSTAIT – Tait Equation of State

The EOSTAIT model defines a equation of state based on the Tait model in combination with a cavitation model where the pressure *p* is defined as follows:

No cavitation $(\rho > \rho_c)$,

$$p = a_0 + a_1(\eta^\gamma - 1)$$

Cavitation $(\rho \le \rho_c)$,

$$p = p_c$$

where η = ρ/ρ_0

ρ = overall material density

ρ_0 = reference density

ρ_c = critical density which produces the cavitation pressure p_c

The pressure can not fall below the cavitation pressure $p_c = a_0 + a_1(((\rho_c)/(\rho_0))^\gamma - 1)$, although the density can continue to decrease below its critical value ρ_c.

The EOSTAIT equation of state can als be used to model viscous fluids, see als Section 2.5.7 on page 2-76.

2.5.6.4 EOSJWL – JWL Equation of State

This equation of state can be used only with Eulerian elements.

where e = specific internal energy per unit mass

ρ_0 = reference density

ρ = overall material density

p = $A\left(1-\frac{\omega\eta}{R_1}\right)e^{-R_1/\eta} + B\left(1-\frac{\omega\eta}{R_2}\right)e^{-R_2/\eta} + \omega\eta\rho_o e$

$$\eta = \rho/\rho_0$$

and A, B, ω, R_1, and R_2 are constants.

These parameters are defined in Reference 3.

A DETSPH entry must be used to specify the detonation time, the location of the detonation point, and the velocity of a spherical detonation wave. When no DETSPH entry is present, all the material detonates immediately and completely.

2.5.7 Material Viscosity

Viscous fluid material models are available for the ROE solver only. The viscous behavior is referenced from the EOSPOL or EOSTAIT entry. For these viscous material models the stress tensor, t_{ij}, is given by

$$t_{ij} = -p\delta_{ij} + s_{ij} \tag{2-12}$$

$$\frac{dp}{dt} = \frac{K}{\rho}\frac{d\rho}{dt} \tag{2-13}$$

and

$$s_{ij} = 2\mu\frac{de_{ij}^d}{dt} \tag{2-14}$$

where K denotes the bulk modulus, ρ the density, s_{ij} the deviatoric stress tensor, p the pressure, e_{ij}^d the deviatoric strain tensor, and μ the coefficient of viscosity. In the Roe solver the stresses are computed directly from the velocity gradients.

2.5.8 Material Failure

Failure criteria are referenced from the DMAT or DMATEP entry. When the failure criterion is satisfied, the material loses all of its strength.

There are several methods of determining material failure in MSC.Dytran:

FAILMPS	Constant, maximum plastic strain.
FAILEX	User-specified failure.
FAILEX1	User-specified (extended) failure.
FAILEST	Constant, maximum equivalent stress and minimum time step.
FAILMES	Constant, maximum equivalent stress.
FAILPRS	Constant, maximum pressure.
FAILSDT	Constant, maximum plastic strain and minimum time step.

2.5.8.1 FAILMPS – Maximum Plastic Strain Failure Model

The FAILMPS entry defines a failure criterion based on a maximum value of effective plastic strain. The entry can be used with Eulerian and Lagrangian solid elements, shell elements, and Hughes-Liu beams.

2.5.8.2 FAILEX – User Failure Subroutine

The FAILEX entry allows a failure criterion to be described in an external subroutine EXFAIL. The subroutine must be included in the file referenced by the USERCODE FMS statement.

2.5.8.3 FAILEX1 – User Failure Subroutine

The FAILEX1 entry allows a failure criterion to be described in an external subroutine EXFAIL1. The failure model allows for inclusion of degradation of material properties. It is applicable to solid orthotropic materials (DMATOR) only. The subroutine must be included in the file referenced by the USERCODE FMS statement.

2.5.8.4 FAILEST – Maximum Equivalent Stress and Minimum Time Step Failure Model

The FAILEST entry defines properties of a failure model where total failure occurs when the equivalent stress exceeds the specified value and the element time step falls below the specified limit. The entry can only be used with Lagrangian solid elements.

2.5.8.5 FAILMES – Maximum Equivalent Stress Failure Model

The FAILMES entry defines a failure criterion based on a maximum value of the equivalent stress. The entry can only be used with Lagrangian solid elements.

2.5.8.6 FAILPRS – Maximum Pressure Failure Model

The FAILPRS entry defines a failure criterion based on a maximum value of the pressure. The entry can only be used with Lagrangian (orthotropic) elements.

2.5.8.7 FAILSDT – Maximum Plastic Strain and Minimum Time Step Failure Model

The FAILSDT entry defines the properties of a failure model where total failure occurs when the equivalent plastic strain exceeds the specified value and the element time step falls below the specified limit. The entry can only be used with Lagrangian solid (isotropic) elements.

2.5.9 Spallation Models

A spallation model defines the minimum pressure prior to spallation. At present there is only one spallation model, PMINC, that defines a constant spallation pressure.

2.5.9.1 PMINC – Constant Minimum Pressure

A constant minimum pressure must be defined that must be less than or equal to zero. Note that the pressure is positive in compression. If the pressure in an element falls below the minimum pressure, the element spall and the pressure and yield stress are set to zero. The material then behaves like a fluid. When the pressure subsequently becomes positive, the material will no longer be in a spalled state. The pressure can then decrease again to the specified minimum (the spall limit) before spallation occurs again.

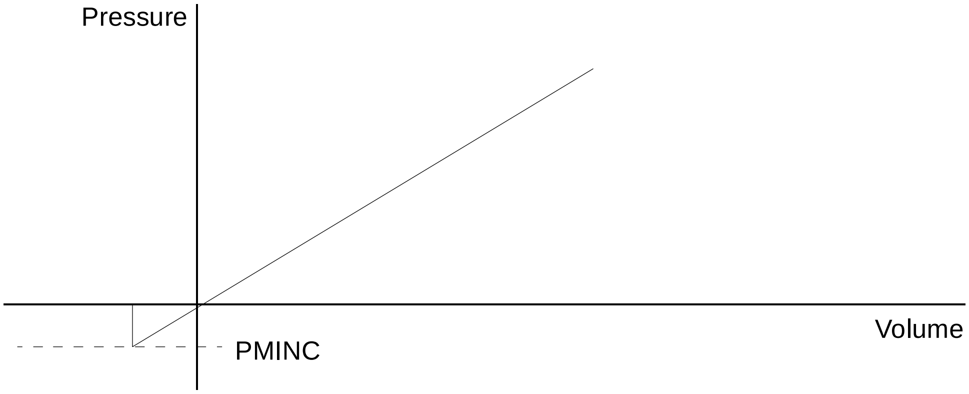


Figure 2-28. Minimum Pressure Cutoff.

2.5.10 Artificial Viscosities

The types of artificial viscosity used in MSC.Dytran are bulk viscosity and hourglass viscosity. The parameters for bulk viscosity are material parameters. The hourglass-viscosity parameters are defined per property.

2.5.10.1 Bulk Viscosity

Artificial bulk viscosity is used to control the formation of shock waves. Shock waves are the propagation of discontinuities in velocity. The simplest example of a shock wave is a “square wave.” An ideal impact between two flat surfaces generates a square wave. Materials that stiffen upon deformation can produce a shock wave from a smooth wave profile. A finite element model of a continuous body cannot numerically represent this propagating discontinuity. When a time integration scheme without algorithmic damping (such as the explicit central difference method) is used to integrate the response, severe oscillations in amplitude trail the shock front. These oscillations can be traced to the limitations imposed by the finite frequency spectrum of the finite element mesh.

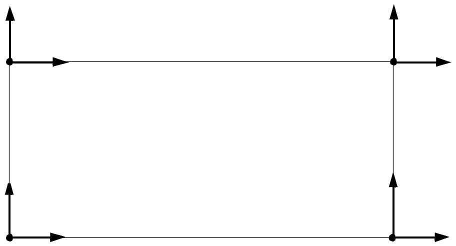
To control the oscillations trailing the shock front, artificial bulk viscosity is introduced. Artificial bulk viscosity is designed to increase the pressure in the shock front as a function of the strain rate. The effect on the shock wave is to keep it smeared over approximately five elements. Reducing the coefficients to steepen the wave front further results in undesirable oscillations trailing the shock, a condition sometimes referred to as “overshoot.”

The bulk viscosity equations contain both linear and quadratic terms that are given default values suitable for most situations. The values of the viscosity coefficients, BULKL for the linear viscosity, and BULKQ for the quadratic viscosity, can be changed on the respective fields of the material entries. A global redefinition of the default values can be achieved by using the parameter BULKL, and BULKQ entries.

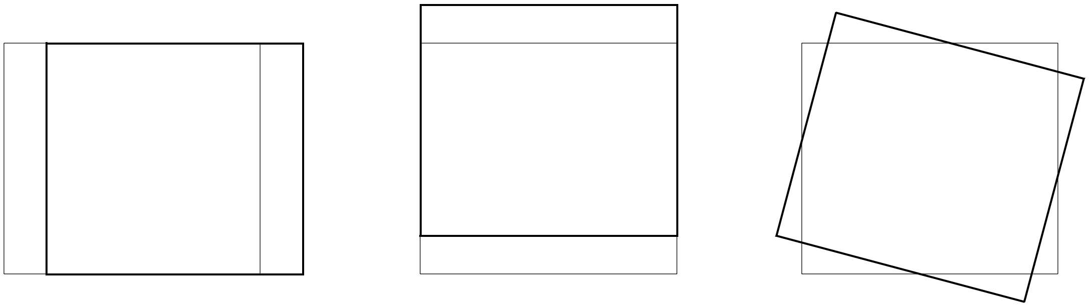
2.5.10.2 Hourglass Damping

The solid and shell elements in MSC.Dytran have only one integration point at the center of the element. This makes the program very efficient since each element requires relatively little processing, but it also introduces the problem of hourglassing.

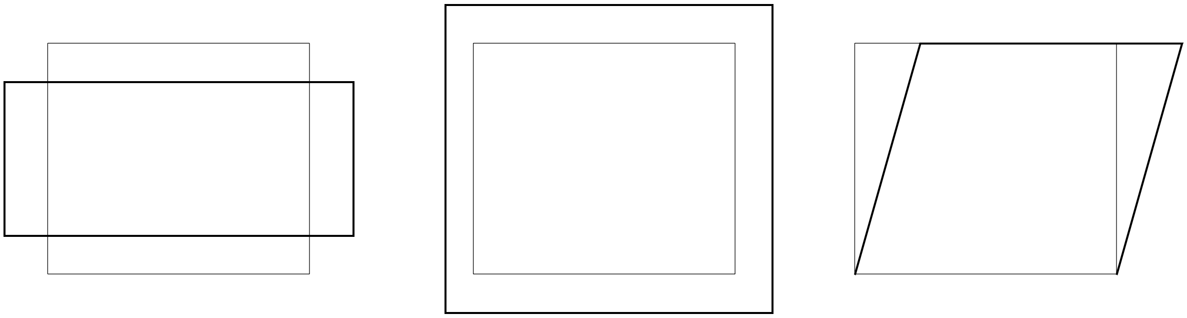
Consider, for simplicity, the two-dimensional membrane action of a CQUAD4 element.



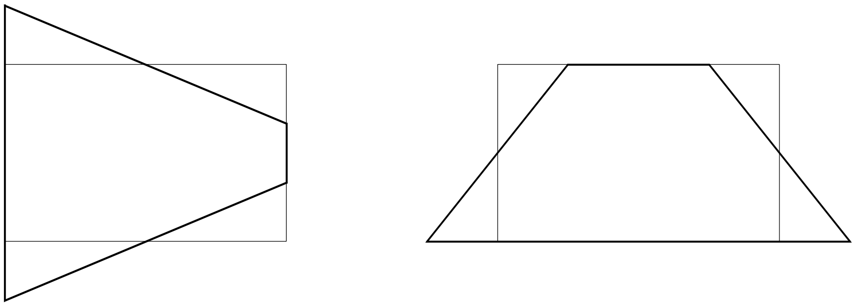
The element has four grid points, each with two degrees of freedom. There are, therefore, a total of eight degrees of freedom and eight modes of deformation. There are three rigid body modes, two translational modes, and one rotational mode.



With a single integration point, two direct and one shear stress are calculated at the center of the element. This means that only three modes of deformation have stiffness associated with them.



Two modes of deformation remain, that correspond to the linear stress terms. With a single integration point, these have no stiffness associated with them and are called the zero energy or hourglass modes.



When no measures are taken to stop these modes from occurring, they rapidly spread through the mesh and degrade the accuracy of the calculation, reduce the time step, and ultimately cause the analysis to abort when the length of the side of an element becomes zero.

Similar zero energy modes exist for the bending deformation of CQUAD4 elements, in CHEXA and CPENTA elements. CTRIA3 and CTETRA elements do not suffer from hourglassing, since no zero energy modes exist in these elements.

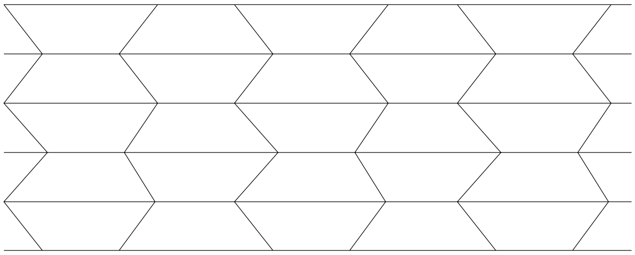


Figure 2-29. Deformation of a Mesh Showing Hourglassing.

Sophisticated methods for controlling hourglassing are available in MSC.Dytran. There are two forms: viscous and stiffness damping. The viscous form damps out hourglass modes and is carefully tuned so that other modes of deformation are not affected. The stiffness form applies forces to restrict the hourglass deformation by controlling the nonlinear part of the strain field that produces hourglassing. Normally the viscous forms work well, but in some instances are not adequate. The stiffness form is more effective but tends to make the elements overly stiff, depending on the input parameters selected.

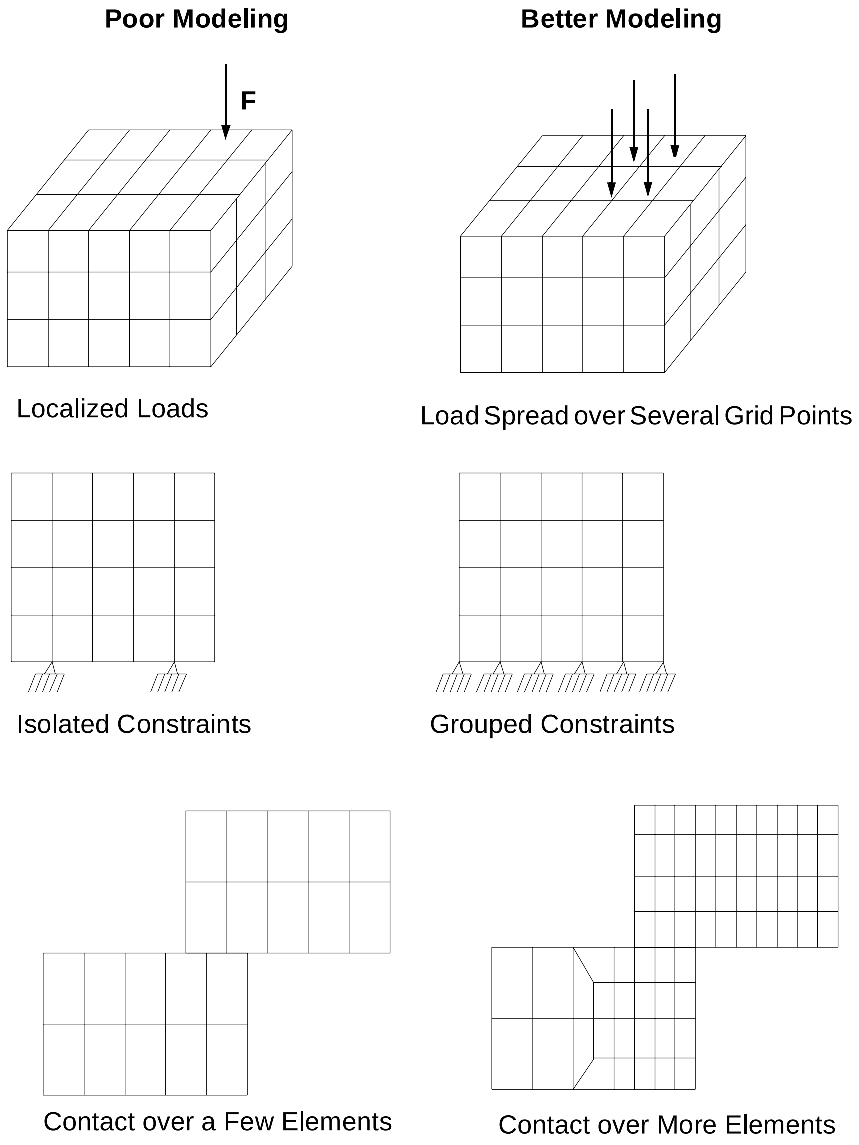
Each of the hourglass forms has slightly different characteristics. The default model is efficient and recommended for general use.

The default hourglass type can be reset using the PARAM option HGTYPE, HGSHELL, or HGSOLID. The hourglass coefficient can also be specified using the PARAM option HGCOEFF, HGCMEM, HGCWRP, HGCTWS, or HGCSOL. In addition, the hourglass type and coefficient can be specified for each individual property using the HGSUPPR entry.

Careful modeling can help prevent the occurrence of hourglassing in a mesh. Try to avoid sharp concentrations of load and isolated constraints. Rather, try to spread the loading and constraint over as large an area as possible. Some examples of how to avoid hourglassing are shown in Figure 2-30.

In the majority of cases, hourglassing does not cause any problem. In those instances where it does begin to occur, adjustment of the type of hourglass control and the hourglass viscosity should allow the analysis to be completed successfully. Extreme cases of hourglassing are normally caused by coarse meshes. The only solution is to refine the mesh.

Increasing the hourglass coefficient helps prevent hourglassing. However, excessively large values can cause numerical problems. Start with the default value and only increase it if excessive hourglassing occurs.



Poor Modeling

Better Modeling

Localized Loads

Load Spread over Several Grid Points

Isolated Constraints

Grouped Constraints

Contact over a Few Elements

Contact over More Elements

Figure 2-30. Hourglass Promotion and Avoidance.

2.6 Lagrangian Constraints

2.6.1 Constraint Definition

The motion of part or all of a mesh can be prescribed by application of constraints.

2.6.2 Single-Point Constraints

A single-point constraint is used to prescribe the motion of a translational or rotational degree of freedom. The constraint is effective throughout the analysis and is used to specify boundary conditions or planes of symmetry.

A single-point constraint is defined by an SPCn entry. The SPC entry defines the constraints on one grid point, while the SPC1 defines the constraints to be applied to a set of grid points. The SPC2 explicitly defines a rotational velocity constraint. SPC3 defines a constraint in a local coordinate system referenced from the SP3 entry. Several sets of SPC entries can be defined in the Bulk Data Section, but only those selected in the Case Control Section using the SPC = n command are incorporated in the analysis.

Single-point constraints can also be defined using the GRID entry. These constraints are present for the entire analysis and do not need to be selected in Case Control. This is valid only for SPC and SPC1.

Since MSC.Dytran is an explicit code, there is no matrix decomposition. Therefore, the problems of singular matrices that occur with some implicit codes do not exist. All, or part of the Lagrangian mesh can be entirely unconstrained and can undergo rigid body motion. MSC.Dytran correctly calculates the motion of the mesh. Similarly, the redundant degrees of freedom, such as the in-plane rotation of shell elements, do not need to be constrained since they do not affect the solution. The only constraints that are needed are those representing the boundary conditions of the model and those necessary for any planes of symmetry.

2.6.3 Contact Surfaces

Contact surfaces provide a very simple and flexible way of modeling the interaction among the parts of the finite element model and allowing continuous contact between deforming or rigid bodies. This gives enhanced convergence over a point-to-point gap and allows parts of the model to slide large distances relative to each other.

There are three types of contact surface:

- General contact and separation.
- Single-surface contact.
- Discrete grid points contacting a surface.

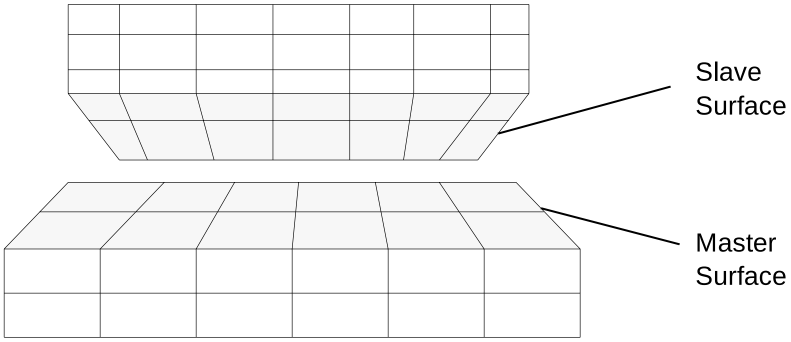
They are defined using the CONTACT entry on which you must specify the type of contact surface, the coefficient of friction, and the entities that might touch the contact surface.

2.6.3.1 General Contact and Separation

This is the most general of the contact surfaces and the one that is used most frequently. It models the contact, separation, and sliding of two surfaces, which can be frictional if required.

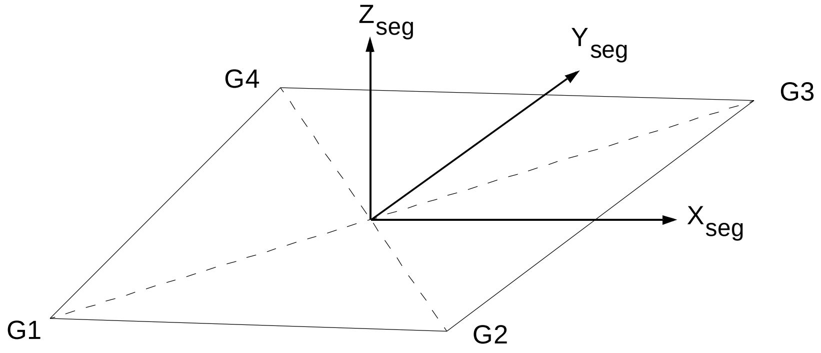
Segments

You must define the two surfaces that may come in contact by specifying the faces of the elements that lie on the surface. Each element face is called a segment of the surface. Segments are specified using the CSEG, CFACE, or CFACE1 entries. They can be attached to either solid or shell elements and can be triangular or quadrilateral. One surface is called the slave surface; the other surface is called the master surface. You must define a set of segments for each.

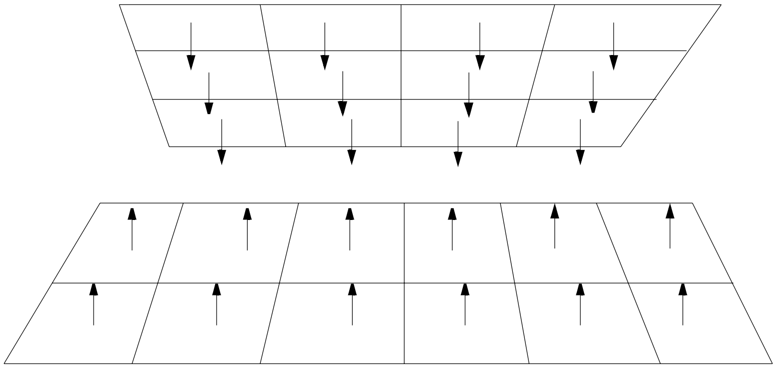


The two surfaces must be distinct and separate. A segment cannot be part of both the slave and master surfaces. The segments can be defined in a number of ways; they can be defined directly using CSEG, CFACE, or CFACE1 entries, or they can be attached to shell or membrane elements chosen by element number, property, or material. CSEG entries can also be defined using CQUAD4 or CTRIA3 entries, and CFACE1 entries can be defined using PLOAD4 entries (see Section 3.2.6 on page 3-4).

The connectivity of the segments is important since it determines from which side contact occurs. The order of the grid points on the CSEG entry defines a coordinate system just like the element coordinate system for the CQUAD4 and CTRIA3 elements described in Section 2.3.3 on page 2-11.



The SIDE field on the CONTACT entry is used to define the side from which contact occurs. In the example below, the z-axes of the segments point towards each other so that the top surfaces contact. The SIDE field should therefore be set to TOP. It is possible to specify that contact can occur from both sides of the segment. This is dangerous, however, in that initial penetrations of the contact surfaces are not detected.



It requires that the normals of a set of segments all point in the same direction. If this is not the case, the REVERSE option on the CONTACT entry automatically reverses the normals of any segments that do not point in the same general direction as the majority of segments on the surface.

Penetration

There must be no initial penetration of the two surfaces. The surfaces must either be coincident or have a gap between them. If there is initial penetration, MSC.Dytran issues a User Warning Message when the INIPEN field on the CONTACT entry is set to ON.

However, the calculation does continue, but the forces are applied to separate the surfaces. If the penetrations are large, these forces are also large and may cause premature termination of the analysis.

The PENTOL field on the CONTACT entry sets a tolerance for the penetration checks. Grid points outside of this tolerance are not initialized into the contact surface and so do not take part in the contact.

Method

The detailed theory is outside the scope of this manual, but it is important that you know how the contact surface works if you are to use it effectively. At each time step, each grid point on the slave surface is checked, and the nearest master segment is located. MSC.Dytran then checks to see if the grid point has penetrated the master segment. If it has not, the calculation continues. If it has penetrated, forces are applied in a direction normal to the master surface forces to prevent further penetration of the segment. The magnitude of the forces depends on the amount of penetration and the properties of the elements on each side of the contact surface. The magnitude of the forces is calculated internally by MSC.Dytran to ensure minimal penetration while retaining a stable solution. The FACT field on the CONTACT entry can be used to scale the magnitude of the forces. This can be useful when two components are forced together by large forces. However, instability may occur when the FACT value is set to a high value.

A friction force is also applied to each of the surfaces, parallel to the surface. The magnitude of the force during sliding is equal to the magnitude of the normal force multiplied by the coefficient of friction. The direction of the friction force is opposite to the relative motion of the surfaces.

The coefficient of friction μ is calculated as follows

$$\mu = \mu_k + (\mu_s - \mu_k)e^{-\beta v}$$

where μ_s = static coefficient of friction

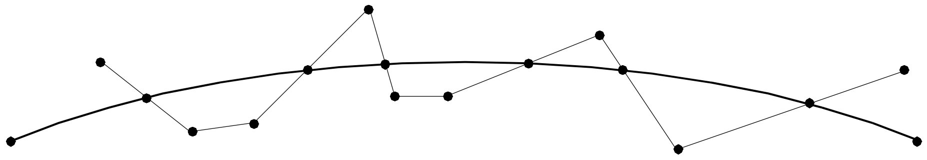
μ_k = kinetic coefficient of friction

β = exponential decay coefficient

v = relative sliding velocity of the slave and master surfaces

You must specify μ_s, μ_k, and β.

The algorithm is not symmetrical, since the slave points are checked for penetration of the master segments but not vice versa. This means that the mesh density of the slave surface should be finer than that of the master surface. If not, penetrations can occur as shown in two dimensions below.



This can lead to hourglassing and incorrect results.

Since the closest segment to each point on the surface is constantly updated, the contact surface works correctly regardless of how far the two surfaces slide relative to each other or how much the shape of the surfaces changes as the mesh deforms.

Efficiency

Generally, contact surfaces are very simple to use and very efficient. However, the penetration checks do take time, and therefore, the number of slave and master segments on each interface should be limited to those where contact might occur.

The UPDATE and SORT fields on the CONTACT entry control the working of the algorithm and allow its cost and accuracy to be adjusted.

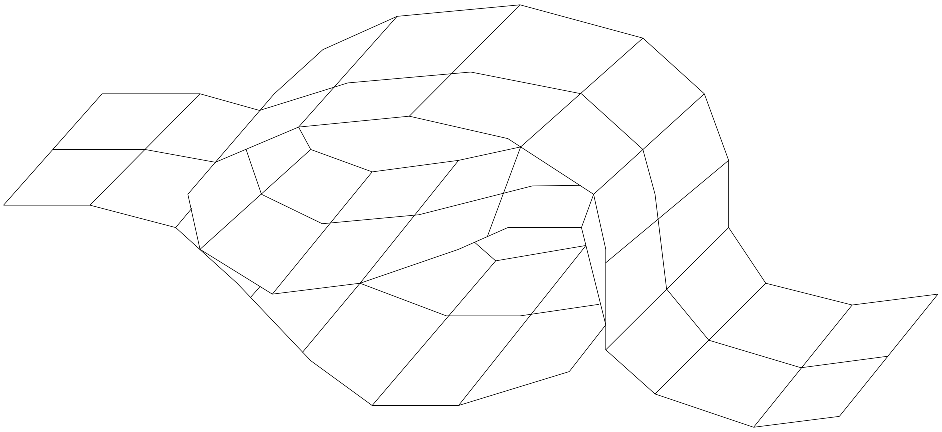
The UPDATE option determines how often the contact forces are recalculated. The problem being examined may be reasonably static and the contact forces may have remained fairly constant during the analysis. In such cases, the forces do not need to be evaluated in every cycle, which saves computational time. When the problem is highly dynamic, however, it is advisable to recalculate contact forces more often to preserve accuracy. The default for this option is 0.0.

The SORT option determines how often the list of points and their nearest segments are updated. In the same way as UPDATE, this depends on the dynamics of the problem. The default for this option is 0.1, which is a conservative number requesting a re-sort frequently.

TSTART and TEND enable you to switch the contact surface on and off at specific times. This means that the contact surfaces are not checked until the contact surface is activated, thus saving computational effort when no contact occurs. By default, the contact surface is active throughout the analysis.

2.6.3.2 Single Surface

The single-surface contact is similar to the general one described in previous sections, but instead of defining slave and master segments, you define one set of slave segments where the slave segments cannot penetrate themselves. This is particularly useful for modeling buckling problems where the structure folds onto itself as the buckles develop and the points of contact cannot be determined beforehand.



This type of contact surface is defined in the same way as the general type, using the CONTACT entry, except that you only define a set of slave segments—the MID field must be left blank. The surface can be frictional by giving nonzero values of the friction coefficients on the CONTACT entry. Friction forces are applied in the same way as for the general contact surface (see Section 2.6.3.1 on page 2-85).

Unlike the general contact surface, the connectivity of the segments does not matter. Contact can occur on either side of the surface automatically. However, the normals of all segments on the surface must point in the same direction although it does not matter in which direction. In most of the meshes, this usually is the case. If not, the REVERSE option automatically reverses the normals of segments that do not point in the same direction as the majority of segments on the surface.

The single-surface algorithm works in much the same way as the master-slave type described in Section 2.6.3.1. The algorithm is particularly efficient, and rather large areas of single surface contact may be defined.

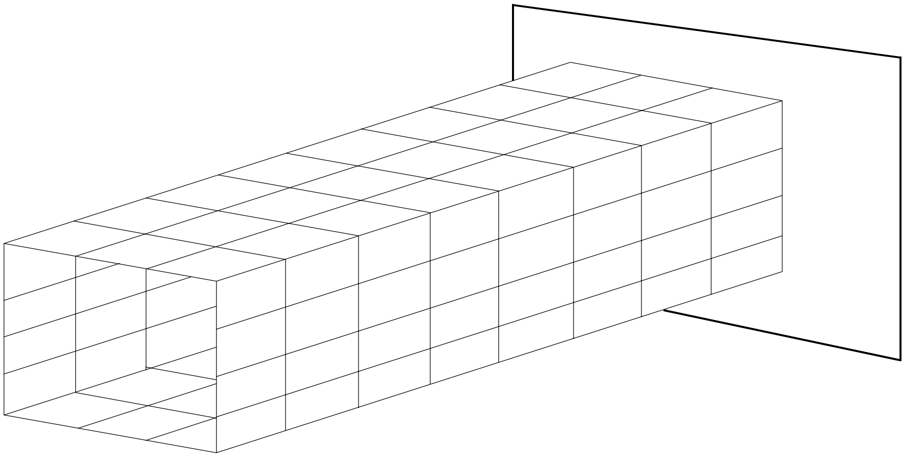
2.6.3.3 Discrete Grid Points

This type of contact surface allows individual grid points to contact a surface. The SID field on the CONTACT entry must be set to GRID. You must supply a list of the slave grid points—which can not penetrate the master surface—using the SET1 entry. The master surface must be defined as a set of segments in the same way as general contact surfaces are defined (see Section 2.5.4.1 on page 2-60).

The slave points can be attached to any type of element. Throughout the analysis the slave points are prevented from penetrating the master surface. When in contact with the master surface, the slave points can slide frictionless or with friction along the surface.

2.6.4 Rigid Walls

A rigid wall is a plane through which specified slave grid points cannot penetrate. The rigid wall provides a convenient way of defining rigid targets in impact analyses.



Any number of rigid walls can be specified using WALL entries. The orientation of each wall is defined by the coordinates of a point on the wall and a vector that is perpendicular to the wall and points towards the model.

At each time step, a check is made to determine whether the slave grid points have penetrated the wall. These slave points are defined using a SET1 entry, and there can be any number of them. Since a check is made for every slave point at each time step, you should specify only those points as slave points that are expected to contact the wall in order to ensure the most efficient solution.

If a slave point is found to have penetrated the wall, it is moved back towards the wall so that its momentum is conserved. If the slave point subsequently moves away from the wall, it is allowed to do so.

Slave points cannot have any other constraint. They can, however, be part of other contact and coupling surfaces.

2.6.5 Tied Connections

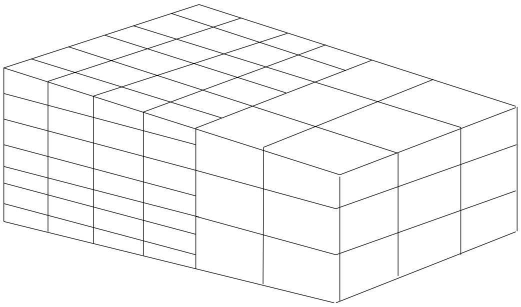
Tied connections are used to join parts of the mesh together. There are three types of connections:

- Two surfaces tied together.
- Grid points tied to a surface.
- Shell edge tied to a shell surface.

All are defined using the RCONN entry and are described in the following sections.

2.6.5.1 Two Surfaces Tied Together

With this type of connection, two surfaces are permanently joined together during the analysis. This provides a convenient method of mesh refinement. It is better to use this method of mesh refinement than to use CPENTA or CTETRA elements, which are too stiff. Naturally, tied contact surfaces should not be close to any critical regions or areas that are highly nonlinear. Otherwise, you may use them wherever convenient.



You need to define the slave and master surfaces that are to be tied together by specifying the faces of elements that lie on the surface. Each element face or segment can be attached to either solid or shell elements and can be either quadrilateral or triangular.

You must define two surfaces that comprise a master and slave surface by specifying the faces of the elements that lie on the surface. Each element face is called a segment. The segments can be defined using CSEG, CFACE, or CFACE1 entries.

The way the tied surface works is not symmetrical, so your choice of slave and master surface is important. The slave segments must always be attached to the finer mesh, and the master segments are attached to the coarser mesh. To use tied surfaces to connect two meshes that change their mesh density so that in one area one mesh is finest while in another area the other is finest, use more than one tied connection to join them together.

If this rule is not followed, some grid points will penetrate the other mesh, hourglassing will be excited, and spurious results will occur in the region of the tied connection.

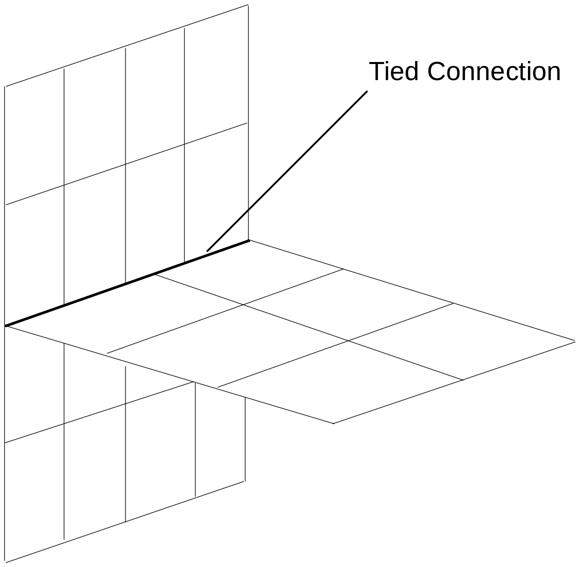
2.6.5.2 Grid Points Tied to a Surface

Individual grid points can be tied to a surface using this type of tied connection. In this case, the SID field of the RCONN entry must be set to GRID, and you must give a list of all grid points that are to be tied. The master surface must be defined as a set of segments in the same way as for the two surface connections described in the previous section. In addition, the OPTION field must be set to NORMAL.

During the analysis, each grid point will be tied to the surface; i.e., its position relative to the surface will not change. Only the translational degrees of freedom are tied. If, for example, a shell element is attached to a tied grid point, then the shell can rotate relative to the surface.

2.6.5.3 Shell Edge Tied to a Shell Surface

This type of connection is used to connect the edge of one set of shell elements to the surface of another set.



The SID field of the RCONN entry must be set to GRID, and you must give a list of all the grid points that lie on the edge of the first set of shell elements. The master surface is defined as a set of segments in the same way as the two surface connection described in Section 2.6.5.1 on page 2-91, except that the segments can only be attached to shell elements. In addition, the OPTION field must be set to SHELL.

In addition, the list of grid points can consist of any type of six DOF grid points (CBEAMs, CTRIAs, etc.).

Similar to the previous connection, the slave grid points are tied to the surface during the analysis; i.e., their position relative to the surface will not change. The difference is that, in this case, the rotational degrees of freedom are also coupled so that the angle between the two sets of shells will be maintained.

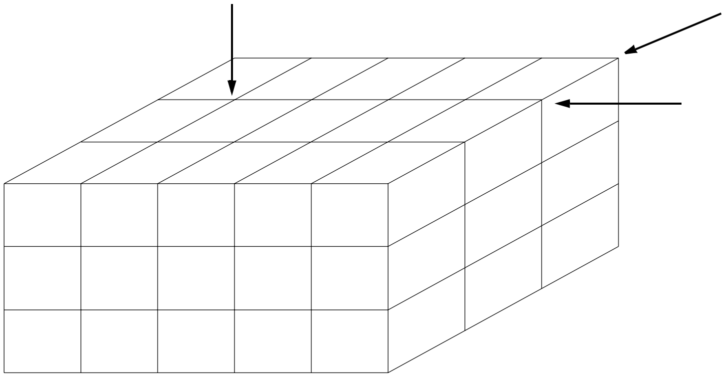
2.7 Lagrangian Loading

2.7.1 Loading Definition

This section covers the different ways that the analysis model can be loaded. The facilities available are:

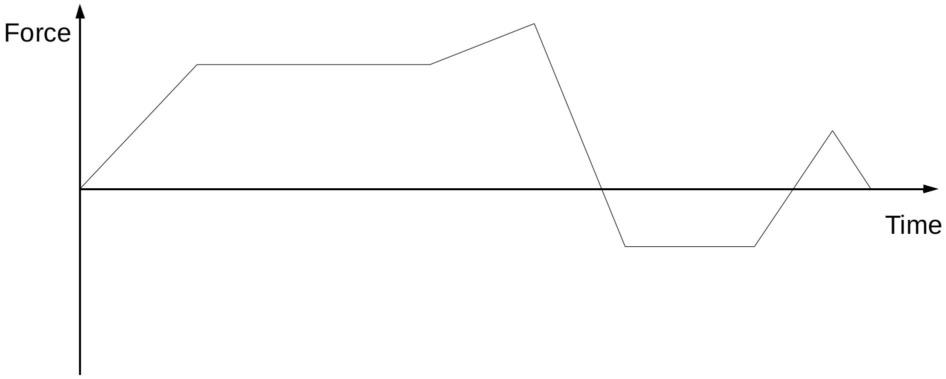
- Concentrated loads and moments at grid points.
- Pressure loads.
- Enforced motions.
- Initial conditions.

2.7.2 Concentrated Loads and Moments



Concentrated loads and moments can be applied to any grid point using the DAREA, FORCE, FORCE1, FORCE2, MOMENT, MOMENT1, or MOMENT2 entries in combination with a TLOADn entry.

The TLOAD1 entry can reference a TABLEXX entry which is used to specify how the force F(t) varies with time t.



The TLOAD2 entry always defines a variation with time by a function of which the coefficients are explicitly defined on the TLOAD2 entry.

The TLOADn entry also references a set of loading entries. These select the type of load, the grid point that is to be loaded, the direction of the load, and a scale factor to be applied to the curve of force versus time. The actual load applied *P(t)* is given by

$$P(t) = AF(t)$$

where *A* is the scale factor.

The types of concentrated load that can be applied are discussed in the following section.

FORCE, FORCEn, or DAREA – Fixed-Direction Concentrated Loads

The FORCE, FORCEn, and DAREA entries define fixed direction loads. In other words, the direction of the force is constant throughout the analysis and does not change as the structure moves.

FORCE, FORCEn, and DAREA entries have the same effect but define the loading in different ways. With the DAREA entry, you specify the grid point, the direction in the basic coordinate system in which the load acts, and the scale factor. With the FORCE or FORCEn entry, you define the grid point, the components of a vector giving the loading direction, and the scale factor. In this case, the magnitude of the vector also acts as a scale factor, so the force in direction *i* is given by

$$P_i = AN_iF_i(t)$$

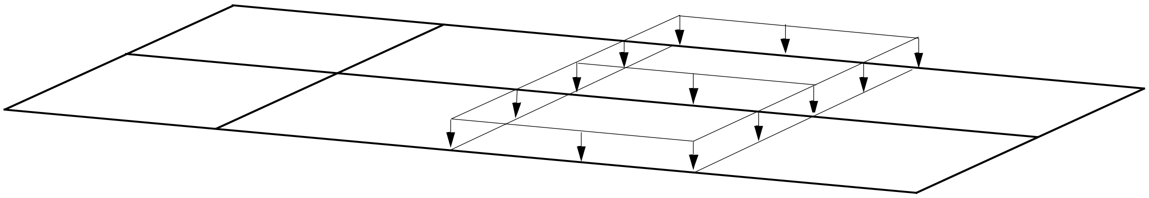
The concentrated load or enforced motion on a rigid body can be specified by defining the load at the rigid body center of gravity. To do so, set the TYPE field of the TLOAD1 or TLOAD2 entries to 13 and 12, respectively. The G field in the FORCE or MOMENT entry references the property number of the rigid body or MR<id> or FR<id>, where id is the number of a MATRIG or RBE2-FULLRIG entry, respectively.

MOMENT, MOMENTn, or DAREA – Fixed-Direction Concentrated Moments

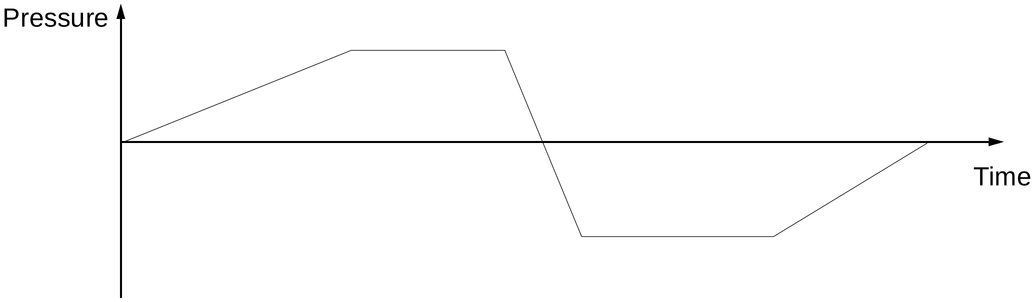
Concentrated moments can be applied using either MOMENT, MOMENTn, or DAREA entries. The difference between the two is the same as that between the FORCE, FORCEn, and DAREA entries described in the previous section.

2.7.3 Pressure Loads

Pressure loads are applied to the faces of solid elements and to shell elements. Pressure loads are defined using the PLOAD or PLOAD4 entry in combination with a TLOADn entry.



The TLOAD1 entry references a TABLEXX entry on which you specify the variation of the pressure *P(t)* with time *t*.



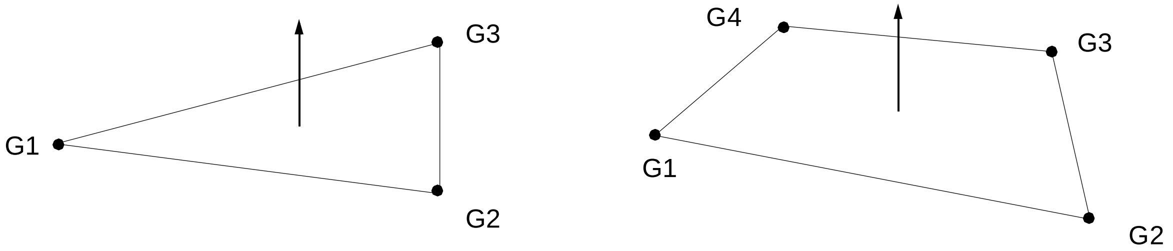
The TLOAD2 entry defines the variation of the pressure *P(t)* with time *t* based on an equation, which is defined by the TLOAD2 entry.

TLOAD2 also references a set of PLOAD and/or PLOAD4 entries. Each entry selects the face of the element to be loaded by its grid points and defines the scale factor to be applied to the curve of pressure versus time. The actual pressure acting on the element p_{el} is given as follows

$$p_{el}(t) = Ap(t)$$

where *A* is the scale factor.

The direction of positive pressure is calculated according to the right-hand rule using the sequence of grid points on the PLOAD entry. For PLOAD4 entries, the pressure is inwards for solid elements and in the direction of the element normal vector for shell elements.



2.7.4 Enforced Motion

This facility specifies the enforced motion of a degree of freedom at grid points by defining the grid-point velocity with time. The enforced motion is applied in a way similar to concentrated loads, using a DAREA, FORCE, RFORCE, GRAV, or FORCEEX entry in combination with a TLOAD1 or TLOAD2 entry.

You must specify that the TLOAD1 or TLOAD2 entry defines enforced motion. TLOAD1 references a TABLEXX entry that gives the variation of velocity *V(t)* with time *t*. The TLOAD2 entry implicitly defines a function of time. It also references a set of DAREA and/or FORCE entries that define the grid point being excited and the direction of the excitation. FORCE and DAREA entries have the same effect but define the excitation in different ways. With the DAREA entry, you specify the grid point, the direction in which the excitation is applied, and the scale factor *S*.

For enforced velocity, the velocity of the grid point $V_g(t)$ is given by

$$V_g(t) = S\ V(t)$$

With the FORCE entry, you define the grid point, the components of a vector *N* giving the excitation direction, and the scale factor *S*. In this case, the magnitude of the vector also acts as a scale factor, so the velocity of the grid point $V_g(t)$ is given by

$$V_g(t) = S\ N\ V(t)$$

If you want to specify the motion of a grid point in terms of its displacement, you must differentiate the motion to produce a velocity versus time characteristic that can be used by MSC.Dytran.

The FORCEEX entry allows the enforced motion of grid points to be defined in an external subroutine. The load number specified on the FORCEEX entry must be referenced in a TLOAD1 entry that specifies enforced motion, i.e., loading type 2. The subroutine EXTVEL containing the enforced motion specification must be included in the file referenced by the USERCODE FMS statement. The RFORCE entry defines enforced motion due to a centrifugal acceleration field. This motion affects all structural elements present in the problem. The GRAV defines an enforced motion due to a gravitational acceleration field. This motion affects all Lagrangian and Eulerian elements.

Grid points with enforced motion cannot be:

- Attached to a rigid body.
- A slave point for a rigid wall.
- Contact or rigid connection with rigid ellipsoids.

The motion of a rigid body can be specified by defining enforced motion of the rigid-body center of gravity. To do so, set the TYPE field of the TLOAD1 and TLOAD2 entries to 12. The G field on the DAREA, FORCE, or MOMENT entry references the property number of the rigid body, MR<id> or FR<id>, where id is the property number of the MATRIG entry or the RBE2-FULLRIG entry respectively.

2.7.5 Initial Conditions

The initial velocity of grid points can be defined using TIC, TICGP, TIC1, and TIC2 entries. This allows the initial state of the model to be set prior to running the analysis. It is important to recognize the difference between initial velocities and enforced velocities. Enforced velocities specify the motion of grid points throughout the transient analysis. Initial velocities, on the other hand, specify the velocity of grid points at the beginning of the analysis. Thereafter, the velocities are determined by the calculation.

Where TIC1 and TIC2 set only the initial grid-point velocity, the TICGP entry can be used to set the initial value of any valid grid point variable. It can also refer to a local coordinate system by including the CID1 and/or CID2 entry in the list.

Element variables can also be given initial values using the TICEL entry. Any valid element variable can be defined for a set of elements.

2.8 Eulerian Loading and Constraints

2.8.1 Loading Definition

The implementation of loading and constraints within Eulerian meshes is somewhat different than that in a Lagrangian mesh. Eulerian constraints apply to element faces within the mesh rather than to the grid points. MSC.Dytran allows you to set the initial conditions for material in Eulerian elements, constrain material with fixed barriers, apply gravitational body forces, apply pressure boundaries to element faces, apply flow boundaries where material enters or leaves the mesh, and couple the mesh so that the material interacts with the Lagrangian parts of the model.

If an exterior—or free face—of an Eulerian mesh does not have a specific boundary condition, then, by default, it forms a barrier through which the material cannot flow. The default can be redefined by using a FLOWDEF entry.

2.8.2 Flow Boundary

A flow boundary defines the physical properties of material flowing in or out of Eulerian elements and the location of the flow. The FLOW entry is referenced by a TLOAD1. The TYPE field on the TLOAD1 must be set to 4.

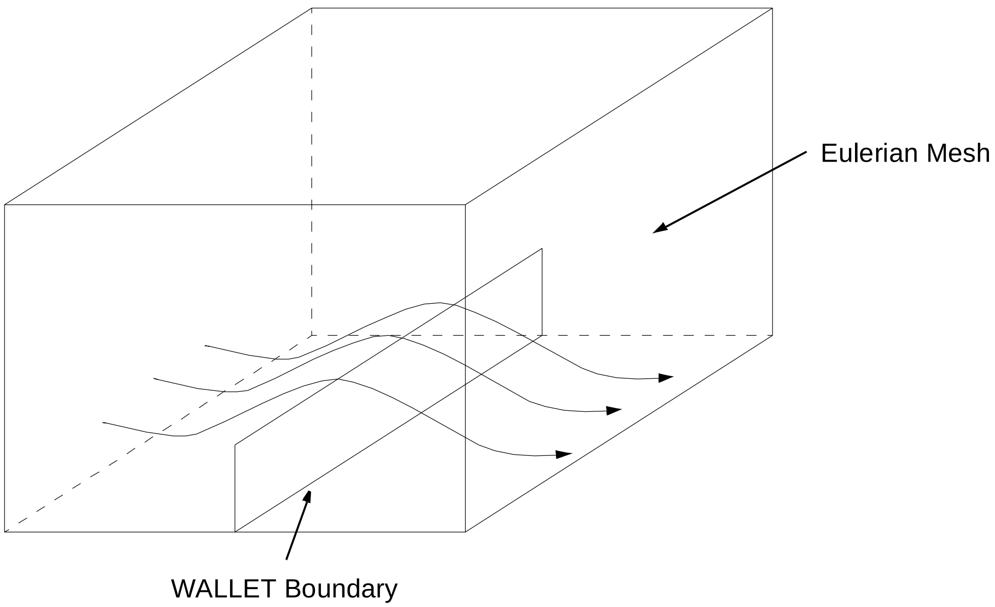
The FLOW entry references a set of segments, specified by CFACE, CFACE1, or CSEG entries, through which the material flows. The subsequent fields allow you to specify the x, y, or z velocity, the pressure, and the density or specific internal energy of the flowing material. If only the pressure is defined, this gives a pressure boundary. Any of the variables that are not specified take the value in the element that the material is flowing into or out of.

The FLOWEX entry specifies a similar flow boundary through a set of faces. However, the physical details of the flow are determined from a user subroutine.

2.8.3 Rigid Wall

The WALLET entry defines a wall that is equivalent to a Lagrangian rigid wall. This is a barrier to material transport in an Eulerian mesh. The barrier is defined by a set of faces generated from a CFACE, CFACE1, or CSEG entry through which no material can flow.

This is the default condition for any exterior faces of the Eulerian mesh that do not have a FLOW boundary specified. However, the WALLET entry can be used to specify rigid walls within an Eulerian mesh.



2.8.4 Initial Conditions

The initial conditions of Eulerian elements can be defined using the TICEL or TICEUL entry. This allows the initial state of the model to be set prior to running the analysis. It is important to recognize the difference between initial conditions and enforced conditions. Enforced conditions specify the loading and constraints of material throughout the transient analysis. Initial conditions, on the other hand, specify the state of the material only at the beginning of the analysis. Thereafter, the material state is determined by the calculation.

The TICEL entry defines transient initial conditions for elements. Any valid element variable can be given an initial value.

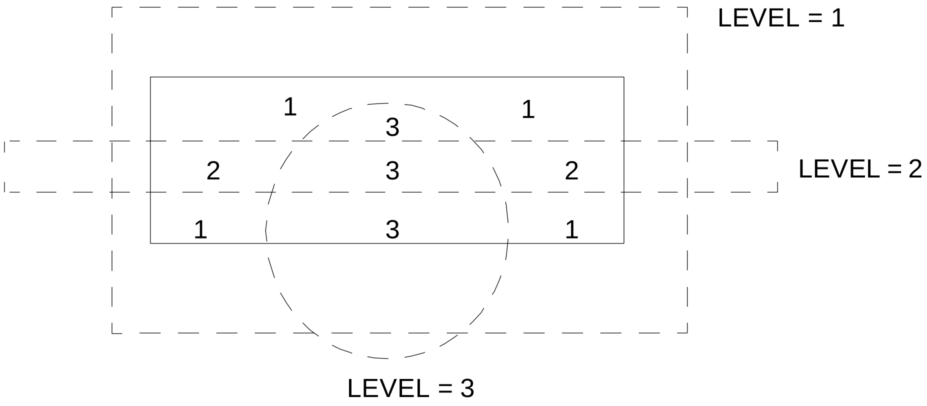
The TICEUL entry defines transient initial conditions for geometrical regions in the Euler mesh. The TICEUL entry must be used together with the PEULER1 property definition.

With the TICEUL entry, it is possible to generate initial conditions in cylindrical or spherical geometry shapes and in sets of elements.

Each geometrical region (cylinder, sphere or set of elements) has a level number. This allows the creation of regions of arbitrary shape by allowing the regions to overlap. An element that lies in two or more geometrical regions is assigned to the region that has the highest level number.

Think of geometrical regions as shapes cut out of opaque paper. Position the region of the lowest level number on the mesh. Then, place the next higher region on top of the first and continue until all the regions are in place. When the last region is placed, you have a map indicating to which region each element in the problem is assigned.

The following figure shows how three different geometrical regions can be used to create regions of arbitrary shape. The solid line represents the boundary of the mesh. Region one (LEVEL = 1) is the large dashed rectangle. Region two (LEVEL = 2) is the long narrow rectangle. Region three (LEVEL = 3) is a circular region. The numbers on the diagram indicate how the elements in different parts of the mesh are assigned to these three regions.



If two or more regions with the same level number but different initial value sets or materials overlap, then the regions are ambiguously defined. This results in an error.

2.8.5 Detonation

Eulerian elements that reference a JWL equation of state (EOSJWL) have to be detonated. A DETSPH entry must be present that defines a spherical detonation wave. You define the location of the detonation point, the time of detonation, and the speed of the detonation wave. MSC.Dytran then calculates the time at which each explosive element detonates. Elements that do not have a JWL equation of state are unaffected.

2.8.6 Body Forces

If the GRAV entry is specified, the Eulerian material also has body forces acting on the material mass. The GRAV entry defines an acceleration in any direction. All Eulerian material present in the problem is affected.

2.9 General Coupling

Definition

The objective of coupling is to enable the material modeled by the Eulerian and Lagrangian meshes to interact. Initially, the two solvers are entirely separate. Lagrangian elements that happen to lie within an Eulerian mesh do not affect the flow of the Eulerian material and no forces are transferred from the Eulerian material to the Lagrangian elements. Coupling computes the interaction of the two sets of elements and thus enables complex fluid-structure interaction problems to be analyzed.

The first task in coupling the Eulerian and Lagrangian sections of a model is to create a surface on the Lagrangian structure. It is this surface that is used to transfer the forces between the two solvers. The surface acts as a boundary to the flow of material in the Eulerian mesh. At the same time, the stresses in the Eulerian elements cause forces to act on the coupling surface, distorting the Lagrangian elements.

The SURFACE entry defines a multifaceted surface on the Lagrangian structure. The element faces in this surface can be identified by a set of CFACEs, CFACE1s, CSEGs, element numbers, property numbers, material numbers, or any combination of these. The method of defining of the surface is therefore extremely flexible and can be adapted to individual modeling needs.

The coupling is activated using the COUPLE entry. This specifies that the surface is to be used for Euler-Lagrange coupling. The COVER field indicates whether the inside or the outside of the coupling surface is covered, which means that it does not contain Eulerian material. For problems where the Eulerian material is inside a Lagrangian structure (for example, an inflating air bag), COVER should be set to OUTSIDE since the Eulerian elements outside the coupling surface are covered. For problems where the Eulerian material is outside the Lagrangian structure (for example a projectile penetrating soft material), the inside of the coupling surface is covered, and COVER should be set to INSIDE.

The coupling surface must have a positive volume to meet MSC.Dytran’s internal requirements, which means that the normals of all the segments of the surface should point outwards. However, if this is not the case, the REVERSE field can be used to have MSC.Dytran automatically reverse the direction of any segments with inward pointing normals.

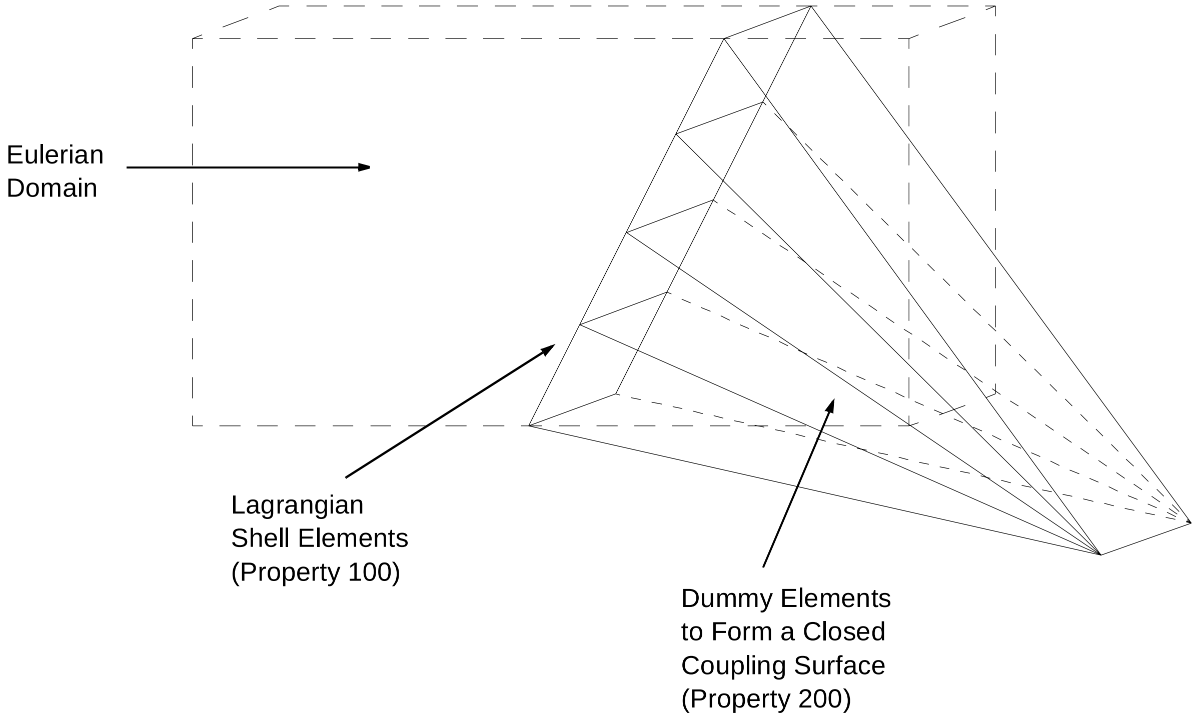
A fast coupling algorithm can be switched on by using PARAM,FASTCOUP. The restriction, however, is that the Eulerian mesh must be aligned with the basic coordinate system axes.

Closed Volume

The coupling surface must form a closed volume. This is fundamental to the way the coupling works in MSC.Dytran. It means that there can be no holes in the surface and the surface must be closed.

In order to create a closed volume, it may be necessary to artificially extend the coupling surface in some problems. In the example shown below, a plate modeled with shell elements is interacting with an Eulerian mesh. In order to form a closed coupling surface, dummy shell elements are added behind the plate. The shape of these dummy shell elements does not matter particularly. However, it is best to use as few as possible to make the solution as efficient as possible.

The closed volume formed by the coupling surface must intersect at least one Euler element, otherwise the coupling surface is not recognized by the Eulerian mesh.


Eulerian Domain
Lagrangian Shell Elements (Property 100)
Dummy Elements to Form a Closed Coupling Surface (Property 200)

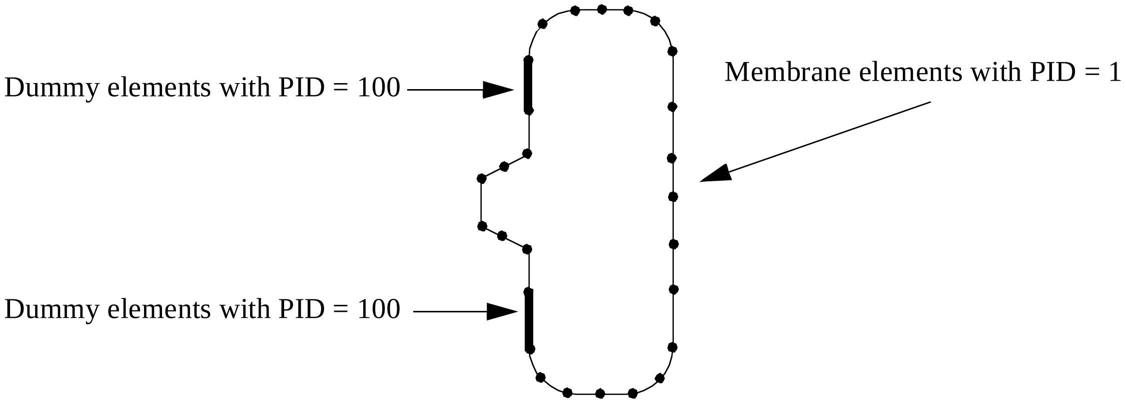
```
COUPLE, 1, 10, INSIDE
SURFACE, 10, , PROP, 10,
SET1, 10, 100, 200
PSHELL, 100, 100, 0.05
PSHELL1, 200, , DUMMY
```

Care must be taken when doing so, however. The additional grid points created for the dummy elements do not move, since they are not connected to any structural elements. When the shell elements move so far that they pass beyond these stationary grid points, the coupling surface turns inside out and has a negative volume, causing MSC.Dytran to terminate.

Porosity

A newer and better methodology for modeling porosity has been implemented for air bags. Please see Section 2.17 on page 2-127. The PORFLOW entry as described in this section is still supported as a general purpose capability.

The coupling surface or parts of it can be made porous by referring to a COUPOR entry from the COUPLE entry. This will be further explained by means of an example. The example models an air bag with porous material and two holes:


Dummy elements with PID = 100
Membrane elements with PID = 1
Dummy elements with PID = 100

The required input is:

```
PSHELL1, 1, , MEMB, , , , , ,+
+, 1.E-3
PSHELL1, 100, , DUMMY
SURFACE, 11, , PROP, 11
SET1, 11, 1, 100

SUBSURF, 22, 11, PROP, 100
SET1, 1, 1
SET1, 100, 100

COUPLE, 1, 11, OUTSIDE, , , 55
COUPOR, 1, 55, , PORFLOW, 42, CONSTANT, 0.009
COUPOR, 2, 55, 22, PORFLOW, 42, CONSTANT, 1.0
PORFLOW, 42, , MATERIAL, 33, PRESSURE, 1.E-5, METHOD, PRESSURE
```

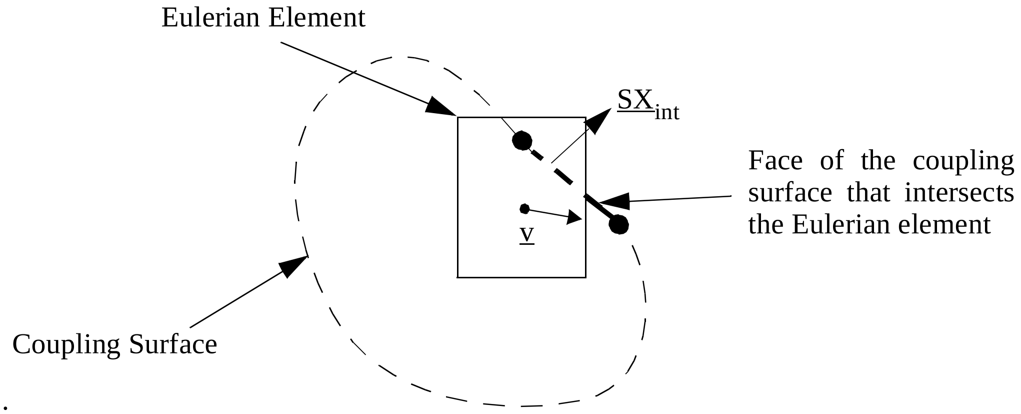
Two different algorithms are available to calculate the mass transport through the coupling surface. The desired method can be activated from the PORFLOW entry.

1. Velocity Method

   This algorithm is activated by:

   PORFLOW, 42, , MATERIAL, 33, PRESSURE, 1.E-5, METHOD, VELOCITY

The transport of mass through the porous area is based on the velocity of the gas in the Eulerian elements, relative to the moving coupling surface.



The volume of the Eulerian material transported through the faces of the coupling surface that intersect an Eulerian element is equal to

$$V_{trans} = -dt \cdot \text{COEFFV} \cdot (\underline{v} \cdot \underline{SX}_{int}) \tag{2-15}$$

where V_{trans} = transported volume during one time step ($V_{trans} > 0$ for outflow; $V_{trans} < 0$ for inflow)

dt = time step

COEFFV = porosity coefficient

$\underline{v}$ = velocity vector of the gas in the Eulerian mesh

$\underline{SX}_{int}$ = area of the face of the coupling surface that intersects the Eulerian element $\|\underline{SX}_{int}\|$ is equal to the area of the face that lies inside the Eulerian element.

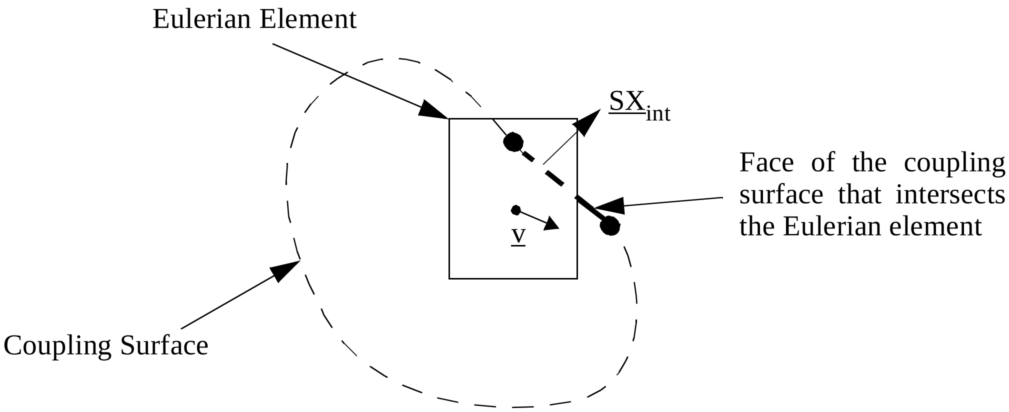
The transported mass through the porous area is equal to the density of the gas times the transported volume.

2. Pressure method

This algorithm is activated by:

PORFLOW, 42, , MATERIAL, 33, PRESSURE, 1.E5, METHOD, PRESSURE.

The transport of mass through the porous area is based on the pressure difference between the gas in the Eulerian element and the outside pressure. The outside pressure is the pressure as specified on the PORFLOW entry.


Eulerian Element
SXint
Face of the coupling surface that intersects the Eulerian element
v
Coupling Surface

The volume of the Eulerian material transported through the faces of the coupling surface that intersect an Eulerian element is equal to:

$$V_{trans} = dt \cdot \text{COEFFV} \cdot (\underline{SX}_{int} \cdot \underline{SX}_{int}) \cdot \sqrt{\frac{2 \cdot p \cdot \rho \cdot \gamma}{(\gamma - 1)} \cdot \left(\left(\frac{p_{exh}}{p}\right)^{\left(\frac{2}{\gamma}\right)} - \left(\frac{p_{exh}}{p}\right)^{\left(\frac{\gamma+1}{\gamma}\right)}\right)} \tag{2-16}$$

where V_{trans} = transported volume during one time step ($V_{trans} > 0$ for outflow; $V_{trans} < 0$ for inflow)

dt = time step

COEFFV= porosity coefficient

$\underline{SX}_{int}$ = area of the face of the coupling surface that intersects the Eulerian element $\|\underline{SX}_{int}\|$ is equal to the area of the face that lies inside the Eulerian element

p = pressure of the gas in the Eulerian element

ρ = density of the gas in the Eulerian element

γ = adiabatic exponent (= C_p / C_v)

p_{exh} = pressure at the face

The pressure at the face is approximated by the one-dimensional isentropic expansion of the gas to the critical pressure or the environmental pressure according to

$$p_{exh} = \begin{cases} p_{env} & (p_{env} > p_c) \\ p_c & (p_{env} < p_c) \end{cases} \tag{2-17}$$

where p_c is the critical pressure.

$$p_c = p \cdot \left(\frac{2}{\gamma + 1}\right)^{\left(\frac{\gamma}{\gamma - 1}\right)} \qquad (2\text{-}18)$$

In case the outside pressure is larger than the pressure of the gas, inflow through the coupling surface will occur. This porosity model can only be used for materials that use the gamma law equation of state.

Efficiency

The coupling algorithm requires a very large number of calculations to determine how the coupling surface intersects the Eulerian mesh. The coupling surface should therefore be as small as possible to make the solution efficient.

A subcycling technique can also be used to improve the efficiency. When subcycling is used, the geometry of the coupling surface is not updated every time step but only when necessary based on the motion of the surface. Subcycling is automatically controlled by MSC.Dytran. The frequency of geometrical updates varies during the calculation, depending on the motion of the coupling surface. The parameters COSUBCYC and COSUBMAX control the subcycling process. Subcycling is switched on using the COSUBCYC parameter.

2.10 Multiple Coupling Surfaces with Failure

In combination with the fast coupling algorithm, the PARAM,FASTCOUP entry, and the Roe solver (the PARAM,LIMITER,ROE entry, see Section 2.21 on page 2-136), it is possible to define multiple coupling surfaces which can fail or be deactivated. With each coupling surface, an Eulerian region that consists of a mesh that is aligned with the basic coordinate system-axes, is defined by using the MESHID or SET1ID field on the COUPLE1 entry.

Failure of a coupling surface is possible by defining a failure model for the Lagrangian elements (CQUADs and/or CTRIAs) attached to the coupling surface, and by setting the PARAM,FASTCOUP, ,FAIL entry (see Section 2.5.8 on page 2-77). When an element fails, and it is attached on both sides to a coupling surface with an Eulerian region, mass can flow through the coupling surface by defining an interaction between those coupling surfaces using the COUP1INT entry. However, when you do not define an interaction, the coupling surface can still fail, but in that case, default surrounding variables will be taken to calculate the in- or outflow. These variables can be defined by the COUP1FL entry.

In case of multiple coupling surfaces it is also possible to deactivate a coupling surface and its associated Eulerian region at a certain time using the TDEAC field on the COUPLE1 entry. The deactivation will stop the calculation of the coupling surface and its associated Eulerian region, but the calculation of the Lagrangian structure will continue. Activation of the coupling surface is not possible.

2.11 Arbitrary Lagrange-Euler (ALE) Coupling

Definition

As stated for the general coupling, ALE also acts to enable the material modeled by the Eulerian and Lagrangian meshes to interact. The two meshes initially must be coupled to each other by an ALE interface surface. The Lagrangian and Eulerian grid points in the interface surface coincide in physical space but are distinct in logical space. The interface serves as a boundary for the flowing Eulerian material during the analysis. The Eulerian material exerts pressure on the Lagrangian part of the interface that is distributed as forces to the Lagrangian grid points.

The interface moves as the Lagrangian structure deforms. Thus, the Eulerian mesh boundary also moves. In order to preserve the original Eulerian mesh and have it follow the structural motion, the Eulerian grid points can be defined as ALE grid points. In that case the motion of the ALE interface is propagated through the Eulerian mesh by the ALE motion algorithm.

In an ALE calculation the Eulerian material flows through the mesh while the mesh can also move. The material can have a velocity relative to the moving mesh, which makes it an Eulerian formulation.

The ALE interface can not be used in combination with Eulerian single material elements with strength and in combination with the Roe Solver.

Efficiency

Since the ALE coupling does not require geometrical calculations during the analysis, it is potentially faster than the general coupling. However, the deformation of the structure at the interface should be smooth but not necessarily small. Bird-strike analyses are typical ALE applications. The deformation is usually large but smooth in time.

2.12 Dynamic Relaxation

Definition

Dynamic Relaxation (DR) is a process that uses a damping concept to find the steady-state part of a dynamic solution to a transient response. In general, problems, especially those with highly nonlinear geometric and material behavior, can be treated with an explicit DR method. In many cases, however, the number of iterations needed to reach convergence can be very large.

MSC.Dytran offers two possible ways of dynamic relaxation to find a static solution of a structural mechanic problem. The static part of the dynamic solution is found by introducing damping in the iterative solution scheme that is used to solve the equations of motion.

Alpha Damping (VISCDMP)

The Alpha-type of dynamic relaxation uses a single damping parameter that is introduced in the central difference integration scheme of the equations of motion

$$v^{n+1/2} = v^{n-1/2}(1-\alpha) + a^n \Delta t^n \tag{2-19}$$

where v denotes the grid-point velocity, a is the acceleration, Δt is the time step, and α is the dynamic relaxation parameter (the damping coefficient). The DR parameter can be individually defined for each available structural element type in MSC.Dytran and is input on the VISCDMP entry.

The choice of the DR parameter(s) depends on the natural frequencies of the system. The critical damping α should be taken to be approximately 5/3 times the critical damping (or 5/3 times the natural frequency times the time step).

Global, C-Matrix, or System Damping (VDAMP)

Dynamic relaxation that uses global damping as the damping device is based on a mass-spring-damper system. The equation of motion reads

$$Ma^n + Cv^n + F_{int}{}^n = F_{ext}{}^n \tag{2-20}$$

The dynamic relaxation scheme uses the following C-matrix

$$C = \frac{2\beta}{\Delta t}M \tag{2-21}$$

All matrices are diagonal. Thus, each degree of freedom can be written as

$$m_i a_i^n + \frac{2\beta}{\Delta t} m_i v_i^n = (f_{ext}^n - f_{int}^n)_i \tag{2-22}$$

A central difference time integration scheme is applied, yielding

$$\begin{aligned} a_i^n &= (V_i^{n+1/2} - V_i^{n-1/2})\Delta t^{-1} && \text{(a)} \\ v_i^n &= 1/2(V_i^{n+1/2} + V_i^{n-1/2}) && \text{(b)} \end{aligned} \tag{2-23}$$

Combining Eqs. (2-22) and (2-23) leads to the following expression for the updated velocity

$$v_i^{n+1/2} = \frac{1-\beta}{1+\beta} v_i^{n+1/2} + \frac{\Delta t}{1+\beta}\left(\frac{f_{ext}^n - f_{int}^n}{m_i}\right) \tag{2-24}$$

where the parameter β is input on the VDAMP entry.

Equation (2-21) can also be written as

$$m_i a_i + b_i v_i = -k_i d_i \tag{2-25}$$

that describes the dynamic motion of a damped, single-degree-of-freedom system. The natural frequency of such a system is found to be

$$\omega_i = \sqrt{k_i/\mathrm{m}_i} \tag{2-26}$$

Critical damping is defined by

$$b_i^{crit} = 2m_i\omega_i = 2\sqrt{k_i m_i} \tag{2-27}$$

Or, in terms of the dynamic relaxation parameter

$$\beta_i^{crit} = \omega_i \Delta t = \sqrt{k_i/\mathrm{m}_i}\ \Delta t \tag{2-28}$$

For a system with one degree of freedom, with a constant time step, and with $(f_{ext} - f_{int}) = -kd$, the dynamic relaxation parameter β can be related directly to a percentage of critical damping. This is shown in the following example. Such a direct relation is not possible for structures that have a lot of different natural frequencies. In those cases, the dynamic relaxation parameter should be set so that it corresponds to the lowest natural frequency. Also, the time step changes during the calculation, making it less easy to relate the relaxation parameter to a natural frequency.

Remarks

Always be very careful when using damping in general, especially if there are large nonlinearities in the solution. Nonlinear solutions are path dependent, and artificially introducing a source of viscosity (damping) might interfere with the solution path.

In regard to the efficiency of the dynamic relaxation, keep in mind that it can require a large number of time steps to reach convergence, as mentioned previously. This is the case in those problems where the ratio between the largest and the smallest natural frequency is large. In such cases, the stable explicit time step is very small compared to the period corresponding to the largest natural frequency. It is very often advantageous to use an implicit code such as MSC.Nastran in these situations to find the static part of the solution and use this as an initial state. MSC.Dytran also supports this capability (NASINIT).

Example

m = 1 kg

k = 225 kg/sec2 (=N/m)

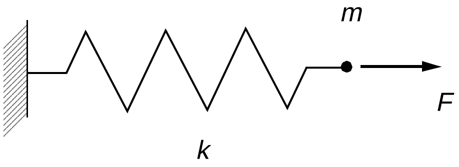
f = 50 N

Δt= 1 msec

Natural frequency $\omega = \sqrt{k/\mathrm{m}} = 15$ (rad/sec)

Period $T = 2\pi/\omega = 0.4188$ (sec)

Critical damping $\beta = \omega\Delta t = 0.015$



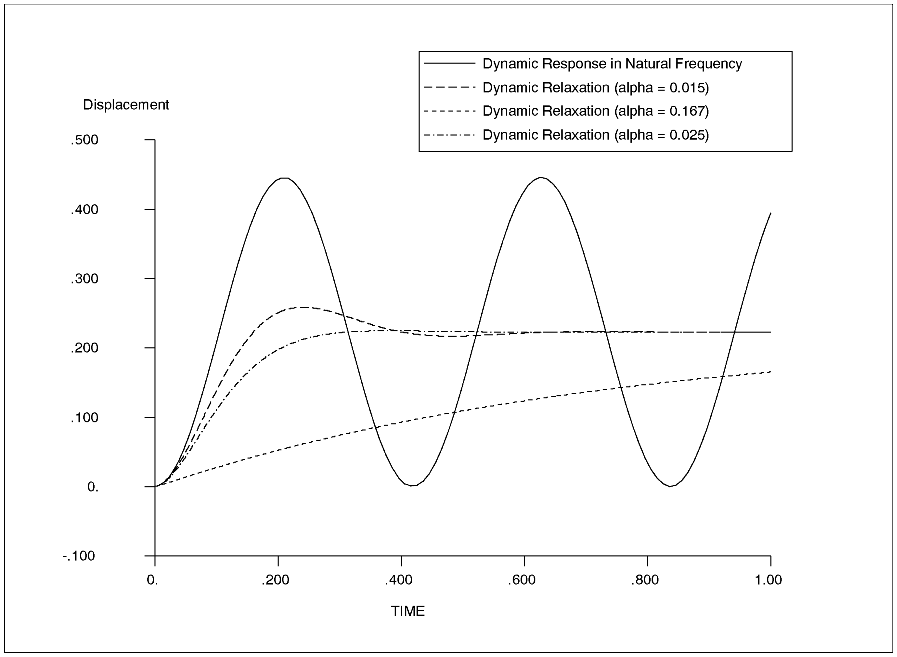


Figure 2-31. Solution for Different Values of the Dynamic Relaxation Parameter (α).

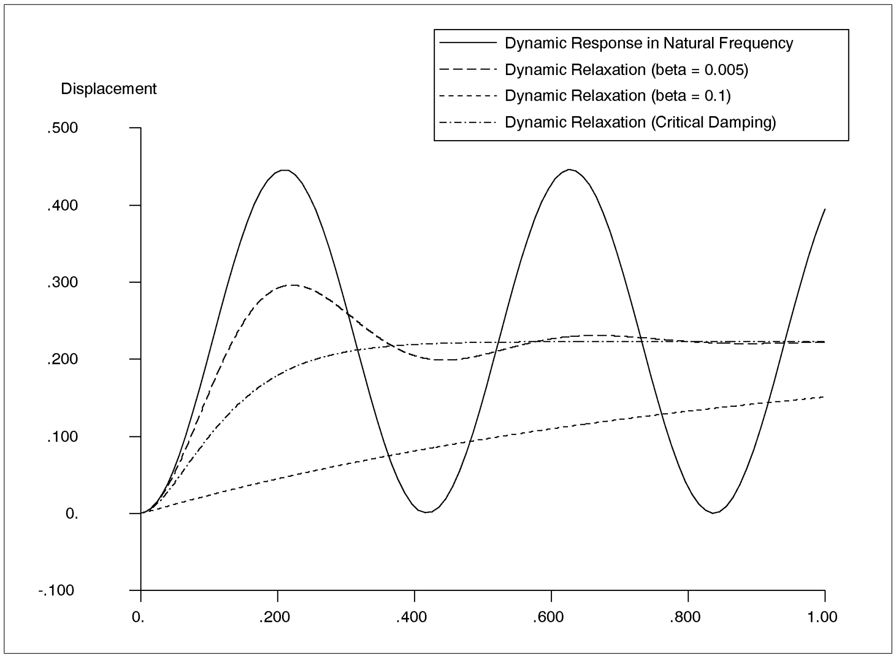


Figure 2-32. Solution for Different Values of the Dynamic Relaxation Parameter (β).

2.13 Seat Belts

2.13.1 Definition

A seat belt constraint system can be modeled within MSC.Dytran using a special belt element. The element has the following characteristics:

- Tension-only nonlinear spring with mass.
- User-defined loading and unloading path.
- Damping is included to prevent high-frequency oscillations.
- Possible to prestress and/or feed additional slack.

A special contact algorithm is available to model the contact between the belt elements and an occupant model.

2.13.2 Seat Belt Material Characteristics

You can specify the following material characteristics on a PBELT entry:

Loading and Unloading Curves

The loading/unloading curves are defined in a TABLED1 entry specifying the force as a function of strain. The strain is defined as engineering strain

$$\varepsilon^n = \frac{I^n - I^o}{I^o}$$

where I^n is the length at time *n* and I^o is the length at time zero.

The loading and unloading curves must start at (0, 0).

Upon unloading, the unloading curve is shifted along the strain axis until it intersects the loading curve at the point from which unloading commences. An example of a typical load, unload, and reload sequence is shown in Figure 2-33.

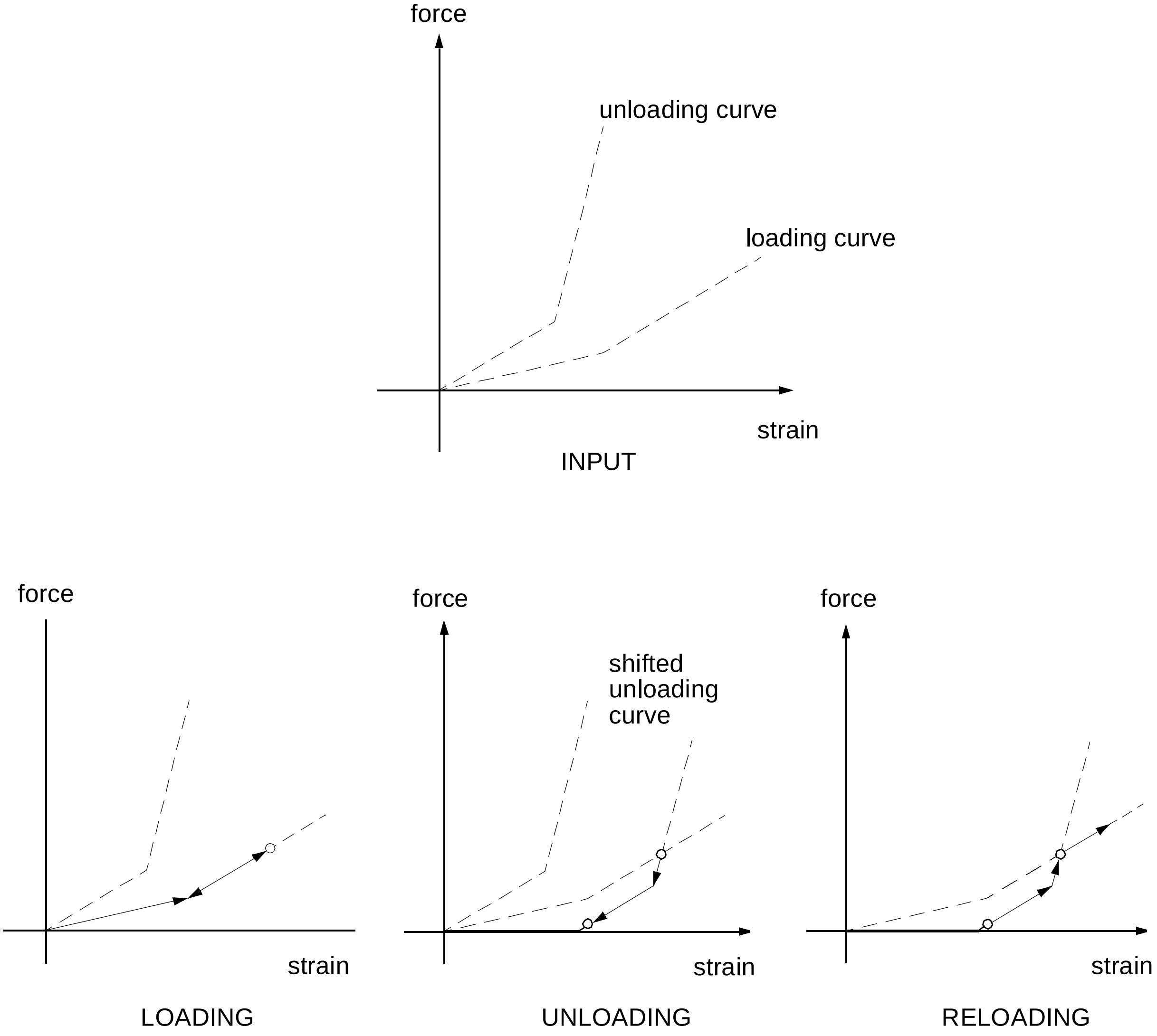


Figure 2-33. Seat Belt Loading and Unloading Characteristics.

The unloading table is applied for unloading and reloading until the strain again exceeds the point of intersection. At further loading, the loading table will be applied.

Seat Belt Element Density

The density of the belt elements is entered as mass per unit length. The density is used during initialization to distribute the mass to the grid points. The grid points masses are used to calculate damping and contact forces.

Damping Forces

A damping force is added to the internal force to damp high-frequency oscillations. The damping force $\vec{F}_D$ is equal to

$$\vec{F}_D = \alpha_1 M \cdot \frac{\vec{V}_{G1} - \vec{V}_{G2}}{\Delta t}$$

where α_1 is the damping factor CDAMP1 as defined on the PBELT entry, *M* is the element mass, $\vec{V}_{G1}$ and $\vec{V}_{G2}$ denote the velocity of grid point 1 and grid point 2 of the element respectively. Δt is the time step.

The damping force $\vec{F}_D$ is limited to

$$\vec{F}_D = max(\vec{F}_D, \alpha_2 \vec{F}_S)$$

where α_2 is the damping coefficient CDAMP2 as defined on the PBELT entry, and $\vec{F}_S$ is the internal force in the element.

Slack

Additional slack can be fed into the belt elements as a function of time. The slack is specified in engineering strain and will be subtracted from the element strain at time *n* as

$$\varepsilon^n = \varepsilon^n - \varepsilon^n_{slack}$$

where ε^n_{slack} denotes the slack strain as found from the TABLED1 definition in the input file.

The force in the element will be zero until the element strain exceeds the slack.

Prestress

The seat belt elements can be prestressed as a function of time. The prestress strain is specified in engineering strain and will be added to the element strain at time *n* as

$$\varepsilon^n = \varepsilon^n + \varepsilon^n_{prestress}$$

where $\varepsilon^n_{prestress}$ is the prestress strain as found from the TABLED1 definition in the input file.

As a result, the elements will build up a tensile force.

2.14 Drawbead Model

The success of the deep-drawing process strongly depends on the extent of slip of the blank at the interface, controlled by the blank holder force and the friction conditions at the interface between the blank and the blank holder and die. Besides a blank holder, drawbeads are commonly used in sheet metal forming processes to provide an additional local control of plastic sheet deformation and thereby the amount of sheet material moving into the die cavity.

The modeling of drawbeads by a finite element mesh is often not feasible. Apart from the drawbead geometry, the region of the blank which slides through the drawbead itself would require a very fine FE mesh. This will increase the total number of elements and decrease the time step of the calculation significantly.

The drawbead option in the contact is an efficient way to locate the grid points in the blank that are moving across the drawbead line, and therefore, need to be restrained by a drawbead force.

You can input a list of grid points to define the position of the drawbead. The list of grid points must be ordered along the drawbead line and is used to define a row of dummy rod elements. The dummy rod elements representing the drawbead must be connected to the tool from which the drawbead restraining force will be applied on the blank. The connection between the drawbead grid points and the tool is achieved by using the rigid connection (RCONN entry).

The restraining force per unit of drawbead length must be supplied by the user and entered by means of the drawbead option in the contact entry. MSC.Dytran will calculate the drawbead length associated with each drawbead grid point. At every time step, the appropriate drawbead force is applied as a localized restraining force on the blank.

Example of Modeling Procedure

CRODs to locate the DRAWBEAD line. The stiffness is chosen such that the rods will not determine the time step. The mass is chosen such that the rods will not add significant mass to the blank holder.

```
CROD, 501, 5, 5001, 5002
SET1, 51, 5001, 5002
PROD, 1, 5, 1.E-20
MAT1, 5, 1.E-20, , 0.3, 1.E-10
```

Define a rigid connection (RCONN) between the drawbead grid points (GRIDset = 51) and the tool (SURFace ID = 11). Note that gaps between the GRIDset and the tool will be automatically closed (CLSGAP = YES).

```
RCONN, 1, GRID, SURF, 51, 11, , , ,+
+, , , , , , , , ,+
+, YES
```

Define the drawbead restraining force per unit length on the CONTACT entry. The force is applied via GRIDset = 51 on the blank (SURFace ID = 1). Note that the contact thickness is taken into account.

```
CONTACT, 1, GRID, SURF, 51, 1, , , ,+
+, DRAWBEAD, , , , 1.0, , , ,+
+, , , , , , , , ,+
+, , , , , , , , ,+
+, <force/length>
```

2.15 Application Sensitive Default Setting

2.15.1 Introduction

MSC.Dytran is capable of handling an extensive variety of applications. Due to the variety it is sometimes difficult for you to make the correct choice of default settings according to the application at hand. Application sensitive default settings make it easier to select the appropriate element formulations or numerical algorithms to achieve the best solution possible in terms of accuracy and CPU time. By default, MSC.Dytran attempts to provide you with the most accurate solution possible. This default setting can also be achieved by including a SETTING,SID,STANDARD entry in your input file. An overview is given in Section 2.15.4 on page 2-123. which settings are automatically done when the default applies.

2.15.2 Overview of Default Definition

Element Formulation

Shell elements:

- Key-Hoff elements are used with three integration points through the element thickness.
- The element thickness is strain dependent.
- The element transverse shear stresses are computed assuming a linear distribution of the stress.
- Shear-locking is avoided.

Solid elements:

- One-point Gauss integration (PSOLID).

Hourglass Suppression Method

Shell elements:

- Flanagan-Belytschko viscous (FBV) where the warping coefficient is equal to 0.1 and rigid body rotation correction is not active.

Solid elements:

- Flanagan-Belytschko stiffness (FBS).

Method for Material Plasticity Behavior

Shell elements:

- Plasticity is treated by an iterative scheme, using as many iterations as necessary (the total number of iterations is limited to 20).

Note: The method for material plasticity behavior does not apply to all material models available. For example, the SHEETMAT material model applies a special algorithm that does not require an iterative method.

2.15.3 Application Type Default Setting

In addition to the STANDARD definition, four other application types are available to influence the default settings:

- CRASH – The defaults set for optimal crash-type analysis.
- SHEETMETAL – The defaults set for optimal sheet metal forming analysis.
- SPINNING – The defaults set for optimal fast rotating structures.
- FAST – The defaults set for optimal fast, but not necessarily the most accurate solutions.
- VERSION2 – The defaults set to pre-Version 3.0.

The resulting default settings are listed below for each of the above mentioned applications.

2.15.3.1 Crash

Element Formulation

Shell elements:

- BLT (Belytschko-Lin-Tsay) elements are used with three integration points through the element thickness.
- The element thickness is strain dependent.
- The element transverse shear stresses are assumed constant through the element thickness.
- Shear-locking is not avoided.

Solid elements:

- One-point Gauss integration (PSOLID).

Hourglass Suppression Method

Shell elements:

- Flanagan-Belytschko viscous (FBV) where the warping coefficient is equal to 0.1 and rigid body rotation correction is not active.

Solid elements:

- Flanagan-Belytschko stiffness (FBS).

Method for Material Plasticity Behavior

Shell elements:

- Plasticity is treated by an iterative scheme, using as many iterations as necessary (the total number of iterations is limited to 20).

2.15.3.2 Sheet Metal

Element Formulation

Shell elements:

- BLT (Belytschko-Lin-Tsay) elements are used with five integration points through the element thickness.
- The element thickness is strain dependent.
- The element transverse shear stresses are computed assuming a constant distribution of the stress.
- Shear-locking is not avoided.

Solid elements:

- One-point Gauss integration (PSOLID).

Hourglass Suppression Method

Shell elements:

- Flanagan-Belytschko viscous (FBV) where the warping coefficient is equal to 0.1 and rigid body rotation correction is not active.

Solid elements:

- Flanagan-Belytschko stiffness (FBS).

Method for Material Plasticity Behavior

Shell elements:

- Plasticity is treated by an iterative scheme, using as many iterations as necessary (the total number of iterations is limited at 20).

2.15.3.3 Spinning

Element Formulation

Shell elements:

- Key-Hoff elements are used three integration points through the element thickness.
- The element thickness is strain dependent.
- The element transverse shear stresses are computed assuming a linear distribution of the stress through the element thickness.
- Shear-locking is avoided.

Solid elements:

- One-point Gauss integration (PSOLID).

Hourglass Suppression Method

Shell elements:

- DYNA method with the warping coefficient set to zero and rigid body rotation correction is active.

Solid elements:

- Original DYNA suppression method.

Method for Material Plasticity Behavior

Shell elements:

- Plasticity is treated by an iterative scheme, using as many iterations as necessary (the total number of iterations is limited to 20).

2.15.3.4 Fast

Element Formulation

Shell elements:

- BLT (fast Belytschko-Lin-Tsay) elements are used with three integration points through the element thickness.
- The element thickness is constant.
- The element transverse shear stresses are assumed to be constant through the element thickness.
- Shear-locking is not avoided.

Solid elements:

- One-point Gauss integration (PSOLID).

Hourglass Suppression Method

Shell elements:

- Flanagan-Belytschko viscous (FBV) where the warping coefficient is equal to 0.1 and rigid body rotation correction is not active.

Solid elements:

- Original DYNA suppression method.

Method for Material Plasticity Behavior

Shell elements:

- Plasticity is treated as a one step radial scale back scheme.

2.15.3.5 Version2

Element Formulation

Shell elements:

- Bely (original MSC/DYNA Belytschko-Lin-Tsay) elements are used with three integration points through the element thickness.
- The element thickness is constant.

- The element transverse shear stresses are assumed to be constant through the element thickness.
- Shear-locking is not avoided.

Solid elements:

- One-point Gauss integration (PSOLID).

Hourglass Suppression Method

Shell elements:

- Flanagan-Belytschko viscous (FBV) where the warping coefficient is equal to 0.1 and rigid body rotation correction is not active.

Solid elements:

- Original DYNA suppression method.

Method for Material Plasticity Behavior

Shell elements:

- Plasticity is treated as a one step radial scale back scheme.

2.15.4 Hierarchy of the Scheme

MSC.Dytran has many more ways to influence the setting of defaults and to select a certain numerical algorithm. For consistency, the application sensitive defaults work in a hierarchical order. This is explained in the following sections.

2.15.4.1 Global and Property Specific Default Definition

The application sensitive defaults can be specified on a global level; i.e., by including a SETTING entry with an application type definition in the input file, but also for specific properties. For example, if your application is CRASH but you have some spinning parts in your model, you can define the global defaults by including the entry SETTING,SID1,CRASH and the specific default setting for the property by SETTING,SID2, and SPINNING,SHELL,PID2. This will result in a global setting of defaults according to CRASH, except for the shell elements that have property number PID2 that will use the defaults necessary for a SPINNING application.

2.15.4.2 Shell Formulation

The shell formulation can always be globally changed using the entry PARAM,SHELLFORM, (formulation type) irrespective of the SETTING entries present in the input file. The ways of shell formulation definition in order of increasing priority is SETTING, PARAM,SHELLFORM, PSHELL1 or PCOMPA.

The thickness of the elements can be made strain independent by including the PARAM,SHTHICK,NO entry in the input file. All application types will then use this as the default, except for SHEETMETAL.

The method for material plasticity can be altered by including the entry PARAM,SHPLAST, (RAD,VECT,ITER) in the input file. All application types will then apply this setting as the default except for VERSION2 which always applies the radial return method (RADIAL).

2.15.4.3 Hourglass Suppression Method

The method to prevent hourglass modes from occurring can also be defined using the HGSUPPR entry in the input file. If there are any HGSUPPR entries in the input file, these will always prevail using the hierarchical order within the hourglass definition scheme. The same applies to the hourglass method constants that can also be specifically defined on a global or on a property level.

2.16 Mass Scaling

The explicit dynamics procedure of MSC.Dytran uses relatively small time steps dictated by the shortest natural period of the mesh: the analysis cost is in direct proportion to the size of the mesh. There are two types of problems where the cost-effectiveness of the analysis can be increased:

- If a mesh consists of a few, very small (or stiff) elements, the smallest (or stiffest) element determines the time step for all elements of the mesh.
- If a few severely distorted elements are obtained by the analysis, the most distorted element determines the time step for all elements of the mesh. This may even end up with a too small stable time step.

Speedup of those problems can be achieved by using mass scaling (PARAM,SCALEMAS). Mass scaling is based on adding numerical mass to an element so that its time step will never become less than the minimum allowable time step defined by you. Note that mass scaling can be risky in areas where either inertia effects are relevant or contact with other parts is expected to occur.

2.16.1 Mass Scaling Used for Problems Involving a Few Small Elements

It is common practice that meshing of real-life problems may involve some relatively small elements: elements frequently localized in a kind of transition region and meant to connect large structural parts to each other. Those elements will determine the time step of the whole calculation although they might be present in the model to a very limited extent. Speedup can be realized by using mass scaling. Some guidelines:

- Make a run for one cycle and retrieve the time step of all elements by requesting ELDLTH.
- By using a postprocessing program, see which elements are determining the time step and filter out the elements whose time steps exceed a user-defined minimum (DTMIN).
- See what the impact would be of specifying this new time step (DTMIN). Select the value of DTMIN such that hardly any elements would be scaled in the area of interest (for example, as much as possible outside the impact region in a crash simulation).

2.16.2 Mass Scaling Used for Problems Involving a Few Severely Distorted Elements

There are conceivable application areas where elements are distorted to such a high extent that a few of them will determine the time step for all elements of the mesh. For example, crushing of a subfloor structure frequently involves failure modes associated with the occurrence of severely distorted elements. Modeling this kind of crushing behavior without including a failure mechanism might end up with a stable time step that is too small. Since those elements are often present in a relatively small region, the mass scaling method might be a good means to artificially speed up the calculation without losing the capability to model the global crushing behavior. Note that to prevent severely distorted elements, it is recommended that a proper failure mechanism be included, instead of coping with the distorted elements by making use of the mass scaling method. Some guidelines are as follows:

- Since you do not know in advance which elements will become too distorted, you should first run the analysis as far as possible (without defining PARAM,SCALEMAS). You should request the time step of all elements (ELDLTH).
- If the problem ends up with a too small stable time step, the analysis will finish prematurely. See which elements are so severely distorted and decide what a reasonable minimum time step (DTMIN) might be without affecting elements in the area of interest. See the guidelines of the previous section.
- Rerun the analysis specifying PARAM,SCALEMAS if the region of highly distorted elements is relatively small compared to the whole model.
- If there is too much mass added to the grid points of those elements, the model might show significantly different inertia effects, and subsequently, different global structural response. In order to avoid this, no more mass will be added if the numerically added mass exceeds a certain percentage (MXPERC).
- To limit the amount of overhead time spent on checking against its mass scaling criterion, the checking is only done every specified number of STEPS.

2.17 Porosity in Air Bags

2.17.1 Definition and Input File Entries

Porosity is defined as the flow of gas through the air bag surface. There are two ways to model this:

- Holes: The air bag surface contains a discrete hole.
- Permeability: The air bag surface is made from material that is not completely sealed.

The same porosity models are available for both the uniform pressure air bag model as the Eulerian coupled air bag model. The porous flow can be either to and from the environment or into and from another uniform pressure model.

The following table shows the available porosity models and their usage:

Entry	Flow Through	Flow To/From
PORHOLE	hole	environment
PERMEAB	permeable area	environment
PORFGBG	hole	another uniform pressure model airbag
PERMGBG	permeable area	another uniform pressure model airbag

The following entries are required to activate the different porosity models. The id's are chosen arbitrarily, but are unique for each cross reference. See the individual manual pages for further explanation of the fields. The model incorporates a switch from the Eulerian coupled air-bag model to the uniform pressure air-bag model at 50 msecs.

Flow Through a Hole to the Environment

air-bag surface	: SURFACE,1,.....
porous area	: SUBSURF,10,1,....
uniform pressure model	: GBAG,20,1,,,30
porosity for subsurface	: GBAGPOR,40,30,10,PORHOLE,50, ,<coeffv>
Eulerian coupled model	: COUPLE,60,1,OUTSIDE, ,70
porosity for subsurface	: COUPOR,80,70,10,PORHOLE,50, ,<coeffv>
hole characteristics	: PORHOLE,50,,,BOTH,<penv>,<rhoenv>,<sieenv>

Euler to GBAG switch (at 50.E-3 secs)	: GBAGCOU,101,60,20,50.E-3,1.E20

Flow Through Permeable Area to the Environment

air-bag surface	: SURFACE,1,.....
porous area	: SUBSURF,10,1,....
uniform pressure model	: GBAG,20,1,,,30
porosity for entire bag	: GBAGPOR,40,30,0,PERMEAB,50
porosity for subsurface	: GBAGPOR,40,30,10,PERMEAB,50
Eulerian coupled model	: COUPLE,60,1,OUTSIDE,,,70
porosity for entire bag	: COUPOR,80,70,0,PERMEAB,50
porosity for subsurface	: COUPOR,80,70,10,PERMEAB,50
permeab characteristics:	
linear	: PERMEAB,50,1.E-4, ,BOTH,<penv>,<rhoenv>,<sieenv>
tabular	: PERMEAB,50, ,90,BOTH,<penv>,<rhoenv>,<sieenv>
	: TABLED1,90,.....
Euler to GBAG switch (at 50.E-3 secs)	: GBAGCOU,101,60,20,50.E-3,1.E20

Flow Through a Hole to Another Uniform Pressure Air Bag

<u>air bag 1</u>

air-bag surface	: SURFACE,1,.....
porous area	: SUBSURF,10,1,....
uniform pressure model	: GBAG,20,1,,,30
porosity for subsurface	: GBAGPOR,40,30,10,PORFGBG,50, ,<coeffv>
Eulerian coupled model	: COUPLE,60,1,OUTSIDE, , ,70
porosity for subsurface	: COUPOR,80,70,10,PORFGBG,50, ,<coeffv>
hole characteristics	: PORFGBG,50,,,BOTH,1020
Euler to GBAG switch (at 50.E-3 secs)	: GBAGCOU,101,60,20,50.E-3,1.E20

air bag 2

air-bag surface	: SURFACE,1001,.....
porous area	: SUBSURF,1010,1001,....
uniform pressure model	: GBAG,1020,1001

Note that the porosity characteristics need to be defined for air bag 1 only. The gas will automatically flow from bag 1 into bag 2 and vice versa.

Flow Through a Permeable Area to Another Uniform Pressure Air Bag

air bag 1

air-bag surface	: SURFACE,1,.....
porous area	: SUBSURF,10,1,....
uniform pressure model	: GBAG,20,1,,,30
porosity for entire bag	: GBAGPOR,40,30,0,PERMGBG,50
porosity for subsurface	: GBAGPOR,40,30,10,PERMGBG,50
Eulerian coupled model	: COUPLE,60,1,OUTSIDE,,,70
porosity for entire bag	: COUPOR,80,70,0,PERMGBG,50
porosity for subsurface	: COUPOR,80,70,10,PERMGBG,50
PERMGBG characteristics:	
linear	: PERMGBG,50,1.E-4, ,BOTH,1020
tabular	: PERMGBG,50, ,90,BOTH,1020
	: TABLED1,90,.....
Euler to GBAG switch (at 50.E-3 secs)	: GBAGCOU,101,60,20,50.E-3,1.E20

air bag 2

air-bag surface	: SURFACE,1001,.....
porous area	: SUBSURF,1010,1001,....
uniform pressure model	: GBAG,1020,1001

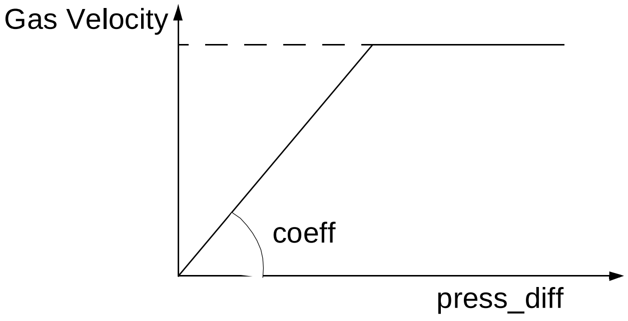
Note that the porosity characteristics need to be defined for air bag 1 only. The gas will automatically flow from bag 1 into bag 2 and vice versa.

2.17.2 Permeability

Permeability is defined as the velocity of gas through a surface area depending on the pressure difference over that area.

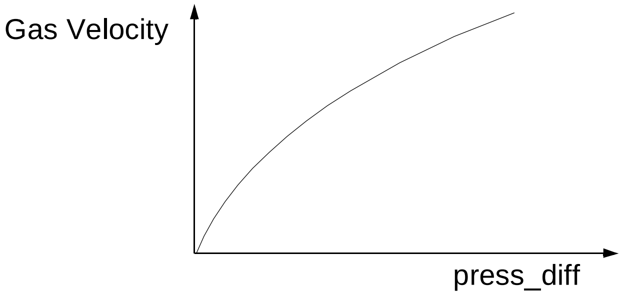
On the PERMEAB and PERMGBG entries, permeability can be specified by either a coefficient or a pressure dependent table:

a. Coefficient: Massflow = coeff*pressure_difference



$$\text{coeff} = \frac{\delta(\text{massflow})}{\delta(\text{press\_diff})}$$

b. Table:



pressure dependent table

The velocity of the gas flow can never exceed the sonic speed:

$$V_{max} = V_{sonic} = \sqrt{\gamma R T_{crit}}$$

where γ is the gasconstant of in- or outflowing gas, and T_{crit} is the critical temperature.

The critical temperature can be calculated as follows:

$$\frac{T_{crit}}{T_{gas}} = \frac{2}{(\gamma + 1)}$$

where T_{gas} is the temperature of outflowing gas.

2.17.3 Holes

Flow through holes as defined on the PORHOLE or PORFGBG entries is based on the theory of one-dimensional gas flow through a small orifice. The formulas to calculate the velocity of the gas are the same as for the PORFLOW with the pressure method. The formulas are given in Section 2.9 on page 2-101, General Coupling.

2.18 Inflator in Air Bags

There are several methods available to define an inflator in airbag analyses. The most enhanced and the most preferred method is described here.

For both the uniform pressure model (GBAG) and the Euler coupled model (COUPLE), the inflator location and area are defined by means of a subsurface (SUBSURF), which must be part of the GBAG and/or COUPLE surface. The characteristics of the inflator are specified on an INFLATR entry. This entry references tables for the mass flow rate and the temperature of the inflowing gas.

A model can be defined containing both an Euler coupled model (COUPLE), as a uniform pressure model for the airbag (GBAG). It is possible to define these two options with identical inflator characteristics. This allows use of the GBAGCOU entry to switch from the Euler coupled model to the Uniform pressure model during the calculation. When the same airbag surface is referenced from both a COUPLE and a GBAG entry, the GBAGCOU switch must be present in the input file.

An inflator in an airbag analysis specified using the following input:

air-bag surface	: SURFACE,1,.....
inflator area	: SUBSURF,10,1,....
uniform pressure model	: GBAG,20,1,,,,40,....
inflator for subsurface	: GBAGINFL,50,40,10,INFLATR,50, ,<coeffv>
Eulerian coupled model	: COUPLE,60,1,OUTSIDE,,,,,,+ +070
inflator for subsurface	: COUINFL,90,70,10,INFLATR,50,,<coeffv>
inflator characteristics	: INFLATR,50,130,,912.,1.4,286.
Euler to GBAG switch (at 50.E-3 secs)	: GBAGCOU,101,60,20,50.E-3,1.E20

Note that it is possible to define multiple inflators per airbag module, by defining a set of COUINFL and/or GBAGINFL entries with the same value in the third field. This is the set ID. Each inflator can reference its own tables for massflow rate and temperature.

2.19 Heat Transfer in Air Bags

For air bags with high temperature, energy is exchanged with the environment. There are two ways to define heat transfer in airbags, convection (HTRCONV) and radiation (HTRRAD).

The heat-transfer rates due to convection and radiation are defined by:

1. Convection:

$$q_{conv} = h(t)A(T - T_{env})$$

   where $h(t)$ is the time-dependent heat-transfer coefficient, A is the (sub)surface area for heat transfer, T is the temperature inside the air bag, and T_{env} is the environment temperature.

2. Radiation:

$$q_{rad} = eAs[T^4 - T^4_{env}]$$

   where e is the gas emissivity, A the (sub)surface area for heat transfer, T is the temperature inside the air bag, and T_{env} the environment temperature.

Both types can be defined independently for the whole air bag surface, or for parts of the surface by means of SUBSURFs.

Example

air-bag surface:	SURFACE,1,.....
subsurface with heat transfer:	SUBSURF,2,1,....
subsurface with heat transfer:	SUBSURF,10,1,....
uniform pressure model:	GBAG,20,1,,,,,30
convection for whole surface:	GBAGHTR,40,30, ,HTRCONV,50,,<coeffv>
radiation for whole surface:	GBAGHTR,41,30, ,HTRRAD,50,,<>
convection for subsurface 10:	GBAGHTR,42,30,10,HTRCONV,51,,<coeffv>
radiation for subsurface 10:	GBAGHTR,43,30,10,HTRRAD,51,,<coeffv>
radiation for subsurface 2:	GBAGHTR,44,30, 2,HTRRAD,52,,<coeffv>

Eulerian coupled model:	COUPLE,60,1,OUTSIDE,,,,,,+ +,70
convection for whole surface:	COUHTR,80,70, ,HTRCONV,50,,<coeffv>
radiation for whole surface:	COUHTR,81,70, ,HTRRAD,50,,<coeffv>
convection for subsurface 10:	COUHTR,82,70,10,HTRCONV,51,,<coeffv>
radiation for subsurface 10:	COUHTR,83,70,10,HTRRAD,51,,<coeffv>
radiation for subsurface 2:	COUHTR,84,70, 2,HTRRAD,52,,<coeffv>
convection characteristics:	HTRCONV,50, 7.,,297.
convection characteristics:	HTRCONV,51,52.,,297.
radiation characteristics:	HTRRAD,50,.15,,297.,5.676-8
radiation characteristics:	HTRRAD,51,.6,,297.,5.676-8
radiation characteristics:	HTRRAD,52,.4,,297.,5.676-8
Euler to GBAG switch: (at 50.E-3 secs)	GBAGCOU,101,60,20,50.E-3,5.

2.20 Initial Metric Method for Airbags

2.20.1 Introduction

The Initial Metric Method is typically useful for airbag modeling. When using out-of-plane folding technique, the membrane elements can deform quite significantly. The final shape of the deformed bag can be negatively influenced. In order to overcome this problem, MSC.DYTRAN offers a way to initialize strains inside elements such that the final shape is preserved. It is called the Initial Metric Method, further called the IMM method. Elements can be initialized smaller than the original state, but also can be initialized larger.

- For elements that are initialized smaller, stresses will only start to build up after the original state has been reached.
- Elements that are larger will have a positive IMM strain. When growing larger, their Young's modulus is assumed to be twice as large during one time step. When shrinking there will be no stresses applied until the original state is reached.

IMM can also be applied when scaling the model of the airbag, such that the model will fit inside the inflator housing.

2.20.2 Usage

The IMM needs two models of the same airbag. One model is called Initial state and the second is called Original or Reference state.

- The Initial state model has to be part of the main input file. This state can be visualized in output requests.
- The Original state model has to be supplied to MSC.Dytran in a different file. MSC.Dytran will read this file and use the data to initialize IMM strains on the elements of the Initial state model.

The following line will activate the Initial Metric Method in MSC.Dytran (see also Section 3.5.2 on page 3-11):

```
dytran jid=<dytran_input_file> imm=<original_input_file>
```

The Initial Metric Method can be used in combination with MADYMO or ATB and can be used in any contact type. Although IMM can only be used for triangular membrane elements. The IMM-strains can be visualized in an archive file or time-history file. The variables are EXXIMM, EYYIMM and EXYIMM. A value of zero denotes that during the run, the original state was reached and IMM for that particular element is not active anymore.

2.21 Roe Solver

For gas and fluid flow, a state-of-the-art Eulerian solver is available that is based on the ideas of Professor Philip Roe. The Roe solver is based on the solution of so-called Riemann problems at the faces of the Eulerian elements. The mathematical procedure amounts to a decomposition of the problem in a discrete wave propagation problem. By including the physics of the Riemann solution at the faces, a qualitatively better solution is obtained. The Roe solver is also known as an approximate Riemann solver.

The solver can be either first or second order accurate in space. Second order spatial accuracy is obtained by applying a so-called MUSCL scheme in combination with a nonlinear limiter function. The MUSCL approach guarantees that no spurious oscillations near strong discontinuities in the flow field will occur. The scheme is total variation diminishing (TVD), meaning it does not produce new minima or maxima in the solution field. The Roe solver is activated by the PARAM,LIMITER,ROE entry in the input file.

The time integration in the Roe solver is performed by a multi-stage time integrator, also know as a Runge-Kutta type scheme. Higher order temporal accuracy can be achieved by applying multiple stages in the time integration. The PARAM,RKSCHEME entry activates the multi-stage time integration scheme.

When a coupling surface is required you have to use the COUPLE1 entry. Multiple coupling surfaces with failure can be requested if the fast coupling algorithm is used by setting the PARAM,FASTCOUP, ,FAIL entry (see Section 2.9 on page 2-101).

There are some limitations in the current implementation. The JWL equation of state is not yet supported. To analyze the effect of explosives, the blast wave approach is recommended. Eulerian elements must be completely filled with materials, so void or partial void elements are not allowed. The ALE interface to Lagrangian structures, and air bag applications are also not yet supported.

2.22 Underwater Shock Analysis (USA)

The underwater shock analysis capability is extended by the coupling of MSC.Dytran with the boundary element code USA (References 9 and 10). The coupling calculates the transient response of a totally or partially submerged structure to a spherical shock wave of arbitrary pressure profile and source location. The solution method is staggered where the structural response is calculated by MSC.Dytran and the fluid pressure response by USA. The computational model for the fluid, which is treated as an infinite or semi-infinite acoustic medium, is constructed through the use of the Doubly Asymptotic Approximation (DAA).

The MSC.Dytran-USA interface has two phases. First, there is an initialization phase of USA where the following data is given to USA by MSC.Dytran:

- Grid-point locations.
- Element connectivity.
- DOF table.
- Mass matrix.

Second, a transient phase has the function to exchange data in a staggered fashion during the combined run. The following data will be issued to USA by MSC.Dytran:

- Grid-point displacements.
- Grid-point velocities.
- Grid-point internal forces.

Finally, USA returns the following data back to MSC.Dytran:

- Pressure on elements or forces on grid points.

Example

The interface between MSC.Dytran and USA is modeled using a SURFACE entry, which is referred to by a USA entry.

```
USA, 1, 4000
SURFACE, 4000, , PROP, 40
SET1, 40, 1, 3, 5
```

It is also possible to model the interface by a SET1 entry of one-dimensional element type of grid-points or, a combination of a SURFACE and SET1 entry.

```
USA, 1, , 2
SET1, 2, 1, THRU, 11
```

Archive file output can be requested for the interface (See Section 3.7). It is also possible to get time-history file output for specific elements in the surface. Time-history output is achieved by defining a surface gauge using the SGAUGE entry. You need to define the grid-point numbers of the element on the surface.

```
CQUAD4, 94, 1, 104, 105, 109, 108
 SGAUGE, 3, 104, 105, 109, 108
```

For a detailed description of how to execute the MSC.Dytran-USA coupling see Appendix G.

CHAPTER 3

Running the Analysis

3.1 Analysis Sequence

The steps involved in running a successful analysis with MSC.Dytran are essentially the same as those involved in running a normal static analysis with any other finite element analysis code. The main difference is that a static analysis is usually run in one operation, whereas an MSC.Dytran analysis is often run in a number of stages.

The main steps involved are as follows:

1. Modeling.
2. Data translation.
3. Data check.
4. Analysis.
5. Results translation.
6. Results postprocessing.

Steps 4 through 6 are repeated for each stage of the analysis.

3.2 Using a Modeling Program with MSC.Dytran

You can produce an MSC.Dytran model in exactly the same way as you would produce any other finite element model, using MSC.Patran or another modeling package. There are a number of things to remember when modeling, and these are discussed in this section. Once your model is complete, you should use the translator supplied with your modeling package to write an MSC.Nastran input file. Since much of the input for MSC.Dytran is the same as MSC.Nastran, the MSC.Nastran style of input can be used with only minor modifications for MSC.Dytran. Virtually all modeling packages can write MSC.Nastran input files; if your modeling package does not do this, then you must write your own translator to translate the data into a form that MSC.Dytran can understand.

The vast majority of the data for an MSC.Dytran analysis can usually be created by the modeling package including the following entries:

Grid points	GRID
1-D elements	CBAR, CBEAM, CROD
Solid elements	CHEXA, CPENTA, CTETRA
Shell and membrane elements	CQUAD4, CTRIA3
Properties	PSOLID, PSHELL
Single-point constraints	SPC, SPC1
Concentrated loads	FORCE, MOMENT
Pressure loads	PLOAD, PLOAD4

Since most modeling packages are usually set up to prepare data for linear elastic analyses, a complete MSC.Dytran input file cannot be created. You need to edit the file to add features, such as rigid walls or nonlinear material properties, that cannot be created by the modeling program. Such additions are usually quite minor.

When you are creating your MSC.Dytran model, there are some points you should remember. These points are discussed in the following section.

3.2.1 Grid Points

MSC.Dytran can accept any system of grid point numbering, so there is no need to renumber your model. Try to adopt a numbering scheme where a grid point’s number is indicative of its position. Since MSC.Dytran does not have a bandwidth requirement, do not perform any bandwidth optimization on your model.

3.2.2 Elements

Remember that MSC.Dytran only has linear, solid, shell, and membrane elements, so do not create any elements with midside nodes. The elements must be given the correct property and material numbers. Choose the element numbers to help you decide where an element is. Gaps in the numbering system do not cause a problem. Since MSC.Dytran has no wavefront requirement, wavefront optimization is not necessary. The order in which you define the grid points on shell elements is important since it determines the top and bottom surfaces. All shell elements in a region should have the same top and bottom surfaces, so that when you plot the results you are looking at the same side of the real structure. Some modeling programs can plot the element coordinate system. Use this to check that the element z-axes are all pointing in the same direction.

The Lagrangian elements in MSC.Dytran can undergo large deformations during the analysis. Therefore, the analysis should be started with as little distortion as possible. Many of the highly automated mesh generations produce element shapes that can be viewed as distorted elements when the elements are used in highly nonlinear analyses. You should use the automated mesh generators with care, and be prepared to modify the shapes of some of the elements that are produced.

Some automated mesh generators only produce triangles and tetrahedra. You should avoid using such generators, since the tetrahedra have very poor performance. The triangular shells give correct answers in bending, but are stiffer than quadrilateral elements. Triangular elements tend to make the analysis more expensive since two triangles are necessary to model one quadrilateral.

Use the geometry checking option of the modeler to ensure that the elements have reasonable shapes, since MSC.Dytran analyzes elements of almost any shape. Use any other checking options available in the modeler because errors in the model can prove to be expensive.

Eulerian elements can be modeled in the same way as Lagrangian elements, since they can have general connectivity and arbitrary shape. Since material flows through an Eulerian mesh, it is important that the mesh extends far enough to accommodate the motion of the material.

Some actions in the modeling package can produce elements that have (mathematically) negative volumes. The parameter GEOCHECK performs a check on all three dimensional elements and corrects them if necessary.

3.2.3 Properties and Materials

Select an appropriate property type within the modeler. Depending on the modeler it may be necessary to translate the available properties (normally PSHELL/PSOLID) into appropriate MSC.Dytran property types (e.g., PEULER, PEULER1). The MSC.Patran modeling package does enable the user to define interactively any of the MSC.Dytran property types and should be used if possible at this stage of the modeling process.

3.2.4 Constraints

Single-point constraints can be applied to any Lagrangian grid point in the model, but the enforced displacement must be zero. Nonzero displacements are not valid in MSC.Dytran. Eulerian grid points should not be constrained unless this is desired in an ALE calculation. A variety of different types of constraints (SPC1, SPC2, SPC3) are available, enabling you to define constraints with respect to both stationary and moving coordinate systems.

3.2.5 Loading

You can apply Lagrangian concentrated loads and pressures in the modeler, and they are translated to FORCE, MOMENT, and PLOAD4 entries. Most modeling packages only apply static loads, not dynamic ones. Therefore, you must add a TLOADn entry giving the variation of the load with time later. When you apply loads in the modeling package, the magnitude of the load or pressure that you specify is actually the scale factor for the load-time curve.

The faces of Eulerian elements with flow boundaries can be specified in the modeler (see Section 3.2.6 on page 3-4). However, the TLOAD1 and FLOW entries must be added later when using modeling packages other than MSC/XL.

3.2.6 Modeling of Surfaces and Faces

The ability to model surfaces is an important part of preparing MSC.Dytran input data since surfaces (see SURFACE, SUBSURF) are used to model contact surfaces, coupling surfaces, rigid bodies of arbitrary shape, rigid connections between parts of the mesh, arbitrary Lagrange-Euler interfaces (known as ALE surfaces), gas bags (often used in air bag calculations).

These surfaces are built up from the faces of elements using one of three “element face” types: CSEG, CFACE, or CFACE1 entries. The only difference between these three types is the way in which the element face is defined. Of these three face types, the CFACE entry is the optimal entry resulting in the quickest data translation and processing.

Defining these faces is probably the greatest area of incompatibility between most modeling packages and MSC.Dytran. There are often a large number of CSEG, CFACE, or CFACE1 entries in a typical input file. It is important to be able to check graphically that the segments are all present and correct. Few modeling packages can write CSEG, CFACE, or CFACE1 entries.

The exception is MSC/XL, which can both generate and display CFACEs, enabling the user to visualize the surfaces on the screen. The MSC/XL user can, therefore, interactively define CFACEs and visualize the surfaces/boundary conditions (e.g., FLOW boundary conditions) that refer to these CFACEs together with the entries, which in turn refer to the surfaces (e.g., RIGID, CONTACT, and COUPLE entries).

If the user wishes to use a modeling package other than MSC/XL, MSC.Dytran contains several “tricks” enabling the user to define CSEGs and CFACE1s in a relatively straightforward manner:

CSEGs – Trick 1

MSC.Dytran converts CQUAD4 and CTRIA3 entries to CSEG entries if the following rules are followed:

1. The PID of the PSHELLn entry that is referenced by the CQUAD4 or CTRIA3 element is the set number (SID) of the CSEG entry produced.
2. The thickness (T) on the PSHELLn entry is set to 9999.

Note that a CQUAD4 or CTRIA3 with a thickness of 9999 only gives you a CSEG entry. It does not give a shell element as well. The following two forms of input, therefore, give identical results:

```
CSEG, 100, 10, 1, 2, 22, 21
CSEG, 101, 10, 2, 3, 23, 22
```

and

```
CQUAD4, 100, 10, 1, 2, 22, 21
CQUAD4, 101, 10, 2, 3, 23, 22
PSHELL, 10,, 9999
```

In the second case, you do not obtain any shell elements since the CQUAD4 entries are converted to CSEG entries as they are read in. This is because their thickness is 9999. Note that the material ID number of the PSHELL entry (field 3) is left blank and is no longer required data.

For example, to create the contact surface below, create CQUAD4 elements on the faces of all elements on the bottom surface of the upper block. Then give them a thickness of 9999.0. Next, create CQUAD4 elements on the faces of the elements on the top surface of the lower block and give them a thickness of 9999.0. Finally, add the following CONTACT and SURFACE entries to the translated input file. Your input file will contain:

```
$ Bottom surface of upper block
CQUAD4, 100, 10, 1, 2, 3, 4
CQUAD4, 101, 10, 5, 6, 7, 8
PSHELL, 10,, 9999.
$ Top surface of lower block
CQUAD4, 200, 20, 101, 102, 103, 104
CQUAD4, 201, 20, 105, 106, 107, 108
PSHELL, 20,, 9999.
CONTACT, 1, SURF, SURF, 1, 2
SURFACE, 1,, SEG, 10
SURFACE, 2,, SEG, 20
```

Although this may seem a little confusing at first, you will quickly become accustomed to it. All you need to remember is that a CQUAD4 or CTRIA3 element with a thickness of 9999 is not a shell element. Naturally, the other property data for such entries is irrelevant and is ignored.

CSEGs – Trick 2

Trick 1 enables the user to specify CSEGs for any kind of element (2-D or 3-D) via a PSHELL/PSHELL1 entry and corresponding CQUAD4/CTRIA3 entries using a thickness of 9999. In many applications, you only need to generate surfaces constructed from the faces of shell or membrane elements (an example is an air-bag analysis). In these cases, you can directly generate the surface by selecting the elements by property, material, or element number (see the SURFACE entry; TYPE=PROP, MAT or ELEM). MSC.Dytran then internally generates a CSEG for each of the specified shell/membrane elements, avoiding an explicit definition of each CSEG entry. One drawback of this “trick” is the inability to visualize the faces and surface. Tricks 1 and 3 are the preferred methods since you can at least visualize the CSEGs and CFACE1s entries.

CFACE1 – Trick 3

When defining the boundary conditions of Eulerian meshes, it is more efficient to use CFACE1 than CSEG. MSC.Dytran converts PLOAD4 pressure entries to CFACE1 entries if the pressure is 9999. The following two forms of input are, therefore, identical:

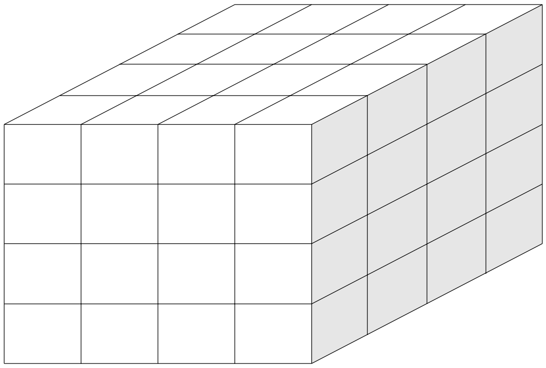
```
CFACE1, 1, 100, 1279, 7377,  7477
CFACE1, 2, 100, 3042, 6672, 5943
```

and

```
PLOAD4, 100, 1279, 9999., , , , 7377,  7477
PLOAD4, 100, 3042, 9999., , , , 6672,  5943
```

No pressure loads are applied since the pressure is 9999. This means that they are converted to CFACE1 entries as they are read in. Remember that PLOAD4 entries with a thickness of 9999. are not pressure loads but are CFACE1 entries.

In the following example, a nonreflecting boundary is applied to the shaded faces of a Eulerian mesh.



```
TLOAD1, 100, 110,  , 4
FLOW, 110,  200, MATERIAL, 10
PLOAD4, 200, 4, 9999., , , , 5, 115
PLOAD4, 200, 14, 9999., , , , 15, 125. . .
```

Since the PLOAD4 entries have a pressure of 9999., they are not pressure loads but are CFACE1 entries.

3.3 Translating the Data

After you have created your model using a modeling package, the data must be converted to a form that MSC.Dytran can understand. Use the translator in the modeling program to write an MSC.Nastran input file (virtually all modeling packages can write this type of file).

The exact form and completeness of the resulting input file varies depending on the modeler used. If your modeler adds MSC.Nastran Executive and Case Control to the top of the input file, delete them since the commands are different for MSC.Dytran. If you do not delete them, MSC.Dytran ignores them since they are valid only for MSC.Nastran. Similarly, any PARAM entries written in the Bulk Data Section for use with MSC.Nastran DMAP are ignored by MSC.Dytran.

If you are not using one of MSC's modeling packages, you must add some additional information to the file. First, if you are using any of the following, add the File Management Section:

- Output file types.
- User subroutines (if required).
- Restart control (if required).
- Rezone control (if required).

You can also add an Executive Section containing the following information:

- CPU time limit.

Next, add a Case Control Section giving the following information:

- Termination time/time step for the analysis.
- Sets of constraints and loading to be used.
- Type and frequency of the results to be output.
- Other Case Control options.

Finally, add any additional features that could not be created during the modeling stage. These are problem dependent, but will usually include some of the following:

- Material properties (DMATxx).
- Eulerian properties (PEULER, PEULER1).
- Contact surfaces (CONTACT).
- Coupling surfaces (COUPLE).
- ALE surfaces (ALE).

- Rigid bodies (RIGID, RELLIPS, MATRIG, RBE2-FULLRIG).
- Rigid walls (WALL).
- Dynamic loading (TLOAD1, TLOAD2).
- Eulerian boundary conditions (FLOW, WALLET).
- Initial conditions (TIC, TICGP, TICEL).
- Cross sections (SECTION).

After these modifications are completed, you have a ready-to-run MSC.Dytran input file.

3.4 Checking the Data

3.4.1 Data Check

Once you have an MSC.Dytran input file, always run a data check before running the main analysis. A data check performs the following tasks:

- Reads and checks the input file.
- Sorts out the data and prints any error or warning messages.
- Performs two time steps.

No action is required to carry out a data check, since it is the default option for all new analyses. You should carefully check the output to ensure that the model is correct.

The importance of checking your input data cannot be overemphasized. The nonlinear, dynamic nature of MSC.Dytran analyses requires large amounts of computer resources. MSC.Dytran can also analyze models that might be rejected by other codes. Input errors can be both expensive and time consuming. A little extra time spent thoroughly checking the data will be well worth it in the long term.

3.5 Executing MSC.Dytran

The input for MSC.Dytran is the same regardless of the computer being used. However, the way you actually run the program is machine dependent. The *MSC.Dytran Installation and Operations Guide*, Version 4.7 describes which operating systems are currently supported along with other computer specific information.

In some instances, the execution of MSC.Dytran may be customized to a particular machine configuration. If you have not used MSC.Dytran on a particular machine before, contact your system administrator for details on how to run it.

3.5.1 Running MSC.Dytran Using XDYTRAN

The easiest way to start an MSC.Dytran analysis job interactively is to use the graphical user interface program, XDYTRAN. For more information on how to use XDYTRAN, see Appendix A, "Using XDYTRAN", on page A-1. To start a batch job, use the run script documented in Section 3.5.2.

3.5.2 Running MSC.Dytran Using the dytran Command

MSC.Dytran can also be run using the command procedure `dytran`. The dytran procedure processes its command line arguments and creates a second command procedure, which is submitted to a batch queue, or is executed interactively. Here is how to use the dytran procedure:

```
dytran      jid=JobId  size=small|medium|large  rid=RestartId  atb=AtbId
            bat=yes|no  notify=yes|no  exe=executable  imm=IMMfile
            user_rout=user_routines_file  ask_rem=yes|no
            print=print_prefix  output=out_prefix
```

Examples

```
% dytran  jid=mydata  exe=my_exe.exe
% dytran  jid=mydata  bat=no ( interactive run )
% dytran  jid=mydata  bat=no print=printout
% dytran  jid=mydata
```

Command Line Arguments	Purpose
jid=JobId	Name of the input file (file type is detected). Script will look for the file JobId.dat. No default is applicable here.
rid=RestartId	Restart file of a previous analysis from which to restart. Script will look for the file RestartId.RST. No default is applicable here.
size=small/medium/large	Defines the size of allocated memory by the executable. You can choose small, medium or large.
atb=AtbId	Script will look for the ATB input file with name AtbId.ain. No default is applicable here.
imm=IMMfile	Name of an Initial Metric Method file name (if any). Script will look for IMMfile.dat No default applicable here.
bat=yes/no	Indicates the way in which the job will be run. bat=yes means run job in background bat=no means run job interactively Default is yes.
notify=yes/no	If notify=yes, a message will be sent to the terminal when the job finishes. If notify=no, no message is sent. Default is yes.
exe=my_exe.exe	This command line argument enables you to use your own modified MSC.Dytran executable with name my_exe.exe. Default executable is <installation_dir>/dytran47/dytranexe/dytran.exe.
user_rout=my_file.f	The Fortran file with name my_file.f in which user routines are stored, will be compiled and linked with the dytran library. The executable produced will be used to perform the calculation. When the input file is a JobId.dat file, the USERCODE statement is looked for in this file and user_rout is set accordingly. Specifying user_rout on the command line will overrule the USERCODE statement.
ask_rem=yes/no	Parameter with which you can specify that you want be asked to remove all files beginning with the generic name derived from the JobId. Ask_rem=yes means that you will be asked to remove these files or restart with a different JobId. Default is yes.

Command Line Arguments	Purpose
print=my_file	Command line argument that can be used in case of interactive runs, that forces the output to be written to a file instead of the terminal screen. Example: with print=my_file the output will be written to a file named JID_MY_FILE.OUT.
output=prefix	Specifies the prefix to be used for all output files. Default is the JobId.

Notes:
1. Jid must be specified
2. Keywords are separated by a blank.
3. Keywords may appear in any order.
4. Dytran output file will be named JobId.OUT.
5. Archive files will have the extension .ARC.
6. Time history files will have the extension .THS.
7. Restartoutput files will have the extension .RST.
8. All output files will be prefixed with the JobId.
9. All output filenames will be in uppercase letters.
10. ATB output files will have the extension .AOU.
11. Only CTRIA3 and GRID are valid cards in an IMM file.

3.5.3 Stopping MSC.Dytran

There are several methods of stopping an MSC.Dytran job after it is running. The method used depends on how the job is running (batch or interactive) and whether the contents of the output files is important.

When Output Files are Unimportant

If the program is running interactively, a <control-c> command may be typed at the terminal. If the program is running in batch, then the following command may be issued:

```
%kill -9 pid
```

where <pid> is the process ID of the MSC.Dytran job. The process ID can be obtained using the UNIX command “ps”.

When Output Files are Important

To provide a systematic, forced shutdown, MSC.Dytran will trap the UNIX signal SIGUSR1. When this signal is caught, the program writes its restart files and terminates when the current time step is complete. To terminate the program, enter the following command:

```
%kill -USR1 <pid>
```

where <pid> is the process ID of the MSC.Dytran job. If the job is running interactively, you will have to do this from another terminal.

3.6 Memory Requirements

3.6.1 MSC.Dytran Memory Requirements

MSC.Dytran executes almost entirely in core. During the input phase, the input file is read and the data is stored in arrays within the program.

The default size of MSC.Dytran is preset to 40 megabytes of memory. This size is appropriately set for smaller types of analysis problems. However, the default MSC.DYTRAN executable may not be large enough to solve medium or large size problems. Beside the default executable size, MSC.Dytran is delivered with two larger size executable, that can be used to run medium and large size problems.

If you can not run your problem using the default executable size, you can increase the size of the memory to be used by choosing one of the other modules, see also the “size” argument in Section 3.5.2, "Running MSC.Dytran Using the dytran Command", on page 3-11. To use an executable size which will run medium or large size problems, use one of the following commands:

```
%dytran jid=mydata size=medium
dytran jid=mydata size=large
```

The executable for medium size problems uses 90 MB of memory, and large size problems use 160 MB.

If you can not run your problem using the maximum supplied executable size, you can increase the size of the memory of the executable by using a user-defined subroutine, see Section 3.6.2, "Memory Management Using a User-Defined Subroutine", on page 3-16.

Warning: If you increase the amount of memory too much, and your system does not have enough swap space, you will not be able to run MSC.Dytran and will receive one of the following messages, depending on your system:

```
Not enough swap space.
No memory.
Killed.
```

You must either decrease the size of the parameters in the user-defined subroutine, or increase the physical memory or swap space on your system.

3.6.2 Memory Management Using a User-Defined Subroutine

The memory definition using the “size” option, will satisfy the needs of most users. If you need to increase the amount of memory beyond the large size, you can customize MSC.Dytran with help of a subroutine called `memory_large.f`. The memory allocation routine `memory_large.f` resides on the `<installation_directory>/dytran47/dytranexe` directory and must be copied to the current working directory.

Inside the FORTRAN subroutine `memory_large.f` are four parameter statements defining the size of the arrays used in the program. The current definitions are:

LIMI	=	12 000 000	Storage space for integers.
LIMX	=	11 000 000	Storage space for real variables.
LIMC	=	1 500 000	Storage space for character variables.
LIML	=	750 000	Storage space for logical variables.

To create an executable with a larger memory definition, you must increase the values of the four parameters in the `memory_large.f` subroutine. To compile `memory_large.f` and link the new MSC.Dytran executable, you must include a USERCODE statement, which references the file containing the FORTRAN coding of `memory_large.f`, in the FMS section of the input data file.

```
USERCODE=memory_large.f
```

The dytran command procedure then automatically performs the following steps as part of the batch or interactive run:

1. Compile `memory_large.f`.
2. Link the `memory_large.o` routine with the MSC.Dytran library to create a modified executable.
3. Run the analysis.

Note that the modified version of MSC.Dytran is compiled and linked again each time you run an analysis with user code. If you want to use the same modified executable many times without compiling and linking each time, it is more efficient to use the exe option to run with the modified executable:

```
%dytran jid=run12 exe=my_executable.exe
```

3.6.3 Memory Management Using XDYTRAN

The easiest way to modify the memory usage of MSC.Dytran and to create a new executable of MSC.Dytran, is to use the graphical user interface program XDYTRAN. Using the “Customizing Memory” option, the user can allocate new sizes for the memory usage by MSC.Dytran. When one or more of the sizes have been changed, XDYTRAN will write out a new version of memory.f in the same directory where the customized executable will be generated. It will automatically be linked in by XDYTRAN.

Note: You can use the modified executable many times by choosing the newly created customized executable in the “Processes” subwindow in XDYTRAN, when you create a process to run the job. For more information on how to use XDYTRAN, see Appendix A, "Using XDYTRAN", on page A-1.

3.7 User-Modifiable MSC.Dytran

MSC.Dytran allows you to add your own subroutines to the program. The FORTRAN coding for the subroutines should be stored in one file that is in the same directory as the MSC.Dytran data file.

3.7.1 User-Modifiable MSC.Dytran Using XDYTRAN

To create a new executable of MSC.Dytran that includes user subroutines, the “Select File” option must be used. When the file with user subroutines is selected, XDYTRAN automatically links in the user subroutines to create a new version of MSC.Dytran.

Note: You can use the modified executable many times by choosing the newly created customized executable in the “Processes” subwindow in XDYTRAN, when you created a new process to run the job. For more information on how to use XDYTRAN, see Appendix A, "Using XDYTRAN", on page A-1.

3.7.2 User-Modifiable MSC.Dytran Using the dytran Command Procedure

To create a new executable of MSC.Dytran that includes user subroutines, a USERCODE statement that references the file containing the FORTRAN coding must be included in the FMS section of the input data file.

```
USERCODE=user.f
```

If at the same time, you want to change the memory usage by MSC.Dytran, you must place a new version of memory.f on the working directory. The memory.f subroutine should not be included in the file that contains the user subroutines. Section 3.6.1 explains where the default memory.f routine resides on your system, and how it should be changed.

The MSC.Dytran command procedure then automatically performs the following steps as part of the batch or interactive run:

1. Compile user.f (and memory.f when present).
2. Link the routines with MSC.Dytran to create a modified executable.
3. Run the analysis.

Note that the modified version of MSC.Dytran is compiled and linked again each time you run an analysis with user code. If you want to use the same modified executable many times without compiling and linking each time, it is more efficient to use the “exe” option of the dytran command procedure to run with the modified executable:

```
% dytran jid=run12 exe=dytran.exe
```

3.8 Files Created by MSC.Dytran

MSC.Dytran creates a number of files during the analysis. The names of these files are generally identical for the UNIX System. Some differences may exist for other systems and are given in the *MSC.Dytran Installation and Operations Guide*.

In the main sections of this manual, generic names are used when referring to a particular MSC.Dytran file. These generic reference names and the actual generic file names are given below:

File	Generic Reference Name	Generic Name
Input	dat	file_name.dat
Messages	MSG	FILE_SUMMARY.MSG
Output	PRT	file_name.OUT
Archive	ARC	output_file_#.ARC
Time History	THS	output_file_#.THS
Restart	RST	output_file_#.RST
Warning and Errors	ERR	ERROR_SUMMARY.MSG
Data Ignored	IGN	NASTRAN_IGNORE.MSG
Neutral Input	NIF	file_name.NIF

Input File

The input file contains all the input data and must be present in order to run MSC.Dytran. It is a text file with up to 80 characters on each line.

Message File

The message file contains information every time data is written to an archive or restart file.

Output File

The output file is a text file suitable for printing on a line or laser printer, or viewing with a standard editor. It contains messages produced by MSC.Dytran as well as a summary of the calculation at every time step.

Archive Files

MSC.Dytran can create any number of archive files containing results at times during the analysis. Archive files are binary and can only be read by the translator XDEXTR. They contain a complete description of the geometry and connectivity of the analysis model as well as the requested results.

Time History Files

MSC.Dytran can also create any number of time history files containing results for particular grid points and elements during the calculation. They are also binary and can be read only by the supplied translator. Time history files only contain results.

Restart Files

Restart files are binary files that contain the information necessary to restart the analysis. Any number can be created; however, since a full representation of the analysis is stored every time restart data is written, restart files can become very large, and you may run out of disk space if you write restart data too often.

Neutral Input File

This is an internal file used to store the input data after initial sorting. Usually, it can be deleted.

Error File

MSC.Dytran produces a single error file containing a summary of all warnings and errors issued during reading and subsequent data processing.

Data Ignored File

In many cases, MSC.Dytran issues messages indicating differences between MSC.Nastran entries and the corresponding MSC.Dytran interpretation for example. If these are not fatal, they are written to this file.

3.9 Outputting Results

3.9.1 Input Commands

MSC.Dytran is very flexible in the way results are output for postprocessing. Any number of output files can be created, and you can choose exactly what entities, such as grid points and elements, and which results are stored in each file. Data can be written at any time or time step during the analysis.

You need to specify the following for a complete output specification:

1. Type of the file.
2. How often it is saved.
3. How often data is written.
4. What entities (e.g., grid points, elements) are stored.
5. What results are output.

Type of the File - TYPE Entry

There are six types of files selected using the TYPE FMS statement:

ARCHIVE

This is usually used for storing connectivity and results for all or part of the model at particular times during the analysis. It can then be used to plot deformed shapes, contour plots, or arrow plots.

```
TYPE (logical_file) = ARCHIVE
```

TIMEHIS

This is usually used for storing only results from a few key entities during the analysis. It does not contain the connectivity capability, and so it can be used only to create time history plots of the results.

```
TYPE (logical_file) = TIMEHIS
```

RESTART

This is not a results file but is used for restarting MSC.Dytran and in fact is similar to an ARCHIVE file.

```
TYPE (logical_file) = RESTART
```

STEPSUM

A one-line time step summary is written to the output(file). It is useful for checking the characteristics of the analysis.

```
TYPE (logical_file) = STEPSUM
```

MATSUM	Output of the state of a material at selected time steps. TYPE (logical_file) = MATSUM
MRSUM	Information about MATRIG and RBE2-FULLRIG assemblies at selected time steps. TYPE (logical_file) = MRSUM
EBDSUM	Information about the Eulerian boundaries at selected time steps. TYPE (logical_file) = EBDSUM

The file that a particular command refers to is identified by a logical filename. This logical filename forms part of the actual filename on most computers (see the *MSC.Dytran Installation and Operations Guide*).

How Often It Is Saved - SAVE Entry

Every time data is stored, it is usually stored in the same file. However, after data is stored, a set number of times specified by the SAVE FMS statement, a new file is opened and the old file is closed and saved; e.g., SAVE (logical_file) = 20 means that twenty sets of results are stored in the file before it is closed and a new file is opened.

On most types of computers the files are identified by the time-step number when data was first written to the file. See the *MSC.Dytran Installation and Operations Guide* for details of the filenames on your computer.

Once a file has been saved, the results stored in it are available for postprocessing even when the analysis is still running.

How Often Data Is Written – STEPS/TIMES Entry

Results can be output at intervals during the analysis specified either in terms of the time (TIMES entry), or the time-step numbers (STEPS entry). You specify a list of the times or time steps at which output is required. For example:

```
STEPS (logical_file) = 100, 200, 300
```

A range can be specified as <start>, THRU, <end>, BY, <increment>. The end of the calculation can be identified by the word END. For example,

```
TIMES (logical_file) = 0, THRU, END, BY, 1.0E-3
```

Grid Points, Elements Stored/Results Output

The items for which variables are to be stored are specified through a SET entry and one of the following:

GRIDS	Grid points to be stored.
ELEMENTS	Elements to be stored.
RIGIDS	Rigid bodies to be stored.
RELS	Rigid ellipsoids to be stored.
MATS	Materials to be stored.
GBAGS	Gas bags to be stored.
CONTS	Contact surfaces to be stored.
CSECS	Cross sections to be stored.
CPLSURFS	Coupling surfaces to be stored.
SURFACES	Surfaces to be stored.
SUBSURFS	Subsurfaces to be stored.
USASURFS	USA surfaces to be stored.
SGAUGES	Surface gauges to be stored.

The results required are then specified using the following commands:

GPOUT	Grid point results.
ELOUT	Element results.
RBOUT	Rigid body results.
RELOUT	Rigid ellipsoid results.
MATOUT	Material results.
GBAGOUT	Gas bag results.
CONTOUT	Contact surface results.
CSOUT	Cross-section results.
CPLSOUT	Coupling-surface results.
SURFOUT	Surface results.
SUBSOUT	Subsurface results.
USASOUT	USA-surface results.
SGOUT	Surface gauge results.

For example, the following commands store the xx-stress for elements 200 through 210:

```
ELEMENTS (FILE7) = 20
SET 20 = 200, THRU, 210
ELOUT (FILE7) = TXX
```

The following commands store the x-velocity and y-force components for grid points 101, 209, and 1005.

```
GRIDS(ARC1) = 40
SET 40 = 101, 209, 1005
GPOUT(ARC1) = XVEL, YFORCE
```

The results available for output are listed in Section 3.9.2 on page 3-24. Results for rigid bodies, rigid ellipsoids, gas bags, materials, contact surfaces, cross sections and sgauges can only be stored in time-history files. Results for coupling surfaces and USA surfaces can only be stored in archive files.

3.9.2 Result Types

The data stored by MSC.Dytran and available for output varies with the element type or, in the case of grid points, the type of element to which the grid points are attached. The following fundamental element types are available in MSC.Dytran:

- One-dimensional elements.
- Lagrangian solid elements.
- Quadrilateral shell elements.
- Triangular shell elements.
- Triangular membrane elements.
- Dummy quadrilateral and triangular elements.
- Eulerian solid elements (hydrodynamic).
- Eulerian solid elements (with shear strength).
- Eulerian solid elements (multimaterial hydrodynamic).
- Eulerian solid elements (multimaterial with shear strength).
- Rigid bodies.

A particular archive or time-history file can contain information for the grid points or elements of one or more of these element types.

Sections 3.9.2.1 on page 3-25 and 3.9.2.2 on page 3-33 list all the results types available. The keyword is the description put on the Case Control commands (GPOUT, ELOUT, etc.), to write the data to archive or time-history files.

3.9.2.1 Grid-Point Results (GPOUT)

One-Dimensional Elements

Keyword	Description
XPOS	x-coordinate.
YPOS	y-coordinate.
ZPOS	z-coordinate.
XVEL	z-translational velocity.
YVEL	y-translational velocity.
ZVEL	z-translational velocity.
XFORCE	x-force = external + internal.
YFORCE	y-force = external + internal.
ZFORCE	z-force = external + internal.
XDIS	x-translational displacement.
YDIS	y-translational displacement.
ZDIS	z-translational displacement.
PMASS	Grid-point mass.
PMOMI	Grid-point inertia.
XAVEL	x-angular velocity.
YAVEL	y-angular velocity.
ZAVEL	z-angular velocity.
XMOMENT	x-moment = external + internal.
YMOMENT	y-moment = external + internal.
ZMOMENT	z-moment = external + internal.
EXRVEL	Initial enforced x-angular velocity (Nastran Initialization).
EYRVEL	Initial enforced y-angular velocity (Nastran Initialization).
EZRVEL	Initial enforced z-angular velocity (Nastran Initialization).
EXVEL	Initial enforced x-velocity (Nastran Initialization).
EYVEL	Initial enforced y-velocity (Nastran Initialization).
EZVEL	Initial enforced z-velocity (Nastran Initialization).
XFCON	Constraint x-force (SPC3 and FORCE3 only).

Keyword	Description
YFCON	Constraint y-force (SPC3 and FORCE3 only).
ZFCON	Constraint z-force (SPC3 and FORCE3 only).
XMCON	Constraint x-moment (SPC3 and FORCE3 only).
YMCON	Constraint y-moment (SPC3 and FORCE3 only).
ZMCON	Constraint z-moment (SPC3 and FORCE3 only).
DLTPNT	Nodal time step.
XACC	x-translational acceleration.
YACC	y-translational acceleration.
ZACC	z-translational acceleration.
RPOS	Resultance coordinate.
RVEL	Resultance velocity.
RACC	Resultance acceleration (see Remark 3).
RFORCE	Resultance force = external + internal.
RDIS	Resultance translational displacement.
RAVEL	Resultance angular velocity.
RMOMENT	Resultance moment = external + internal.
RFCON	Resultance constraint force.
RMCON	Resultance constraint moment.

Remarks:

1. The constraint forces and moments (XFCON, YFCON, ZFCON, XMCON, YMCON, ZMCON) are only output when the constraint is an SPC3 or a FORCE3. In all other cases the result will be zero.

2. All variables starting with R (Resultance) are calculated as follows:

$$Rxx = \sqrt{Xxx^2 + Yxx^2 + Zxx^2}$$

3. If filtering of the acceleration is required, it should be done on the individual components (XACC, YACC, ZACC) and not on the resultant (RACC). This is because the resultant value is always positive, while the components can be either positive or negative.

Quadrilateral and Triangular Shells

Keyword	Description
XPOS	x-coordinate.
YPOS	y-coordinate.
ZPOS	z-coordinate.
XDIS	x-translational displacement.
YDIS	y-translational displacement.
ZDIS	z-translational displacement.
XVEL	x-translational velocity.
YVEL	y-translational velocity.
ZVEL	z-translational velocity.
XAVEL	x-angular velocity.
YAVEL	y-angular velocity.
ZAVEL	z-angular velocity.
XFORCE	x-force = external + internal.
YFORCE	y-force = external + internal.
ZFORCE	z-force = external + internal.
XMOMENT	x-moment = external + internal.
YMOMENT	y-moment = external + internal.
ZMOMENT	z-moment = external + internal.
EXVEL	Initial enforced x-velocity (Nastran Initialization).
EXRVEL	Initial enforced x-angular velocity (Nastran Initialization).
EYRVEL	Initial enforced y-angular velocity (Nastran Initialization).
EZRVEL	Initial enforced z-angular velocity (Nastran Initialization).
EYVEL	Initial enforced y-velocity (Nastran Initialization).
EZVEL	Initial enforced z-velocity (Nastran Initialization).
XFCON	Constraint x-force (SPC3 and FORCE3 only).
YFCON	Constraint y-force (SPC3 and FORCE3 only).
ZFCON	Constraint z-force (SPC3 and FORCE3 only).
XMCON	Constraint x-moment (SPC3 and FORCE3 only).
YMCON	Constraint y-moment (SPC3 and FORCE3 only).

Keyword	Description
ZMCON	Constraint z-moment (SPC3 and FORCE3 only).
DLTPNT	Nodal time step.
PMASS	Grid-point mass.
PMOMI	Grid-point inertia.
XACC	x-translational acceleration.
YACC	y-translational acceleration.
ZACC	z-translational acceleration.
RPOS	Resultance coordinate.
RVEL	Resultance velocity.
RACC	Resultance acceleration (see Remark 3).
RFORCE	Resultance force = external + internal.
RDIS	Resultance translational displacement.
RAVEL	Resultance angular velocity.
RMOMENT	Resultance moment = external + internal.
RFCON	Resultance constraint force.
RMCON	Resultance constraint moment.

Remarks:

1. The constraint forces and moments (XFCON, YFCON, ZFCON, XMCON, YMCON, ZMCON) are only output when the constraint is an SPC3 or a FORCE3. In all other cases the result will be zero.

2. All variables starting with R (Resultance) are calculated as follows:

   $$Rxx = \sqrt{Xxx^2 + Yxx^2 + Zxx^2}$$

3. If filtering of the acceleration is required, it should be done on the individual components (XACC, YACC, ZACC) and not on the resultant (RACC). This is because the resultant value is always positive, while the components can be either positive or negative.

Triangular Membranes

Keyword	Description
XPOS	x-coordinate.
YPOS	y-coordinate.
ZPOS	z-coordinate.
XDIS	x-displacement.
YDIS	y-displacement.
ZDIS	z-displacement.
XVEL	x-velocity.
YVEL	y-velocity.
ZVEL	z-velocity.
XFORCE	x-force = external + internal.
YFORCE	y-force = external + internal.
ZFORCE	z-force = external + internal.
PMASS	Grid-point mass.
XFCON	Constraint x-force (SPC3 and FORCE3 only).
YFCON	Constraint y-force (SPC3 and FORCE3 only).
ZFCON	Constraint z-force (SPC3 and FORCE3 only).
DLTPNT	Nodal time step.
XACC	x-translational acceleration.
YACC	y-translational acceleration.
ZACC	z-translational acceleration.
RPOS	Resultance coordinate.
RVEL	Resultance velocity.
RACC	Resultance acceleration (see Remark 3).
RFORCE	Resultance force = external + internal.
RDIS	Resultance translational displacement.
RFCON	Resultance constraint force.

Remarks:

1. The constraint forces (XFCON,YFCON,ZFCON) are only output when the constraint is an SPC3 or a FORCE3. In all other cases the result will be zero.

2. All variables starting with R (Resultance) are calculated as follows:

$$Rxx = \sqrt{Xxx^2 + Yxx^2 + Zxx^2}$$

3. If filtering of the acceleration is required, it should be done on the individual components (XACC, YACC, ZACC) and not on the resultant (RACC). This is because the resultant value is always positive, while the components can be either positive or negative.

Dummy QUAD4s and TRIA3s – Rigid Bodies

Keyword	Description
XPOS	x-coordinate.
YPOS	y-coordinate.
ZPOS	z-coordinate.
XVEL	x-velocity.
YVEL	y-velocity.
ZVEL	z-velocity.
XFORCE	x-force = external + internal.
YFORCE	y-force = external + internal.
ZFORCE	z-force = external + internal.
XLOCAL	x-coordinate in the rigid body coordinate system.
YLOCAL	y-coordinate in the rigid body coordinate system.
ZLOCAL	z-coordinate in the rigid body coordinate system.
PMASS	Grid-point mass.
DLTPNT	Nodal time step.
XACC	x-translational acceleration.
YACC	y-translational acceleration.
ZACC	z-translational acceleration.
RPOS	Resultance coordinate.
RVEL	Resultance velocity.
RACC	Resultance acceleration (see Remark 2).
RFORCE	Resultance force = external + internal.

Remarks:

1. All variables starting with R (Resultance) are calculated as follows:

   $$Rxx = \sqrt{Xxx^2 + Yxx^2 + Zxx^2}$$

2. If filtering of the acceleration is required, it should be done on the individual components (XACC, YACC, ZACC) and not on the resultant (RACC). This is because the resultant value is always positive, while the components can be either positive or negative.

Lagrangian Solids

Keyword	Description
XPOS	x-coordinate.
YPOS	y-coordinate.
ZPOS	z-coordinate.
XVEL	x-velocity.
YVEL	y-velocity.
ZVEL	z-velocity.
XFORCE	x-force = external + internal.
YFORCE	y-force = external + internal.
ZFORCE	z-force = external + internal.
PMASS	Grid-point mass.
EXVEL	Initial enforced x-velocity (Nastran Initialization).
EYVEL	Initial enforced y-velocity (Nastran Initialization).
EZVEL	Initial enforced z-velocity (Nastran Initialization).
XFCON	Constraint x-force (SPC3 and FORCE3 only).
YFCON	Constraint y-force (SPC3 and FORCE3 only).
ZFCON	Constraint z-force (SPC3 and FORCE3 only).
DLTPNT	Nodal time step.
XACC	x-translational acceleration.
YACC	y-translational acceleration.
ZACC	z-translational acceleration.
RPOS	Resultance coordinate.
RVEL	Resultance velocity.
RACC	Resultance acceleration (see Remark 3).
RFORCE	Resultance force = external + internal.
RFCON	Resultance constraint force.

Note: For all Lagrangian grid points that belong to a rigid surface, use XFCON, YFCON, and ZFCON to store the local coordinates in the rigid body system.

Remarks:

1. The constraint forces (XFCON, YFCON, ZFCON) are only output when the constraint is an SPC3 or a FORCE3. In all other cases the result will be zero.
2. All variables starting with R (Resultance) are calculated as follows:

   $$Rxx = \sqrt{Xxx^2 + Yxx^2 + Zxx^2}$$

3. If filtering of the acceleration is required, it should be done on the individual components (XACC, YACC, ZACC) and not on the resultant (RACC). This is because the resultant value is always positive, while the components can be either positive or negative.

Eulerian Solids (Hydrodynamic, Multimaterial Hydrodynamic, and Multimaterial with Shear Strength)

Keyword	Description
XPOS	x-coordinate.
YPOS	y-coordinate.
ZPOS	z-coordinate.
RPOS	Resultance coordinate.
XVG	x-velocity.
YVG	y-velocity.
ZVG	z-velocity.
RVG	Resultance velocity.
APOS	Weight factor alpha.
BPOS	Weight factor beta.
CPOS	Weight factor gamma.
XAG	x-acceleration.
YAG	y-acceleration.
ZAG	z-acceleration.
RAG	Resultance acceleration.
RACC	Resultance acceleration (see Remark 2).

Remarks:

1. All variables starting with R (Resultance) are calculated as follows:

$$Rxx = \sqrt{Xxx^2 + Yxx^2 + Zxx^2}$$

2. If filtering of the acceleration is required, it should be done on the individual components (XACC, YACC, ZACC) and not on the resultant (RACC). This is because the resultant value is always positive, while the components can be either positive or negative.

Eulerian Solids (Shear Strength)

Keyword	Description
XPOS	x-coordinate.
YPOS	y-coordinate.
ZPOS	z-coordinate.
RPOS	Resultance coordinate.
RACC	Resultance acceleration (see Remark 2).

Remark:

1. All variables starting with R (Resultance) are calculated as follows:

   $$Rxx = \sqrt{Xxx^2 + Yxx^2 + Zxx^2}$$

2. If filtering of the acceleration is required, it should be done on the individual components (XACC, YACC, ZACC) and not on the resultant (RACC). This is because the resultant value is always positive, while the components can be either positive or negative.

3.9.2.2 Element Results (ELOUT)

One-Dimensional Elements

Keyword	Description
MASS	Element mass.
YHAT1X	x-component of the y-axis of the local element coordinate system at end 1.
YHAT1Y	y-component of the y-axis of the local element coordinate system at end 1.
YHAT1Z	z-component of the y-axis of the local element coordinate system at end 1.
ZHAT1X	x-component of the z-axis of the local element coordinate system at end 1.
ZHAT1Y	y-component of the z-axis of the local element coordinate system at end 1.
ZHAT1Z	z-component of the z-axis of the local element coordinate system at end 1.
YHAT2X	x-component of the y-axis of the local element coordinate system at end 2.
YHAT2Y	y-component of the y-axis of the local element coordinate system at end 2.
YHAT2Z	z-component of the y-axis of the local element coordinate system at end 2.
ZHAT2X	x-component of the z-axis of the local element coordinate system at end 2.
ZHAT2Y	y-component of the z-axis of the local element coordinate system at end 2.

Keyword	Description
ZHAT2Z	z-component of the z-axis of the local element coordinate system at end 2.
XFORCE	x-force resultant in local element coordinate system.
YFORCE	y-force resultant in local element coordinate system.
ZFORCE	z-force resultant in local element coordinate system.
XMOMENT	x-moment resultant in local element coordinate system.
YMOMENT	y-moment resultant in local element coordinate system.
ZMOMENT	z-moment resultant in local element coordinate system.
FIBL1	Area properties - variable 1.
FIBL2	Area properties - variable 2.
FIBL3	Area properties - variable 3.
FIBL4	Area properties - variable 4.
EDIS	Element distortional energy.
FAIL	Failure time.
ELDLTH	Stable time step of element (according to CFL-criteria).
ELTIME	Time at which element is updated (subcycling).
ALPSTBL	Stable alpha of element (subcycling).
ELGROUP	Subcycle group of element.
EXUSER1	User variable 1.
EXUSER2	User variable 2.
STRAIN	Element strain (belt elements only).
RFORCE	Resultance force.
RMOMENT	Resultance moment.

Note: The Hughes-Liu beam can have variables at each integration point (layer) requested for output. A list of valid variable names is listed on page 3-37.

Remarks:

1. For corotational CELASx/CDAMP1 the direction vector is stored in ZHAT2X, ZHAT2Y, ZHAT2Z.

2. All variables starting with R (Resultance) are calculated as follows:

$$Rxx = \sqrt{Xxx^2 + Yxx^2 + Zxx^2}$$

Quadrilateral Shell Elements

Keyword	Description
MASS	Element mass.
THICK	Thickness.
AREA	Area.
FAIL	Failure time.
EDIS	Element distortional energy.
QHOUR1	Hourglass force 1.
QHOUR2	Hourglass force 2.
QHOUR3	Hourglass force 3.
QHOUR4	Hourglass force 4.
QHOUR5	Hourglass force 5.
EHRG	Hourglass energy.
ELDLTH	Stable time step of element (according to CFL criteria).
ELTIME	Time at which element is updated (subcycling).
ALPSTBL	Stable alpha of element (subcycling).
ELGROUP	Subcycle group of element.
EXUSER1	User variable 1.
EXUSER2	User variable 2.
Q1	Cosine of angle between element and material coordinate systems.
Q2	Sine of angle between element and material coordinate systems.

Triangular Shell Elements

Keyword	Description
MASS	Element mass.
THICK	Thickness.
AREA	Area.
FAIL	Failure time.
EDIS	Element distortional energy.
ELDLTH	Stable time step of element (according to CFL criteria).
ELTIME	Time at which element is updated (subcycling).
ALPSTBL	Stable alpha of element (subcycling).
ELGROUP	Subcycle group of element.
EXUSER1	User variable 1.
EXUSER2	User variable 2.
Q1	Cosine of angle between element and material coordinate systems.
Q2	Sine of angle between element and material coordinate systems.

Note: Both the triangular and the quadrilateral shell elements use integration points through the element thickness in the solution sequence. The results at the integration points (sublayers) can be individually requested for output. A list of valid sublayer variable names is listed on page 3-37.

Sublayer Variables

The variable name consists of the normal name of the variable with the number of the integration layer added to the end.

TXX03 - TXX of third integration sublayer
TXZ32 - TXZ of thirty-second integration sublayer.

For convenience, the inner, middle, and outer sublayers can be referred to as -IN, -MID, and -OUT, rather than the specific integration sublayer number. Thus, TXX-IN refers to TXX at the inner sublayer (sublayer 1) and TXX-OUT refers to TXX at the outer sublayer (sublayer “max”). Note that the numbering of sublayers is in the positive direction of the element normal.

The program verifies whether the layer number at which output is requested (either by means of the -IN, -MID, or -OUT definition or by a sublayer number) is a valid one according to the property definition of the elements. If a sublayer number is not valid, the result will be set to zero.

DMATEP, YLDVM, and FAILMPS/FAILEX entries are used to specify the material data for the Hughes-Liu beam. The following specifications indicate which of the variables are relevant depending on the YLDVM entry. If the variable is not relevant but is requested for output, the value stored will be zero.

Hughes-Liu beam (form = HUGHES on PBEAM1 entry) and

- No YLDVM or FAILMPS/FAILEX entry
  TXX, TYY, TZZ, TXY, TYZ, and TZX
- YLDVM with only YIELD, EH fields (simple bilinear stress-strain curve)
  TXX, TYY, TZZ, TXY, TYZ, TZX, EFFPL, EFFST, and FAIL
- YLDVM with TABLE, TYPE fields (piecewise linear stress-strain curve) and/or TABY/ D, P fields (strain-rate-dependent yield stress)
  TXX, TYY, TZZ, TXY, TYZ, TZX, EFFPL, EFFST, FAIL, and EFFSR

For shell elements (CQUAD4, CTRIA3) with isotropic elastoplastic material (DMATEP definition) the default set of sublayer variables that are available for output are the stress and strain components, the effective stress, and the effective plastic strain. A restricted set can be obtained by setting the PARAM, STRNOUT to NO. In that case, the sublayer variables for output are the stress components and effective plastic strain.

Sublayer Variables Keywords and Descriptions

Keyword	Description
EFFPL	Effective plastic strain.
EFFST	Effective stress.
TXX	xx-stress.
TYY	yy-stress.
TZZ	zz-stress.
TXY	xy-stress.
TYZ	yz-stress.
TZX	zx-stress.
EPSXX	xx-strain.
EPSYY	yy-strain.
EPSZZ	zz-strain.
EPSXY	xy-strain.
EPSYZ	yz-strain.
EPSZX	zx-strain.

Keyword	Description
FAIL	Sublayer-failure flag.
EFFSR	Strain rate.
EDIS	Distortional energy.
FBCFI	Fiber-compression index.
FBTFI	Fiber-tension index.
MXCFI	Matrix-compression index.
MXTFI	Matrix-tension index.
SHRFI	Shear-failure index.
Q1XX	Composite-fiber orientation.
Q2XX	Composite-matrix orientation.
USR1L	User variable 1
USR2L	User variable 2
USR3L	User variable 3
USR4L	User variable 4
USR5L	User variable 5
USR6L	User variable 6
USR7L	User variable 7
USR8L	User variable 8
USR9L	User variable 9
USR10L	User variable 10
FIBFL	Fiber-failure flag.
MTXFL	Matrix-failure flag.
SHRFL	Shear-failure flag.
TYZLIN	Linear stress in yz-direction.
TZXLIN	Linear stress in zx-direction.
EPSMX	Major principal strain.
EPSMN	Minor principal strain.
EPZZ	Effective plastic strain in transverse direction.
FLP	Forming limit parameter.
YLDRAD	Yield radius.

Triangular Membrane Elements

Keyword	Description
MASS	Element mass.
THICK	Thickness.
TXX	xx-stress (element coordinate system).
TYY	yy-stress (element coordinate system).
TXY	xy-stress (element coordinate system).
EFFPLS	Effective plastic strain.
EFFSTS	Equivalent stress.
AREA	Original area.
SLEN21	Side length 21.
SLEN32	Side length 32.
SLEN31	Side length 31.
X2HAT	x-coordinate point 2 in corotational frame.
Y2HAT	y-coordinate point 2 in corotational frame.
X3HAT	x-coordinate point 3 in corotational frame.
Y3HAT	y-coordinate point 3 in corotational frame.
EPSXX	xx-membrane strain.
EPSYY	yy-membrane strain.
EPSXY	xy-membrane strain.
FAC1	Mass fraction of the first grid point.
FAC2	Mass fraction of the second grid point.
FAC3	Mass fraction of the third grid point.
ECF3X	x-direction of corotational base-vector 3.
ECF3Y	y-direction of corotational base-vector 3.
ECF3Z	z-direction of corotational base-vector 3.
SMDFER	Ratio of current and original area.
EXUSER1	User variable 1.
EXUSER2	User variable 2.
ELTIME	Time at which element is updated.
ALPSTBL	Stable alpha of element (subcycling).
ELGROUP	Subcycling group of element.

Dummy Quads and Trias

Keyword	Description
ZUSER	User-specified.

Lagrangian Solid Elements

Keyword	Description
TXX	xx-stress.
TYY	yy-stress.
TZZ	zz-stress.
TXY	xy-stress.
TYZ	yz-stress.
TZX	zx-stress.
EFFSTS	Effective stress.
PRESSURE	Pressure.
EFFPLS	Effective plastic strain.
SIE	Specific internal energy.
EDIS	Distortional energy.
TDET	Detonation time.
MASS	Element mass.
VOLUME	Volume.
HALFQ	Half of the artificial viscosity.
SNDSPEED	Sound speed.
EXUSER1	User variable 1.
EXUSER2	User variable 2.
ELTIME	Time at which element is updated.
ALPSTBL	Stable alpha of element (subcycling).
ELGROUP	Subcycle group of element.

Lagrangian Solid Elements; Nonhydrodynamic Materials Only

Keyword	Description
EPSXX	xx-centroidal strain.
EPSYY	yy-centroidal strain.
EPSZZ	zz-centroidal strain.
EPSXY	xy-centroidal strain.
EPSYZ	yz-centroidal strain.
EPSZX	zx-centroidal strain.

Lagrangian Solid Elements; Orthotropic Materials Only

Keyword	Description
AX	x-component of material axis a.
AY	y-component of material axis a.
AZ	z-component of material axis a.
BX	x-component of material axis b.
BY	y-component of material axis b.
BZ	z-component of material axis b.
CX	x-component of material axis c.
CY	y-component of material axis c.
CZ	z-component of material axis c.
EXX	xx-Young’s modulus.
EYY	yy-Young’s modulus.
EZZ	zz-Young’s modulus.
EXY	xy-Young’s modulus.
EYZ	yz-Young’s modulus.
EZX	zx-Young’s modulus.
GXY	xy-shear modulus.
GYZ	yz-shear modulus.
GZX	zx-shear modulus.
EMAX	Maximum Young’s modulus.

Eulerian Solid Elements (Hydrodynamic)

Keyword	Description
XVEL	x-velocity.
YVEL	y-velocity.
ZVEL	z-velocity.
VOLUME	Element volume.
MASS	Mass of material 1.
DENSITY	Density of material 1.
SIE	Specific internal energy of material 1.
PRESSURE	Pressure.
Q	Artificial viscosity.
ENERGY	Total energy (kinetic + internal) of material 1.
DIV	Divergence.
VOID	Void fraction.
FMAT	Material fraction of material 1.
SSPD	Speed of sound.
FBURN	Burn fraction.
TDET	Detonation time.
BFTIME	Burn time.
XMOM	x-momentum.
YMOM	y-momentum.
ZMOM	z-momentum.
FVUNC	Volume uncovered fraction.
QDIS	Characteristic element length.
FVUOLD	Old volume uncovered fraction.
EXUSER1	User variable 1.
EXUSER2	User variable 2.
MFLR	Total massflow rate (sum of MFLR-POR, MFLR-PER and MFLR-INF).
MFL	Total massflow (sum of MFLR-POR, MFLR-PER and MFLR-INF).
MFLR-POR	Massflow rate by PORFGBG (flow between the Euler element and GBAGs through holes) or PORHOLE (flow between the Euler element and the environment through holes).

Keyword	Description
MFL-POR	Massflow by PORFGBG (flow between the Euler element and GBAGs through holes) or PORHOLE (flow between the Euler element and the environment through holes).
MFLR-PER	Massflow rate by PERMGBG (flow between the Euler element and GBAGs through permeable (SUB)SURFACEs) or PERMEAB (flow between the Euler element and the environment through permeable (SUB)SURFACEs).
MFL-PER	Massflow by PERMGBG (flow between the Euler element and GBAGs through permeable (SUB)SURFACEs) or PERMEAB (flow between the Euler element and the environment through permeable (SUB)SURFACEs).
MFLR-INF	Massflow rate by inflators (inflow by INFLATRs and EXFLOW,INFLATR3).
MFL-INF	Massflow by inflators (inflow by INFLATRs and EXFLOW,INFLATR3).
DQDT	Total heat transfer rate (sum of DQDT-CNV and DQDT-RAD).
DQ	Total heat transfer (sum of DQ-CNV and DQ-RAD).
DQDT-CNV	Heat transfer rate by HTRCONV (transfer from the Euler element to the environment through (SUB)SURFACEs)
DQ-CNV	Heat transfer by HTRCONV (transfer from the Euler element to the environment through (SUB)SURFACEs).
DQDT-RAD	Heat transfer rate by HTRRAD (transfer from the Euler element to environment through (SUB)SURFACEs).
DQ-RAD	Heat transfer by HTRRAD (transfer from the Euler element to the environment through (SUB)SURFACEs).

Remarks:

1. All variables starting with MFL (MassFLow) have following sign conventions:

   MFLxx > 0 means inflow ; this leads to an increase of mass in the Euler element.
   MFLxx < 0 means outflow ; this leads to an decrease of mass in the Euler element.

2. All variables starting with DQ have following sign conventions:

   DQxx > 0 means inflow ; ; this leads to an increase of heat transfer in the Euler element.
   DQxx < 0 means outflow ; ; this leads to an decrease of heat transfer in the Euler element.

Eulerian Solid Elements (Viscous Hydrodynamic)

Keyword	Description
XVEL	x-velocity.
YVEL	y-velocity.
ZVEL	z-velocity.
VOLUME	Element volume.
MASS	Mass of material 1.
DENSITY	Density of material 1.
SIE	Specific internal energy of material 1.
PRESSURE	Pressure.
Q	Artificial viscosity.
ENERGY	Total energy (kinetic + internal) of material 1.
DIV	Divergence.
VOID	Void fraction.
FMAT	Material fraction of material 1.
SSPD	Speed of sound.
FBURN	Burn fraction.
TDET	Detonation time.
BFTIME	Burn time.
XMOM	x-momentum.
YMOM	y-momentum.
ZMOM	z-momentum.
FVUNC	Volume uncovered fraction.
QDIS	Characteristic element length.
SXX	xx-deviatoric stress.
SYY	yy-deviatoric stress.
SZZ	zz-deviatoric stress.
TXX	xx-stress.
TYY	yy-stress.
TZZ	zz-stress.
TXY	xy-stress.
TYZ	yz-stress.

Keyword	Description
TZX	zx-stress.
EFFSTS	Effective stress.
DUDX	x-velocity gradient in x-direction.
DUDY	x-velocity gradient in y-direction.
DUDZ	x-velocity gradient in z-direction.
DVDX	y-velocity gradient in x-direction.
DVDY	y-velocity gradient in y-direction.
DVDZ	y-velocity gradient in z-direction.
DWDX	z-velocity gradient in x-direction.
DWDY	z-velocity gradient in y-direction.
DWDZ	z-velocity gradient in z-direction.
FVUOLD	Old volume uncovered fraction.
EXUSER1	User variable 1.
EXUSER2	User variable 2.
MFLR	Total massflow rate (sum of MFLR-POR, MFLR-PER and MFLR-INF).
MFL	Total massflow (sum of MFLR-POR, MFLR-PER and MFLR-INF).
MFLR-POR	Massflow rate by PORFGBG (flow between the Euler element and GBAGs through holes) or PORHOLE (flow between the Euler element and the environment through holes).
MFL-POR	Massflow by PORFGBG (flow between the Euler element and GBAGs through holes) or PORHOLE (flow between the Euler element and the environment through holes).
MFLR-PER	Massflow rate by PERMGBG (flow between the Euler element and GBAGs through permeable (SUB)SURFACEs) or PERMEAB (flow between the Euler element and the environment through permeable (SUB)SURFACEs).
MFL-PER	Massflow by PERMGBG (flow between the Euler element and GBAGs through permeable (SUB)SURFACEs) or PERMEAB (flow between the Euler element and the environment through permeable (SUB)SURFACEs).
MFLR-INF	Massflow rate by inflators (inflow by INFLATRs and EXFLOW,INFLATR3).
MFL-INF	Massflow by inflators (inflow by INFLATRs and EXFLOW,INFLATR3).
DQDT	Total heat transfer rate (sum of DQDT-CNV and DQDT-RAD).
DQ	Total heat transfer (sum of DQ-CNV and DQ-RAD).
DQDT-CNV	Heat transfer rate by HTRCONV (transfer from the Euler element to the environment through (SUB)SURFACEs)

Keyword	Description
DQ-CNV	Heat transfer by HTRCONV (transfer from the Euler element to the environment through (SUB)SURFACEs).
DQDT-RAD	Heat transfer rate by HTRRAD (transfer from the Euler element to environment through (SUB)SURFACEs).
DQ-RAD	Heat transfer by HTRRAD (transfer from the Euler element to the environment through (SUB)SURFACEs).

Remarks:

1. All variables starting with MFL (MassFLow) have following sign conventions:

   MFLxx > 0 means inflow ; this leads to an increase of mass in the Euler element.
   MFLxx < 0 means outflow ; this leads to an decrease of mass in the Euler element.

2. All variables starting with DQ have following sign conventions:

   DQxx > 0 means inflow ; ; this leads to an increase of heat transfer in the Euler element.
   DQxx < 0 means outflow ; ; this leads to an decrease of heat transfer in the Euler element.

Eulerian Solid Elements (with Shear Strength)

Keyword	Description
XVEL	x-velocity.
YVEL	y-velocity.
ZVEL	z-velocity.
TXX	xx-stress (basic coordinate system).
TYY	yy-stress (basic coordinate system).
TZZ	zz-stress (basic coordinate system).
TXY	xy-stress (basic coordinate system).
TYZ	yz-stress (basic coordinate system).
TZX	zx-stress (basic coordinate system).
EFFSTS	Effective stress.
PRESSURE	Pressure.
SXX	xx-deviatoric stress (basic coordinate system).
SYY	yy-deviatoric stress (basic coordinate system).
SZZ	zz-deviatoric stress (basic coordinate system).
SXY	xy-deviatoric stress (basic coordinate system).
SYZ	yz-deviatoric stress (basic coordinate system).
SZX	zx-deviatoric stress (basic coordinate system).
SINWX	x-stress rotation angle.
SINWY	y-stress rotation angle.
SINWZ	z-stress rotation angle.
EFFPLS	Effective plastic strain.
EPSXXD	xx-strain rate.
EPSYYD	yy-strain rate.
EPSZZD	zz-strain rate.
EPSXYD	xy-strain rate.
EPSYZD	yz-strain rate.
EPSZXD	zx-strain rate.
VOLUME	Element volume.
MASS	Element mass.
DENSITY	Element density.
SIE	Specific internal energy.
EDIS	Distortional energy.
ENERGY	Total energy (internal + kinetic).

Keyword	Description
Q	Artificial viscosity.
DIV	Divergence.
VOID	Void fraction.
FMAT	Material fraction.
SSPD	Speed of sound.
FBURN	Burn fraction.
TDET	Detonation time.
BFTIME	Burn time.
XMOM	x-momentum.
YMOM	y-momentum.
ZMOM	z-momentum.
FVUNC	Volume uncovered fraction.
QDIS	Characteristic element length.
FVUOLD	Old uncovered fraction.
EXUSER1	User variable 1.
EXUSER2	User variable 2.

Eulerian Solid Elements (Multimaterial Hydrodynamic)

Keyword	Description
XVEL	x-velocity.
YVEL	y-velocity.
ZVEL	z-velocity.
VOLUME	Element volume.
MASS	Mass of material 1.
DENSITY	Density of material 1.
SIE	Specific internal energy.
PRESSURE	Pressure.
Q	Artificial viscosity.
ENERGY	Total energy (kinetic + internal) of material 1.
DIV	Divergence of material 1.
VOID	Void fraction.
FMAT	Material fraction of material 1.
SSPD	Speed of sound.
FBURN	Burn fraction.

Keyword	Description
TDET	Detonation time.
BFTIME	Burn time.
XMOM	x-momentum.
YMOM	y-momentum.
ZMOM	z-momentum.
FVUNC	Volume uncovered fraction.
QDIS	Characteristic element length.
FVUOLD	Old volume uncovered fraction.
MASST	Total mass of element.
EXUSER1	User variable 1.
EXUSER2	User variable 2.

Note: Multimaterial Editing

The different materials in an element are in pressure equilibrium, but they have their own density, specific internal energy, and material volume fraction. The variables stored in element data storage relate to the material in the element that is stored as the first one internally. Multimaterial editing can be performed for a material with material user number xx using the following variables:

MASSxx	Mass of material xx.
DENSITYxx	Density of material xx.
SIExx	Specific internal energy of material xx.
ENERGYxx	Total energy of material xx.
DIVxx	Divergence of material xx.
FMATxx	Volume fraction of material xx.

Leading zeros in material user numbers are not acceptable.

Eulerian Solid Elements (Multimaterial with Shear Strength)

Keyword	Description
XVEL	x-velocity.
YVEL	y-velocity.
ZVEL	z-velocity.
VOLUME	Element volume.
MASS	Mass of material 1.
DENSITY	Density of material 1.
SIE	Specific internal energy of material 1.
PRESSURE	Pressure.
Q	Artificial viscosity.
ENERGY	Total energy (kinetic + internal) of material 1.
DIV	Divergence.
VOID	Void fraction.
EPSXXD	xx-strain rate.
EPSYYD	yy-strain rate.
EPSZZD	zz-strain rate.
EPSXYD	xy-strain rate.
EPSYZD	yz-strain rate.
EPSZXD	zx-strain rate.
SINWX	x-stress rotation angle.
SINWY	y-stress rotation angle.
SINWZ	z-stress rotation angle.
SXX	xx-deviatoric stress.
SYY	yy-deviatoric stress.
SZZ	zz-deviatoric stress.
SXY	xy-deviatoric stress.
SYZ	yz-deviatoric stress.
SZX	zx-deviatoric stress.
TXX	xx-stress.
TYY	yy-stress.

Keyword	Description
TZZ	zz-stress.
TXY	xy-stress.
TYZ	yz-stress.
TZX	zx-stress.
EFFPLS	Effective plastic strain.
EFFSTS	Effective stress.
EDIS	Distortional energy of material 1.
VOLOLD	Old volume.
EXUSER1	User variable1.
EXUSER2	User variable2.

Note: Multimaterial Editing

For more information on multimaterial editing, refer to the note on page 3-49.

3.9.2.3 Material Results (MATOUT)

Keyword	Description
XMOM	x-component of momentum.
YMOM	y-component of momentum.
ZMOM	z-component of momentum.
EKIN	Kinetic energy.
EINT	Internal energy.
EDIS	Distortional energy.
VOLUME	Volume.
MASS	Mass.
EHRG	Hourglass energy.

3.9.2.4 Eulerian Boundary Results (EBDOUT))

Keyword	Description
XMOM	x-component of transported momentum.
YMOM	y-component of transported momentum.
ZMOM	z-component of transported momentum.
FX	x-component of force acting on boundary.
FY	y-component of force acting on boundary.
FZ	z-component of force acting on boundary.
MFL	Total mass flow through boundary.
MFLR	Mass flow rate through boundary.
ENERGY	Energy transport through boundary.
XVEL	x-component of velocity through boundary.
YVEL	y-component of velocity through boundary.
ZVEL	z-component of velocity through boundary.
XIMP	x-component of impulse acting on boundary.
YIMP	y-component of impulse acting on boundary.
ZIMP	z-component of impulse acting on boundary.

3.9.2.5 Rigid Body Results (RBOUT)

Keyword	Description
XCG	x-coordinate of center of gravity.
YCG	y-coordinate of center of gravity.
ZCG	z-coordinate of center of gravity.
INERTIA1	First principal moment of inertia.
INERTIA2	Second principal moment of inertia.
INERTIA3	Third principal moment of inertia.
A11	x-component of the first principal axis.
A21	y-component of the first principal axis.
A31	z-component of the first principal axis.
A12	x-component of the second principal axis.
A22	y-component of the second principal axis.
A32	z-component of the second principal axis.
A13	x-component of the third principal axis.
A23	y-component of the third principal axis.
A33	z-component of the third principal axis.
XVEL	x-translational velocity.
YVEL	y-translational velocity.
ZVEL	z-translational velocity.
XAVEL	x-angular velocity.
YAVEL	y-angular velocity.
ZAVEL	z-angular velocity.
XLAVEL	x-angular velocity in the local coordinate system.
YLAVEL	y-angular velocity in the local coordinate system.
ZLAVEL	z-angular velocity in the local coordinate system.
XFORCE	x-force.
YFORCE	y-force.
ZFORCE	z-force.
XMOMENT	x-moment.
YMOMENT	y-moment.

Keyword	Description
ZMOMENT	z-moment.
XACC	x-translational acceleration.
YACC	y-translational acceleration.
ZACC	z-translational acceleration.
RCG	Resultance coordinate of center of gravity.
RVEL	Resultance translational velocity.
RACC	Resultance acceleration.
RFORCE	Resultance force = external + internal.
RAVEL	Resultance angular velocity.
RMOMENT	Resultance moment = external + internal.
RLAVEl	Resultance angular velocity in the local coordinate system.

Remark:

1. All variables starting with R (Resultance) are calculated as follows:

$$Rxx = \sqrt{Xxx^2 + Yxx^2 + Zxx^2}$$

3.9.2.6 Rigid Ellipsoid Results (RELOUT)

Keyword	Description
GEOMETRY	The ellipsoid is covered with dummy elements, which are written to an archive file so the ellipsoid can be visualized when postprocessing.
XIMP	x-impulse constraint.
YIMP	y-impulse constraint.
ZIMP	z-impulse constraint.
WORK	Work done by constraint.
XPULSE	x-pulse acting on ellipsoid.
YPULSE	y-pulse acting on ellipsoid.
ZPULSE	z-pulse acting on ellipsoid.
XPUMOM	x-pulse moment acting on ellipsoid.
YPUMOM	y-pulse moment acting on ellipsoid.
ZPUMOM	z-pulse moment acting on ellipsoid.
XCGEO	x-coordinate of the geometrical center.

Keyword	Description
YCGEO	y-coordinate of the geometrical center.
ZCGEO	z-coordinate of the geometrical center.
A11	x-component of the first principal axis (a-axis).
A21	y-component of the first principal axis (a-axis).
A31	z-component of the first principal axis (a-axis).
A12	x-component of the second principal axis (b-axis).
A22	y-component of the second principal axis (b-axis).
A32	z-component of the second principal axis (b-axis).
A13	x-component of the third principal axis (c-axis).
A23	y-component of the third principal axis (c-axis).
A33	z-component of the third principal axis (c-axis).
INERTIA1	First principal moment of inertia.
INERTIA2	Second principal moment of inertia.
INERTIA3	Third principal moment of inertia.
MASS	Mass.
XVEL	x-component of translational velocity of geometrical center.
YVEL	y-component of translational velocity of geometrical center.
ZVEL	z-component of translational velocity of geometrical center.
OMEGAA	x-component of angular velocity in the local coordinate system (around a-axis).
OMEGAB	y-component of angular velocity in the local coordinate system (around b-axis).
OMEGAC	z-component of angular velocity in the local coordinate system (around c-axis).
A	Half length of the local x-axis.
B	Half length of the local y-axis.
C	Half length of the local z-axis.
CI	“Stiffness” of ellipsoid, used in contact logic.
TXPULS	Accumulated xpulse (used for coupling to the external program).
TYPULS	Accumulated ypulse (used for coupling to the external program).
TZPULS	Accumulated zpulse (used for coupling to the external program).
TXPMOM	Accumulated xpumom (used for coupling to the external program).
TYPMOM	Accumulated ypumom (used for coupling to the external program).
TZPMOM	Accumulated zpumom (used for coupling to the external program).

Keyword	Description
XAVEL	x-component of angular velocity in the global coordinate system.
YAVEL	y-component of angular velocity in the global coordinate system.
ZAVEL	z-component of angular velocity in the global coordinate system.
XCG	x-coordinate of the center of gravity.
YCG	y-coordinate of the center of gravity.
ZCG	z-coordinate of the center of gravity.
XACC	x-translational acceleration.
YACC	y-translational acceleration.
ZACC	z-translational acceleration.
RCG	Resultance coordinate of center of gravity.
RVEL	Resultance translational velocity.
RACC	Resultance acceleration.
RAVEL	Resultance angular velocity in the global coordinate system.
OMEGAR	Resultance angular velocity in the local coordinate system.
RIMP	Resultance impulse constraint.
RPULSE	Resultance pulse acting on ellipsoid.
RPUMOM	Resultance pulse moment acting on ellipsoid.
RCGEO	Resultance coordinate of the geometrical center.

Remark:

1. All variables starting with *R* (Resultance), including OMEGAR, are calculated as follows:

$$Rxx = \sqrt{Xxx^2 + Yxx^2 + Zxx^2} .$$

3.9.2.7 Rigid Plane Results (PLNOUT))

Keyword	Description
GEOMETRY	The rigid plane is covered with dummy elements, which are written to an archive file so the plane can be visualized when postprocssing.

Remark:

1. Only planes referenced from MADYMO with the RPLEX card can be asked for output.

3.9.2.8 Gas Bag Results (GBAGOUT)

Keyword	Description
CDEX	Discharge coefficient for the exhaust openings.
AEX	Total area of the exhaust openings.
PEX	Pressure level above which the flow out of the air bag through the holes starts.
CDLEAK	Discharge coefficient for the permeability of the bag.
ALEAK	Effective leak area.
PSTOP	Pressure level below which the flow out of the gas bag stops.
PENV	Environmental pressure.
RGAS	Specific gas constant.
FLGAS	Inflow of gas from inflator.
TGAS	Temperature of the inflowing gas.
CPGAS	Specific heat capacity of the gas at a constant pressure.
VOLUME	Volume inside the gas bag.
VOLY	Volume inside the gas bag, calculated by the y-plane projection.
VOLZ	Volume inside the gas bag, calculated by the z-plane projection.
VFLUX	Rate of change of the gas-bag volume.
TEMP	Temperature inside the gas bag.
TFLUX	Rate of change of temperature inside the gas bag.
PRESSURE	Pressure of the gas bag during the previous steps.
VOLPREV	Volume at previous time step.
MASS	Mass inside the gas bag.
MFLR	Total massflow rate (sum of MFLR-POR, MFLR-PER and MFLR-INF).
MFL	Total massflow (sum of MFLR-POR, MFLR-PER and MFLR-INF).
MFLR-POR	Massflow rate by PORFGBG (flow between the GBAG and another GBAG through holes) or PORHOLE (flow between the GBAG the environment through holes).
MFL-POR	Massflow by PORFGBG (flow between the GBAG and another GBAG through holes) or PORHOLE (flow between the GBAG the environment through holes).
MFLR-PER	Massflow rate by PERMGBG (flow between the GBAG and another GBAG through permeable (SUB)SURFACEs) or PERMEAB (flow between the GBAG and the environment through permeable (SUB)SURFACEs).
MFL-PER	Massflow by PERMGBG (flow between the GBAG and another GBAG through permeable (SUB)SURFACEs) or PERMEAB (flow between the GBAG and the environment through permeable (SUB)SURFACEs).

Keyword	Description
MFLR-INF	Massflow rate by inflators (inflow by INFLATRs and EXFLOW,INFLATR3).
MFL-INF	Massflow by inflators (inflow by INFLATRs and EXFLOW,INFLATR3).
DQDT	Total heat transfer rate (sum of DQDT-CNV and DQDT-RAD).
DQ	Total heat transfer (sum of DQ-CNV and DQ-RAD).
DQDT-CNV	Heat transfer rate by HTRCONV (transfer from the GBAG to the environment through (SUB)SURFACEs).
DQ-CNV	Heat transfer by HTRCONV (transfer from the GBAG to the environment through (SUB)SURFACEs).
DQDT-RAD	Heat transfer rate by HTRRAD (transfer from the GBAG to environment through (SUB)SURFACEs).
DQ-RAD	Heat transfer by HTRRAD (transfer from the GBAG to the environment through (SUB)SURFACEs).

Remarks:

1. All variables starting with MFL (MassFLow) have following sign convention:

   MFLxx > 0 means inflow ; this leads to an increase of mass in the Euler element.
   MFLxx < 0 means outflow ; this leads to an decrease of mass in the Euler element.

2. All variables starting with DQ have following sign convention:

   DQxx > 0 means inflow ; ; this leads to an increase of heat transfer in the Euler element.
   DQxx < 0 means outflow ; ; this leads to an decrease of heat transfer in the Euler element.

3.9.2.9 Contact Surface Results (CONTOUT)

Keyword	Description
DMIN	Overall smallest distance between a slave node and a master face.
XFORCE	x-component of contact force.
YFORCE	y-component of contact force.
ZFORCE	z-component of contact force.
FMAGN	Magnitude of contact force.
XACC	x-component of acceleration.
YACC	y-component of acceleration.
ZACC	z-component of acceleration.
AMAGN	Magnitude of acceleration.

3.9.2.10 Cross Section Results (CSOUT)

Keyword	Description
XFORCE	x-component of cross-section force.
YFORCE	y-component of cross-section force.
ZFORCE	z-component of cross-section force.
FMAGN	Magnitude of cross-section force.

Cross sections can be arbitrarily defined by lists of grid points and elements by the SECTION entry as explained in Chapter 4, Section 4.6 on page 4-78. The output that can be requested for the cross section consists of the x-, y-, and z-components of the total force on the cross section, where the components are defined in the basic system. The total force (the magnitude of the force) can also be requested for output. See the Case Control Section CSOUT and CSECS entries for the definition.

The list of grid points is used to define the geometry of the cross section. The list of elements is used to define the orientation of the cross section and may be a mixture of different element types. Each grid point must therefore be attached to one of the elements. Cross-section data can only be written in the form of a time-history file.

3.9.2.11 Coupling-Surface Results (CPLSOUT)

Any Eulerian variable can be requested for output on a coupling surface.

The variable output on a coupling-surface element is computed as the sum of the variables of the Eulerian elements times the intersection area of the coupling-surface element with the Eulerian element, divided by the total area of the coupling-surface element.

3.9.2.12 Surface Results (SURFOUT)

A SURFACE referenced from a COUPLE entry will enclose a number of Eulerian elements either completely or partly. The Euler elements enclosed have certain variables available for output (e.g., the material MASS, DENSITY, TEMPTURE, etc.). The sum or the average of these values are available as SURFACE output. The table below lists which variables are summed up and which variables are averaged.

This option is only available for the Single Material Hydrodynamic Euler processor: PEULER with TYPE = HYDRO.

The same variables are made available as SURFACE output when the SURFACE is referenced from a GBAG entry. This makes the output transparent in case of airbag analyses where a switch is made from a Euler coupled model to a GBAG model during the calculation. When a variable is not used in the GBAG approach (e.g., XVEL, XMOM) it will get a value equal to zero.

Apart from these variables related to the Eulerian elements, also the total AREA of the SURFACE is available.

Furthermore, some variables that apply to the GBAG logic only (e.g., MFLR-GBG as described in Section 3.9.2.8 on page 3-57, Gas Bag Results (GBAGOUT)) are available.

The table below lists which variables are summed up, and which variables are averaged. Please refer to the following sections for a more detailed description of each variable:

Section 3.9.2.2 on page 3-33	Element Results (ELOUT) Eulerian Solid Elements (Hydrodynamic)
Section 3.9.2.8 on page 3-57	Gas Bag Results (GBAGOUT)

Some variables of the Eulerian elements are not listed here. These are available for output and are summed up, but they are not as useful and thus, are not listed.

Keyword	Description
AREA	Total Area of the SURFACE.
VOLUME	Total Volume enclosed by the SURFACE.
MASS	Sum of MASS in the intersected Euler elements or GBAG.
ENERGY	Sum of ENERGY in the intersected Euler elements or GBAG.

Keyword	Description
XMOM	Sum of XMOM in the intersected Euler elements or GBAG.
YMOM	Sum of YMOM in the intersected Euler elements or GBAG.
ZMOM	Sum of ZMOM in the intersected Euler elements or GBAG.
MFLR	Sum of MFLR in the intersected Euler elements or GBAG.
MFLR-INF	Sum of MFLR-INF in the intersected Euler elements or GBAG.
MFLR-POR	Sum of MFLR-POR in the intersected Euler elements or GBAG.
MFLR-PER	Sum of MFLR-PER in the intersected Euler elements or GBAG.
MFL	Sum of MF in the intersected Euler elements or GBAG.
MFL-INF	Sum of MFL-INF in the intersected Euler elements or GBAG.
MFL-POR	Sum of MFL-POR in the intersected Euler elements or GBAG.
MFL-PER	Sum of MFL-PER in the intersected Euler elements or GBAG.
MFLR-GBG	Sum of MFLR-GBG in the GBAG.
MFL-GBG	Sum of MFL-GBG in the GBAG.
DQDT	Sum of DQDT in the intersected Euler elements or GBAG.
DQDT-CNV	Sum of DQDT-CNV in the intersected Euler elements or GBAG.
DQDT-RAD	Sum of DQDT-CNV in the intersected Euler elements or GBAG.
DQ	Sum of DQ in the intersected Euler elements or GBAG.
DQ-CNV	Sum of DQ-CNV in the intersected Euler elements or GBAG.
DQ-RAD	Sum of DQ-RAD in the intersected Euler elements or GBAG.
DENSITY	Average value of DENSITY in the intersected Euler elements or GBAG.
PRESSURE	Average value of PRESSURE in the intersected Euler elements or GBAG.
TEMPTURE	Average value of TEMPTURE in the intersected Euler elements or GBAG.
SIE	Average value of SIE in the intersected Euler elements or GBAG.
Q	Average value of Q in the intersected Euler elements or GBAG.
DIV	Average value of DIV in the intersected Euler elements or GBAG.
FMAT	Average value of FMAT in the intersected Euler elements or GBAG.
SSPD	Average value of SSPD in the intersected Euler elements or GBAG.
FVUNC	Average value of FVUNC in the intersected Euler elements or GBAG.
QDIS	Average value of QDIS in the intersected Euler elements or GBAG.
XVEL	Average value of XVEL in the intersected Euler elements or GBAG.
YVEL	Average value of YVEL in the intersected Euler elements or GBAG.
ZVEL	Average value of ZVEL in the intersected Euler elements or GBAG.

Remarks:

1. All variables starting with MFL (MassFLow) have the following sign conventions:

   MFLxx > 0 means inflow ; this leads to an increase of mass in the Euler element.
   MFLxx < 0 means outflow ; this leads to an decrease of mass in the Euler element.

2. All variables starting with DQ have the following sign conventions:

   DQxx > 0 means inflow ; this leads to an increase of heat transfer in the Euler element.
   DQxx < 0 means outflow ; this leads to an decrease of heat transfer in the Euler element.

3.9.2.13 Subsurface Results (SUBSOUT)

A SUBSURFACE that belongs to a SURFACE referenced from a COUPLE entry intersects a number of Eulerian elements. The intersected Euler elements have certain variables available for output (e.g., the material MASS, DENSITY, TEMPTURE, etc.). The sum or the average of these values are available as SUBSURFACE output. The table below lists which variables are summed up and which variables are averaged.

This option is only available for the Single Material Hydrodynamic Euler processor: PEULER with TYPE = HYDRO.

The same variables are made available as SUBSURFACE output when the SURFACE it belongs to is referenced from a GBAG entry. This makes the output transparent in case of airbag analyses where a switch is made from an Euler coupled model to a GBAG model during the calculation. When a variable is not used in the GBAG approach (e.g., XVEL, XMOM) it will get a value equal to zero.

Apart from these variables related to the Eulerian elements, the total AREA of the SUBSURFACE is also available.

Furthermore, some variables that apply to the GBAG logic only (e.g., MFLR-GBG as described in Section 3.9.2.8 on page 3-57, Gas Bag Results (GBAGOUT)) are available.

The table below lists which variables are summed up, and which variables are averaged. Please refer to the following sections for a more detailed description of each variable:

Section 3.9.2.2 on page 3-33	Element Results (ELOUT) Eulerian Solid Elements (Hydrodynamic)
Section 3.9.2.8 on page 3-57	Gas Bag Results (GBAGOUT)

Some variables of the Eulerian elements are not listed here. These are available for output and are summed up, but they are not as useful and thus, are not listed.

Keyword	Description
AREA	Total Area of the SUBSURFACE.
PRESDIFF	Average value of PRESSURE difference between intersected Euler elements or GBAG and environment.
TEMPDIFF	Average value of TEMPERATURE difference between intersected Euler elements or GBAG and environment.
VALPMB	Total PERMEABILITY value of subsurface (VOLUME/AREA/SECOND).
MASS	Sum of MASS in the intersected Euler elements or GBAG.
ENERGY	Sum of ENERGY in the intersected Euler elements or GBAG.
XMOM	Sum of XMOM in the intersected Euler elements or GBAG

Keyword	Description
YMOM	Sum of YMOM in the intersected Euler elements or GBAG.
ZMOM	Sum of ZMOM in the intersected Euler elements or GBAG.
MFLR	Sum of MFLR in the intersected Euler elements or GBAG.
MFLR-INF	Sum of MFLR-INF in the intersected Euler elements or GBAG.
MFLR-POR	Sum of MFLR-POR in the intersected Euler elements or GBAG.
MFLR-PER	Sum of MFLR-PER in the intersected Euler elements or GBAG.
MFL	Sum of MFL in the intersected Euler elements or GBAG.
MFL-INF	Sum of MFL-INF in the intersected Euler elements or GBAG.
MFL-POR	Sum of MFL-POR in the intersected Euler elements or GBAG.
MFL-PER	Sum of MFL-PER in the intersected Euler elements or GBAG.
MFLR-GBG	Sum of MFLR-GBG in the GBAG.
MFL-GBG	Sum of MFL-GBG in the GBAG.
DQDT	Sum of DQDT in the intersected Euler elements or GBAG.
DQDT-CNV	Sum of DQDT-CNV in the intersected Euler elements or GBAG.
DQDT-RAD	Sum of DQDT-CNV in the intersected Euler elements or GBAG.
DQ	Sum of DQ in the intersected Euler elements or GBAG.
DQ-CNV	Sum of DQ-CNV in the intersected Euler elements or GBAG.
DQ-RAD	Sum of DQ-RAD in the intersected Euler elements or GBAG.
DENSITY	Average value of DENSITY in the intersected Euler elements or GBAG.
PRESSURE	Average value of PRESSURE in the intersected Euler elements or GBAG.
TEMPTURE	Average value of TEMPTURE in the intersected Euler elements or GBAG.
SIE	Average value of SIE in the intersected Euler elements or GBAG.
Q	Average value of Q in the intersected Euler elements or GBAG.
DIV	Average value of DIV in the intersected Euler elements or GBAG.
FMAT	Average value of FMAT in the intersected Euler elements or GBAG.
SSPD	Average value of SSPD in the intersected Euler elements or GBAG.
FVUNC	Average value of FVUNC in the intersected Euler elements or GBAG.
QDIS	Average value of QDIS in the intersected Euler elements or GBAG.
XVEL	Average value of XVEL in the intersected Euler elements or GBAG.
YVEL	Average value of YVEL in the intersected Euler elements or GBAG.
ZVEL	Average value of ZVEL in the intersected Euler elements or GBAG.

Remarks:

1. All variables starting with MFL (MassFLow) have the following sign conventions:

   MFLxx > 0 means inflow ; this leads to an increase of mass in the Euler element.
   MFLxx < 0 means outflow ; this leads to an decrease of mass in the Euler element.

2. All variables starting with DQ have following sign conventions:

   DQxx > 0 means inflow ; ; this leads to an increase of heat transfer in the Euler element.
   DQxx < 0 means outflow ; ; this leads to an decrease of heat transfer in the Euler element.

3.9.2.14 USA Surface Results (USASOUT)

Keyword	Description
PRESSURE	Pressure supplied by USA.

USA surface data can only be written in the form of an archive file.

3.9.2.15 Surface Gauge Results (SGOUT)

Keyword	Description
PRESSURE	Pressure supplied by USA.

Surface gauge data can only be written in the form of a time-history file.

3.10 Restarts

3.10.1 Creating Restart Files

A restart is used when the analysis is proceeding correctly and you want to run the next stage. You must ensure that restart files are available from the initial run. The following FMS statements and Case Control commands write restart data to a file (logical name RST1) at 0.5E–3 and 1.0E–3 seconds.

```
TYPE(RST1) = RESTART
CEND
TIMES(RST1) = 0.5E-3, 1.0E-3
```

It is not necessary to define which grid points and elements or which data is stored in a restart file. In the example above, all the restart information is stored in one file. The FMS statement SAVE can be used to create more than one file. The STEPS Case Control command can also be used to write restart data on a time-step basis. The following input writes restart data to a file (logical name RST2) every 100 time steps, and a new file is created each time data is written.

```
TYPE(RST2) = RESTART
SAVE(RST2) = 1
CEND
STEPS (RST2) = 0, THRU, END, BY, 100
```

3.10.2 Restarting a Previous Analysis

To restart an analysis, you need an input file that only contains the FMS, Executive, and Case Control Sections for the job and includes the FMS statement RESTART. Only those Case Control options that you want to change must be included. The Bulk Data Section must be empty.

The restart file from the previous analysis is specified using the FMS statement RSTFILE, and the step from which the analysis is to continue is specified using the RSTBEGIN statement. To continue an analysis to a new termination time, your input file is:

```
RESTART
RSTFILE = filename
RSTBEGIN = step number
CEND
ENDTIME = new finish time
BEGIN BULK
ENDDATA
```

For example,

```
RESTART
RSTFILE = RUN1_RST1_322.RST
RSTBEGIN = 636
CEND
ENDTIME = 2.0E-3
BEGIN BULK
ENDDATA
```

It is possible to remove certain types of elements from the calculation when restarting. The RSTDROP parameter allows all of the Eulerian elements, Lagrangian elements, and coupling surfaces to be removed from the calculation. All of the elements of that type are removed. For example, it is not possible to remove only a few Lagrangian or Eulerian elements. Otherwise, no data can be changed when restarting except the ENDTIME, ENDSTEP, the parameters STEPMIN, RSTDROP, and parameter options.

3.10.3 Prestress Analysis

A prestress analysis based on an MSC.Nastran solution can be performed as follows:

```
PRESTRESS
BULKOUT = file-name-1
SOLUOUT = file-name-2
NASDISP = file-name-3
CEND
BEGIN BULK
NASINIT, ..., ..., ...,
....
ENDDATA
```

3.11 Controlling the Analysis

Executing the transient dynamic analysis is very simple since most of the control is automatic. For most analyses, you need to do very little. The analysis is performed, and the time step is continually adjusted by MSC.Dytran to ensure a stable solution with a minimum use of computer resources. However, there are many ways to override the automatic control and manually select the parameters that control the analysis. This is done using Case Control commands or the PARAM entry. Details for each parameter that can be set using the PARAM entry are provided in Section 4.7 on page 4-437. A brief description is given below.

Modifying the Time Step

Perhaps the most critical factor affecting the solution is the time step. By default, the time step is calculated by MSC.Dytran so that it is smaller than the time taken for a stress wave to cross the smallest element. This ensures a stable solution. The time step is recalculated at every iteration. The automatic time step works well for the vast majority of analyses, and you should change it only when you have a very good reason to do so.

The time step can be scaled using the parameter STEPFCT. The internally calculated time step is multiplied by the scale factor that you specify to get the time step actually used. The scale factor cannot be greater than 1.0, otherwise, the solution becomes unstable.

The initial time step must be specified using the parameter INISTEP. The specified time step is used for the first iteration; thereafter, the internally calculated time step is used.

The parameter MAXSTEP allows you to specify a maximum time step for the analysis. The time step cannot exceed this value.

The parameter MINSTEP lets you specify a minimum time step. When the calculated time step falls below this value, the analysis terminates.

Blending of Eulerian Elements

When using coupling surfaces, the surface may cut through Eulerian elements so that only a very small proportion of the element is uncovered. To prevent such elements from controlling the time step, the Eulerian elements are blended with adjacent elements. The parameter FBLEND allows you to specify the uncovered fraction at which elements are blended when they would otherwise control the time step.

Coupling Subcycling

Subcycling techniques are used to improve the efficiency of the coupling algorithms. The geometry of the coupling surface is only updated when required, which will depend on the motion of the surface.

The parameter COSUBMAX lets you specify the maximum number of time steps before the geometry of the coupling surface is forced to be updated. The parameter COSUBCYC allows you to control the rate of growth of the subcycling interval. When either of these parameters is specified, the subcycling is activated.

Element Subcycling

The element subcycling algorithm partitions the elements into groups of equal time steps. Each group of elements is then updated with the group time step. Especially when a few small elements determine the time step, large CPU savings can be achieved. The element subcycling algorithm can be activated with the parameter ELSUBCYC.

Limits

Various limits can be set that affect the analysis. The parameter RHOCUT defines a minimum density for Eulerian elements. When the density of an element is less than the minimum density, the element is considered empty. Each of the Eulerian solvers has its own density cutoff. These can be defined separately, although in most cases, the automatic setting is sufficient. The parameter VELCUT sets a velocity cutoff. The parameter VELMAX allows you to specify a maximum velocity in Eulerian meshes. This can be useful for near-vacuous flows. Finally, the parameter SNDLIM specifies the minimum value for the speed of sound.

3.12 Terminating the Analysis

The analysis terminates when any of the following conditions occur.

Termination Time Reached

When the time reaches the time specified by the ENDTIME Case Control command, the analysis terminates.

Termination Step Reached

When the step number reaches the number specified by the ENDSTEP Case Control command, the analysis terminates.

Insufficient CPU Time

If the CPU time (in minutes) specified on the TIME statement in Executive Control is exceeded, the analysis terminates.

Time Step Too Small

If the time step falls below the value specified by the PARAM option MINSTEP, the analysis terminates.

User Signal

On most UNIX systems, the user can send a signal to force a wrap-up of the analysis. The analysis terminates with the normal output as requested at END.

3.13 Translating the Results

MSC.Dytran can store results in either archive or time-history files, and any number of these files can be created. The format of these files is unique to MSC.Dytran to ensure that the very large quantity of data that they often contain is stored as efficiently as possible.

In order to gain access to these results you have to use a special program to translate the results into a form that your postprocessing program can understand. At present a translator is provided for MSC/XL, I-DEAS/Version 6, MSC.Patran, and FEMB. See Appendix B for details on how to use it.

The SAVE FMS statement controls how often each archive or time-history file is closed and saved and a new file opened. Once a file is saved, the results stored in the file can be postprocessed even when the analysis is still running. When SAVE is set to 1, each set of results is stored in a separate file, and the data can be postprocessed as soon as it is written.

3.14 Postprocessing

You can postprocess your results using MSC.Patran or the postprocessor you normally use for standard finite element analyses. Essentially, the usual techniques can be used, but you should consider the following suggestions.

Plot the Time Variation of Results

Use the results from the time-history files to see how particular parameters vary during the analysis. Time-variation plots can be used to identify the key times during the analysis that demand more detailed postprocessing.

Use Real Displacements

When plotting deformed shapes, set the magnification factor for the displacement to 1.0, so that real displacements, not magnified ones, are plotted. Since MSC.Dytran is a large displacement code, you can get some very odd-looking plots if you scale the deformations. In particular, contact surfaces may appear to be penetrated and parts of the mesh may seem to overlap.

Plot Contours on the Deformed Shape

Try to plot the contours of the results on the deformed shape for Lagrangian elements. This produces much more meaningful results when the deformations are large.

Plot Material Contours

The location of material within an Eulerian mesh can be determined by plotting contours of the material fraction (FMAT).

Plot the Effective Plastic Strain

MSC.Dytran outputs the effective plastic strain, a scalar measure of how much permanent deformation has occurred. The quantity is very useful for showing the amount of deformation in a component and the areas that have yielded.

Plot the Velocity Fields

Use the vector or arrow plots to view the velocity fields in the structure. Arrow plots give you a rapid indication of the way in which a structure is moving. Arrow plots are essential in understanding the flow characteristics in Eulerian meshes.

Animate the Analysis

Some postprocessing programs have the facility to animate the progression of the analysis. This can be very useful in obtaining an overall impression of what is happening during the analysis and for showing the results of the analysis to clients, colleagues, or management.

3.15 User Subroutines

User-written subroutines are a powerful feature in MSC.Dytran that allow you to customize the program to your particular needs and provide capabilities that are not possible with the standard program.

The following user subroutines may be used:

EXALE	ALE grid point output.
EEXOUT	Element output.
EXBRK	Failure model for breakable join.
EXCOMP	Constitutive model for composites with failure.
EXELAS	Spring model.
EXEOS	Equation of state.
EXFAIL	Failure model.
EXFAIL1	General failure model for orthotropic solid elements.
EXFLOW	Flow boundary condition.
EXFLOW2	Flow boundary condition for multimaterial Euler.
EXFUNC	Time-dependent function.
EXINIT	Initial condition.
EXPBAG	Pressure in a gas bag.
EXPLD	Pressure load.
EXSPR	Spring model.
EXTLU	Declaration of FORTRAN LU numbers.
EXTVEL	Lagrangian velocity constraint.
EXVISC	Damper model.
EXYLD	Yield model.
GEXOUT	Grid point output.

The user-written subroutines have been made very simple to use. Some knowledge of programming in FORTRAN is required to write the subroutine, but the incorporation of the routines into MSC.Dytran is automatic on most types of computers. Any MSC.Dytran user with a working knowledge of FORTRAN should have no problems using this facility.

Care should be exercised when using user-written subroutines, however. It is possible to corrupt the data stored within MSC.Dytran, rendering the results meaningless. You should only use user subroutines if you are experienced in the use of MSC.Dytran and fully understand the implications of what you are doing.

3.15.1 Loading the User Subroutines with MSC.Dytran

The user-written subroutines must be in a file in the user area. The name of the file is immaterial, but it is best associated with the name of the analysis. In general, FORTRAN subroutine filenames have the extension f.

The File Management Section of the data file must contain a USERCODE statement that references the file containing the FORTRAN coding for the user-written subroutines.

For example,

```
USERCODE = user.f
```

This causes MSC.Dytran to:

1. Compile the user-written subroutines with the correct compiler options.
2. Link the user-written subroutines with MSC.Dytran.
3. Run the MSC.Dytran analysis.

On most types of computers, the procedure is automatic. See Section 3.7.1 on page 3-18 for the exact procedure.

3.15.2 User Access to Element and Grid-Point Data from User Subroutines

Within certain user-written subroutines, you have easy access to the data stored for an element or a grid point. The restriction is that the user-written subroutine must have the list of user numbers of the elements or grid points involved. In this way, you can store or retrieve data for a list of elements or grid points. You can apply calls to the subroutines included in the MSC.Dytran library delivered on the release tape.

To retrieve grid point data, the following subroutines are available:

retrieve_grid point_int_var	(for integer data)
retrieve_grid point_float_var	(for float data)

To retrieve element data, the following subroutines are available:

retrieve_element_int_va	(for integer data)
retrieve_element_float_var	(for float data)

To store grid point data, the following subroutines are available:

store_grid point_int_var	(for integer data)
store_grid point_float_var	for integer data)

To store element data, the following subroutines are available:

store_element_int_var	(for integer data)
store_element_float_var	(for float data).

An example of user access to MSC.Dytran data is given below:

```
      SUBROUTINE  EEXOUT
     +(EDTNAM,LENNAM,NZONE,CZONE,NZTYPE,LBIZON,LBXZON)
*
*     single or double defined below
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
*     dimension arguments, local arrays and data type
      DIMENSION NZONE(*),LBIZON(*),LBXZON(*)
      CHARACTER*(*) EDTNAM
      CHARACTER*(*) CZONE(*)
      CHARACTER*16  CVRNAM
*
      DIMENSION XMASS(128)
*
*     the length of the element string
      LENELM = 10
*
*     get data for the mass
*     LENELM = Length of the string of elements for data retrieval
*     NZONE  = Array holding the user numbers of the elements
*     XMASS  = Array to hold the Mass of the string of elements
*     CVRNAM = Character variable holding the variable name
*
      CVRNAM = ’Mass’
      CALL RETRIEVE_ELEMENT_FLOAT_VAR(LENELM,NZONE,XMASS,CVRNAM)
*
*     increase the mass data by one
      DO 100 N = 1,10
        XMASS(N) = XMASS(N) + 1.
 100  CONTINUE
*
*     store the new data for the mass
*     LENELM = Length of the string of elements for data retrieval
*     NZONE  = Array holding the user numbers of the elements
*     XMASS  = Array to hold the Mass of the string of elements
*     CVRNAM = Character variable holding the variable name
*
      CALL STORE_ELEMENT_FLOAT_VAR(LENELM,NZONE,XMASS,CVRNAM)
*
      RETURN
      END
```

3.15.3 User-Written Subroutine Descriptions

This section contains a description of the arguments for the user-written subroutines. The descriptions are arranged in alphabetical order.

The calling sequence provides an example of how to use the subroutine together with the argument list. Each argument is described together with its type (Real, Integer, or Character).

Some user subroutines are vectorized. As a result, the subroutines can be called more than once every time step.

EXALE

User Defined ALE Grid Point Motion

User-defined ALE grid-point motion.

Calling Sequence:

```
CALL              EXALE (CNAME, LENNAM, TIME, NCYCLE, ISTART, IEND,
                  IUSER, XPOS, YPOS, ZPOS, XVG, YVG, ZVG)
```

Input:

CNAME	Character string. Name specified on the ALEGRID entry.
LENNAM	Integer variable. Length of CNAME.
TIME	Real variable. Time at the current time step.
NCYCLE	Integer variable. Number of the current time step.
ISTART, IEND	Integer variables. Grid-point loop counters.
IUSER(*)	Integer array. Grid-point numbers.
XPOS(*), YPOS(*), ZPOS(*)	Real arrays. Grid-point coordinates in basic coordinate system.
XVG(*), YVG(*), ZVG(*)	Real arrays. Grid-point velocity components during last time step.

Output:

XVG(*), YZG(*), ZVG(*)	Real arrays. Grid-point velocity components for current time step.

(Continued)

Remarks:

1. This subroutine must be included if there are any ALEGRID entries with the TYPE set to USER.
2. The subroutine is used to calculate the grid-point motion in an ALE calculation according to a user-specified prescription.
3. The precision of the calculations should be appropriate for the computer being used. See the *MSC.Dytran Installation and Operations Guide*.
4. This routine is part of a vectorized process. As a result, the routine can be called more than once per time step.

Example:

```
      SUBROUTINE EXALE( CNAME,LENNAM,TIME,NCYCLE,ISTART,IEND,
     +                  IUSER,XPOS,YPOS,ZPOS,XVG,YVG,ZVG )
*
*     single or double defined below
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
*     declare argument as arrays and data-type here....
      CHARACTER*(*) CNAME
*
*     -----------------------------------------------------------------
*
*           cname  = name of the exale definition
*           lennam = length of the character string
*           time   = current problem time
*           ncycle = current time step number
*           istart = start of the grid point loop
*           iend   = end of the grid point loop
*           iuser  = array with grid point user numbers
*           xpos   = x-position of the list of grid points
*           ypos   = y-position of the list of grid points
*           zpos   = z-position of the list of grid points
*           xvg    = x-velocity of the list of grid points
*           yvg    = y-velocity of the list of grid points
*           zvg    = z-velocity of the list of grid points
*
*     -----------------------------------------------------------------
*
*
*     local dimensions and declarations
*
      DIMENSION IUSER(*)
      DIMENSION XPOS(*),YPOS(*),ZPOS(*)
      DIMENSION XVG(*) ,YVG(*) ,ZVG(*)
```

(Continued)

```
*
*     data statements
*
*     statement functions
*
*     executable statements
*
      FACTOR = 1.51
      X = 18.02775637
*
*     compute cosine and sine for this cycle
      RXCOS = X*COS(FACTOR*TIME)
      RXSIN = X*SIN(FACTOR*TIME)
*
*     jump to the motion prescription according to the name
      IF (CNAME(1:LENNAM) .EQ. ’EXALE1’) GOTO 1000
      IF (CNAME(1:LENNAM) .EQ. ’EXALE2’) GOTO 2000
*
 1000 CONTINUE
      DO 100 NP = ISTART,IEND
        XVG(NP) = RXCOS*RXSIN
        YVG(NP) = 0.0
        ZVG(NP) = RXSIN
 100  CONTINUE
*
      GOTO 9900
*
 2000 CONTINUE
      DO 200 NP = ISTART,IEND
        XVG(NP) = RXCOS
        YVG(NP) = 0.0
        ZVG(NP) = RXSIN*RXCOS
 200  CONTINUE
*
 9900 CONTINUE
*
*
      RETURN
      END
```

User-defined element output.

Calling Sequence:

```
CALL              EEXOUT (NAME, LENNAM, NEL, CEL, NETYPE, LIEL, LXEL)
```

Input:

NAME
Character string.
Output name specified on the ELEXOUT entry.

LENNAM
Integer variable.
Length of NAME.

NEL (*)
Integer array.
Element number.

CEL (*)
Character *8 array.
Unused.

NETYPE
Integer variable.
Type of element.

2	One-dimensional element.
3	Triangular shell.
4	Quadrilateral shell.
5	Triangular membrane.
6	Dummy triangle.
7	Dummy quadrilateral.
8	Lagrangian solid.
9	Eulerian solid (hydrodynamic).
10	Eulerian solid (with strength).
11	Eulerian solid (multimaterial).

LIEL(*)
Integer array.
Base address of element in the main integer storage array ILGDAT.

LXEL(*)
Integer array.
Base address of element in the main real storage array XLGDAT.

Remarks:

1. This subroutine must be included if there are any ELEXOUT Case Control commands.
2. The subroutine can be used to calculate results based on the data available in MSC.Dytran.
3. The precision of the calculations should be appropriate for the computer being used. See the *MSC.Dytran Installation and Operations Guide*.

(Continued)

4. This subroutine is vectorized. All the input data is stored in arrays that must be dimensioned. The start and end of the arrays are given by the variables LST and LFIN in the common block /LOCLOP/. All of the entries in the arrays between LST and LFIN must be output. See the following examples.

Example 1:

This example calculates the magnitude of the velocity in Eulerian elements and stores the result in the user variable EXUSER2.

```
      SUBROUTINE EEXOUT
      +(NAME, LENNAM, NEL, CEL, NETYPE, LIEL, LXEL)
*
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
*
      DIMENSION NEL (*), LIEL (*), LXEL (*)
      CHARACTER *(*) CEL (*), NAME
*
      COMMON/LOCLOP/LST, LFIN
      COMMON/ILGMEM/IDUM1,IDUM2,IDUM3,IDUM4,ILGDAT(1)
      COMMON/XLGMEM/XLGDAT(1)
*
      IF (NETYPE.NE.9) GOTO 9900
*
*     The magnitude of the velocity of the Eulerian elements is computed
*     and stored in the user variable EXUSER2
*
      DO 100 NZ = LST, LFIN
            XVEL = XLGDAT (LXEL (NZ)+1)
            YVEL = XLGDAT (LXEL (NZ)+2)
            ZVEL = XLGDAT (LXEL (NZ)+3)
            VEL = XVEL*XVEL + YVEL*YVEL + ZVEL*ZVEL
            XLGDAT (LXEL (NZ) + 25) = SQRT (VEL)
*
 100  CONTINUE
*
 9900 RETURN
      END
```

(Continued)

Example 2:

This example shows how the shell element sublayer data can be retrieved from memory to organize user-defined editing. The example applies to any shell element either defined by a PSHELLn or PCOMPn entry.

```
      SUBROUTINE EEXOUT
     +(EDTNAM,LENNAM,NZONE,CZONE,NZTYPE,LBIZON,LBXZON)
*
*     single or double defined below
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
      DIMENSION NZONE(*),LBIZON(*),LBXZON(*)
      CHARACTER*(*) EDTNAM
      CHARACTER*(*) CZONE(*)
*
      COMMON /LOCLOP/ LST,LFIN
      COMMON /NCYVAR/ IDUM1,NCYCLE
      COMMON /XCYVAR/ RDUM1,RDUM2 ,RDUM3,RDUM4,RDUM5,TIME
*
      CHARACTER*16 CVAR
      DIMENSION CVAR(6)
*
      DIMENSION XVAR(1024)
      DIMENSION DATA(5,1024,6)
*
      LOGICAL LHERE
*
      DATA LHERE /.TRUE./
*
*     Define the sublayer output here by variable name
*
      CVAR(1) = ’TXX’
      CVAR(2) = ’TYY’
      CVAR(3) = ’TXY’
      CVAR(4) = ’FAIL’
      CVAR(5) = ’EXY’
      CVAR(6) = ’MXTFI’
*
***************************************************************************
*     Make a loop over the sublayers, variables
*
*     The routine will retrieve the variable from the designated
*     sublayer for the entire string of elements in one call
*
*     The data array will contain all requested data after the
*     loops over the sublayers and the variables requested
***************************************************************************
```

(Continued)

```
*
*     Loop over the sublayers
*
      DO 300 ISUB = 1,5
*
*       Loop over the variables
*
        DO 200 NVAR = 1,6
*
*         Call a predefined user routine
*
          CALL GET_ELEMENT_SUBL_VARS
     +         (NZONE,XVAR,CVAR(NVAR),ISUB)
*
*         Arguments: element list, float data list, variable name list,
*                    and sublayer number
*
*         Make a loop over the elements in the edit list
*
          DO 100 NZ = LST,LFIN
*
*           Store all data for the list in the data array
*
            DATA(ISUB,NZ,NVAR) = XVAR(NZ)
*
 100      CONTINUE
 200    CONTINUE
 300  CONTINUE
*
*     If we come here for the first time write the header
*
      IF (LHERE) THEN
        OPEN(UNIT=90,FILE=’SUBLAYERS’,STATUS=’UNKNOWN’)
        WRITE(90,’(9A)’,IOSTAT=IOS)
     +  ’     Time      ’,
     +  ’   Element     ’,
     +  ’   Sublayer    ’,
     +  ’     Txx       ’,
     +  ’     Tyy       ’,
     +  ’     Txy       ’,
     +  ’     Fail      ’,
     +  ’     Exy       ’,
     +  ’    Mxtfi      ’
*
*       And a dummy line
*
```

(Continued)

```
          WRITE(90,’(A)’) ’ ’
          LHERE = .FALSE.
        ENDIF


*
*       Write it all to a file
*
        DO 400 ISUB=1,5
          DO 500 NZ=LST,LFIN
            NZON = NZONE(NZ)
            WRITE(90,’(E15.8,2I15,6E15.8)’,IOSTAT=IOS)
     +      TIME,NZON,ISUB,(DATA(ISUB,NZ,NVAR),NVAR=1,6)
 500      CONTINUE
 400    CONTINUE
*
 9900 CONTINUE
*
        RETURN
        END
```

EXBRK

User Defined Failure of Breakable Joins

The EXBRK user subroutine defines the failure criterion for a breakable join between pairs of grid points of one-dimensional and/or shell elements.

Calling Sequence:

```
CALL            EXBRK
                +  (TIME, ICYCLE, NMSETS, ILIST, IFAIL, CSETNM,
                +   FX1, FY1, FZ1, FX2, FY2, FZ2,
                +   XM1, YM1, ZM1, XM2, YM2, ZM2,
                +   FAIL1, FAIL2, FAIL3, FAIL4, FAIL5, FAIL6,
                +   ICONN1, ICONN2)
```

Input:

TIME
Real variable.
Current time in computation.

ICYCLE
Integer variable.
Current time step number.

NMSETS
Integer variable.
Number of join pairs in the user-defined string.

ILIST
Integer array.
Contains the set numbers of the join pairs in the string.

CSETNM
Character array.
Contains the name of the user-defined criterion for the join pairs.

FX1, FY1, FZ1
Real arrays.
Contain the force components of the first grid point of the join pairs.

FX2, FY2, FZ2
Real arrays.
Contain the force components of the second grid point of the join pairs.

XM1, YM1, ZM1
Real arrays.
Contain the moment components of the first grid point of the join pairs.

XM2, YM2, ZM2
Real arrays.
Contain the moment components of the second grid point of the join pairs.

ICONN1, ICONN2
Integer arrays.
Contain data on the first and second grid point of the join pair. Data concerns grid point user number, number of connected elements, and the connected element user numbers.

(Continued)

Output:

IFAIL Integer array.
Global failure flag for the join pairs.

FAIL1-FAIL6 Real arrays.
Contain the component failure flags for the join pairs. This data is used for degradation of the join. Forces and moments are multiplied by these values.

Remarks:

1. This subroutine must be included if there are any references to the EXBRK in the input data.
2. The subroutine is called every time step. The forces and moments on the pair of grid points are passed to the routine. You must return the global failure flag and the component failure switches that are used by the code upon return.
3. The precision of the calculations should be appropriate for the computer being used. See the *MSC.Dytran Installation and Operations Guide.*
4. There can be more than one failure criterion defined in the EXBRK user routine. The criteria are distinguished by their user-defined names.

Example:

This subroutine defines a failure criterion on the force x-component.

```
      SUBROUTINE EXBRK
     +     (TIME, ICYCLE, NMSETS, ILIST, IFAIL, CSETNM,
     +      FX1, FY1, FZ1, FX2, FY2, FZ2,
     +      XM1, YM1, ZM1, XM2, YM2, ZM2,
     +      FAIL1, FAIL2, FAIL3, FAIL4, FAIL5, FAIL6,
     +      ICONN1, ICONN2)
*
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
*
      DIMENSION ILIST(*), IFAIL(*)
      DIMENSION ICONN1(8,*), ICONN2(8,*)
      DIMENSION FX1(*), FY1(*), FZ1(*), FX2(*), FY2(*), FZ2(*)
      DIMENSION XM1(*), YM1(*), ZM1(*), XM2(*), YM2(*), ZM2(*)
      DIMENSION FAIL1(*), FAIL2(*), FAIL3(*)
      DIMENSION FAIL4(*), FAIL5(*), FAIL6(*)
      CHARACTER*16 CSETNM(*)
*
      FXMAX = 12000.**2
      DO 100 N=1, NMSETS
```

(Continued)

```
        IF (IFAIL(N) .EQ. 0) GOTO 100
        IF (CSETNM(N) .EQ. ’CRIT_1’) THEN
          DFX = (FX1(N) - FX2(N))**2
          IF  (DFX  .GE.  FXMAX)  THEN
            IFAIL(N) = 0
            WRITE (6,*)
     +        ’grid point pair (Point1 = ’, ICONN1(1, N),
     +        ’, Point2 = ’, ICONN2(1,N), ’)   failure.’
            WRITE (6,*)
     +        ’Time at failure: time = ’, TIME,
     +        ’Cycle at failure: cycle = ’, ICYCLE
            WRITE (6,*)
     +        ’Point1 number of connected elements: ’, ICONN1(2,N),
     +        ’element numbers’, (ICONN1(I,N), I=3,2+ICONN1(2,N))
            WRITE (6,*)
     +        ’Point2 number of connected elements: ’, ICONN2(2,N),
     +        ’element numbers ’, (ICONN2(I,N), I=3,2+ICONN2(2,N))
          ENDIF
        ENDIF
 100  CONTINUE
*
      RETURN
      END
```

Defines an orthotropic failure model for shell composites.

Calling Sequence:

```
CALL            EXCOMP (CNAME, YMX, YMY, XNUY, SXY, SYZ, SZX, FBTEN,
                FBCOM, XMXTEN, XMXCOM, SHRF, CAPA, XMAT, TIME, NSTEP,
                IPREC, LAST, NADVAR, ISUBLY, LBUSER, DLTH, SIG1, SIG2,
                SIG4, SIG5, SIG6, D1, D2, D3, D4, D5, D6, DOUT1,
                DOUT2, DOUT4, EFAIL, EFT, EFC, ESF, EMT, EMC, Q1, Q2,
                FAIL, FAIL2, USRVAR)
```

Input:

CNAME Material name (character).

YMX Young’s modulus in fiber direction.

YMY Young’s modulus in matrix direction.

XNUY Poisson’s ratio ν_{yx} .

SXY In-plane shear modulus.

SYZ Transverse shear modulus.

SZX Transverse shear modulus.

FBTEN Fiber tensile strength.

FBCOM Fiber compressive strength.

XMXTEN Matrix tensile strength.

XMXCOM Matrix compressive strength.

SHRF Shear strength.

CAPA Shear correction factor.

XMAT Not used.

TIME Current problem time.

NSTEP Step number.

IPREC Single/double precision check:
1 = library is single precision.
2 = library is double precision.

(Continued)

LAST	Length of element string.
NADVAR	Number of additional variables (see MAT8A Bulk Data).
ISUBLY	Sublayer number.
LBUSER	List of pointers to the user variables.
DLTH	Time step.
SIG1	Sigma xx in fiber system.
SIG2	Sigma yy in fiber system.
SIG4	Sigma xy in fiber system.
SIG5	Sigma yz in fiber system.
SIG6	Sigma zx in fiber system.
D1	Strain increment xx.
D2	Strain increment yy.
D3	Strain increment zz.
D4	Shear angle = 2.0 strain increment xy.
D5	Strain increment yz.
D6	Strain increment zx.
DOUT1	Total xx-strain for output
DOUT2	Total yy-strain for output
DOUT4	Total xy-strain for output
EFAIL	User fail switch.
EFT	User fail switch.
EFC	User fail switch.
ESF	User fail switch.
EMT	User fail switch.
EMC	User fail switch.
Q1	Fiber axis relative to element system.

(Continued)

Q2	Matrix axis.
USRVAR	User variable.

Output:

New Stresses

SIG1	Sigma xx in fiber system.
SIG2	Sigma yy in fiber system.
SIG4	Sigma xy in fiber system.
SIG5	Sigma yz in fiber system.
SIG6	Sigma zx in fiber system.

Fail Switches

FAIL	Overall element fail switch.
FAIL2	One-dimensional element time step suppression.

Remarks:

1. This subroutine must be included if EXCOMP is specified on a MAT8A Bulk Data entry.
2. The subroutine returns the stress tensor and failure flags.
3. Failure flags FAIL and FAIL2 are used by MSC.Dytran to zero out the hourglass forces and to enforce time step skipping for “1D elements” (if requested).
4. The total strains are supplied only if requested on the PCOMPA entry. Do not use the total strains when they are turned off.
5. Additional sublayer variables are only available when requested on the PCOMPA entry. The pointers LBUSER are set to a large value if the variables are not defined.
6. The precision of the calculations should be appropriate for the computer being used. See the *MSC.Dytran Installation and Operations Guide*.
7. If IPREC = 1, the MSC.Dytran object library is single precision; if IPREC = 2, it is double precision.
8. If your input refers to EXCOMP on a MAT8A entry and the material number (MID) is set to 99999999, the demo example will run.
9. EXCOMP does not necessarily need to define a composite material. Any material model can be programmed. EXCOMP is used for each sublayer on the PCOMPA Bulk Data entry that refers to it.

(Continued)

Example:

This subroutine swaps the stresses into the user variables:

```
      SUBROUTINE EXCOMP
     +   ( CNAME , YMX , YMY , XNUY , SXY , SYZ , SZX ,
     +     FBTEN , FBCOM , XMXTEN , XMXCOM , SHRF ,
     +     CAPA , XMAT , TIME , NSTEP , IPREC ,
     +     LAST , NADVAR , ISUBLY , LBUSER ,DLTH ,
     +     SIG1 , SIG2 , SIG4 , SIG5 , SIG6 ,
     +     D1 , D2 , D3 , D4 , D5 , D6 ,
     +     DOUT1 , DOUT2 , DOUT4 , EFAIL ,
     +     EFT , EFC , ESF , EMT,EMC, Q1, Q2 ,
     +     FAIL , FAIL2 , USRVAR )
*
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
      CHARACTER*(*) CNAME
      DIMENSION LBUSER(*)
      DIMENSION
     +     SIG1(*) , SIG2 (*), SIG4 (*),
     +     SIG5 (*), SIG6 (*),
     +     D1(*) , D2(*) , D3(*) ,
     +     D4(*) , D5(*) , D6(*) ,
     +     DOUT1(*) , DOUT2(*) , DOUT4(*) , EFAIL(*) ,
     +     EFT(*) , EFC(*) , ESF(*) ,
     +     EMT(*),EMC(*), Q1(*), Q2(*) ,
     +     FAIL(*) , FAIL2(*) , USRVAR(*)
*
*     input :
*     cname    -  material name (character)
*     ymx      -  youngs modulus in fiber dir
*     ymy      -  youngs modulus in matrix dir
*     xnuy     -  poisson ration nuyx
*     sxy      -  inplane shear modulus
*     syz      -  transverse shear modulus
*     szx      -  transverse shear modulus
*     fbten    -  fiber tensile strength
*     fbcom    -  fiber compressive strength
*     xmxten   -  matrix tensile strength
*     xmxcom   -  matrix compressive strength
*     shrf     -  shear strength
*     capa     -  shear correction factor
*     xmat     -  not used
*     time     -  current problem time
*     nstep    -  step number
*     iprec    -  singel/double precision check
*                 1 - library is single precision
*                 2 - library is double precision
```

(Continued)

```
*      last      -  length of element string
*      nadvar    -  number of additional vars
*                   ( see mat8a bulk data )
*      isubly    -  sublayer number
*      lbuser    -  list of pointer into usrvar
*      dlth      -  time step
*      sig1      -  sigma xx in fiber system
*      sig2      -  sigma yy in fiber system
*      sig4      -  sigma xy in fiber system
*      sig5      -  sigma yz in fiber system
*      sig6      -  sigma zx in fiber system
*      d1        -  strain increment xx
*      d2        -  strain increment yy
*      d3        -  strain increment zz
*      d4        -  shear angle = 2.0 x strain increment xy
*      d5        -  strain increment yz
*      d6        -  strain increment zx
*      efail     -  user fail switch
*      eft       -  user fail switch
*      efc       -  user fail switch
*      esh       -  user fail switch
*      emt       -  user fail switch
*      emc       -  user fail switch
*      q1        -  fiber axis rel to element sys
*      q2        -  matrix axis
*
*      output :
*      new stresses
*      sig1      -  sigma xx in fiber system
*      sig2      -  sigma yy in fiber system
*      sig4      -  sigma xy in fiber system
*      sig5      -  sigma yz in fiber system
*      sig6      -  sigma zx in fiber system
*      fail switches
*      fail      -  overall element fail switch
*      fail2     -  one-dimensional time step suppression
*
*      notes :
*
*      - if nadvar = 0 do not use usrvar arrays
*      - dout1-dout4 are not usable if strain output option
*        on pcompa card was set to -no-
*      - the program expects fail to be set to zero if the
*        element ( all sublayers) has failed. the time step
*        will be skipped for a failed element and all
*        forces (also hourglass) will be set to zero
*
```

(Continued)

```
*     example :
*
*     swap the sigmas into usrvar
*
*     do lv = 1,last
*       usrvar(lbuser(1) + lv ) = sig1(lv)
*       usrvar(lbuser(2) + lv ) = sig2(lv)
*     enddo
*
*
      DATA INIT/1/
*
*     note that sys_print is equivalent to a fortran print statement
*
*     this demo only works if cname = 99999999 , set on the mat8a
*     bulk data entry
      IF ( CNAME.NE.’99999999’ ) THEN
        CALL SYS_PRINT (’USER SUPPLIED EXCOMP IS MISSING ....’)
      ENDIF
*
*     checks done only at first step
*
*     make sure we have the strain output on and 6
*     defined user variables
      IF ( INIT.EQ.NSTEP ) THEN
*
*       check is for a single precision library . if double
*       precision then check against 2
        IF ( IPREC.NE.IDEFPR ) THEN
          CALL SYS_PRINT (’PRECISION IS WRONG IN EXCOMP ’)
          STOP
        ENDIF
*
        IF ( NADVAR.LT.6 ) THEN
          CALL SYS_PRINT(’FOR THIS EXCOMP DEMO TO RUN YOU MUST DEFINE ’)
          CALL SYS_PRINT(’AT LEAST 6 SUBLAYER USER VARIABELS ON THE ’)
          CALL SYS_PRINT(’MAT8A BULK DATA ENTRY FOR MATERIAL ’)
          CALL SYS_PRINT( CNAME )
          STOP
        ENDIF
*
        CALL SYS_PRINT(’YOU ARE USING THE DEMO EXCOMP ’)
        CALL SYS_PRINT(’RATHER THAN YOUR OWN VERSION’)
        CALL SYS_PRINT(’RELINK MSC.Dytran WITH YOUR EXCOMP CODING ’)
*
        DO 100 NV = 1,3
          DO 110 LV = 1,LAST
            USRVAR(LBUSER(NV) + LV) = -1.E20
```

(Continued)

```
 110      CONTINUE
 100    CONTINUE
*
        DO 200 NV = 4,6
          DO 210 LV = 1,LAST
            USRVAR(LBUSER(NV) + LV) = 1.E20
 210      CONTINUE
 200    CONTINUE
*
*       see if strain output is on
        IF ( DOUT1(1) .EQ. 123456789. ) THEN
          CALL SYS_PRINT('FOR THIS EXCOMP DEMO TO RUN YOU MUST DEFINE ')
          CALL SYS_PRINT('STRAIN OUTPUT OPTION ON ON THE ')
          CALL SYS_PRINT('PCOMPA BULK DATA ENTRY WHICH HOLDS MATERIAL ')
          CALL SYS_PRINT (CNAME)
          STOP
        ENDIF
      ENDIF
*
      XNUX = XNUY * YMX/YMY
      PXY = 1./(1. - XNUY*XNUX)
      C11 = YMX*PXY
      C12 = PXY*XNUX*YMY
      C22 = PXY*YMY
      C44 = SXY
      DO 1000 LV = 1,LAST
        SIG1(LV) = SIG1(LV) + C11*D1(LV) + C12*D2(LV)
        SIG2(LV) = SIG2(LV) + C12*D1(LV) + C22*D2(LV)
        SIG4(LV) = SIG4(LV) + C44*D4(LV)
        SIG5(LV) = SIG5(LV) + SYZ*CAPA*D5(LV)
        SIG6(LV) = SIG6(LV) + SZX*CAPA*D6(LV)
 1000 CONTINUE
*
*     save max in user vars as an example
      DO 2000 LV = 1,LAST
*
*       save maximum stress
        USRVAR(LBUSER(1) + LV) = MAX( USRVAR(LBUSER(1)+LV),SIG1(LV) )
        USRVAR(LBUSER(2) + LV) = MAX( USRVAR(LBUSER(2)+LV),SIG2(LV) )
        USRVAR(LBUSER(3) + LV) = MAX( USRVAR(LBUSER(3)+LV),SIG4(LV) )
*
*       save minimum strain
        USRVAR(LBUSER(4) + LV) = MIN( USRVAR(LBUSER(4)+LV),DOUT1(LV) )
        USRVAR(LBUSER(5) + LV) = MIN( USRVAR(LBUSER(5)+LV),DOUT2(LV) )
        USRVAR(LBUSER(6) + LV) = MIN( USRVAR(LBUSER(6)+LV),DOUT4(LV) )
 2000 CONTINUE
*
 5000 CONTINUE
*
      RETURN
      END
```

EXELAS

User-Defined CELAS1 Spring Element

Returns the force and stiffness in CELAS1 spring elements.

Calling Sequence:

```
CALL                EXELAS (N, M, IX, IC, PROP, HISV, FORCEO, C, DI, V, A,
                    UREL, DUREL, VREL, XMASS, FORCE, STIFF)
```

Input:

N
Integer variable.
Element number.

M
Integer variable.
Property number.

IX(2)
Integer array.
Connectivity:
IX(1) = grid point at end 1.
IX(2) = grid point at end 2.

IC(2)
Integer array.
Component:
IC(1) = component at end 1 (between 1 and 6).
IC(2) = component at end 2 (between 1 and 6).

PROP(7)
Real array.
Properties as input on the PELASEX entry.

HISV(6)
Real array.
History variables for the element. This array can be used by the user to store variables from one time step to the next.

FORCEO
Real variable.
Force in the element at the previous time step.

C(3,2)
Real array.
Deformed coordinates in the basic coordinate system:
C(1:3,1) x, y, z, coordinates at end 1.
C(1:3,2) x, y, z, coordinates at end 2.

(Continued)

DI (6,2)
Real array
Incremental displacements in the basic coordinate system:
DI(1:3,1) = x, y, z, translational displacements of end 1.
DI(4:6,1) = x, y, z, rotational displacements of end 1.
DI(1:3,2) = x, y, z, translational displacements of end 2.
DI(4:6,2) = x, y, z, rotational displacements 2.

These are incremental displacements, i.e., the displacements for this time step only.

V(6,2)
Real array.
Velocities in the basic coordinate system:
V(1:3,1) = x, y, z, translational velocities of end 1.
V(4:6,1) = x, y, z, rotational velocities of end 1.
V(1:3,2) = x, y, z, translational velocities of end 2.
V(4:6,2) = x, y, z, rotational velocities of end 2.

A(6,2)
Real array.
Accelerations in the basic coordinate system:
A(1:3,1) = x, y, z translational accelerations of end 1.
A(4:6,1) = x, y, z rotational accelerations of end 1.
A(1:3,2) = x, y, z, translational accelerations of end 2.
A(4:6,2) = x, y, z, rotational accelerations of end 2.

UREL
Real variable.
Relative displacement of the element; i.e., the displacement of end 2 in the spring direction minus the displacement of end 1.

DUREL
Real variable.
Relative incremental displacement of the element VREL real variable.

VREL
Real variable.
Relative velocity of the end points of the element in the direction of the element.

XMASS(2)
Real array.
Mass of the grid points at ends 1 and 2.

Output:

FORCE
Real variable.
Force in the element.

STIFF
Real variable.
Current stiffness of the element.

(Continued)

Remarks:

1. This subroutine must be included if the PELASEX entry is specified in the Bulk Data Section.
2. The velocities (V) and accelerations (A) of the end points can be updated by the user subroutine when required.
3. The stiffness is used by MSC.Dytran to estimate the time step. A nonzero value must be returned.
4. The precision of the calculations should be appropriate for the computer being used. See the *MSC.Dytran Installation and Operations Guide*.

Example:

This example defines a stiffness and a corresponding force for a spring element.

```
      SUBROUTINE EXELAS
     +(N,M,IX,IC,PROP,HISV,FORCEO,C,DI,V,A,UREL,DUREL,
      +VREL,XMASS,FORCE,STIFF)
*
*     single or double defined below
      IMPLICIT DOUBLE PRECSION (A-H,O-Z)
*
*     declare argument as arrays and data-type here....
      DIMENSION IX(2),IC(2),PROP(7),HISV(6),C(3,2),
     +            DI(6,2),V(6,2),A(6,2),XMASS(2)
*
*     define the stiffness and the force on the spring
      STIFF = 1.E3
      FORCE = STIFF * DUREL
*
      RETURN
      END
```

EXEOS

User-Specified Equation of State

Calculates the equation of state for Lagrangian and Eulerian solid elements.

Calling Sequence:

```
CALL            EXEOS (CNAME, LENNAM, ISTART, IEND, RHO, DV, DEVIS,
                XMASS, FBURN, POLD, SIEOLD, PNEW, SIENEW, CLNEW,
                GRUNGM)
```

Input:

CNAME
Character string.
Name specified on the EOSEX entry.

LENNAM
Integer variable.
Length of CNAME.

ISTART, IEND
Integer variables.
Grid-point loop counters.

RHO(*)
Real array.
Density of element.

DV(*)
Real array.
Change in volume of element.

DEVIS(*)
Real array.
Viscous work term of element.

XMASS(*)
Real array.
Mass of element.

FBURN(*)
Real array.
Burn fraction of element (not for Lagrangian elements).

POLD(*)
Real array.
Old pressure of element.

SIEOLD(*)
Real array.
Old specific internal energy of element.

(Continued)

Output:

PNEW(*)
Real array.
New pressure of element.

SIENEW(*)
Real array.
New specific internal energy of element.

CLNEW(*)
Real array.
New soundspeed of element.

GRUNGM(*)
Real array.
Gruneisen gamma of element (only for multi-material elements).

Remarks:

1. This subroutine must be included if there are any EOSEX entries.
2. The subroutine is used to calculate the pressure PNEW for Lagrangian and/or Eulerian elements according to a user-specified prescription. In most cases the equation of state is energy dependent, the new pressure must be solved therefore simultaneously with the energy equation:

$$\mathrm{SIENEW} = \mathrm{SIEOLD} + \mathrm{DEVIS} - \frac{1}{2}\left((\mathrm{POLD} + \mathrm{PNEW})\frac{\mathrm{DV}}{\mathrm{XMASS}}\right)$$

$$\mathrm{PNEW} = f(\mathrm{RHO}, \mathrm{SIENEW})$$

It may be necessary to solve these equations by iteration.

The soundspeed CLNEW should be computed as:

$$\mathrm{CLNEW}^2 = \left.\frac{d\Sigma_l}{d\rho}\right|_S = \frac{\frac{4}{3}G + \left.\frac{dp}{d\eta}\right|_S}{\rho_0}$$

where S denotes that the adiabatic derivative is required, Σ_l is the longitudinal stress, ρ is the density, p is the pressure, G is the shear modulus, ρ_0 is the reference density and $\eta = \rho/\rho_0$. Care must be used in defining this quantity, because an error can cause an instability in the calculation.

The Gruneisen gamma GRUNGM must be computed for multi-material elements and can be calculated as follows:

$$p = A \cdot f\left(\frac{\rho}{\rho_0}\right) + B \cdot f(\rho, e)$$

where B is the Gruneisen gamma.

(Continued)

3. The precision of the calculations should be appropriate for the computer being used. See the *MSC.Dytran Installation and Execution Guide*.

4. This routine is part of a vectorized process. As a result, the routine can be called more than once per time step.

Example:

This example calculates the equation of states of an explosive in water using multi-material Eulerian elements.

```
      SUBROUTINE EXEOS(CNAME,LENNAM,ISTART,IEND,RHO,DV,DEVIS,XMASS,
     +                  FBURN,POLD,SIEOLD,PNEW,SIENEW,CLNEW,GRUNGM)

*
*     single or double defined below
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
*     declare argument as arrays and data-type here....
      CHARACTER*(*) CNAME
*
*     -----------------------------------------------------------------
*
*     input:
*           cname  = name of the exale definition
*           lennam = length of the character string
*           istart = start of the grid point loop
*           iend   = end of the grid point loop
*           rho    = density
*           dv     = change in volume
*           devis  = viscous work term
*           xmass  = mass
*           fburn  = burn fraction (not used for Lagrangian elements)
*           pold   = old pressure
*           sieold = old specific internal energy
*
*    output:
*           pnew   = new pressure
*           sienew = new specific internal energy
*           clnew  = new soundspeed
*           grungm = Gruneisen gamma
*
*     -----------------------------------------------------------------
*
*     local dimensions and declarations
*
```

(Continued)

```
      DIMENSION RHO(*),DV(*),DEVIS(*)
      DIMENSION XMASS(*),FBURN(*),POLD(*),SIEOLD(*)
      DIMENSION PNEW(*),SIENEW(*),CLNEW(*),GRUNGM(*)
*
      ZERO = 0.0
      ONE  = 1.0
      HALF = 0.5
*
*     first water
      IF (CNAME(1:LENNAM) .NE. ’WATER’) GOTO 1000
*
*     variables
      E = 2.2E9
      RHOREF = 1000.
      HVLM   = 1.0/1.1 - 1.0
      SSPD = SQRT(E/RHOREF)
*
*     the simple bulk equation of state
      DO 100 NZ = ISTART, IEND
        TMU = (RHO(NZ)-RHOREF)/RHOREF
        AMU = MAX(TMU,HVLM)
        IF (XMASS(NZ) .LE. 0.0) GOTO 100
*
*       calculation of new pressure, specific internal energy
*       and soundspeed
        PNEW(NZ)   = E*AMU
        SIENEW(NZ) = SIEOLD(NZ) + DEVIS(NZ) -
     +               HALF*(POLD(NZ)+PNEW(NZ))*DV(NZ)/XMASS(NZ)
        CLNEW(NZ)  = SSPD
*
*       gruneisen gamma can be neglected
        GRUNGM(NZ) = ZERO
 100  CONTINUE
*
 1000 CONTINUE
*
*     next the explosive
      IF (CNAME(1:LENNAM) .NE. ’JWL’) GOTO 9900
*
*     variables
      A = 6.17E11
      B = 1.69E10
      R1 = 4.4
      R2 = 1.2
      OMEGA = .25
      RHOREF = 1770.
      OR1    = ONE / R1
      OR2    = ONE / R2
```

(Continued)

```
      BOR1   = A*OMEGA*OR1
      BOR2   = B*OMEGA*OR2
      DMIN6  = 1.E-6
      SMALL  = 1.E-20
*
*     the jwl equation of state
      DO 200 NZ = ISTART,IEND
        IF (XMASS(NZ) .LE. ZERO) GOTO 200
        ETA = RHO(NZ)/RHOREF
        TERM1 = -R1 / ETA
        TERM2 = -R2 / ETA
        EXP1  = EXP(TERM1)
        EXP2  = EXP(TERM2)
        TERM1 = A*(ONE-OMEGA*ETA*OR1)
        TERM2 = B*(ONE-OMEGA*ETA*OR2)
        TAA   = TERM1*EXP1+TERM2*EXP2
        AA    = TAA*FBURN(NZ)
        TCC   = EXP1*(R1*TERM1/ETA**2-BOR1)+
     +          EXP2*(R2*TERM2/ETA**2-BOR2)
        CC    = TCC*FBURN(NZ)
        BB    = OMEGA*ETA*FBURN(NZ)
        DD    = OMEGA*FBURN(NZ)
        DVOVH = DV(NZ)*RHO(NZ)/XMASS(NZ)
        TERM3 = MAX(SMALL,ONE+HALF*BB*DVOVH/ETA)
        TERM4 = SIEOLD(NZ) + DEVIS(NZ) -
     +          HALF*(POLD(NZ)+AA)*DVOVH/RHO(NZ)
*
*       calculation of new pressure, specific internal energy
*       and speed of sound
        SIENEW(NZ) = TERM4/TERM3
        PNEW(NZ)   = AA + RHOREF*MAX(ZERO,SIENEW(NZ)*BB)
        SSPD       = CC/RHOREF + SIENEW(NZ)*DD +
     +               PNEW(NZ)*BB*RHOREF/RHO(NZ)/RHO(NZ)
        SSPD       = MAX(DMIN6,SSPD)
        CLNEW(NZ) = SQRT(SSPD)
*
*       in this case the gruneisen gamma should be calculated
        GRUNGM(NZ) = BB*RHOREF/RHO(NZ)
 200  CONTINUE
*
 9900 CONTINUE
*
      RETURN
      END
```

EXFAIL

User-Defined Failure Model

Returns a failure flag FFAIL to MSC.Dytran for all elements in the string (LST....LFIN).

Calling Sequence:

```
CALL            EXFAIL (MATNAM, LENNAM, EKPLAS, EFFSTS, SPRES, EDIS,
                RHO, FFAIL)
```

Input:

MATNAM
Character string.
Name of the material.

LENNAM
Integer variable.
Length of MATNAM.

EPLAS(*)
Real array.
Plastic strain of an element.

EFFSTS(*)
Real array.
Effective stress of an element.

PRES(*)
Real array.
Pressure of an element.

EDIS(*)
Real array.
Distortional energy of an element.

RHO(*)
Real array.
Density of an element.

Output:

FFAIL (*)
Real array.
Failure flag of an element:
FFAIL = 0   Element failed.
FFAIL = 1   Element not failed.

Remarks:

1. The subroutine must be included if there are any FAILEX entries in the input.
2. The pressure array is only used for Eulerian material with strength.
3. The precision of the calculations should be appropriate for the computer being used. See the *MSC.Dytran Installation and Operations Guide*.

(Continued)

Example:

In this example, the material will fail when the maximum plastic strain exceeds 50%. The routine returns the FFAIL flag to the code (FFAIL = 1 no failure and FFAIL = 0 failure).

```
      SUBROUTINE EXFAIL
     +      (MATNAM, LENNAM, EPLAS, EFFSTS, PRES, SIE, RHO, FFAIL)
C
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
C
      DIMENSION EPLAS (*), EFFSTS (*), PRES (*), SIE (*), RHO (*), FFAIL (*)
C
      CHARACTER*80 MATNAM
C
      COMMON /LOCLOP/ LST,LFIN
C
C     Example of failure if the maximum plastic strain exceeds 50%
C
      DO 100 NZ = LST, LFIN
            IF (EPLAS (NZ) . GT. 0.5) THEN
                  FFAIL (NZ) = 0.
            ELSE
                  FFAIL (NZ) = 1.
            ENDIF
100   CONTINUE
*
      RETURN
      END
```

EXFAIL1

User-Defined Orthotropic Failure Model

The EXFAIL1 user routine defines a general failure model for orthotropic three-dimensional elements.

Calling Sequence:

```
CALL             EXFAIL1
                 +  (MATNAM, LENNAM, IZONE,
                 +   TXX, TYY, TZZ, TXY, TXZ, TYZ,
                 +   DEPSXX, DEPSYY, DEPSZZ, DEPSXY, DEPSXZ, DEPSYZ,
                 +   EPSXX, EPSYY, EPSZZ, EPSXY, EPSXZ, EPSYZ,
                 +   EXX, EYY, EZZ, EXY, EXZ, EYZ,
                 +   GXY, GYZ, GZX,
                 +   USRVR1, USRVR2,
                 +   TSTEP, FFAIL)
```

Input:

MATNAM
Character string.
Name of the material.

LENNAM
Integer variable.
Length of MATNAM.

IZONE
Integer array.
Element user number.

TXX, TYY, TZZ
Real arrays.
Normal stress components.

TXY, TXZ, TYZ
Real arrays.
Shear stress components.

DEPSXX, DEPSYY, DEPSZZ
Real arrays.
Normal strain increments.

DEPSXY, DEPSXZ, DEPSYZ
Real arrays.
Shear strain increments.

EPSXX, EPSYY, EPSZZ
Real arrays.
Normal (last cycle) strains.

EPSXY, EPSXZ, EPSYZ
Real arrays.
Shear (last cycle) strains.

(Continued)

XX, EYY, EZZ,
EXY, EXZ, EYZ,
GXY, GYZ, GZX — Real arrays.
Elasticity matrix components.

USRVR1, USRVR2 — Real arrays.
User variables.

Output:

TSTEP — Real array.
Element time step.

FFAIL — Real array.
Element failure flag.
FFAIL = 0 Element failed.
FFAIL= 1 Element not failed.

Remarks:

1. The subroutine must be included if there are any FAILEX1 entries in the input data.
2. The FAILEX1 entry can only be used in combination with orthotropic solid (Lagrangian) elements.
3. The access to the element's elasticity matrix allows for inclusion of degradation of the material on an element basis. Any changes made to the elasticity matrix components of an element is stored in element memory and is used in the next time step in the evaluation of the new stress state. When the properties of the elasticity matrix depend on the strain, a full constitutive model is defined by the user.
4. The strains are the last time step strain. To get the current strain, the increments must be added. The increments are used to detect the direction of loading (i.e., loading or unloading).
5. Changes made to the strain tensor are not stored.
6. The stress tensor is always represented in the material coordinate system, which is based on element topology for the materials that refer to FAILEX1.
7. Any changes made to the stress tensor components are stored in element memory. Note that this can result in an inconsistent relation of stress state and strain field.
8. The user variables are used to store element data that is not standard part of MSC.Dytran storage. When these variables are used by other user subroutines, this may cause definition conflicts. The content of the user variables is stored at return from the EXFAIL1 user subroutine. Additional user variables can be defined by the parameter PARAM,VARACTIV.
9. The precision of the calculations should be appropriate for the computer being used. See the *MSC.Dytran Installation and Operations Guide*.

(Continued)

Example:

In the example that follows, a failure model is defined that is based on maximum strain, depending on the direction of the strain. It includes degradation of the material before failure.

```
      SUBROUTINE EXFAIL1
     +  (MATNAM, LENNAM, IZONE,
     +   TXX, TYY, TZZ, TXY, TXZ, TYZ,
     +   DEPSXX, DEPSYY, DEPSZZ, DEPSXY, DEPSXZ, DEPSYZ,
     +   EPSXX, EPSYY, EPSZZ, EPSXY, EPSXZ, EPSYZ,
     +   EXX, EYY, EZZ, EXY, EYZ, EXZ, GXY, GYZ, GZX,
     +   USRVR1, USRVR2,
     +   TSTEP, FFAIL)

*
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
      SAVE IERM
*
*     stress tensor
      DIMENSION TXX(*), TYY(*), TZZ(*), TXY(*), TXZ(*), TYZ(*)
*
*     strain increments
      DIMENSION DEPSXX(*), DEPSYY(*), DEPSZZ(*)
      DIMENSION DEPSXY(*), DEPSXZ(*), DEPSYZ(*)
*
*     last time step total strain tensor
      DIMENSION  EPSXX(*), EPSYY(*), EPSZZ(*)
      DIMENSION  EPSXY(*), EPSXZ(*), EPSYZ(*)
*
*     elasticity matrix
      DIMENSION  EXX(*), EYY(*), EZZ(*)
      DIMENSION  EXY(*), EYZ(*), EXZ(*)
      DIMENSION  GXY(*), GYZ(*), GZX(*)
*
*     element user numbers
      DIMENSION IZONE(*)
*
*     current time step and element failure flag
      DIMENSION  TSTEP(*), FFAIL(*)
*
*     user variables
      DIMENSION USRVR1(*), USRVR2(*)
*
      COMMON /LOCLOP/ LST,LFIN

*
```

(Continued)

```
      CHARACTER*80 MATNAM
      CHARACTER*80 CFLRNM
      LOGICAL LFIRST
*
*     set some constants
      ZERO = 0.
      ONE  = 1.
*
*     set the failure name for  groups
      IF (MATNAM(1:LENNAM) .EQ. ’100’)  THEN
        CFLRNM = ’COMPOSITE’
      ENDIF
*
*     start by checking on the material name....
*
*     is it the composite...
      IF (CFLRNM .EQ. ’COMPOSITE’) THEN
*
*       set some material parameters
        XYMIN = 0.05
        XYMAX = 0.08
*
*       loop over the elements in the list....
*
        DO 100 NZ=LST,LFIN
*
*         start by getting the user number....
          NZONEU = IZONE(NZ)
*
*         Assume xx-fiber direction
*
*         variables....strains are n-1 cycle strains
          EPSXX(NZ) = EPSXX(NZ) + DEPSXX(NZ)
          EPSYY(NZ) = EPSYY(NZ) + DEPSYY(NZ)
          EPSZZ(NZ) = EPSZZ(NZ) + DEPSZZ(NZ)
          EPSXY(NZ) = EPSXY(NZ) + DEPSXY(NZ)
          EPSXZ(NZ) = EPSXZ(NZ) + DEPSXZ(NZ)
          EPSYZ(NZ) = EPSYZ(NZ) + DEPSYZ(NZ)
*
*         Strain to Failure
*
*         tensile & compressive 1% (Ultimate Failure Strain)
*         .... fiber direction ....
          IF (ABS(EPSXX(NZ)) .GT. 0.01 ) FFAIL(NZ) = ZERO
*
*         inplane shear (epsxy) 5%
          IF (EPSXY(NZ) .GT. 0.05)
```

(Continued)

```
*
*           compute damage
            XDMGE1 =
     +      MIN(MAX((EPSXY(NZ)-XYMIN)/(XYMAX-XYMIN),ZERO),ONE)
*
*           degradation
            FACTOR  = (ONE-XDMGE1)/(ONE-USRVR1(NZ))
            EXY(NZ) = EXY(NZ) * FACTOR
            EYY(NZ) = EYY(NZ) * FACTOR
            EYZ(NZ) = EYZ(NZ) * FACTOR
            GXY(NZ) = GXY(NZ) * FACTOR
*
*           store in user variable 1
            USRVR1(NZ) = MAX(XDMGE1,USRVR1(NZ))
*
          ENDIF
*
*         if fully damaged --> failure
          IF (USRVR1(NZ) .GE. 0.836) THEN
            FFAIL(NZ) = ZERO
            USRVR1(NZ) = 0.836
          ENDIF
*
*         next element
 100    CONTINUE
      ENDIF
*
      RETURN
      END
```

Returns the velocity, pressure, density, and specific internal energy at an Eulerian user-defined flow boundary.

Calling Sequence:

```
CALL            EXFLOW (FLNAME, LENNAM, NELEM, PELEM, QELEM, UXELEM,
                UYELEM, UZELEM, RHOEL, SIEEL, PFACE, UXFACE, UYFACE,
                UZFACE, RHOFAC, SIEFAC)
```

Input:

FLNAME
Character string.
Name of the boundary.

LENNAM
Integer variable.
Length of FLNAME.

NELEM(*)
Integer array.
Element number.

PELEM(*)
Real array.
Pressure in the element.

QELEM(*)
Real array.
Artificial quadratic viscosity of element.

UXELEM(*)
Real array.
x-velocity of element.

UYELEM(*)
Real array.
y-velocity of element.

UZELEM(*)
Real array.
z-velocity of element.

RHOEL(*)
Real array.
Density of element.

SIEEL(*)
Real array.
Specific internal energy of element.

Output:

PFACE(*)
Real array.
Pressure at boundary.

UXFACE(*)
Real array.
x-velocity at boundary.

(Continued)

UYFACE(*) Real array.
y-velocity at boundary.

UZFACE(*) Real array.
z-velocity at boundary.

RHOFAC(*) Real array.
Density of inflowing material.

SIEFAC(*) Real array.
Specific internal energy of inflowing material.

Remarks:

1. This subroutine must be included if there are any FLOWEX entries in the input file.
2. The pressure and velocity at the boundary must be specified. If there is flow into the mesh, the density and specific internal energy must also be defined.
3. This subroutine is called twice every time step for every Euler face referenced on the FLOWEX entry. The first call is for the material transport calculation, the second is for the impulse calculation.
4. This subroutine is vectorized. All the input data is stored in arrays, which must be dimensioned. The start and end of the arrays are given by the variables LST and LFIN in the common block /LOCLOP/. Calculations must be done for all of the entries in the arrays between LST and LFIN. See the following example.
5. The precision of the calculations should be appropriate for the computer being used. See the *MSC.Dytran Installation and Operations Guide.*

Example:

This example simulates a nonreflecting boundary by defining the velocity and pressure at the boundary to be the same as that in the element.

```
      SUBROUTINE EXFLOW (FLNAME, LENNAM, NELEM, PELEM, QELEM,
     +      UXELEM, UYELEM, UZELEM, RHOEL, SIEEL, PFACE, UXFACE,
     +      UYFACE, UZFACE, RHOFAC, SIEFAC)
      IMPLICIT DOUBLE PRECISION (A-H, 0-Z)
      DIMENSIONNELEM(*), PELEM(*), QELEM(*), UXELEM(*),
     +            UYELEM(*), UZELEM(*), RHOEL(*), SIEEL(*)
      DIMENSIONPFACE(*), UXFACE(*), UYFACE(*), UZFACE(*),
     +            RHOFAC(*), SIEFAC(*)
      CHARACTER*(*) FLNAME
      COMMON /LOCLOP/LST, LFIN
C
C     Do the vector loop from the LST to LFIN
      DO 100 I = LST, LFIN
```

(Continued)

```
            PFACE (I) = PELEM (I)
            UXFACE(I) = UXELEM (I)
            UYFACE (I) = UYELEM(I)
            UZFACE (I) = UZELEM (I)
            RHOFAC(I)= RHOEL (I)
            SIEFAC(I) = SIEEL (I)
100   CONTINUE
      RETURN
      END
```

EXFLOW2

User-Defined Flow Boundary

Returns the velocity, pressure, density, and specific internal energy at an Eulerian user-defined flow boundary.

Calling Sequence:

```
CALL            EXFLOW2 (FLNAME, LENNAM, TIME, NCYCLE, NELEM, PELEM,
                QELEM, UXELEM, UYELEM, UZELEM,RHOEL, SIEEL, PFACE,
                UXFACE, UYFACE, UZFACE, RHOFAC, SIEFAC, SX, SY, SZ,
                CMATNO, IFLWTP)
```

Input:

FLNAME — Character string.
Name of the boundary.

LENNAM — Integer variable.
Length of FLNAME.

NELEM(*) — Integer array.
Element number.

PELEM(*) — Real array.
Pressure in the element.

QELEM(*) — Real array.
Artificial quadratic viscosity of element.

UXELEM(*) — Real array.
x-velocity of element.

UYELEM(*) — Real array.
y-velocity of element.

UZELEM(*) — Real array.
z-velocity of element.

RHOEL(*) — Real array.
Density of element.

SIEEL(*) — Real array.
Specific internal energy of element.

SX(*) — Real array.
Face area x-component.

(Continued)

SY(*)	Real array. Face area y-component.
SZ(*)	Real array. Face area z-component.

Output:

PFACE(*)	Real array. Pressure at boundary.
UXFACE(*)	Real array. x-velocity at boundary.
UYFACE(*)	Real array. y-velocity at boundary.
UZFACE(*)	Real array. z-velocity at boundary.
RHOFAC(*)	Real array. Density of inflowing material.
SIEFAC(*)	Real array. Specific internal energy of inflowing material.
CMATNO	Character array. Material name of material for in- or outflow at the faces in the list.
IFLWTP	Integer variable. Flow type switch: 0 in/outflow. 1 outflow. 2 inflow.

Remarks:

1. This subroutine is valid for multimaterial Euler only. For hydrodynamic single material, or single material with strength, use EXFLOW.
2. This subroutine must be included if there are any FLOWEX entries in the input file and the Euler processor used is the multimaterial Euler processor.
3. The pressure and velocity at the boundary must be specified. If there is flow into the mesh, the density and specific internal energy must also be defined.
4. This subroutine is called twice every time step for every Euler face referenced on the FLOWEX entry. The first call is for the material transport calculation, the second is for the impulse calculation.

(Continued)

5. This subroutine is vectorized. All the input data is stored in arrays that must be dimensioned. The start and end of the arrays are given by the variables LST and LFIN in the common block /LOCLOP/. Calculations must be done for all of the entries in the arrays between LST and LFIN. See the following example.

6. The precision of the calculations should be appropriate for the computer being used. See the *MSC.Dytran Installation and Operations Guide.*

Example:

```
     SUBROUTINE EXFLOW2
     + (FLNAME, LENNAM, TIME, NCYCLE, IUSRZN,
     +  PZON, QZON, UXZON, UYZON, UZZON, RHOZON, SIEZON,
     +  PFAC, UXFAC, UYFAC, UZFAC, RHOFAC, SIEFAC,
     +  SX, SY, SZ, CMATNO, IFLWTP )
*
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
      DIMENSION PZON(*),QZON(*),UXZON(*),UYZON(*),UZZON(*),
     +          RHOZON(*),SIEZON(*)
      DIMENSION PFAC(*),UXFAC(*),UYFAC(*),UZFAC(*),RHOFAC(*),SIEFAC(*)
      DIMENSION SX(*),SY(*),SZ(*)
      DIMENSION IUSRZN(*)
*
      CHARACTER*80 FLNAME
      CHARACTER*8  CMATNO(*)
*
      COMMON/LOCLOP/LST,LFIN
*
      CHARACTER*80 FLNAME
*
      DATA SMALL /1.E-15/
      DATA ZERO /0./
      DATA ONE  /1./
*
*
*     mass flow
      DATA DMASS   /10./
*
      DO 100 NF = LST,LFIN
*
        FACX = ONE
        FACY = ONE
        FACZ = ONE
        IF (ABS(SX(NF)).LE.SMALL) FACX = ZERO
        IF (ABS(SY(NF)).LE.SMALL) FACY = ZERO
        IF (ABS(SZ(NF)).LE.SMALL) FACZ = ZERO
```

(Continued)

```
*
*       Material at Inflow
        CMATNO (NF) = '100'
*
*       Density at Inflow
        RHOFAC (NF) = 1000.
*
*       Internal Energy at Inflow
        SIEFAC (NF) = 2.E5
*
*       Pressure on the face is the element pressure
        PFAC   (NF) = PZON(NF)
*
*       normal on the face points outward
*           .... transport of material ....
*
        UXFAC  (NF) =
     +         -FACX * DMDT / ( RHOFAC(NF) * SX(NF) )
        UYFAC  (NF) =
     +         -FACY * DMDT / ( RHOFAC(NF) * SY(NF) )
        UZFAC  (NF) =
     +         -FACZ * DMDT / ( RHOFAC(NF) * SZ(NF) )
*
  100 CONTINUE
*
      RETURN
      END
EXFUNC
```

EXFUNC

User-Defined Function

Defines the functions to create time dependency in dynamic excitation.

Calling Sequence:

```
CALL                 EXFUNC (CFNAME, XVAL, YVAL, NMSTR)
```

Input:

CFNAME
Character variable.
The name of the function defined on input.

XVAL (NMSTR)
Real array.
The x-value that the function requires (time).

NMSTR
Number of values.

Output:

YVAL (NMSTR)
Real array.
The value to be returned by the function.
Note that the y-value is multiplied by the scale factor defined on the load entry.

Remarks:

1. This subroutine must be included if there are any TABLEEX entries in the input.
2. The subroutine is called every time step. The time is passed to the subroutine. The outcome (y-value) is returned.
3. The precision of the calculations should be appropriate for the computer being used. See the *MSC.Dytran Installation and Operations Guide*.
4. There can be more than one function defined in the EXFUNC user routine; these functions can be distinguished by their names.

Example:

This subroutine defines six different functions that can be referred to from the input by a TABLEEX entry. These functions can be used for a variety of dynamic loads.

```
     SUBROUTINE EXFUNC
     + (FNAME, XVAL, YVAL, NMSTR)
*
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
      DIMENSION XVAL(*),YVAL(*)
*
      CHARACTER*16 CFNAME
*
```

(Continued)

```
      IF (CFNAME.EQ.’FUNC1’) THEN
        YVAL = LOG (ABS (XVAL) )
      ELSE IF (CFNAME. EQ.’FUNC2’) THEN
        YVAL = SIN (XVAL*6.28)*COS (XVAL*3.14)
      ELSE IF = (CFNAME.EQ.’FUNC3’) THEN
        YVAL = XVAL*XVAL*XVAL+2*XVAL
      ELSE IF (CFNAME.EQ.’FUNC4’) THEN
        YVAL = XVAL
      ELSE IF (CFNAME.EQ.’FUNC5’) THEN
        YVAL = ABS (XVAL)
      ELSE IF (CFNAME.EQ.’FUNC6’) THEN
        YVAL = EXP (XVAL)*LOG10 (ABS (XVAL-1.) )
      ELSE
        CONTINUE
      ENDIF
*
      RETURN
      END
```

EXINIT

User-Defined Initial Condition

Defines an initial condition for elements and/or grid points at the beginning of the analysis.

Calling Sequence:

```
CALL                EXINIT (CNAME, LENNAM, TIME, NCYCLE, NGPEL, NUMENT,
                    + ISTART, IEND)
```

Input:

CNAME
Character variable.
Name specified on the TICEEX or TICGEX entry.

LENNAM
Integer variable.
Number of characters in CNAME.

TIME
Real variable.
Time at the current time step.

NCYCLE
Integer variable.
Number of the current time step.

NGPEL(*)
Integer array.
Element or grid point user number.

NUMENT
Integer variable.
Length of array and number of elements or grid points defined on the TICEEX or TICGEX entry.

ISTART, IEND
Integer variables.
Element loop counters.

Remarks:

1. This subroutine must be included if there are any TICEEX or TICGEX entries.
2. This subroutine is used to initialize the variables of elements and/or grid points.

Example:

This example shows how to initialize a gravitational field in water.

```
      SUBROUTINE EXINIT
     +(CNAME, LENNAM, TIME, NCYCLE, NGPEL, NUMENT, LST, LFIN)
*
*     single or double defined below
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
*     declare argument as arrays and data-type her
```

(Continued)

```
      CHARACTER*(*) CNAME
*
*     -----------------------------------------------------------------
*
*     cname  = name of the exinit definition
*     lennam = length of the character string
*     time   = current problem time
*     ncycle = current time step number
*     ngpel  = gridpoint or element user number
*     nument = array length
*     lst    = start of the element point loop
*     lfin   = end of the element point loop
*
*     -----------------------------------------------------------------
*
*     parameter constants
*
*     global commons
      CHARACTER*16 CVAR
*
*     local dimensions and declarations
      DIMENSION NGPEL(*)
      DIMENSION IPU(1),XPVAR(1)
      DIMENSION IPN(1),NZU(1)
      DIMENSION IZVAR(8)
      DIMENSION NZONEU(NUMENT),XVAR(NUMENT)
*
      DATA ACCG /9.81/
      DATA YSURF /5.75/
      DATA RHOREF /1000./
      DATA BULK /2.2E9/
*
*     check if we have the right initial condition entry
      IF (CNAME(1:LENNAM) .NE. ’INEL1’) GOTO 9900
*
*     loop over the elements
      NZV = 0
      DO 200 NZ = LST,LFIN
*
      NZU(1) = NGPEL(NZ)
*
      CVAR = ’NODE1’
      CALL RETRIEVE_ELEMENT_INT_VAR(1,NZU,IPN,CVAR)
      IZVAR(1) = IPN(1)
      CVAR = ’NODE2’
      CALL RETRIEVE_ELEMENT_INT_VAR(1,NZU,IPN,CVAR)
      IZVAR(2) = IPN(1)
      CVAR = ’NODE3’
```

(Continued)

```
      CALL RETRIEVE_ELEMENT_INT_VAR(1,NZU,IPN,CVAR)
      IZVAR(3) = IPN(1)
      CVAR = ’NODE4’
      CALL RETRIEVE_ELEMENT_INT_VAR(1,NZU,IPN,CVAR)
      IZVAR(4) = IPN(1)
      CVAR = ’NODE5’
      CALL RETRIEVE_ELEMENT_INT_VAR(1,NZU,IPN,CVAR)
      IZVAR(5) = IPN(1)
      CVAR = ’NODE6’
      CALL RETRIEVE_ELEMENT_INT_VAR(1,NZU,IPN,CVAR)
      IZVAR(6) = IPN(1)
      CVAR = ’NODE7’
      CALL RETRIEVE_ELEMENT_INT_VAR(1,NZU,IPN,CVAR)
      IZVAR(7) = IPN(1)
      CVAR = ’NODE8’
      CALL RETRIEVE_ELEMENT_INT_VAR(1,NZU,IPN,CVAR)
      IZVAR(8) = IPN(1)
*     Find the eight nodes of the zone
      YMID = 0.0
      DO 100 IC = 1,8
          IPI = IZVAR(IC)
*         get user numbers
          CALL PX_GET_USR_PNT_FROM_INT_PNT
     +       ( IPU(1) , IPI )
*         get ypos
          CVAR = ’YPOS’
          CALL RETRIEVE_GRIDPOINT_FLOAT_VAR(1,IPU,XPVAR,CVAR)
          YMID = YMID + XPVAR(1)
 100  CONTINUE
*     Compute the z-coordinate of the center of the zone and
*     compute the pressure for the the distance under the water level
      YMID   = YMID/8.
      DH     = YSURF - YMID
      PRES   = RHOREF * ACCG * DH
*     To this pressure belongs a density
      RHO    = RHOREF + PRES*RHOREF/BULK
*
*     Only change the density in non_void zones
      IUNUS = ISVOID(NZU(1))
      IF(IUNUS.EQ.0) then
          NZV = NZV+1
          NZONEU(NZV) = NZU(1)
          XVAR(NZV) = RHO
      ENDIF
 200  CONTINUE
*
      CVAR=’DENSITY’
      CALL STORE_ELEMENT_FLOAT_VAR(NZV,NZONEU,XVAR,CVAR)
*
*     all statements ultimately end at label 9900
 9900 CONTINUE
      RETURN
      END
```

Defines the pressure within a closed volume bounded by membrane elements.

Calling Sequence:

```
CALL              EXPBAG (NAME, LENNAM, TIME, VOLUME, PRES)
```

Input:

NAME
Character variable.
Name of the gas bag.

LENNAM
Integer variable.
Number of characters in NAME.

TIME
Real variable.
Problem time.

VOLUME
Real variable.
Volume inside the gas bag.

Output:

PRES
Real variable.
Pressure inside the gas bag.

Remarks:

1. This subroutine must be included if there are any GBAGEX entries in the input file.
2. The subroutine is called every time step. The volume of the gas bag is calculated and passed to the subroutine. The pressure in the gas bag is returned.
3. The precision of the calculations should be appropriate for the computer being used. See the *MSC.Dytran Installation and Operations Guide*.

(Continued)

Example:

This subroutine simulates an air bag with the pressure inside initially at 100 N/m2 and updated using the equation P ∗ V = constant.

```
      SUBROUTINE EXPBAG
     +(PBNAME, LENNAM, TIME, VOLUME, PRES)
C
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
C
      SAVE IFIRST, CONST
      CHARACTER *(*) PBNAME
      DATA IFIRST /0/
C
      IF (IFIRST.EQ.0) THEN
            PRES = 1000.
            CONST = PRES * VOLUME
            IFIRST = 1
      ELSE
            PRES = CONST/VOLUME
      ENDIF
C
      RETURN
      END
```

Defines the pressure on a set of faces.

Calling Sequence:

`CALL` `EXPLD (NAME, LENNAM, TIME, PRES, SIGN)`

Input:

NAME
Character variable.
Name of a set of pressures.

LENNAM
Integer variable.
Number of characters in NAME.

TIME
Real variable.
Problem time.

SIGN
Real variable.
Unused.

Output:

PRES
Real variable.
The magnitude of the pressure.

Remarks:

1. This subroutine must be included if there are any PLOADEX entries in the input file.
2. The precision of the calculations should be appropriate for the computer being used. See the *MSC.Dytran Installation and Operations Guide*.

Example:

```
      SUBROUTINE EXPLD (NAME, LENNAM, TIME, PRES, SIGN)
      IMPLICIT DOUBLE PRESCISION (A-H, O-Z)
      CHARACTER *(*) NAME
C
      PRES = 725. * SQRT (TIME)
      RETURN
      END
```

EXSPR

User-Defined CSPR Spring Element

Returns the force and stiffness in CSPR spring elements.

Calling Sequence:

```
CALL                EXSPR (N, M, IX, IC, PROP, HISV, FORCEO, C, DI, V, A,
                    UREL, DUREL, VREL, XMASS, FORCE, STIFF)
```

Input:

N
Integer variable.
Element number.

M
Integer variable.
Property number.

IX(2)
Integer array.
Connectivity:
IX(1) = grid point at end 1.
IX(2) = grid point at end 2.

IC(2)
Integer array.
Unused.

PROP(7)
Real array.
Properties as input on the PSPREX entry.

HISV(6)
Real array.
History variables for the element. This array can be used by the user to store variables from one time step to the next.

FORCEO
Real variable.
Force in the element at the previous time step.

C(3,2)
Real array.
Deformed coordinates in the basic coordinate system:
C(1:3,1) = x-, y-, z-coordinates at end 1.
C(1:3,2) = x-, y-, z-coordinates at end 2.

DI (6,2)
Real array.
Incremental displacements in the basic coordinate system:
DI(1:3,1) = x, y, z, translational displacements of end 1.
DI(4:6,1) = x, y, z, rotational displacements of end 1.
DI(1:3,2) = x, y, z, translational displacements of end 2.
DI(4:6,2) = x, y, z, rotational displacements of end 2.

These are incremental displacements; i.e., the displacements for this time step only.

(Continued)

V(6,2) Real array.
Velocities in the basic coordinate system:
V(1:3,1) = x, y, z, translational velocities of end 1.
V(4:6,1) = x, y, z, rotational velocities of end 1.
V(1:3,2) = x, y, z, translational velocities of end 2.
V(4:6,2) = x, y, z, rotational velocities of end 2.

A(6,2) Real array.
Accelerations in the basic coordinate system:
A(1:3,1) = x, y, z, translational accelerations of end 1.
A(4:6,1) = x, y, z, rotational accelerations of end 1.
A(1:3,2) = x, y, z, translational accelerations of end 2.
A(4:6,2) = x, y, z, rotational accelerations of end 2.

UREL Real variable.
Relative displacement of the element; i.e., the displacement of end 2 in the spring direction minus the displacement of end 1.

DUREL Real variable.
Relative incremental displacement of the element.

VREL Real variable.
Relative velocity of the end points of the element in the direction of the element.

XMASS(2) Real array.
Mass of the grid points at ends 1 and 2.

Output:

FORCE Real variable.
Force in the element.

STIFF Real variable.
Current stiffness of the element.

Remarks:

1. This subroutine must be included if the PSPREX entry is specified in the Bulk Data Section.
2. The velocities (V) and accelerations (A) of the end points can be updated by the user subroutine when required.
3. The stiffness is used by MSC.Dytran to estimate the time step. A nonzero value must be returned.
4. The precision of the calculations should be appropriate for the computer being used. See the *MSC.Dytran Installation and Operations Guide*.

(Continued)

Example:

This example defines the stiffness and the corresponding force for a spring element.

```
      SUBROUTINE EXSPR
     +(N,M,IX,IC,PROP,HISV,FORCEO,C,DI,V,A,UREL,DUREL,
      +VREL,XMASS,FORCE,STIFF)
*
      single or double defined below
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
*     declare argument as arrays and data-type here....
      DIMENSION IX(2),IC(2),PROP(7),HISV(6),C(3,2),
     +            DI(6,2),V(6,2),A(6,2),XMASS(2)
*
*     define the stiffness and the corresponding force
      RMASS = 1./(XMASS(1) + XMASS(2))
      STIFF = RMASS * (XMASS(1)*1.E3 + XMASS(2)*2.E3)
      FORCE = STIFF * DUREL
*
      RETURN
      END
```

The EXTLU user subroutine declares FORTRAN logical unit (LU) numbers for usage by other user subroutines.

Calling Sequence:

```
CALL                EXTLU (LUUSR, LUMAX)
```

Input:

LUMAX — Integer.
Maximum LU number allowed.

Output:

LUUSR — Integer array.
To store declared LU number.

Remarks:

1. LUMAX is set by MSC.Dytran. This value can be used to check whether the user-defined LU number does not exceed the maximum allowable LU number. LUMAX should not be changed in the user subroutine.
2. Declared FORTRAN LU numbers are reserved within MSC.Dytran and are used for files you need in other user subroutines.
3. It is advised to define a common block in the EXTLU subroutine where the user-defined LU numbers are kept. This common block can be included in any other user subroutine that utilizes external user-defined files. For example:

   COMMON /MYLU/ LU01, LU02, LU03, LU04, LU05

Example:

```
      SUBROUTINE EXTLU (LUUSR,LUMAX)
*
*     User Subroutine to declare FORTRAN LU numbers for
*     exclusive usage in any User Subroutines.
*
*     Subroutine EXTLU is always called by the program
*
*     MSC.Dytran checks whether the user declaration is valid
*
      DIMENSION LUUSR(LUMAX)
      COMMON /MYLU/ LU01,LU02,LU03,LU04,LU05
*
*     E.g. Declare LU numbers 80 and 81 as user exclusive LU’s
*     Any LU number greater than LUMAX is illegal
```

(Continued)

```
*
      LU01 = 80
      LU05 = 81
      LUUSR(LU01) = LU01
      LUUSR(LU05) = LU05
*
*     The above statements reserve LU01 and LU05 as user exclusive LU’s
*
      RETURN
      END
```

Constrains the velocity of Lagrangian grid points.

Calling Sequence:

```
CALL            EXTVEL (NAME, LENNAM, NGP, XPOS, YPOS, ZPOS, XVEL,
                YVEL, ZVEL, XAVEL, YAVEL, ZAVEL,PMASS)
```

Input:

NAME — Character string.
Velocity boundary name.

LENNAM — Integer variable.
Number of characters in NAME.

NGP — Integer variable.
Grid point number.

XPOS — Real variable.
Old x-coordinate of point.

YPOS — Real variable.
Old y-coordinate of point.

ZPOS — Real variable.
Old z-coordinate of point.

XVEL — Real variable.
Tentative x-translational velocity of the point.

YVEL — Real variable.
Tentative y-translational velocity of the point.

ZVEL — Real variable.
Tentative z-translational velocity of the point.

XAVEL — Real variable.
Tentative x-angular velocity of the point.

YAVEL — Real variable.
Tentative y-angular velocity of the point.

ZAVEL — Real variable.
Tentative z-angular velocity of the point.

(Continued)

Output:

XVEL Real variable.
Constrained x-translational velocity of the point.

YVEL Real variable.
Constrained y-translational velocity of the point.

ZVEL Real variable.
Constrained z-translational velocity of the point.

XAVEL Real variable.
Constrained x-angular velocity of the point.

YAVEL Real variable.
Constrained y-angular velocity of the point.

ZAVEL Real variable.
Constrained z-angular velocity of the point.

PMASS Real variable.
Grid point mass.

Remarks:

1. This subroutine must be included if there are any FORCEEX entries in the input file.
2. The subroutine returns the constrained velocities of each grid point.
3. EXTVEL is called once for every grid point referenced on FORCEEX entries.
4. The precision of the calculations should be appropriate for the computer being used. See the *MSC.Dytran Installation and Operations Guide*.

Example:

This example constrains the x-velocity of grid points with an x coordinate that is positive.

```
      SUBROUTINE EXTVEL
   +(NAME, LENNAM, NGP, XPOS, YPOS, ZPOS, XVEL, YVEL, ZVEL, XAVEL, YAVEL, ZAVEL)
*
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
*
      CHARACTER*(*) NAME
*
*     This routine puts the x-velocity to zero when the x-position of the point
*     is positive.
*
      IF (XPOS.GT.0) XVEL = 0.
*
      RETURN
      END
```

Returns the force in CVISC damper elements.

Calling Sequence:

```
CALL          EXVISC (N, M, IX, IC, PROP, HISV, FORCEO, C, DI, V, A,
              UREL, DUREL, VREL, XMASS, FORCE)
```

Input:

N
Integer variable.
Element number.

M
Integer variable.
Property number.

IX(2)
Integer array.
Connectivity:
IX(1) = grid point at end 1.
IX(2) = grid point at end 2.

IC(2)
Integer array.
Unused.

PROP(7)
Real array.
Properties as input on the PVISCEX entry.

HISV(6)
Real array.
History variables for the element. This array can be used by the user to store variables from one time step to the next.

FORCEO
Real variable.
Force in the element at the previous time step.

C(3,2)
Real array.
Deformed coordinates in the basic coordinate system:
C(1:3,1) x, y, z, coordinates at end 1.
C(1:3,2) x, y, z, coordinates at end 2.

DI(6,2)
Real array.
Incremental displacements in the basic coordinate system:
DI(1:3,1) x, y, z, translational displacements of end 1.
DI(4:6,1) x, y, z, rotational displacements of end 1.
DI(1:3,2) x, y, z, translational displacements of end 2.
DI(4:6,2) x, y, z, rotational displacements of end 2.

These are incremental displacements; i.e., the displacements for this time step only.

(Continued)

V(6,2) Real array.
Velocities in the basic coordinate system:
V(1:3,1) x, y, z, translational velocities of end 1.
V(4:6,1) x, y, z, rotational velocities of end 1.
V(1:3,2) x, y, z, translational velocities of end 2.
V(4:6,2) x, y, z, rotational velocities of end 2.

V(6,2) Real array.
Accelerations in the basic coordinate system:
A(1:3,1) x, y, z translational accelerations of end 1.
A(4:6,1) x, y, z rotational accelerations of end 1.
A(1:3,2) x, y, z translational accelerations of end 2.
A(4:6,2) x, y, z rotational accelerations of end 2.

UREL Real variable.
Relative displacement of the element; i.e., the displacement of end 2 in the damper direction minus the displacement of end 1.

DUREL Real variable.
Relative incremental displacement of the element.

VREL Real variable.
Relative velocity of the end points of the element in the direction of the element.

XMASS(2) Real array.
Mass of the grid points at ends 1 and 2.

Output:

FORCE Real variable.
Force in the element.

Remarks:

1. This subroutine must be included if the PVISCEX entry is specified in the Bulk Data Section.

2. The velocities (V) and accelerations (A) of the end points can be updated using the user subroutine if required.

3. The precision of the calculations should be appropriate for the computer being used. See the *MSC.Dytran Installation and Operations Guide*.

(Continued)

Example:

This example defines the damping force for a danger element.

```
      SUBROUTINE EXVISC
     +(N,M,IX,IC,PROP,HISV,FORCEO,C,DI,V,A,UREL,DUREL,
      + VREL,XMASS,FORCE)
*
*     single or double defined below
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
*
*     declare argument as arrays and data-type here....
      DIMENSION IX(2),IC(2),PROP(7),HISV(6),C(3,2),
     +            DI(6,2),V(6,2),A(6,2),XMASS(2)
*
*     define the force on the damper element
      VELX = V(1,1) - V(1,2)
      VELY = V(2,1) - V(2,2)
      VELZ = V(3,1) - V(3,2)
*
      FORCE = 1.E-3 * SQRT(VELX*VELX + VELY*VELY + VELZ*VELZ)
*
      RETURN
      END
```

EXYLD

User-Specified Yield Behavior

The EXYLD user subroutine defines the yield stress, YLDSQ, for Lagrangian solid elements or Eulerian elements with shear strength, for all elements in the string (ISTART ... IEND).

Calling Sequence:

```
CALL             EXYLD (MATNAM, LENNAM, TIME, NCYCLE, DLTH, IZONE,
                 PRES, EDIS, SIE, RHO, FBURN, EFFPLS, ZMASS, EFFSTS,
                 TWOJ2, EFFSR, USRVR1, USRVR2, RELV, SXXO, SYYO, SZZO,
                 SXYO, SYZO, SXZO, SXXT, SYYT, SZZT, SXYT, SYZT, SXZT,
                 DEXX, DEYY, DEZZ, DEXY, DEYZ, DEZX, TDET, YLDSQ,
                 ISTART, IEND)
```

Input:

MATNAM
Character string.
Name of the material.

LENNAM
Integer variable.
Length of MATNAM.

TIME
Real variable.
Time at the current time step.

NCYCLE
Integer variable.
Cycle number of the current time step.

DLTH
Real variable.
Time step increment at the current time step.

IZONE(*)
Integer array.
Element number.

PRES(*)
Real array.
Pressure in the element.

EDIS(*)
Real array.
Distortional energy of the element.

SIE(*)
Real array.
Specific internal energy of the element.

RHO(*)
Real array.
Density of the element.

FBURN(*)
Real array.
Burn fraction of the element.

(Continued)

EFFPLS(*) Real array.
Effective plastic strain of the element.

ZMASS(*) Real array.
Mass of the element.

EFFSTS(*) Real array.
Old effective stress of the element.

TWOJ2(*) Real array.
Trial second invariant at current time of the element.

EFFSR(*) Real array.
Effective strain rate of the element.

USRVR1(*) Real array.
User variable 1 of the element.

USRVR2(*) Real array.
User variable 2 of the element.

RELV(*) Real array.
Relative volume of the element.

SXXO(*)...SXZO(*) Real arrays.
Old deviatoric stresses of the element.

SXXT(*)...SXZT(*) Real arrays.
Trial deviatoric stresses at current time of the element.

DEXX(*)...DEZX(*) Real arrays.
Strain rate components of the element.

TDET(*) Real array.
Detonation time of the element.

ISTART Integer.
First element in string.

IEND Integer.
Last element in string.

Output:

YLDSQ(*) Real array.
Yield stress of the element.

(Continued)

Remarks:

1. The subroutine must be included if there are any YLDEX entries in the input.
2. The FBURN array is only used for Eulerian material with strength. The IZONE array and the USRVR1-2 arrays can only be used for Lagrangian material.
3. The precision of the calculations should be appropriate for the computer being used. See the *MSC.Dytran Installation and Operations Guide*.

Example:

In this example, the yield stress will be computed as a function of the effective strain rate and the pressure. The routine returns the yield stress YLDSQ to the code. The USRVR1 and USRVR2 arrays are used to store the yield stress and the effective strain rate in memory so that they can be requested as output.

```
      SUBROUTINE EXYLD
     +      (MATNAM, LENNAM, TIME, NCYCLE, DLTH, IZONE, PRES, EDIS, SIE,
     +       RHO, FBURN, EFFPLS, ZMASS, EFFSTS, TWOJ2, EFFSR, USRVR1,
     +       USRVR2, RELV, SXXO, SYYO, SZZO, SXYO, SYZO, SXZO, SXXT,
     +       SYYT, SZZT, SXYT, SYZT, SXZT, DEXX, DEYY, DEZZ, DEXY,
     +       DEYZ, DEZX, TDET, YLDSQ, ISTART, IEND)
C
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
C
      DIMENSION IZONE(*)
      DIMENSION YLDSQ(*)
      DIMENSION EDIS(*),ZMASS(*),EFFSTS(*),TDET(*)
      DIMENSION SIE(*),RHO(*),FBURN(*),EFFPLS(*),EFFSR(*)
      DIMENSION TWOJ2(*),PRES(*),RELV(*)
      DIMENSION USRVR1(*), USRVR2(*)
      DIMENSION SXXO(*),SYYO(*),SZZO(*),SXYO(*),SYZO(*),SXZO(*)
      DIMENSION SXXT(*),SYYT(*),SZZT(*),SXYT(*),SYZT(*),SXZT(*)
      DIMENSION DEXX(*),DEYY(*),DEZZ(*),DEXY(*),DEYZ(*),DEZX(*)
C
      CHARACTER*80 MATNAM
C
C
C     Example of yield stress as a function of
C     effective strain rate and pressure.
C
      DO 100 NZ = ISTART, IEND
C
C           define the yield stress.
            YLDSQ(NZ) = EFFSR(NZ)*SQRT(ABS(PRES(NZ)))
```

(Continued)

```
C
C           store the yield stress and effective strain
C           rate in the user vars, so that they can be
C           requested as output.
            USRVR1(NZ) = YLDSQ(NZ)
            USRVR2(NZ) = EFFSR(NZ)
100   CONTINUE
*
      RETURN
      END
```

GEXOUT

User-Defined Grid Point Output

User-defined grid-point output.

Calling Sequence:

```
CALL              GEXOUT (NAME, LENNAM, NGP, CGP, NGTYPE, LIGRD, LXGRD)
```

Input:

NAME
Character string.
Output name specified on the GPEXOUT entry.

LENNAM
Integer variable.
Length of NAME.

NGP(*)
Integer array.
Grid-point number.

CGP(*)
Character *8 array.
Unused.

NGTYPE
Integer variable.
Type of element to which the grid point is attached:

2	One-dimensional element.
3	Triangular shell.
4	Quadrilateral shell.
5	Triangular membrane.
6	Dummy triangle.
7	Dummy quadrilateral.
8	Lagrangian solid.
9	Eulerian solid (hydrodynamic).
10	Eulerian solid (with strength).
11	Eulerian solid (multimaterial).

LIGRD(*)
Integer array.
Base address of grid point in the main integer storage array ILGDAT.

LXGRD(*)
Integer array.
Base address of grid point in the main real storage array XLGDAT.

Remarks:

1. This subroutine must be included if there are any GPEXOUT Case Control commands.

2. The subroutine can be used to calculate results based on the data available in MSC.Dytran.

(Continued)

3. The precision of the calculations should be appropriate for the computer being used. See the *MSC.Dytran Installation and Operations Guide.*

4. This subroutine is vectorized. All the input data is stored in arrays, which must be dimensioned. The start and end of the arrays is given by the variables LST and LFIN in the common block /LOCLOP/. All of the entries in the arrays between LST and LFIN must be output. See the following example.

5. Access to grid-point variables is possible by including calls to the subroutines listed in Section 3.15.2 on page 3-75 using the variable names from Section 3.9.2 on page 3-24.

Example:

This example outputs the total force on a grid point to the primary output (unit 6).

```
      SUBROUTINE GEXOUT
      +(NAME, LENNAM, NGP, CGP, NGTYPE, LIGRD, LXGRD)
*
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
*
      DIMENSION NGP (*), LIGRD(*), LXGRD(*)
      CHARACTER *(*) NAME
      CHARACTER*8 CGP(*)
*
      COMMON/LOCLOP/LST, LFIN
      COMMON/XLGMEM/XLGDAT(1)
*
      IF (NGTYPE.NE.8) GOTO 9900
*
*     The total force on each Lagrangian node is printed out.
*
      DO 100 NG = LST, LFIN
      FTOT = XLGDAT(LXGRD(NG) +7)**2+
      +        XLGDAT(LXGRD(NG)+8)**2+
      +        XLGDAT(LXGRD(NG)+9)**2
      FTOT = SQRT (FTOT)
      WRITE (6, 101) NGP (NG), FTOT
 101  FORMAT (1X, ’Force on node’, I5, ‘is‘, E13.5)
 100  CONTINUE
*
 9900 RETURN
      END
```

3.16 Prestress Analysis

In some cases, a prestressed initial state is needed for a structure in order to get the correct results in a transient follow-up problem; e.g., a bird striking a rotating turbine fan blade.

An efficient way to solve the static prestress problem is by using MSC.Nastran. From this solution, it is possible to initialize MSC.Dytran such that the correct initial, prestressed state will be achieved for a transient dynamic analysis.

There are two ways of initializing a prestressed state:

1. Direct MSC.Nastran initialization.

   Both the displacement and the stress field are read from the MSC.Nastran output data and transferred directly into MSC.Dytran as an initial state for the structural elements. It is defined by including the NASTINP FMS command referring to the MSC.Nastran solution file.

2. MSC.Nastran initialization via an intermediate MSC.Dytran prestress analysis.

   The MSC.Nastran computed displacement field is used to obtain a stable prestressed state in MSC.Dytran (the prestress analysis). This MSC.Dytran solution will be applied in the subsequent transient analysis and will act as a stable initial state for the structure.

   Entries involved in the prestress analysis are:

   - PRESTRESS
   - SOLUOUT
   - BULKOUT
   - NASINIT
   - NASDISP

   The entry required to effect the initialization of the transient analysis:

   - SOLINIT

   The parameters involved are:

   - INITFILE
   - INITNAS

3.16.1 An Example MSC.Nastran Input Data

The following is an example of MSC.Nastran input data that can be used to obtain a solution that allows for geometric and material nonlinearities.

```
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$ Example Input Deck for an Initialization for Centrifugal Loading           $
$                                                                            $
$ The Output will be an XL-database                                          $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$
$ Executive control statements
INIT DBALL LOGICAL=(DBALL(80000))
TIME 600
SOL 66
COMPILE SOL66 SOUIN=MSCSOU NOLIST NOREF $
CEND
TITLE=       Example MSC.NASTRAN Input Deck
SUBTITLE=       PRE-STRESS CALCULATION
  SPC=1
  ECHO = NONE
  MAXLINES = 2000000
  SEALL=ALL
  DISP(PRINT,PLOT)=ALL
  STRESS(PRINT,PLOT)=ALL
  NLPARM = 10
SUBCASE 1
LABEL=CENTRIFUGAL LOADING OF 10000 RPM
  LOAD=1
BEGIN BULK
$
$ Include Geometry and Constraints From Files
INCLUDE SOLID.DAT
INCLUDE SPC1.DAT
$
$ Material definition
PSOLID,1,1,0
MAT1,1,1.156E11,,0.318,4527.
$
$ Gridpoint for the rforce loading...
GRID,100000,0,-.1,0.0,0.0,0,0
$
$ DATA DECK
PARAM,POST,0
$
$ Not Lumped But Coupled Masses
PARAM,COUPMASS,1
$
$ Define Large Displacements
PARAM,LGDISP,1
```

```
$
PARAM,DBDICT,2
PARAM,OUNIT,12
$
$ The extra point on the axis of rotation is fixed
SPC1,1,123546,100000
$
$ Define the Centrifugal Loading
RFORCE,1,100000,0,85.,0.0,0.0,1.0,2
$
$ Definition of the Solution Sequence and Convergence Criteria
NLPARM,10,10,,,,,UPW
$
ENDDATA
```

CHAPTER 4

Input Data

4.1 General Description of the Input File

Input to MSC.Dytran takes the form of a data file where each line can contain up to 80 characters. The file contains all the information to define the analysis model and control the analysis.

The input to MSC.Dytran is similar, but not identical, to that for MSC.Nastran and MSC/DYNA. If you are familiar with MSC.Nastran, learning to use MSC.Dytran will be very easy although you should note the areas in which the two programs differ. These differences are summarized in Section 4.2 on page 4-3.

The input data is split into four main sections, which must come in the following order:

1. File Management Section (FMS).
2. Executive Control Section.
3. Case Control Section.
4. Bulk Data Section (note that parameter options may appear at any location within the Bulk Data Section).

The File Management Section contains information about the files used during the analysis and to control restarting. The Executive Control Section is not used often in MSC.Dytran, since the program does not have an Executive System like MSC.Nastran.

The Case Control Section controls the analysis, specifies the type of input and output required, selects the constraints and loading from the Bulk Data, and allows you to control the way the analysis progresses. A discussion of the functions available in the Case Control Section and a detailed description of the commands that can be used is given in Section 4.5 on page 4-28.

The Bulk Data Section contains all data necessary to define the model, the constraints, loading conditions, and initial conditions. Only one model can be defined in the input data, but several types of constraints and loading can be specified. The constraints and loading actually used in the analysis are selected in the Case Control Section. The Bulk Data Section is discussed in Section 4.6 on page 4-78 together with a detailed description of the entries.

The File Management, Executive Control, and Case Control Sections use a free-format input, which means that the data can appear anywhere on the line with individual items separated by commas or spaces. The Bulk Data Section can also be in free format and can optionally be in fixed format. In cases where additional precision is required, large format can be used, where each entry occupies two lines in the input file. Free, fixed, and large format can be mixed as needed in the input file on a line-by-line basis.

Comments can appear anywhere in the input file by placing a $ at the start of the comment. A full description of the various input formats is given in Section 4.6.2 on page 4-78.

The input data can be present in several separate files. In this case, you can use the INCLUDE command or entry, available in both the Case Control and Bulk Data Sections, to direct MSC.Dytran to read the appropriate file. The mechanism can be used to store the infrequently changed Bulk Data in one file, while the File Management, Executive Control, and Case Control Sections, which are usually modified more often, can be stored in another file.

4.2 Similarity with MSC.Nastran

The input for MSC.Dytran is similar to the input for MSC.Nastran, since the vast majority of the input for the two codes is identical. There are, however, a number of differences arising from the fundamental differences between the two programs, and the fact that there are features available in MSC.Dytran that are not available in MSC.Nastran and vice versa.

Similarity to MSC.Nastran has a number of advantages for anyone who works with both programs:

- You only need to learn one form of input.
- Models used for MSC.Nastran analyses can be reused with minor modifications for MSC.Dytran.
- MSC.Dytran can be used with a wide range of modeling packages.

It is important to remember that MSC.Nastran and MSC.Dytran are completely different programs even though they offer similar input. A CQUAD4 shell element in MSC.Dytran has nothing in common with the CQUAD4 shell element in MSC.Nastran, since it differs in formulation, type of integration, and capabilities. Similarly, other features defined using the same entries do not necessarily behave in the same way. The solution method is different, so an identical analysis in MSC.Nastran and MSC.Dytran can give slightly different results, although they will be within engineering accuracy.

Input

MSC.Nastran has a wide range of facilities of which a number are not available in MSC.Dytran. Therefore, there are MSC.Nastran entries that are not valid in MSC.Dytran.

The following entries are compatible with both codes:

Elements

CBAR
CBEAM
CDAMP1
CDAMP2
CELAS1
CELAS2
CHEXA
CQUAD4
CROD
CTETRA
CTRIA3
CVISC

Properties

PBAR
PBEAM
PROD
PSHELL

PCOMP
PDAMP
PELAS
PSOLID
PVISC

Materials

MAT1
MAT8

Loads and Constraints

DAREA
FORCE
FORCE1
FORCE2
GRAV
GRDSET
MOMENT
MOMENT1
MOMENT2
PLOAD
PLOAD4
RFORCE
SPC
TIC

Coordinate Systems

CORD1C
CORD1R
CORD1S
CORD2C
CORD2R
CORD2S

Other Entries

CONM2
GRID
TABLED1
TIME
TITLE
TLOAD1
TLOAD5

The FMS has the same purpose in both MSC.Dytran and MSC.Nastran, but it is less important in MSC.Dytran since all the filenames are defined automatically. The FMS controls restarting and user-written subroutines as well as specification of the type of the output files.

The Executive Control Section exists but is rarely used since MSC.Dytran does not have an Executive System or DMAP.

The Case Control Section has the same function in both MSC.Dytran and MSC.Nastran but uses different commands.

PARAM entries are used by MSC.Dytran but offer different options to those in MSC.Nastran.

MSC.Dytran offers slightly greater flexibility in the way the input file can be defined, as listed below:

- Free-format data can have more than eight characters.
- Continuation mnemonics do not have to be unique.
- Fixed and free-format input can be freely mixed on a line-by-line basis.
- Real numbers can be entered as integers.

However, continuation lines must follow the entry that references them. If you intend on using both MSC.Dytran and MSC.Nastran on a regular basis, use only those options that are available in both programs to avoid confusion and incompatibility.

Loading

Several of the entries used for static loading in MSC.Nastran (such as FORCE, MOMENT, and PLOAD) are used for dynamic transient loading in MSC.Dytran. Instead of being referenced directly from Case Control, they are referenced from a TLOAD1 entry that gives the variation of the load with time.

The DAREA entry, used for transient loading in MSC.Nastran, is also valid in MSC.Dytran.

4.3 File Management Section (FMS)

4.3.1 Introduction

The File Management Section (FMS) controls any file assignments that are required by MSC.Dytran. It also controls restarting. The FMS must be placed at the beginning of the input file, but the individual statements can be in any order within the FMS.

Most of the file assignments are made automatically by MSC.Dytran and cannot be changed by the user. The filenames used are machine dependent and are listed in the *MSC.Dytran Installation and Execution Guide*.

A summary of the statements available in the FMS is given in Section 4.3.2 on page 4-6. Each statement is described in detail in Section 4.3.3 on page 4-7.

4.3.2 Summary

The following statements are valid in the FMS:

<u>Prestress Analysis</u>

PRESTRESS Indicates a prestress analysis.

BULKOUT Selects the file to which grid-point data is to be written.

NASTDISP Selects the MSC.Nastran displacement file to be used.

SOLUOUT Selects a file to which solution data is to be written.

<u>New Analyses</u>

START Indicates a new analysis.

NASTINP Selects an MSC.Nastran solution file from which MSC.Dytran is to be initialized.

SOLINIT Selects an MSC.Dytran prestress solution file from which MSC.Dytran is to be initialized.

NASTOUT Selects a file to which MSC.Dytran writes geometric and material data in MSC.Nastran format.

Restart Control

RESTART	Indicates a restart of a previous analysis.
RSTFILE	Selects the restart file to be used.
RSTBEGIN	Selects the time step at which the calculation is to be restarted.

User Code

USERCODE	Indicates that user-written subroutines are required for the analysis and defines the filename containing the FORTRAN user-written subroutines.

File Selection

TYPE	Defines the format of a file.
SAVE	Defines the interval of saving an output file.

4.3.3 FMS Descriptions

The format of the FMS statements is free field. In presenting the general formats for each statement, the following conventions are used:

- Uppercase letters should be typed as shown.
- Lowercase letters indicate that a value or option can be substituted.
- Brackets [ ] give a choice of different options.

The default value indicates the value that is used if no FMS command is present. The type column indicates the type of data you must supply. This can be I (Integer), R (Real), or C (Character). In addition, a range of permissible values may also be indicated. For example, I > 0 means that you must supply an integer that is greater than zero.

BULKOUT

Prestress Grid Point Output

Defines a file to which grid-point data is written at the end of the prestress analysis.

Format and Example	Defaults
BULKOUT = filename	Required
BULKOUT = GRID.DAT	

Option	Meaning	Type
filename	The filename to be used.	C

Remarks:

1. The Bulk Data file “filename” contains only grid point data of the deformed geometry at the end of the prestress analysis. It can be used to construct an ALE mesh for the final transient dynamic analysis.
2. See also the NASTDISP, PRESTRESS and SOLUOUT File Management Section statements, and the NASINIT Bulk Data entry.

NASTDISP

Prestress MSC.Nastran Displacement File

Specifies an MSC.Nastran displacement file to be used as input for the prestress analysis.

Format and Example	Defaults
NASTDISP = filename	Required
NASTDISP = DISPLACE.DIS	

Option	Meaning	Type
filename	The filename to be used.	C

Remarks:

1. The displacement file must be either in MSC.Patran format, formed by using NASPAT on the MSC.Nastran OUTPUT2 results file, or in the XL format using MSC/XL on the MSC.Nastran XL database.
2. The default file type is MSC/XL format. This can be changed using PARAM,INITNAS.
3. See also the BULKOUT, PRESTRESS, and SOLUOUT File Management Section statements, and the NASINIT Bulk Data entry.

NASTINP

Prestress MSC.Nastran Solution File

Specifies an MSC.Nastran solution file from which MSC.Dytran is to be initialized via element stresses and grid-point displacements.

Format and Example	Defaults
NASTINP = filename1, filename2	Required
NASTINP = ELEMENT. ELS, GRID. DIS	

Option	Meaning	Type
filename1, filename2	The filenames to be used.	C, C

Remarks:

1. The stresses and displacement files are obtained by using NASPAT on the OUTPUT2 results file from MSC.Nastran.
2. Element stresses are defined in the material coordinate system.
3. It is recommended that the MSC.Nastran geometrical problem setup be performed by MSC.Dytran for consistency (see NASTOUT).
4. This option causes MSC.Dytran to read a MASS.DAT file that is automatically generated by the NASTOUT File Management Section statement.

NASTOUT

MSC.Nastran Input File for Prestress Analysis

Specifies that MSC.Dytran write out MSC.Nastran input containing geometry and material definitions including material coordinate systems, if applicable.

Format and Example	**Defaults**
NASTOUT = filename	Required
NASTOUT = NASGEO.DAT	

Option	**Meaning**	**Type**
filename	The filename to be used.	C

Remark:

1. The option causes a MASS.DAT file to be written containing the element initial masses. This file is read when the NASTINP File Management Section statement is used.

PRESTRESS

Prestress Analysis

Indicates a prestress analysis.

Format and Example	**Defaults**
PRESTRESS	Start run.
PRESTRESS	

Remarks:

1. The following entries should be present elsewhere in the File Management Section or Bulk Data Section for a prestress analysis:

NASTDISP	Specifies an MSC.Nastran displacement file to be used as input (FMS).
BULKOUT	Defines an output file to which grid-point data is written at the end of the prestress analysis (FMS).
SOLUOUT	Defines an output file to which solution data is written at the end of the prestress analysis (FMS).
NASINIT	A Bulk Data entry that controls the prestress analysis.

2. The SOLUOUT file is then used to initialize MSC.Dytran for the primary analysis (via a SOLINIT FMS statement).

3. Prestressing is described in Section 3.16 on page 3-142.

RESTART

Restarts a Previous Run

Requests that a previous run be restarted and continued.

Format and Example	Default
RESTART	Start run.
RESTART	

Remarks:

1. The RSTBEGIN File Management Section statement must be present to specify the time step from which the calculation is to be restarted.
2. The RSTFILE File Management Section statement must be present to specify the name of the restart file to be used.
3. Restarting is described in Section 3.10 on page 3-66.

RSTBEGIN

Restart Time Step

Defines the time step at which a calculation is to be restarted.

Format and Example	Default
RSTBEGIN = n	Required
RSTBEGIN = 5000	

Option	Meaning	Type
n	The number of the time step at which the analysis restarts.	I > 0

Remarks:

1. A RESTART File Management Section statement must be present to indicate a restart analysis.
2. A RSTFILE File Management Section statement must be present to specify the name of the restart file to be used.
3. Restarting is described in Section 3.10 on page 3-66.

RSTFILE

Restart File Section

Defines the restart file to be used for restarting.

Format and Example	Default
RSTFILE = filename	Required
RSTFILE = NAME.RST	

Option	Meaning	Type
filename	The filename to be used for restarting. The file must exist in your runtime directory.	C

Remarks:

1. A RESTART File Management Section statement must be present to indicate a RESTART analysis.
2. A RSTBEGIN File Management Section statement must be present to specify the time step at which the calculations are to be restarted.
3. Restarting is described in Section 3.10 on page 3-66.

SAVE

Interval Between Saving an Output File

Defines how often the file is written before it is closed and saved.

Format and Example	Default
SAVE (logical_file) = n	10
SAVE (OUTPUT1) = 6	

Option	Meaning	Type
logical_file	The logical name of the file.	C
n	The number of times an output file is written before it is closed and saved. (See Remark 3.)	I

Remarks:

1. When the file is written the specified number of times, it is closed, saved, and subsequent results are stored in a new file.
2. Results are available for postprocessing when the file has been closed and saved. If the SAVE statement is set to 1, results are stored in individual files and can be postprocessed immediately.
3. If value of n is negative for a RESTART request, the file is overwritten for every restart save. If the n value is positive, a new file is created for every restart save request.
4. If deformed shape plots are made, be aware that the data translator XDEXTR uses the initial geometry written to the archive to compute the displacements for the subsequent steps for which data is written.

SOLINIT

Specify an Initial Solution File from Prestress Analysis

Specifies a solution file used as input for a transient analysis of a prestressed structure.

Format and Example	Default
SOLINIT = filename	Required
SOLINIT = DYTRAN.SOL	

Option	Meaning	Type
filename	The filename to be used.	C

Remarks:

1. The SOLINIT File Management Section statement causes MSC.Dytran to initialize the structural part of the transient problem from a previous prestress analysis.
2. See also the BULKOUT, NASTDISP, PRESTRESS, and SOLUOUT FMS statements, and the NASINIT Bulk Data entry, for performing the prestress analysis.
3. The solution file should correspond to the filename used to write out the solution data at the end of the prestress analysis (see the SOLUOUT File Management Section statement).
4. See PARAM,INITFILE for an overview of the different initialization methods and information on the element types for which prestressing is available.

SOLUOUT

Specifies the Output Solution File from a Prestress

Specifies an output file to which the solution data is written at the end of a prestress analysis.

Format and Example	**Default**
SOLUOUT = filename	Required
SOLUOUT = DYTRAN.SOL	

Option	**Meaning**	**Type**
filename	The filename to be used.	C

Remarks:

1. The solution file is a binary file. It contains all necessary data of the solution at the end of an MSC.Dytran prestress analysis.
2. See also the BULKOUT and PRESTRESS File Management Section statements, and the NASINIT Bulk Data entry.
3. The solution output file should be the same file as used for initializing the primary analysis (see the SOLINIT File Management Section statement).
4. See PARAM,INITFILE for an overview of the different initialization methods and information on the element types for which prestressing is available.

START

Primary Analysis

Indicates the primary analysis.

Format and Example	**Default**
START	Primary analysis.
START	

Remarks:

1. Since the default is a start analysis, this statement can be omitted.
2. See also the PRESTRESS and RESTART File Management Section statements.
3. This entry can be accompanied by using either of the following File Management Section statements:

   SOLINIT The analysis is to be initialized from a previous MSC.Dytran prestress analysis.

   or

   NASTINP The analysis is to be initialized from a previous MSC.Nastran analysis.

TYPE

Type of Output File

Defines the type of an output file.

Format and Example	Default
TYPE (logical_file) = type	ARCHIVE
TYPE (OUTPUT1) = TIMEHIS	

Option	Meaning		Type
logical_file	The logical filename to which the command refers.		C
type	The format of the file:		C
	ARCIVE	Archive file for storing results at a particular time step.	
	TIMEHIS	Time-history file for storing results for particular entities at particular times during the analysis.	
	RESTART	Restart file used to restart the calculation.	
	STEPSUM	One-line time step summary.	
	MATSUM	A material summary at a particular time step.	
	EBDSUM	An Eulerian boundary summary at a particular time step.	

Remarks:

1. Archive files are normally used to store results at one or more time steps during the analysis. Archive files are used in postprocessing to produce deformed shapes, contour plots, and vector plots. Archive files contain the model geometry and results.

2. Time-history files are normally used to store results for particular grid points and elements and are used to produce time-history plots. Only the results are stored.

3. Restart files are used to restart the calculation.

4. The summaries STEPSUM, MATSUM, and EBDSUM are always printed on standard output, irrespective of the value of “logical_file”.

USERCODE

User Subroutine Selection

Defines the file containing user-written subroutines to be used with the analysis.

Format and Example	Default

USERCODE = filename
USERCODE = user.f
	No user code is used.

Option	Meaning	Type
filename	The name of the file containing the user-written FORTRAN subroutines. The file must exist in your working area.	C

Remarks:

1. The USERCODE command causes the user-written subroutines to be compiled and linked into a new, temporary version of MSC.Dytran. On most computers, this is automatic. See the *MSC.Dytran Installation and Execution Guide* for details on how it is performed on your computer.

2. If the USERCODE statement is not present, the standard version of MSC.Dytran is used.

3. See Section 3.15 on page 3-74 for details on how to write and use user-written subroutines.

4.4 Executive Control Section

4.4.1 Introduction

Executive Control is not used extensively by MSC.Dytran since, unlike MSC.Nastran, it does not contain an Executive System, and DMAP is not available. It is retained for compatibility with MSC.Nastran.

The Executive Control Section immediately follows the FMS and is terminated by a CEND statement. The Executive Control statements can appear in any order within the Executive Control Section.

A summary of the statements available is given in Section 4.4.2 on page 4-22. Each statement is described in detail in Section 4.4.3 on page 4-22.

4.4.2 Summary

Currently, five Executive Control statements are available:

LIMGEN	Limit for CSEG generation.
LIMLNK	Limit of cross references (LINK) in the data file.
LIMMEM	Limit of indirectly referenced elements, grids, or faces in the data file.
TIME	CPU time limit for the analysis.
CEND	Marks the end of the Executive Control Section.

4.4.3 Executive Control Descriptions

The format of the Executive Control statements is free field. In presenting the general formats for each statement, the following conventions are used:

- Uppercase letters should be typed as shown.
- Lowercase letters indicate that a value or option can be substituted.
- Brackets [ ] give a choice of different options.

The default value is used if the statement is not present. Where you can supply an option, the type heading indicates the type of data you must supply. This can be I (Integer), R (Real), or C (Character). A restriction on the range of the option may also be included. For example, I > 0 indicates that you must supply an integer that is greater than zero, while 0. < R < 1. indicates that you must supply a real number greater than zero and less than one.

CEND

Terminates the Executive Control Section

Marks the end of the Executive Control Section and the beginning of the Case Control Section.

Format and Example

CEND

Remark:

1. If there are no FMS or Executive Control statements, the input file can start directly with the Case Control Section.

LIMGEN

Limit for CSEG Generation

The LIMGEN statement dimensions the internal MSC.Dytran arrays to hold data for CSEG generation.

Format and Example:	**Default**
LIMGEN = value	250,000
LIMGEN = 1000	

Option	**Meaning**	**Type**
value	Maximum number of CSEGs generated by CQUAD/CTRIA, with a thickness of 9999. and PLOAD4 with 9999.	I > 0

Remarks:

1. The actual number of integer words used in the internal arrays is twice the value of LIMGEN.
2. CQUAD/CTRIA with thicknesses of 9999. or PLOAD4 entries with 9999. are used to generate CSEG entries.

LIMLNK

Limit of Cross References

The LIMLNK statement dimensions the internal MSC.Dytran arrays to hold data for the links in the data file.

Format and Example:	**Default**
LIMLNK = value	50,000
LIMLNK = 100	

Option	**Meaning**	**Type**
value	Maximum number of limits in the data file.	

Remarks:

1. The actual number of integer words used for the internal arrays is four times LIMLNK.
2. A link is generated if one field on an input entry refers to another input entry; e.g., the MID field on the property entry refers to a material definition.

LIMMEM

Limit of Indirectly Referenced Elements, Grids, or Faces

The LIMMEM statement dimensions the internal MSC.Dytran arrays to hold data for the indirect references in the Data File.

Format and Example	Default
LIMMEM = value	50,000
LIMMEM = 100	

Option	Meaning	Type
value	Maximum number of indirect references.	I > 0

Remark:

1. The default value is sufficient for most applications.

TIME

Selects the Maximum CPU Time

The TIME statement is used to set the CPU time of an MSC.Dytran analysis.

Format and Example	Default
TIME = time	1 minute
TIME = 1.5	

Option	Meaning	Type
time	The maximum CPU time for the analysis in minutes.	R > 0

Remarks:

1. When the CPU time specified on the TIME statement is used, the analysis terminates. The analysis may be continued by performing a restart, if a restart file is requested at the end of the analysis.
2. It is not possible to specify a maximum I/O time. I/O time is normally insignificant compared to the CPU time for an MSC.Dytran analysis.
3. The time is specified in minutes. Thus, 1.5 is equivalent to 90 seconds, and 480 gives 8 hours.
4. It is advised to use the TIME statement to control CPU time, rather than specifying a time limit for the batch queue or the job. If you do give a job or batch queue limit, make sure it is significantly longer than specified on the TIME statement to ensure that MSC.Dytran terminates normally and does not corrupt the files.

4.5 Case Control Section

4.5.1 Introduction

The Case Control Section of the input file controls the analysis, makes selections from the Bulk Data Section, and determines what results are output and how often. Case Control immediately follows the CEND statement, marking the end of the Executive Control Section, and is terminated by a BEGIN BULK entry or, in the case of a restart, by an ENDDATA entry. The Case Control commands can be in any order within the section. A summary of the commands available is given in Section 4.5.2 on page 4-28. Each command is described in detail in Section 4.5.3 on page 4-30.

4.5.2 Summary

The following Case Control commands are available:

Analysis Control

ENDSTEP	Termination step for the analysis.
ENDTIME	Termination time for the analysis.
CHECK	Data check.

Data Selection

TLOAD	Selects transient loading.
TIC	Selects transient initial conditions.
SPC	Selects single-point constraints.

Output Control

CORDDEF	Defines the moving rectangular coordinate system for deformation output.
SET	Defines lists of entity numbers for use in output requests.
SETC	Defines lists of names for use in output requests.
TITLE	Defines the title of the analysis.

Output Selection – Entity Specification

GRIDS Defines the grid points for which results are to be written to a file.

ELEMENTS Defines the elements for which results are to be written to a file.

RIGIDS Defines the rigid surfaces or MATRIGs for which results are to be written to a file.

GBAGS Defines the gas bags for which results are to be written to a file.

RELS Defines the rigid ellipsoids for which results are to be written to a file.

PLANES Defines the rigid planes for which results are to be written to a file.

MATS Defines the materials for which results are to be written to a file.

CONTS Defines the contact surfaces for which results are to be written to a file.

CSECS Defines the cross sections for which results are to be written to a file.

CPLSURFS Defines the coupling surfaces for which results are to be written to a file.

SUBSURFS Defines the subsurfaces for which results are to be written to a file.

SURFACES Defines the surfaces for which results are to be written to a file.

USASURFS Defines the USA surfaces for which results are to be written to a file.

SGAUGES Defines the surface gauges for which results are to be written to a file.

EBDS Defines the Eulerian boundary conditions for which results are written to a file.

Output Selection – Variable Specification

GPOUT Defines the grid-point data that is written to a file.

ELOUT Defines the element data that is written to a file.

RBOUT Defines the rigid surface, MATRIG or RBE2-FULLRIG data that is written to a file.

GBAGOUT Defines the gas-bag data that is written to a file.

RELOUT Defines the rigid-ellipsoid data that is written to a file.

PLNOUT Defines the rigid planes data that is written to a file.

MATOUT Defines the material data that is written to a file.

CONTOUT Defines the contact surface data that is written to a file.

CSOUT Defines the cross-section data that is written to a file.

CPLSOUT Defines the coupling-surface data that is written to a file.

SUBSOUT	Defines the subsurface data that is written to a file.
SURFOUT	Defines the surface data that is written to a file.
USASOUT	Defines the USA surface data that is written to a file.
SGOUT	Defines the surface gauge data that is written to a file.
EBDOUT	Defines the Eulerian boundary data that is written to a file.

Output Frequency

TIMES	Lists the times at which output is required.
STEPS	Lists the time steps at which output is required.

User-Defined Output

GPEXOUT	Indicates that user subroutines are used for grid point output.
ELEXOUT	Indicates that user subroutines are used for element output.

Input File Control

INCLUDE	Switches data input to another file.

Miscellaneous

PARAM	Parameter specification.

4.5.3 Case Control Descriptions

The format of the Case Control commands is free field. In presenting the general formats for each statement, the following conventions are used:

- Uppercase letters should be typed as shown.
- Lowercase letters indicate that a value or option must be substituted.
- Brackets [ ] give a choice of different options.

The default value is used if the command is not present. Where you need to supply an option, the type heading indicates the type of data you must supply. This can be I (Integer), R (Real), or C (Character). A restriction on the range of the option may also be included. For example, I > 0 indicates that you must supply an integer greater than zero; 0. < R < 1. indicates that you must supply a real number greater than zero and less than one.

CHECK

Data Check

Selects the data checking option.

Format and Example	Default
CHECK = [YES, NO]	See Remark 2.
CHECK = YES	

Option	Meaning	Type
YES	A data check is performed. The analysis runs for two time steps.	C
NO	The analysis is run after the data is read in and checked.	C

Remark:

1. The data check option performs the following:

   a. Reads the input data.

   b. Checks for errors.

   c. Produces printed output.

   d. Runs two time steps.

   e. Writes the model data to the output files.

2. The default is YES for a new analysis and NO for a restart analysis.

CONTOUT

Contact Surface Data to be Output

Indicates the contact surface results that are to be written to an output file.

Format and Example	Default
CONTOUT (logical_file) = var1, var2, var3... CONTOUT (OUTPUT1) = XFORCE, FMAGN	No data is written.

Option	Meaning	Type
logical_file	The logical name of the file to which the contact surface output is written.	C
var_i	Variable name to be output. See Section 3.9.2.9 on page 3-59.	C

Remarks:

1. The contact surfaces for which data is written are specified using the CONTS command. The contact-surface results that can be requested for output are listed in Section 3.9.2.9 on page 3-59.
2. The frequency of the output is controlled using the TIMES or STEPS command.
3. For a description of how to output results, see Section 3.9.1 on page 3-21.
4. Contact surface data can only be written to time-history files. (See the TYPE FMS statement.)
5. Continuation lines are not allowed when using the CONTOUT command. If the CONTOUT command exceeds 80 characters, a second CONTOUT command (with the same logical_file name) can be used as follows:

   CONTOUT (logical_file) = var1, var2
   CONTOUT (logical_file) = var3

6. For a time-history file, the following entities will be written to the file together with the corresponding results:

   Master-Slave Contact:

C < Contact Surface ID > M:	Forces/accelerations on/of the master surface.
C < Contact Surface ID > S:	Forces/accelerations on/of the slave surface.
C < Contact Surface ID > T:	Difference between the forces/accelerations on/of the master and slave surfaces of the contact set.

   Single-Surface Contact:

C < Contact Surface ID > T:	Forces/accelerations on/of the single surface.

CONTS

Contact Surfaces to be Stored

Defines the contact surfaces for which results are to be output to a file.

Format and Example	Default
CONTS (logical_file) = n CONTS(THS) = 14	No contact-surface data is output.

Option	Meaning	Type
logical_file	The logical name of the file to which the contact-surface output is written.	C
n	Number of a SET command. Only data for contact surfaces that appear in the set are output.	I > 0

Remarks:

1. For a description of how to output results, see Section 3.9.1 on page 3-21.
2. The results are specified using the CONTOUT command. The contact surface results that can be requested for output are listed in Section 3.9.2.9 on page 3-59.
3. The frequency of the output is controlled using the TIMES or STEPS command.
4. Contact-surface data can only be written to time-history files. (See the TYPE FMS statement.)
5. For a time-history file, the following entities will be written to the file together with the corresponding results:

   Master-Slave Contact:

C < Contact Surface ID > M:	Forces/accelerations on/of the master surface.
C < Contact Surface ID > S:	Forces/accelerations on/of the slave surface.
C < Contact Surface ID > T:	Difference between the forces/accelerations on/of the master and slave surfaces of the contact set.

   Single-Surface Contact:

C < Contact Surface ID > T:	Forces/accelerations on/of the single surface.

CORDDEF

Coordinate System for Deformation Output

Defines the moving rectangular coordinate system in which the deformations are written to the archive files.

The CORDDEF entry can be added to any output request of TYPE = ARCHIVE. The grid-point locations written to the archive file are the locations in the coordinate system referenced by the CORDDEF entry.

The option is particularly useful when studying the motion of a structure in a moving coordinate system.

Format and Example	Default
CORDDEF(logical_file) = n	Basic system
CORDDEF(MYFILE) = 19	

Option	Meaning	Type
logical_file	The logical name of the file to which output is written.	C
n	Number of a CORDxR entry.	I ≥ 0

Remarks:

1. For a description of how to output results, see Section 3.9.1 on page 3-21.
2. Note that this entry is applicable only to output requests with TYPE = ARCHIVE.

CPLSOUT

Coupling-Surface Data to be Output

Indicates the coupling-surface results to be written to an output file.

Format and Example	Default
CPLSOUT (logical_file) = var1,var2,var3... CPLSOUT (SRF_1) = PRESSURE, CLUMP, FMAT	No data is written

Option	Meaning	Type
logical_file	The logical file name of the file to which coupling-surface output is written	C
var1	Variable name to be output. See section 3.9.2.11 on page 3-59.	C

Remarks:

1. The coupling surfaces for which output is written are specified using the CPLSURFS command. The coupling-surface results that can be requested for output are all the Eulerian element variables defined in Section 3.9.2.2 on page 3-33.
2. The frequency of the output is controlled by the TIMES or the STEPS command.
3. For a description of how to output results, see Section 3.9.1 on page 3-21.
4. Continuation lines are not allowed when using the CPLSOUT command. When the command line exceeds 80 characters, a second CPLSOUT command (with the same logical file name) can be used as follows:

   ```
   CPLSOUT (SRF_1) = vanr, var2
   CPLSOUT (SRF_1) = var3
   ```

5. Coupling-surface data can only be written to archive files. (See the TYPE FMS statement).

CPLSURFS

Coupling Surfaces to be Output

Defines the coupling surfaces for which results are to be output to a file.

Format and Example	Default
CPLSURFS (logical-file) = n CPLSURFS (SRF_1) = 44	No coupling surface is output.

Option	Meaning	Type
logical_file	The logical name of the file to which the coupling-surface output is written.	C
n	Number of a SET command. Only data for coupling surfaces that appear in the set are output.	I ≥ 0

Remarks:

1. For a description of how to ouput results, see Section 3.9.1 on page 3-21.
2. The results written are specified using the CPLSOUT command. Any Eulerian variable can be requested for output on a coupling surface. CPLSURFS can be applied to all Eulerian elements as well as to forging elements.
3. The frequency of output is controlled by the TIMES or STEPS command.
4. Coupling-surface data can only be written to archive files. (See the TYPE FMS statement.)

CSECS

Cross Sections to be Output

Defines the cross sections for which results are to be output to a file.

Format and Example	Default
CSECS (logical_file) = n CSECS (SEC001) = 17	No cross section is output.

Option	Meaning	Type
logical_file	The logical name of the file to which the cross-section output is written.	C
n	Number of a SET command. Only data for cross sections that appear in the set are output	I > 0

Remarks:

1. For a description of how to output results, see Section 3.9.1 on page 3-21.
2. The results written are specified using the CSOUT command. The cross-section results that can be requested for output are listed in Section 3.9.2.10 on page 3-59.
3. The frequency of output is controlled using the TIMES or STEPS command.
4. Cross-section data can only be written to time-history files. (See the TYPE FMS statement.)

CSOUT

Cross-Section Data to be Output

Indicates the cross-section results to be written to an output file.

Format and Example	Default
CSOUT (logical_file) = var1, var2, var3...	No data is written.
CSOUT (SEC001) = XFORCE, FMAGN	

Option	Meaning	Type
logical_file	The logical file name of the file to which the cross-section output is written.	C
vari	Variable name to be output. See Section 3.9.2.10 on page 3-59.	C

Remarks:

1. The cross sections for which output is written are specified using the CSECS command. The cross-section results that can be requested for output are listed in Section 3.9.2.10 on page 3-59.
2. The frequency of the output is controlled using the TIMES or the STEPS command.
3. For a description of how to output results, see Section 3.9.1 on page 3-21.
4. Continuation lines are not allowed when using the CSOUT command. If the command exceeds 80 characters, a second CSOUT command (with the same logical filename) can be used as follows:

   ```
   CSOUT (SEC001) = var1, var2
   CSOUT (SEC001) = var3
   ```

5. Cross-section data can only be written to time-history files. (See the TYPE FMS statement.)

EBDOUT

Eulerian Boundary Data to be Output

Indicates the Eulerian boundary results to be written to an output file.

Format and Example	Default
EBDOUT (logical_file) = var_1, var_2	No data is written.
EBDOUT (OUTPUT1) = MFL, XMOM	

Option	Meaning	Type
logical_file	The logical name of the file to which the Eulerian boundary output is written.	C
var_i	Variable name to be output. See Section 3.9.2.4 on page 3-52.	C

Remarks:

1. The Eulerian boundaries for which data is written are specified using the EBDS command. The Eulerian boundary results that can be requested for output are listed in Section 3.9.2.4 on page 3-52.

2. The frequency of the output is controlled using the TIMES and STEPS commands.

3. For a description of how to output results, see Section 3.9.1 on page 3-21.

4. Eulerian boundary data can only be written to time-history files.

5. Continuation lines are not allowed when using the EBDOUT command. If the EBDOUT command exceeds 80 characters, a second EBDOUT (with the same logical_file name) can be used as follows:

```
EBDOUT (logical_file) = var 1, var 2
EBDOUT (logical_file) = var 3
```

EBDS

Eulerian Boundaries to be Output

Defines the rigid planes for which results to be output to a file.

Format and Example	Default
EBDS (logical_file) = n EBDS (EBD14) = 14	No Eulerian boundary output.

Option	Meaning	Type
logical_file	The logical name of the file to which the Eulerian boundary output is written.	C
n	Number of a SET command. Only data for Eulerian boundaries that appear in the set are output.	I > 0

Remarks:

1. The Eulerian boundary results to be written are specified using the EBDOUT command. The Eulerian boundary results that can be requested for output are listed in Section 3.9.2.4 on page 3-52.
2. The frequency of the output is controlled using the TIMES and STEPS commands.
3. For a description of how to output results, see Section 3.9.1 on page 3-21.
4. Eulerian boundary results can only be written to time-history files.

ELEMENTS

Elements to be Output

Defines the elements for which results are to be output to a file.

Format and Example	Default
ELEMENTS (logical_file) = n ELEMENTS (TH3) = 10	No element data is written.

Option	Meaning	Type
logical_file	The logical name of the file to which the element output is written.	C
n	Number of a SET command. Only data for elements that appear in the set are output.	I > 0

Remarks:

1. For a description of how to output results, see Section 3.9.1 on page 3-21.
2. The element results written are specified using the ELOUT command. The element results that can be output are listed in Section 3.9.2.2 on page 3-33.
3. The frequency of output is controlled using the TIMES and STEPS commands.

ELEXOUT

User-Defined Element Output

Output element results using a user-written subroutine.

Format and Example	Default
ELEXOUT (output_name)	No user output.
ELEXOUT(USEROUT)	

Option	Meaning	Type
output_name	The name with which the subroutine is called.	C

Remarks:

1. At every time or time step specified by the TIMES or STEPS command, a subroutine named EEXOUT is called for each of the elements listed using the ELEMENTS command allowing the user to calculate specific quantities for output.
2. For a description of how to output results, see Section 3.9.1 on page 3-21.
3. For a description of how to use user-written subroutines, see Section 3.15 on page 3-74.
4. The following commands:

```
ELEXOUT (USEROUT)
ELEMENTS (USEROUT) = 10
SET 10 = 101, THRU, 110
TIMES (USEROUT) = 1.0E-3, 2.0E-3
```

cause the subroutine EEXOUT to be called at times 1.0E–3 and 2.0E–3 for elements 101 through 110 with the user-supplied name USEROUT.

ELOUT

Element Data to be Output

Indicates the element results to be written to an output file.

Format and Example	**Default**
ELOUT (logical_file) = var$_1$, var$_2$, var$_3$, . . .	No data is written.
ELOUT (OUTPUT1) = TXX, TYY, TZZ	

Option	**Meaning**	**Type**
logical_file	The logical name of the file to which the element output is written.	C
var$_i$	Variable name to be output. See Section 3.9.2.2 on page 3-33.	C

Remarks:

1. The elements for which data is written are specified using the ELEMENTS command. The element results that can be requested for output are listed in Section 3.9.2.2 on page 3-33.

2. The frequency of output is controlled using the TIMES and STEPS commands.

3. For a description of how to output the results, see Section 3.9.1 on page 3-21.

4. Continuation lines are not allowed when using the ELOUT command. If the ELOUT command exceeds 80 characters, a second ELOUT command (with the same logical_file name) can be used as follows:

```
ELOUT (logical_file) = var 1, var 2
ELOUT (logical_file) = var 3
```

ENDSTEP

Final Time Step

Defines the time-step number at which the analysis terminates.

Format and Example	Default
ENDSTEP=n	See Remark 4.
ENDSTEP=3000	

Option	Meaning	Type
n	The time-step number at which the transient dynamic analysis terminates.	I ≥ 0

Remarks:

1. The RESTART statement can be used to continue a previous analysis. Therefore, you do not need to set ENDSTEP to the final point you want to reach, but instead, to the point at which you want the analysis to stop.
2. Unless you are very sure of what the analysis will do, you should always run the analysis in stages. Then use the RESTART statement to continue the analysis after you have checked how the mesh deforms.
3. The ENDTIME command can be used to terminate the analysis based on time.
4. If ENDTIME is specified, ENDSTEP is set to a large value (9999999).
5. At least one of the two termination criteria must be specified, either ENDSTEP or ENDTIME.

ENDTIME

Analysis Termination Time

Defines the termination time for the analysis.

Format and Example	**Default**
ENDTIME = time	See Remark 4.
ENDTIME = 30.0E-3	

Option	**Meaning**	**Type**
time	The time, in analysis units, at which the transient dynamic analysis terminates.	R ≥ 0

Remarks:

1. The RESTART statement can be used to continue a previous analysis. Therefore, you do not need to set ENDTIME to the final point you want to reach, but instead, to the point at which you want the analysis to stop.
2. Unless you are very sure of what the analysis will do, you should always run the analysis in stages. Then use the RESTART statement to continue the analysis after you have checked how the mesh deforms.
3. The ENDSTEP command can be used to terminate the analysis based on the number of time steps.
4. If ENDSTEP is specified, ENDTIME is set to large value (99999).
5. At least one of the two termination criteria must be specified, either ENDTIME or ENDSTEP.

GBAGOUT

Gas Bag Data to be Output

Indicates the gas-bag results to be written to an output file.

Format and Example	Default
GBAGOUT (logical_file) = var_1, var_2, . . .	Required
GBAGOUT (OUTPUT) = PRESSURE	

Option	Meaning	Type
logical_file	The logical name of the file to which the gas-bag output is written.	C
var_i	Variable name to be output. See Section 3.9.2.8 on page 3-57.	

Remarks:

1. The gas bags, for which data is written, are specified using the GBAGS command. The gas-bag results that can be requested for output are listed in Section 3.9.2.8 on page 3-57.
2. The frequency of the output is controlled using the TIMES and STEPS commands.
3. For a description of how to output results, see Section 3.9.1 on page 3-21.
4. Gas-bag data can only be written to time-history files. (See the TYPE FMS statement.)
5. Continuation lines are not allowed when using the GBAGOUT command. If the GBAGOUT command exceeds 80 characters, a second GBAGOUT command (with the same logical_file name) can be used as follows:

```
GBAGOUT (logical_file) = var 1, var 2
GBAGOUT (logical_file) = var 3
```

GBAGS

Gas Bags to be Output

Defines the gas bags for which results are to be output to a file.

Format and Example	Default
GBAGS (logical_file) = n GBAGS(THG) = 14	No gas-bag data is output.

Option	Meaning	Type
logical_file	The logical name of the file to which the gas-bag output is written.	C
n	Number of a SET command. Only data for gas bags that appear in the set are output.	I ≥ 0

Remarks:

1. For a description of how to output results, see Section 3.9.1 on page 3-21.
2. The results written are specified using the GBAGOUT command. The gas-bag results that can be requested for output are listed in Section 3.9.2.8 on page 3-57.
3. The frequency of output is controlled using the TIMES and STEPS commands.
4. Gas-bag data can only be written to time-history files. (See the TYPE FMS statement.)

GPEXOUT

User-Defined Grid-Point Output

Output grid-point results using a user-written subroutine.

Format and Example	Default
GPEXOUT (output_name) GPEXOUT (DYTRAN_EXT_GP)	No user output.

Option	Meaning	Type
output_name	Name used when subroutine is called.	C

Remarks:

1. At every time or time step specified by the TIMES or STEPS commands, a subroutine called GEXOUT is called for each of the grid points specified using a GRIDS command that allows you to calculate specific quantities for output.

2. For a description of how to output results, see Section 3.9.1 on page 3-21.

3. For a description of how to use user-written subroutines, see Section 3.15 on page 3-74.

4. The following commands:

   ```
   GPEXOUT (DYTRAN_EXT_GP)
   GRIDS (DYTRAN_EXT_GP) = 3
   SET, 3, 1 THRU 35.
   STEPS (DYTRAN_EXT_GP) = 5, 10, 15
   ```

   cause subroutine GEXOUT to be called at time steps 5, 10, and 15 for grid points 1 through 35 with the user-supplied name DYTRAN_EXT_GP.

GPOUT

Grid-Point Data to be Output

Indicates the grid-point results to be written to an output file.

Format and Example	Default
GPOUT (logical_file) = var_1, var_2, var_3, . . . GPOUT (OUTPUT1) XVEL, XFORCE	No data is written.

Option	Meaning	Type
logical_file	The logical name of the file to which the grid-point output is written.	C
var_i	Variable name to be output. See Section 3.9.2.1 on page 3-25.	C

Remarks:

1. The grid points for which data is written are specified using the GRIDS command. The grid-point results that can be requested for output are listed in Section 3.9.2.1 on page 3-25.
2. The frequency of the output is controlled using the TIMES and STEPS commands.
3. For a description of how to output results, see Section 3.9.1 on page 3-21.
4. Continuation lines are not allowed when using the GPOUT command. If the GPOUT command exceeds 80 characters, a second GPOUT command (with the same logical_file name) can be used as follows:

```
GPOUT (logical_file) = var 1, var 2
GPOUT (logical_file) = var 3
```

GRIDS

Grid Points to be Output

Defines the grid points for which results are to be output to a file.

Format and Example	Default
GRIDS (logical_file) = n	No grid-point output.

Option	Meaning	Type
logical_file	The logical name of the file to which the grid-point output is written.	C
n	Number of a SET command. Only data for grid points that appear in the set are output.	I > 0

Remarks:

1. For a description of how to output results, see Section 3.9.1 on page 3-21.
2. The grid-point results to be written are specified using the GPOUT command. The grid-point results that can be requested for output are listed in Section 3.9.2.1 on page 3-25.
3. The frequency of output is controlled using the TIMES and STEPS commands.

INCLUDE

Starts Reading of a New File

Switches reading of the input data to another file. Once that file has been read, processing returns to the original file immediately after the INCLUDE file.

Format and Example	Default
INCLUDE filename	Read ".dat" file.
INCLUDE INPUT.DAT	

Option	Meaning	Type
filename	The name of the new input file to be used. The name must be appropriate to the machine on which MSC.Dytran is executing.	C

Remarks:

1. The file must be present in the working area where MSC.Dytran is executing.
2. BEGIN BULK and ENDDATA may be included in an INCLUDE file.

MATOUT

Material Data to be Output

Indicates the material results to be written to an output file.

Format and Example	Default
MATOUT (logical_file) = var_1, var_2...	No data is written.
MATOUT (OUTPUT1) = XMOM, YMOM	

Option	Meaning	Type
logical_file	The logical name of the file to which the material output is written.	C
var_i	Variable name to be output. See Section 3.9.2.3 on page 3-51.	C

Remarks:

1. The materials for which data is written are specified using the MATS command. The material results that can be requested for output are listed in Section 3.9.2.3 on page 3-51.
2. The frequency of the output is controlled using the TIMES and STEPS commands.
3. For a description of how to output results, see Section 3.9.1 on page 3-21.
4. Material data can only be written to time-history files.
5. Continuation lines are not allowed when using the MATOUT command. If the MATOUT command exceeds 80 characters, a second MATOUT command (with the same logical_file name) can be used as follows:

```
MATOUT (logical_file) = var 1, var 2
MATOUT (logical_file) = var 3
```

MATS

Materials to be Output

Defines the materials for which results are to be output to a file.

Format and Example	Default
MATS (logical_file) = n MATS (MAT19) = 19	No material output.

Option	Meaning	Type
logical_file	The logical name of the file to which the material output is written.	C
n	Number of a SET command. Only data for materials that appear in the set are output.	I > 0

Remarks:

1. For a description of how to output results, see Section 3.9.1 on page 3-21.
2. The material results to be written are specified using the MATOUT command. The material results that can be requested for output are listed in Section 3.9.2.3 on page 3-51.
3. The frequency of output is controlled using the TIMES and STEPS commands.
4. Material results can only be written to time-history files.

PARAM

Parameter Specification

Defines the values for the parameters that are used during the analysis.

Format and Example	Default
PARAM, name, value PARAM, INISTEP, 1.E-7	See Section 4.7 on page -437.

Option	Meaning	Type
name	Parameter name.	C
value	Value associated with name.	I, R, C

Remark:

1. This command is normally used in the Bulk Data Section. A list of parameters that can be set, along with the parameter names and values, is given in Section 4.7 on page -437.

PLANES

Rigid-Planes to be Output

Defines the rigid planes for which results to be output to a file.

Format and Example	Default
PLANES (logical_file) = n PLANES (OUTPUT1) = 170	No rigid-plane output.

Option	Meaning	Type
logical_file	The logical name of the file to which the rigid-plane output is written.	C
n	Number of a SETC command. Only data for rigid planes that appear in the set are output.	I > 0

Remarks:

1. The rigid-planes results to be written are specified using the PLNOUT command. The rigid-planes results that can be requested for output are listed in Section 3.9.2.7 on page 3-56.
2. The frequency of the output is controlled using the TIMES and STEPS commands.
3. For a description of how to output results, see Section 3.9.1 on page 3-21.
4. A SETC is used to enable output for rigid planes obtained from MADYMO.
5. Rigid-plane data can only be written to archive files. See also the TYPE FMS statement.

PLNOUT

Rigid-Plane Data to be Output

Indicates the rigid-plane results to be written to an output file.

Format and Example	Default
PLNOUT (logical_file) = var_1, var_2	No data is written.
PLNOUT (OUTPUT1) = GEOMETRY	

Option	Meaning	Type
logical_file	The logical name of the file to which the rigid-plane output is written.	C
var_i	Variable name to be output. See Section 3.9.2.7 on page 3-56.	C

Remarks:

1. The rigid planes for which data is written are specified using the PLANES command. The rigid-plane results that can be requested for output are listed in Section 3.9.2.7 on page 3-56.
2. The frequency of the output is controlled using the TIMES and STEPS commands.
3. For a description of how to output results, see Section 3.9.1 on page 3-21.
4. The keyword GEOMETRY causes a mesh to be placed on the rigid planes for visualization purposes in the postprocessor.
5. Plane output can only be used for ARCHIVE output requests.
6. Continuation lines are not allowed when using the PLNOUT command. If the PLNOUT command exceeds 80 characters, a second PLNOUT (with the same logical_file name) can be used as follows:

```
PLNOUT (logical_file) = var 1, var 2
PLNOUT (logical_file) = var 3
```

RBOUT

Rigid-Body Data to be Output

Indicates the rigid-body results to be written to an output file.

Format and Example	Default
RBOUT (logical_file) = var_1, var_2	No data is written.
RBOUT (OUTPUT1) = XVEL, YVEL, XAVEL, YAVEL, ZAVEL	

Option	Meaning	Type
logical_file	The logical name of the file to which the rigid-body output is written.	C
var_i	Variable name to be output. See Section 3.9.2.5 on page 3-53.	C

Remarks:

1. The rigid bodies for which data is written are specified using the RIGIDS command. The rigid-body results that can be requested for output are listed in Section 3.9.2.5 on page 3-53.
2. The frequency of the output is controlled using the TIMES and STEPS commands.
3. For a description of how to output results, see Section 3.9.1 on page 3-21.
4. Continuation lines are not allowed when using the RBOUT command. If the RBOUT command exceeds 80 characters, a second RBOUT (with the same logical_file name) can be used as follows:

```
RBOUT (logical_file) = var 1, var 2
RBOUT (logical_file) = var 3
```

RELOUT

Rigid-Ellipsoid Data to be Output

Indicates the rigid-ellipsoid results to be written to an output file.

Format and Example	Default
RELOUT (logical_file) = var_1, var_2	No data is written.
RELOUT (OUTPUT1) = GEOMETRY	

Option	Meaning	Type
logical_file	The logical name of the file to which the rigid-ellipsoid output is written.	C
var_i	Variable name to be output. See Section 3.9.2.6 on page 3-54.	C

Remarks:

1. The rigid ellipsoids for which data is written are specified using the RELS command. The rigid-ellipsoid results that can be requested for output are listed in Section 3.9.2.6 on page 3-54.
2. The frequency of the output is controlled using the TIMES and STEPS commands.
3. For a description of how to output results, see Section 3.9.1 on page 3-21.
4. The keyword GEOMETRY causes a mesh to be placed on the rigid ellipsoids for visualization purposes in the postprocessor. This keyword can be used only with archive files.
5. Continuation lines are not allowed when using the RELOUT command. If the RELOUT command exceeds 80 characters, a second RELOUT command (with the same logical_file name) can be used as follows:

```
RELOUT (logical_file) = var 1, var 2
RELOUT (logical_file) = var 3
```

RELS

Rigid Ellipsoids to be Output

Defines the rigid ellipsoids for which results are to be output to a file.

Format and Example	Default
RELS (logical_file) = n RELS (FILE_REL) = 170	No rigid-ellipsoid output.

Option	Meaning	Type
logical_file	The logical name of the file to which the rigid-ellipsoid output is written.	C
n	Number of a SETC command. Only data for rigid ellipsoids that appear in the set are output.	I > 0

Remarks:

1. For a description of how to output results, see Section 3.9.1 on page 3-21.
2. The rigid-ellipsoid results to be written are specified using the RELOUT command. The rigid-ellipsoid results that can be requested for output are listed in Section 3.9.2.6 on page 3-54.
3. The frequency of output is controlled using the TIMES and STEPS commands.
4. A SETC is used to enable output for rigid ellipsoids obtained from MADYMO or ATB.

RIGIDS

Rigid Bodies to be Output

Defines the rigid bodies for which results are to be output to a file.

Format and Example	Default
RIGIDS (logical_file) = n	No rigid-body output.
RIGIDS (TH5Z) = 32	

Option	Meaning	Type
logical_file	The logical name of the file to which the user output is written.	
n	Number of a SET command. Only data for rigid bodies that appear in the set are output.	I > 0

Remarks:

1. For a description of how to output results, see Section 3.9.1 on page 3-21.
2. The rigid-body results to be written are specified using the RBOUT command. The rigid-body results that can be requested for output are listed in Section 3.9.2.5 on page 3-53.
3. The SET can refer to a RIGID surface (id), a MATRIG (MR<id>), or an RBE2-FULLRIG (FR<id>).
4. The frequency of output is controlled using the TIMES and STEPS commands.

SET

Set Definition

Defines a list of grid points, elements, etc., for which output is required.

Format and Example	Default
SET n = i_1,[, i_2, i_3 THRU i_4 BY i_5]	Required
SET 77 = 5	
SET 88 = 5, 6, 7, 8, 9, 10 THRU 55 BY 3	
15, 16, 77, 78, 79, 100, THRU 300 BY 2	
SET 99 = 1 THRU 100000	
SET 44 = ALLSHQUAD	

Option	Meaning	Type
n	Set number.	I > 0
i_1, i_2 etc.	Element or grid-point number at which the output is requested.	I > 0
i_3 THRU i_4 BY i_5	Output at numbers i_3 to i_4 ($i_4 > i_3$) with an increment of i_5.	I > 0
ALLSHQUAD	Data is output for all entities associated with quadrilateral shell elements or grid points. (CQUAD4)	C
ALLSHTRIA	Data is output for all entities associated with triangular shell elements or grid points. (CRTIA3)	C
ALLMEMTRIA	Data is output for all entities associated with triangular membrane elements or grid points. (CTRIA3)	C
ALLLAGSOLID	Data is output for all entities associated with Lagrangian solid elements or grid points.	C
ALLEULHYDRO	Data is output for all entities associated with hydrodynamic Eulerian elements or grid points.	C
ALLEULSTRENGTH	Data is output for all entities associated with Eulerian elements or grid points with shear strength.	C
ALLDUMQUAD	Data is output for all entities associated with dummy CQUAD4 elements or grid points.	C
ALLDUMTRIA	Data is output for all entities associated with dummy CTRIA3 elements or grid points.	C
ALLMULTIEULHYDRO	Data is output for all entities associated with Eulerian multimaterial elements or grid points.	C

(Continued)

Option	Meaning	Type
ALLELEM1D	Data is output for all entities associated with one-dimensional elements or grid points.	C
ALLELEMENTS	Data is output for all entities associated with all elements.	C
ALLGRIDPOINTS	Data is output for all entities associated with all grid points.	C
ALLCONTACTS	Data is output for all entities associated with all contacts.	C
ALLCSECS	Data is output for all entities associated with all cross sections.	C

Remarks:

1. A SET command may occupy more than one line in the input file. A comma (,) at the end of a line signifies that the next line is a continuation. Commas cannot end a set.
2. The keyword BY does not have to be used when specifying an i_1 THRU i_2 range since the assumed default is 1.

SETC

List of Names

Defines a list of names (character strings) that are used to specify what output is required.

Format and Example	Default
SETC n = name1, name2, name3,...	Required.
SETC 10 = HUB, RIM	

Option	Meaning	Type
n	Set number.	I > 0
name_i	Character string.	C

Remarks:

1. A SETC command may occupy more than one line of the input file. A comma (,) at the end of a line signifies that the next line is a continuation. Commas cannot end a set.
2. This SETC may be referred to from outside the Case Control Section.
3. The length of the character string must be 16 characters or less.
4. The RELS command uses the SETC instead of the normal SET1, enabling the user to specify character strings rather than integers.

SGAUGES

Surface Gauges to be Stored

Defines the surface gauges for which results are to be output to a file.

Format and Example	Default
SGAUGES (logical_file) = n SGAUGES (SG12) = 245	No surface gauge data is output.

Option	Meaning	Type
logical_file	The logical name of the file to which the surface gauge output is written.	C
n	Number of a SET command. Only data for surface gauges that appear in the set are output.	I > 0

Remarks:

1. For a description of how to output results, see Section 3.9.1 on page 3-21.
2. The results are specified using the SGOUT command. The surface gauge results that can be requested for output are listed in Section 3.9.2.15 on page 3-65.
3. The frequency of the output is controlled using the TIMES or STEPS command.
4. Surface gauge data can only be written to time-history files. (See the TYPE FMS statement).

SGOUT

Surface Gauge Data to be Output

Indicates the surface gauge results to be written to an output file.

Format and Example	**Default**
SGOUT (logical_file) = var1, var2, var3...	No data is written.
SGOUT (SG12) = PRESSURE	

Option	**Meaning**	**Type**
logical_file	The logical name of the file to which the surface gauge output is written.	C
var_i	Variable name to be output. See Section 3.9.2.15 on page 3-65.	C

Remarks:

1. The surfaces gauges for which data is written are specified using the SGAUGES command. The surface gauge results that can be requested for output are listed in Section 3.9.2.15 on page 3-65.
2. The frequency of the output is controlled using the TIMES or STEPS command.
3. For a description of how to output results, see Section 3.9.1 on page 3-21.
4. Surface gauge data can only be written to time-history files. (See the TYPE FMS statement.)
5. Continuation lines are not allowed when using the SGOUT command. When the command line exceeds 80 characters, a second SGOUT command (with the same logical_file name) can be used as follows:

```
SGOUT (logical_file) = var1, var2
SGOUT (logical_file) = var3
```

SPC

Single-Point Constraint Set Selection

Selects the single-point constraints to be used.

Format and Example	Default

SPC = n
SPC = 100
	No SPCs will be used.

Option	Meaning	Type
n	Number of a set of SPC, SPC1, SPC2, and SPC3 entries to be used.	I > 0

Remark:

1. Single-point constraints are not used by MSC.Dytran unless they are selected in the Case Control Section.

STEPS

Time Steps at Which Data is Written

Defines the time steps at which data is written to an output file.

Format and Example	Default
STEPS (logical_file) = i_1, [i_2, $i_{3,}$ THRU, $i_{4,}$ BY, i_5]	Required.
STEPS (OUTPUT1) = 0, THRU, END, BY, 100	

Option	Meaning	Type
logical_file	The logical name of the file to which the user output is written.	C
i_1, i_2, etc.	Time steps at which output is required.	I
i_3 ,THRU, i_4 BY, i_5	Time steps i_3 to i_4 using an increment i_5 ($i_4 > i_3$).	I

Remarks:

1. The keyword END can be used to indicate the end of the calculation.
2. The TIMES command can be used instead to control the output using the values of time.
3. For a description of how to output results, see Section 3.9.1 on page 3-21.
4. A list of steps should be in ascending order.

SUBSOUT

Subsurface Data to be Output

Indicates the subsurface results that are to be written to an output file.

Format and Example	Default
SUBSOUT(logical_file)	var1, var2, var3...
SUBSOUT(SUBSURF)	TEMPTURE, MSFR, PRESSURE

Option	Meaning
logical_file	The logical name of the file to which the subsurface output is written.
vari	Variable name to be output.

Remarks:

1. The subsurfaces for which data is written are specified using the SUBSURFS command. The subsurface data that can be requested for output are listed in Section 3.9.2.13 on page 3-63.
2. The frequency of the output is controlled using the TIMES or STEPS command.
3. For a description of how to output results, see Section 3.9.1 on page 3-21.
4. Subsurface output data can only be written to a time history files. (See the TYPE FMS statement.)

SUBSURFS

Subsurfaces to be Stored

Defines the subsurfaces for which results are to be written to a file.

Format and Example	Default
SUBSURFS(logical_file)	n
SUBSURFS(SUBSURF)	14

Option	Meaning
logical_file	The logical name of the file to which the subsurface output is written.
n	Number of a SET command. Only data for GBAG or COUPLING subsurfaces that appear in the set are output.

Remarks:

1. For a description of how to output results, see Section 3.9.1 on page 3-21.
2. The results are specified using the SUBSOUT command. The subsurf data that can be requested for output are listed in Section 3.9.2.13 on page 3-63.
3. The frequency of the output is controlled using the TIMES or STEPS command.
4. Subsurface output data can only be written to a time history files. (See the TYPE FMS statement).
5. The SUBSURFACEs specified in the SET command need to be part of a SURFACE referenced by a COUPLE or GBAG entry.

SURFOUT

Surface Data to be Output

Indicates the surface results that are to be written to an output file.

Format and Example	Default
SURFOUT(logical_file)	var1, var2, var3...
SURFOUT(SURF_1)	TEMPTURE, MSFR, PRESSURE

Option	Meaning
logical_file	The logical name of the file to which the subsurface output is written.
vari	Variable name to be output.

Remarks:

1. The surfaces for which data is written are specified using the SURFACES command. The surface data that can be requested for output are listed in Section 3.9.2.12 on page 3-60.
2. The frequency of the output is controlled using the TIMES or STEPS command.
3. For a description of how to output results, see Section 3.9.1 on page 3-21.
4. Surface output data can only be written to a time history files. (See the TYPE FMS statement.)

SURFACES

Surfaces to be Stored

Defines the surfaces for which results are to be written to a file.

Format and Example	Default
SURFACES(logical_file)	n
SURFACES(SURF_1)	14

Option	Meaning
logical_file	The logical name of the file to which the surface output is written.
n	Number of a SET command. Only data for GBAG or COUPLING surfaces that appear in the set are output.

Remarks:

1. For a description of how to output results, see Section 3.9.1 on page 3-21.
2. The results are specified using the SURFOUT command. The subsurf data that can be requested for output is listed in Section 3.9.2.13 on page 3-63.
3. The frequency of the output is controlled using the TIMES or STEPS command.
4. Surface output data can only be written to a time history files. (See the TYPE FMS statement).
5. The SURFACEs specified in the SET command need to be referenced by a COUPLE or GBAG entry.

TIC

Transient Initial Condition Selection

Selects the transient initial conditions to be used.

Format and Example	Default

TIC = n
TIC = 42
	No initial conditions are applied.

Option	Meaning	Type
n	Number of a set of TIC, TIC1, TIC2, TICGP, or TICEL to be used.	I > 0

Remark:

1. Initial conditions are not used by MSC.Dytran unless they are selected in the Case Control Section.

TIMES

Times at Which Data is Written

Defines the times at which data is to be written to an output file.

Format and Example	**Default**
TIMES (logical_file) = t_1, [t_2, t_3, THRU, t_4, BY, t_5]	Required.
TIMES (OUTPUT1) = 0.0, THRU, 5.0, BY, 0.5, 0.6, THRU, END, BY, 0.03	
TIMES (ARC) = 1.0E-3, 3.0E-3, 7.-3	

Option	Meaning	Type
logical_file	The logical name of the file to which the user output is written.	C
t_1, t_2, etc.	Times at which output is required.	R
t_3, THRU, t_4 BY, t_5	Times t_3 to t_4 using an increment t_5 ($t_4 > t_3$).	R

Remarks:

1. The keyword END can be used to indicate the END of the calculation.
2. The STEPS command can be used instead to control the output using the time-step numbers.
3. For a description of how to output results, see Section 3.9.1 on page 3-21.
4. A list of times should be in ascending order.

TITLE

Output Title

Defines the title for the analysis.

Format and Example	Default
TITLE = string	No title.
TITLE = ANALYSIS - run 13	

Option	Meaning	Type
string	A string of up to 72 alphanumeric characters giving a title for the analysis.	C

Remark:

1. The title is written to the output files for use in postprocessing.

TLOAD

Transient Load Selection

Selects the transient loading to be applied.

Format and Example	Default

TLOAD = n
TLOAD = 2
	No loads are applied.

Option	Meaning	Type
n	Number of a set of TLOAD1 or TLOAD2 entries.	I > 0

Remark:

1. Loads, pressures, flow boundaries, and enforced motion are not used by MSC.Dytran unless they are selected in the Case Control Section.

USASOUT

USA-Surface Data to be Output

Indicates the USA-surface results to be written to an output file.

Format and Example	Default
USASOUT (logical_file) = var1, var2, var3...	No data is written.
USASOUT (USA1) = PRESSURE	

Option	Meaning	Type
logical_file	The logical name of the file to which the USA-surface output is written.	C
var_i	Variable name to be output. See Section 3.9.2.14 on page 3-65.	C

Remarks:

1. The USA-surfaces for which data is written are specified using the USASURFS command. The USA-surface results that can be requested for output are listed in Section 3.9.2.14 on page 3-65.
2. The frequency of the output is controlled using the TIMES or STEPS command.
3. For a description of how to output results, see Section 3.9.1 on page 3-21.
4. USA-surface data can only be written to archive files. (See the TYPE FMS statement.)
5. Continuation lines are not allowed when using the USASOUT command. When the command line exceeds 80 characters, a second USASOUT command (with the same logical_file name) can be used as follows:

```
USASOUT (logical_file) = var1, var2
USASOUT (logical_file) = var3
```

USASURFS

USA Surfaces to be Stored

Defines the USA-surfaces for which results are to be output to a file.

Format and Example	Default
USASURFS (logical_file) = n USASURFS (USA1) = 99	No USA-surface data is output.

Option	Meaning	Type
logical_file	The logical name of the file to which the USA-surface output is written.	C
n	Number of a SET command. Only data for USA-surfaces that appear in the set are output.	I > 0

Remarks:

1. For a description of how to output results, see Section 3.9.1 on page 3-21.
2. The results are specified using the USAOUT command. The USA-surface results that can be requested for output are listed in Section 3.9.2.14 on page 3-65.
3. The frequency of the output is controlled using the TIMES or STEPS command.
4. USA-surface data can only be written to archive files. (See the TYPE FMS statement.)

4.6 Bulk Data Section

4.6.1 Introduction

The Bulk Data Section of the input file contains all the data to fully describe the analysis model, including the geometry, topology, constraints, and loading. This section must begin with a BEGIN BULK entry. Thereafter, entries can appear in any order except that continuation lines must follow the entry from which they are referenced. Entries can be numbered in any manner that is convenient. Gaps in the numbering are allowed. The input file must finish with an ENDDATA entry.

Many of the entries are the same as those used for MSC.Nastran. However, sometimes not all the fields are used for MSC.Dytran. If data occurs in the unused fields, a User Warning Message is issued and the excess data is ignored (see Section 4.2 on page 4-3). Similarly, any MSC.Nastran entry that is not used by MSC.Dytran is ignored.

4.6.2 Format of Bulk Data Entries

A Bulk Data entry consists of one or more lines in the input file. The first line starts with a mnemonic that identifies the entry and is called the parent entry. Any other lines are called continuations. Each line can be in free or fixed format. In free format, the fields can appear anywhere on the line and are separated by commas or spaces. With fixed format, each field must be located in a set part of the line. There are two types of fixed format: small and large. Small format consists of ten fields, each of which has eight characters. The entire entry is defined on a single line of the input file. Large format splits the entry so that it occupies two lines of the input file. Each line consists of one field of eight characters, four fields of sixteen characters, and one of eight characters. Small- and large-format entries must be in fixed format, that is, the data must be entirely within the columns that make up the field.

Free- and fixed-field lines can be freely mixed in the input file so, for example, a fixed-format entry can have a free-format continuation, or vice versa.

The first field of each Bulk Data entry contains a mnemonic that identifies the type of entry. Fields 2 through 9 contain data, while field 10 is used for a continuation identifier or for user identification if there are no continuation lines. The mnemonic must start in column one of the first field.

Fields 2 through 9 are for data items. The only limitations on data items are that they cannot have embedded blanks and must be of the proper type, i.e., blank, Integer, Real, or Character. A blank is interpreted as a real zero or integer zero as required. Real numbers may be encoded in various ways. For example, the real number 7.0 may be encoded as 7.0, .7E1, 0.7+1, 70.–1, 7+0, 7, etc. Character data values consist of one to eight alphanumeric characters, the first of which must be alphabetic.

Normally, field 10 is reserved for optional user identification. However, in the case of continuation lines, field 10 (except for the first character, which is ignored) is used in conjunction with field 1 of the continuation as an identifier. Some entries do not require continuations, in which case any data placed in field 10 is regarded as a user comment and is ignored. The continuation line contains the symbol + in column one followed by the same identifier (ignoring the first character) that appeared in field 10 of the entry that is being continued. Character values used as continuation mnemonics cannot contain the symbols *, $ or ”,”. Continuation lines must immediately follow their parent entry. Continuation mnemonics need not be unique.

Free-Field Format

With free-field input, the position of the data items on the line is irrelevant. The mnemonic must be followed by a comma; thereafter, individual data items can be separated by spaces or commas. For example:

```
GRID, 7, 0, 0.0, 1.0, 3.7569
GRID, 7 0 0.0 1.0 3.7569
```

Free-field entries must start in column one; data fields can consist of any number of characters as long as the whole entry fits on the line.

A field may be left blank by entering two commas, with or without spaces between them:

```
GRID, 7,, 0.0, 1.0, 3.7569
```

Only those fields containing data need be entered. All the extra fields are given their default values. In the example above, only six fields have been entered, so the last four are set to the default.

Small-Field, Fixed-Format Entry

1	2	3	4	5	6	7	8	9	10
1a	2	3	4	5	6	7	8	9	10a
8	8	8	8	8	8	8	8	8	8

The small-field, fixed-format entry consists of a single line in the input file containing 80 characters and comprising 10 fields, each of which has eight characters. The data in each field must lie completely within the designated columns.

Large-Field, Fixed-Format Entry

The small-field format should be adequate for most applications. Occasionally, however, the input is generated by another computer program or is available in a form where a wider field is desirable. For these cases, the larger field format with a 16-character data field is provided. Two lines of the input file are used as indicated below:

1a	2	3	4	5	10a
8	16	16	16	16	8

1b	6	7	8	9	10b
8	16	16	16	16	8

The large field format is denoted by placing the symbol * after the mnemonic in field 1a and some unique character configuration in the last seven columns in Field 10a. The second line contains the symbol * in column one followed by the same seven characters that appeared after column 73 in field 10a of the first line. The second line may, in turn, be used to point to a large or small field continuation line, depending on whether the continuation line contains the symbol * (for a large field) or the symbol + (for a small field) in column one. The use of multiple and large field lines is illustrated in the following examples:

Small-Field Entry with Small-Field Continuation

1	2	3	4	5	6	7	8	9	10
TYPE									+CONT
+CONT									

Large-Field Entry

TYPE*									*CONT
*CONT									

Large-Field Entry with Large-Field Continuation

TYPE*					+CONT1
*CONT1					+CONT2
*CONT2					+CONT3
*CONT3					

Large-Field Entry Followed by a Small-Field Continuation and a Large-Field Continuation

TYPE*									+CONT1
*CONT1									+CONT2
+CONT2									+CONT3
*CONT3									+CONT4
+CONT4									

Small-Field Entry with Large-Field Continuation

TYPE*					+CONT1
+CONT1					+CONT2
*CONT2					

4.6.3 Summary

This section contains a summary of all the Bulk Data entries under the following subsections:

- Geometry.
- Lagrangian and Eulerian Elements.
- Constitutive Models.
- Rigid Bodies.
- Lagrangian Constraints.
- Lagrangian Loading.
- Eulerian Loading and Constraints.
- Euler/Lagrange Coupling.
- Miscellaneous.

4.6.3.1 Geometry

Grid Points

GRID	Grid-point location, coordinate system selection.
GRDSET	Default options for GRID entries.
GROFFS	Grid-point offset in the local coordinate system.
CONM2	Concentrated grid-point mass and/or inertia.

Coordinate Systems

CORD1R, CORD2R	Rectangular coordinate system definition.
CORD1C, CORD2C	Cylindrical coordinate system definition.
CORD1S, CORD2S	Spherical coordinate system definition.
CORD3R	Moving rectangular coordinate system definition, form 1.
CORD4R	Moving rectangular coordinate system definition, form 2.
CORDROT	Corotational frame definition.

Mesh Generation

MESH	Mesh generator.

4.6.3.2 Lagrangian Elements

Solid Elements

CHEXA	Connection definition for brick element with eight grid points.
CPENTA	Connection definition for wedge element with six grid points.
CTETRA	Connection definition for tetrahedron element with four grid points.
PSOLID	Property definition for CHEXA, CPENTA, CTETRA.

Surface Elements

CQUAD4	Connection definition for a quadrilateral shell element with four grid points.
CTRIA3	Connection definition for a triangular shell or membrane element with three grid points.
PSHELL	Property definition for CQUAD4 and CTRIA3.
PSHELL1	Complex property definition for CQUAD4 and CTRIA3.
PCOMP	Layered composite element property.
PCOMPA	Additional data for layered composite element property.

1-D Elements

CBAR	Connection definition for a line element with two grid points.
CBEAM	Connection definition for a line element with two grid points.
CROD	Connection definition for a line element with two grid points.
CDAMP1	Connection definition for a scalar damper element with two grid points.
CDAMP2	Connection definition for a linear damper element with two grid points.
CELAS1	Connection definition for a scalar spring element with two grid points.
CELAS2	Connection and property definition for a scalar spring element with two grid points.
CSPR	Connection definition for spring element with two grid points.
CVISC	Connection definition for a viscous damper element with two grid points.
PBAR	Property definition for a CBAR element.
PBEAM	Property definition for CBAR and CBEAM.
PBEAM1	Complex property definition for CBAR and CBEAM.
PBELT	Property definition for a belt element, defined by a CROD.
PDAMP	Property definition for CDAMP1 and CDAMP2.
PELAS	Property definition for CELASn.
PELASEX	Property definition for CELASn with user subroutines.
PROD	Property definition for CROD.
PSPR	Property definition for CSPR.

PSPR1	Property definition for nonlinear CSPR.
PSPREX	Property definition for CSPR with user subroutines.
PVISC	Property definition for CVISC.
PVISC1	Property definition for nonlinear CVISC.
PVISCEX	Property definition for CVISC with user subroutines.
PWELD	Property definition for spotwelds (using CROD).

4.6.3.3 Eulerian Elements

Solid Elements

CHEXA	Connection definition for a brick element with eight grid points.
CPENTA	Connection definition for a wedge element with six grid points.
CTETRA	Connection definition for a tetrahedral element with four grid points.
PEULER	Property definition for CHEXA, CPENTA, CTETRA.
PEULER1	Property definition for CHEXA, CPENTA, CTETRA defining geometrical regions.

4.6.3.4 Constitutive Models

DMAT	General constitutive model.
DMATEL	Isotropic elastic material properties.
DMATEP	Elastic or elastoplastic material properties.
DMATOR	Orthotropic material properties.
DYMAT14	Soil and crushable foam material properties.
DYMAT24	Piecewise linear plasticity material properties.
DYMAT26	Orthotropic crushable material properties.
FOAM1	Crushable foam material properties.
FOAM2	Crushable foam material properties.
MAT1	Linear-isotropic material properties.
MAT8	Orthotropic elastic material properties.

MAT8A	Failure properties for orthotropic material properties.
RUBBER1	Mooney-Rivlin model for rubber-like materials.
SHEETMAT	Anisotropic plastic material for sheet metal.

Yield Models

YLDHY	Hydrodynamic yield properties.
YLDVM	von Mises yield properties.
YLDJC	Johnson-Cook yield properties.
YLDMC	Mohr-Coulomb yield properties.
YLDEX	User-defined yield properties.

Shear Models

SHREL	Elastic shear properties.
SHRLVE	Isotropic linear viscoelastic shear properties.

Equations of State

EOSPOL	Polynomial equation of state.
EOSJWL	JWL explosive equation of state.
EOSGAM	Gamma law equation of state.
EOSTAIT	Equation of state based on Tait model.
EXEOS	User-defined equations of state.

Detonation Models

DETSPH	Spherical detonation wave.

Failure Models

FAILEST	Maximum equivalent stress and minimum time-step failure model.
FAILEX	User-specified failure model.
FAILEX1	Extended user-specified failure model.
FAILMES	Maximum equivalent stress failure model.

FAILMPS	Maximum plastic strain failure model.
FAILPRS	Maximum pressure failure model.
FAILSDT	Maximum plastic strain and minimum time-step failure model.

Spallation Models

PMINC	Constant spallation pressure properties.

4.6.3.5 Rigid Bodies

MATRIG	Rigid-body properties.
RBE2	Rigid-body element.
RELEX	MADYMO or ATB ellipsoid to be used with MSC.Dytran.
RPLEX	MADYMO planes to be used in MSC.Dytran.
RELLIPS	Analytical rigid ellipsoid.
RIGID	Rigid-body properties.
SURFACE	Geometry of a rigid body.

4.6.3.6 ATB Interface

ATBACC	Acceleration field applied to ATB segments.
ATBJNT	Interface to ATB joints.
ATBSEG	Interface to ATB segments.
ATBSEGCREATE	Create grid points and elements for ATBSEG.

4.6.3.7 Lagrangian Constraints

Single-Point Constraints

GRDSET	Includes the default for single-point constraints on the GRID entry.
GRID	Includes the single-point constraint definition (permanent SPCs).
SPC	Single-point constraint.
SPC1	Single-point constraint.

SPC2	Rotational velocity constraint.
SPC3	Single-point constraint in the local coordinate system.

Contact Surfaces

CONTACT	Defines contact surfaces.
CONTINI	User-defined initialization of contact state between two subsurfaces.
CONTREL	Defines rigid-ellipsoid contact with Lagrangian grid points or rigid bodies.
SURFACE	Defines a multifaceted surface.
SUBSURF	Defines a multifaceted subsurface.
CSEG	Defines segments of a surface.
CFACE	Defines segments of a surface.
CFACE1	Defines segments of a surface.

Connections

JOIN	Defines a join of grid points of different types.
BJOIN	Defines a breakable join of six DOF grid points.
KJOIN	Defines the kinematic join of shell and solid grid points.
RCONN	Defines a rigid connection.
RCONREL	Defines a connection with rigid ellipsoids.
RJCYL	Cylindrical-joint constraint between rigid bodies.
RJPLA	Planar-joint constraint between rigid bodies.
RJREV	Revolute-joint constraint between rigid bodies.
RJSPH	Spherical-joint constraint between rigid bodies.
RJTRA	Translational-joint constraint between rigid bodies.
RJUNI	Universal-joint constraint between rigid bodies.
RJSTIFF	Rigid-joint stiffness.

Rigid Walls

WALL	Defines rigid walls.

Rigid Body Constraints

RBC3	Rigid-body constraint.
FORCE	Concentrated load or velocity.
MOMENT	Concentrated moment or enforced motion.

4.6.3.8 Lagrangian Loading

Transient Loading

TLOAD1	Defines the transient load.
TLOAD2	Defines the transient time-varying load.
DAREA	Defines the position and scale factor of a concentrated load.
FORCE	Defines the position and scale factor of a concentrated force.
FORCE1	Defines a follower force, form 1.
FORCE2	Defines a follower force, form 2.
MOMENT	Defines the position and scale factor of a concentrated moment.
MOMENT1	Defines a follower moment, form 1.
MOMENT2	Defines a follower moment, form 2.
PLOAD	Defines the position and scale factor of a pressure load.
PLOAD4	Defines the position and scale factor of a pressure load.
RFORCE	Defines the centrifugal load.
GRAV	Defines the gravitational load.

Enforced Motion

TLOAD1	Defines the transient enforced motion.
TLOAD2	Defines the transient time-varying enforced motion.
DAREA	Defines the direction and scale factor of motion.
FORCE	Defines the direction and scale factor of motion.
FORCE3	Defines the direction and scale factor of motion in local coordinate systems.

FORCEEX	Defines the user-specified enforced motion.
MOMENT	Defines the direction and scale factor of motion.

Initial Conditions

TIC	Defines the transient initial velocities of grid points.
TIC1	Defines the transient initial velocities of grid points.
TIC2	Defines the initial rotational velocity field.
TICEL	Defines the transient initial conditions of elements.
TICGP	Defines the transient initial conditions of grid points.

4.6.3.9 Eulerian Loading and Constraints

Single-Point Constraints

ALEGRID	Defines the motion of Eulerian grid points.
SPC	Single-point constraint.
SPC1	Single-point constraint.
SPC2	Rotational velocity constraint.
SPC3	Single-point constraint.

Flow Boundary

TLOAD1	Defines the transient load.
FLOW	Defines the flow boundary.
FLOWEX	Defines the user-specified flow boundary.
FLOWDEF	Defines the free Eulerian faces to be the flow boundary by default.
PORFLOW	Defines a porous flow boundary.
CSEG	Defines the face to which the flow boundary is applied.
CFACE	Defines the face to which the flow boundary is applied.
CFACE1	Defines the face to which the flow boundary is applied.

Wall

WALLET	Defines a wall for Eulerian material flow.

Gravity

GRAV	Defines the gravitational load.

Initial Conditions

TIC2	Defines the initial rotational grid-point velocities.
TICGP	Defines the transient initial grid-point velocities.
TICEL	Defines the transient initial condition for elements.
TICEUL	Defines the transient initial conditions for Eulerian elements.
TICVAL	Defines the transient initial conditions.
CYLINDER, SPHERE	Defines the geometrical shapes.

4.6.3.10 Lagrangian Loading and Constraints

PLOADEX	User-defined pressure load.

4.6.3.11 Euler/Lagrange Coupling

COUP1FL	Defines the surrounding variables when a segment of a coupling surface fails.
COUP1INT	Defines the interaction between two coupling surfaces.
COUPLE	Defines the general coupling between the Eulerian and Lagrangian meshes.
COUPLE1	Defines the general coupling between the Roe solver for single hydro materials and Lagrangian structures.
COUOPT	Defines the coupling options.
COUPOR	Defines the coupling surface or subsurface porosity.
ALE	Defines the arbitrary Lagrange-Euler (ALE) coupling.
GBAG	Gas-bag pressure definition.
GBAGC	Gas-bag connection.
GBAGCOU	General coupling to gas-bag switch.
GBAGEX	User-specified gas-bag pressure.

SURFACE	Defines the coupling surface.
SUBSURF	Defines the subsurface.

4.6.3.12 Miscellaneous

Comments

$	For inserting comments in Bulk Data Section.

Parameters

PARAM	Specifies values for the parameters used in the solution.

Tabular Input

TABLED1	Tabular functions for loads, properties, materials, etc.
TABLED2	Two-dimensional table for yield function definition in forging materials.
TABLEEX	User-defined analytical function for loads, properties, materials, etc.

Sets

SET1	Sets of numbers for use by other entries.
SETC	Sets of names for use by other entries.

Solution Control

ACTIVE	Activates or deactivates elements and interaction.
VISCDMP	Defines dynamic relaxation factors for a damping.

Output

SECTION	Cross section.

Prestress Analysis

NASINIT	Defines the prestress analysis logistics.

Input File Control

INCLUDE	Switches data input to another file.

Bulk Data Control

BEGIN BULK	Marks the end of the Case Control and the beginning of Bulk Data.
ENDDATA	Marks the end of the input data.
CFACE	Face of an element.

4.6.4 Bulk Data Descriptions

This section describes the format of each Bulk Data entry and the contents of each field on the entry. The "Type" column indicates the type of data in the field. This can be I (Integer), R (Real), or C (Character). In addition, there may be limits on the value that can be entered in the field. For example, I > 0 indicates that you must supply an integer with a value greater than zero. The value limitation 0. < R ≤ 1. indicates that you must supply a real number greater than zero and less than or equal to one.

The "Default" column indicates the value that is used if the field is left blank. If the word "Required" appears, there is no default and you must supply a value.

Anything that appears after a $ on a line is treated as a comment and is ignored. If a $ appears as the first character, the entire line is a comment.

Format and Example:

1	2	3	4	5	6	7	8	9	10
$ followed by any characters on the rest of the line									
$ THE WHOLE LINE IS A COMMENT.									
GRID	1		0.0	10.0	130.0	$ THE REST OF THE LINE IS A COMMENT.			

Remark:

1. If a comment is placed in fields which would otherwise contain data, the data in those fields is given the fields’ default values.

BEGIN BULK

The Beginning of the Bulk Data

Marks the end of the Case Control Section and the beginning of the Bulk Data Section in the input file.

Format and Example:

BEGIN BULK

Remark:

1. A BEGIN BULK entry must always be present.

ACTIVE

Activate Elements and Interaction

Allows you to activate parts of the program for a part of the problem time only.

Format and Example:

1	2	3	4	5	6	7	8	9	10
ACTIVE	ID	TYPE	TYPEV						+CONT1
ACTIVE	3	INTERACT	COUPLE						+CONT1

+CONT1	TIME	TIMEV							
+CONT1	TABLE	1							

Field	Contents		Type	Default
ID	Unique active number.		I > 0	Required
TYPE	Type of activity switch.		C	Required
	ELEMENT	Switches are for the element type as defined under TYPEV.		
	INTERACT	Switches are for an algorithm defining the interactions between different parts of the model. The type of algorithm is defined under TYPEV.		
	RIGID	Switches are for rigid entities as defined under TYPEV.		
TYPEV	Depends on the value of TYPE:		C	Required
	<u>TYPE</u>	<u>TYPEV</u>		
	ELEMENT	SHTRIA SHQUAD MEMTRIA DUMTRIA DUMQUAD LAGSOLID EULHYDRO EULSTRENGTH MULTIEULHYDRO ELEM1D		
	INTERACT	CONTACT COUPLE GBAG		
	RIGID	SURFACE		

(Continued)

ACTIVE

Activate Elements and Interaction

Field	Contents		Type	Default
ACTIVE	Activate Elements and Interaction			
TIME	Type specification for switches.		C	Required
	TABLE	Part will be switched on and off, depending on the y-value of the table with ID as specified in TIMEV. The x-value of the table represents the time, the y-value means:		
	ON:	$y > 0$.		
	OFF:	$y \leq 0$.		
TIMEV	Number of a TABLED1 or TABLEEX.		$I > 0$	Required

Remarks:

1. The default is all parts of the program are active at all times.
2. For CONTACT an activity switch is set on the entry itself. These settings overrule settings on the ACTIVE entry.
3. The active option for multimaterial with shear strength is activated by using TYPEV = MULTIEULHYDRO.
4. For COUPLE and EULHYDRO in combination with PARAM,LIMITER,ROE , an activity switch is set on the COUPLE1 entry. These settings overrule the settings on the ACTIVE entry.

Defines the surfaces of an ALE interface.

Format and Example:

1	2	3	4	5	6	7	8	9	10
ALE	AID	SIDLG	SIDEU						
ALE	32	3	5						

Field	Contents	Type	Default
AID	Unique ALE interface number.	I > 0	Required
SIDLG	Number of a SURFACE entry that defines the Lagrangian part of the ALE interface.	I > 0	Required
SIDEU	Number of a SURFACE entry that defines the Eulerian part of the ALE interface.	I > 0	Required

Remarks:

1. SIDLG and SIDEU must reference the SID of a SURFACE entry.
2. The Eulerian and Lagrangian SURFACEs must have a one-to-one correspondence. This means that the Eulerian and Lagrangian grid points in the SURFACEs must coincide in physical but not in logical space.
3. The tolerance used in finding coinciding SURFACE nodes is defined by the parameter ALETOL.
4. ALE is not applicable in combination with the single material Euler solver with a full-stress tensor.

ALEGRID

Eulerian Grid-Point Motion Definition

Definition of ALE motion for Eulerian grid points.

Format and Example:

1	2	3	4	5	6	7	8	9	10
ALEGRID	AID	MINCUT	MAXCUT	TYPE	WEIGHT	NAME			+CONT1
ALEGRID	28	0.	1.	STANDARD	COMPUTED				+CONT1

+CONT1	G1	G2	THRU	G3	BY	G4	-etc.-		
+CONT1	1	2	THRU	15	BY	3			

Field	Contents	Type	Default
AID	ALEGRID number.	I > 0	Required
MINCUT	See Remark 1.	R	0
MAXCUT	See Remark 1.	R	1.E20
TYPE	Indicates the type of motion. (See Remark 2.)	C	SPECIAL
	STANDARD		
	FREE		
	FIXED		
	FLOW		
	SPECIAL		
	USER		
WEIGHT	Method of calculating weight factors. (See Remark 6.)	C	COMPUTED
	EQUAL		
	COMPUTED		
NAME	Name of the user defined motion prescription.	C	None
G1,G2...	Grid points to which the motion applies. THRU indicates the range, while BY allows an increment to be used within this range.	I > 0	Required

(Continued)

Remarks:

1. The MINCUT and MAXCUT parameters define the minimum and maximum allowable grid-point velocity of ALE grid points. Usually the defaults are sufficient.

$$\vec{u}_g = max\left(\text{MINCUT}\frac{\Delta q}{\Delta t}, \left|\vec{u}_g\right|\right) \cdot sign(\vec{u}_g)$$

$$\vec{u}_g = min\left(\text{MAXCUT}\frac{\Delta q}{\Delta t}, \left|\vec{u}_g\right|\right) \cdot sign(\vec{u}_g)$$

where Δq is the element characteristic dimension and *Dt* is the time step.

2. The TYPE definition causes the grid-point motion algorithm to define grid-point velocities as follows:

STANDARD: Each grid point moves to the center of its neighbors.

FREE: The grid points that are defined as FREE move as on a free surface.

The grid-point velocity becomes

$$\vec{u}_g = \vec{u}_{g_{tentative}} + [(\vec{u}_{fs} - \vec{u}_{g_{tentative}}) \cdot \vec{n}] \cdot \vec{n}$$

where $\vec{n}$ is the normal to the free surface. $\vec{u}_{fs}$ is the free-surface velocity defined as:

$$\vec{u}_{fs} = \frac{\sum\vec{v}_i \\ i = 1}{N}$$

with $\vec{v}_i$ the material velocity of the elements connected to the grid point. $\vec{u}_{g_{tentative}}$ is the tentative grid-point velocity.

FIXED: Grid points that are defined as FIXED move as on a fixed wall.

The grid-point velocity becomes

$$\vec{u}_g = u_{g_{tentative}} - (u_{g_{tentative}} \cdot \vec{n} \cdot \vec{n})$$

where $\vec{n}$ is the normal to the wall.

(Continued)

FLOW: Grid points move as on a flow boundary.

The grid-point velocity becomes

$$\vec{u}_g = \vec{u}_{g_{int}} + [(\vec{u}_{g_{tentative}} - \vec{u}_{g_{int}}) \cdot \vec{t}] \cdot \vec{t}$$

where $\vec{g}_{int}$ is the grid-point velocity of the closest internal grid point. The vector tangent to the flow boundary is given by $\vec{t}$.

SPECIAL: MSC.Dytran searches the grid points defined on the ALEGRID entry. It detects which surface boundary condition the grid points are part of. The grid-point motion is corrected correspondingly.

USER: The grid-point motion is defined via a user-written subroutine, EXALE. The name that is defined in the NAME field can be used to distinguish different motion prescriptions in the user subroutine.

3. More than one ALEGRID entry can occur in input, with each one having a different type definition. All ALEGRID entries that have the same AID will be merged into one definition. This requires a consistent definition with respect to the options of all of the ALEGRID entries that have the same AID.
4. The number of relaxation iterations for the grid-point displacement is one by default but can be changed using PARAM,ALEITR.
5. There can be as many continuation lines as necessary.
6. The weight factors determine the grid-point motion. If the option is set to COMPUTED (default), MSC.Dytran computes the weight factors based on geometrical considerations. If the option is set to EQUAL, all weight factors are set to a constant. The latter is automatically done when a local distortion of the Eulerian mesh is encountered that does not allow for the computation of the weight factors.
7. In case the TYPE field is set to USER, all other fields will be ignored, except the NAME field.
8. For a description of user-written subroutines, see Section 3.15 on page 3-74.

ALEGRID1

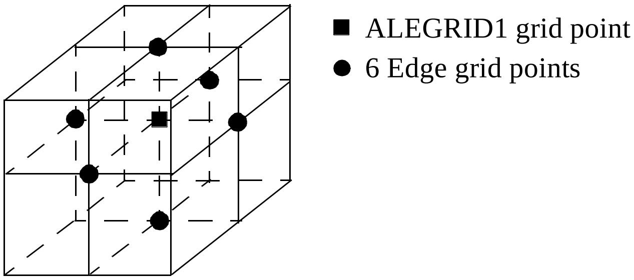
Eulerian Grid-Point Motion Definition

Definition of ALE motion for Eulerian grid points.

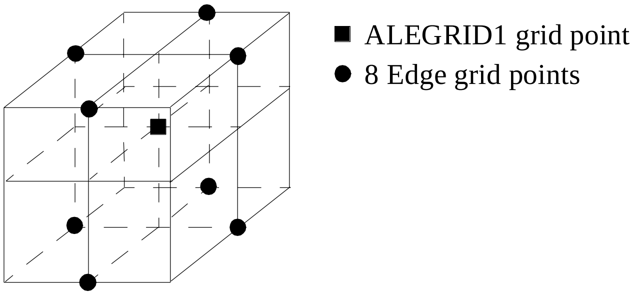
Format and Example:

1	2	3	4	5	6	7	8	9	10
ALEGRID1	AID	SID	EDGE	FDIAG	EDIAG				+CONT1
ALEGRID1	28	11	YES	NO	NO				+CONT1

Field	Contents	Type	Default
AID	ALEGRID1 number.	I > 0	Required
SID	Number of SET1 grid point entries (see Remark 3).	I > 0	Required
EDGE	YES/NO Specifies if the neighboring grid points along the edges need to be taken into account for the mesh motion.	C	YES



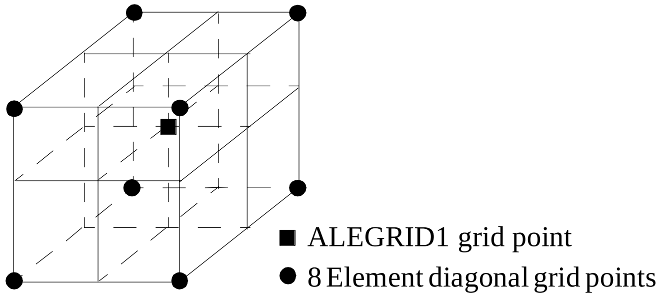
Field	Contents	Type	Default
FDIAG	YES/NO Specifies if the neighboring grid points along the face diagonals need to be taken into account for the mesh motion.	C	NO



Adding these nodes will increase the required amount of memory to store the data, but might improve the motion of the mesh.

(Continued)

Field	Contents	Type	Default
EDIAG	YES/NO Specifies if the neighboring grid points along the element diagonals need to be taken into account for the mesh motion.	C	NO



Remarks:

1. The ALEGRID1 mesh motion algorithm has the following features:

   a. No limit on the number of neighboring grid points exists. This is in contrast to the ALEGRID algorithm where a limit of eight neighboring nodes exists.

   b. The velocity of the grid points is based on a changing weight factor, which is calculated accordingly:

$$u_g = \frac{\Sigma\langle \Delta l_{g_i} \times u_{g_i}\rangle}{\Sigma \Delta l_{g_i}}$$

   where u_g = the velocity component of the Eulerian grid point

   u_{g_i} = the velocity component of the neighbor grid points *i*

   Δl_{g_i} = the distance between the Eulerian grid point and the neighbor grid points *i*

2. When the Lagrangian mesh is moving very fast it can happen that the Eulerian mesh is not properly following the structure, and Eulerian elements get distorted.

   The mesh motion can be improved by increasing the value of PARAM,ALEITR. Multiple iterations per time step will be performed to determine the grid point velocities.

3. Multiple SET1 entries with the same SID are automatically concatenated.

4. A combination of ALEGRID and ALEGRID1 entries in one analysis model is not allowed.

ATBACC

Acceleration Field Applied to ATB Segments

Defines an acceleration field that will be applied to ATB segments.

Format and Example:

1	2	3	4	5	6	7	8	9	10
ATBACC	LID		SCALE	NX	NY	NZ			+CONT1
ATBACC	32		386.088	1.0	0.0	0.0			+CONT1

+CONT1	NAME1	NAME2	NAME3	NAME4	NAME5	NAME6	NAME7	-etc.-	
+CONT1	LT	MT	UT	N	H	RUL	RLL		

Field	Contents	Type	Default
LID	Number of a set of loads.	I > 0	Required
SCALE	ATBACC scale factor.	R ≥ 0.0	1.0
NX, NY, NZ	Components of gravity vector. At least one component must be nonzero.	R ≥ 0.0	0.0
NAMEi	Name of an ATB segment as given in the first field of a B.2 entry in the ATB input file.	C	Required

Remarks:

1. The acceleration *a(t)* is defined as:

$$a(t) = T(t) * \mathrm{SCALE} * N$$

where SCALE is the acceleration scale factor; *N* is the vector defined by NX, NY, and NZ; *T(t)* is the value interpolated at time *t* from the table referenced by the TLOADn entry.

2. LID must be referenced by a TLOADn entry.

3. The type field on the TLOADn entry must be set to zero.

4. More than one ATBACC acceleration field can be defined per problem.

5. This acceleration field is intended to apply a crash pulse to ATB segments that define a crash dummy. The acceleration is multiplied by the mass of the segment and the resulting force is added as an external force.

6. To compare the accelerations of the ATB segments to experiments, the crash pulse needs to be subtracted from the total acceleration. The acceleration of the segments as defined on the H1 entries in the ATB input file are automatically corrected.

ATBJNT

Interface to ATB Joints

This entry can only be used together with the ATBSEG entries that this joint connects. The ATBSEG entries overwrite the position and orientation of the ATB segments as specified in the ATB input file. The ATBJNT entry can be used to create a Bulk Data file containing elements and grid points to visualize the ATB segment together with its joints. This visualization of the joints makes it possible to position the ATB model in any available preprocessor. See also PARAM,ATBSEGCREATE.

Format and Example:

1	2	3	4	5	6	7	8	9	10
ATBJNT	ID	NAME							+CONT1
ATBJNT	1	HN							+CONT1

+CONT1	G0	G1	G2	G3	EID1	EID2	EID3		+CONT2
+CONT1	1010	1011	1012	1013	1004	1005	1006		+CONT2

+CONT2	G4	G5	G6	G7	EID4	EID5	EID6		
+CONT2	2010	2011	2012	2013	2004	2005	2006		

Field	Contents	Type	Default
ID	Unique ATBJNT number.	I > 0	Required
NAME	Name of an ATB joint as given in the first field of a B.3 entry in the ATB input file.	C	Required
G0-G3 G4-G7	An ATB joint connects two segments. A local joint coordinate system is attached to each of these two segments. The position and orientation of these two coordinate systems relative to the segment coordinate systems is given on entry B.3 in the ATB input file. For each joint (J = 1,NJNT) a B.3 entry is defined in the ATB input file. The joint J connects the segment JNT(J) as given on the B.3 entry and the segment J + 1. MSC.Dytran finds the two segments that are connected by the joint with name = NAME. The grid points G0-G3 and G4-G7 define the joint coordinate systems for the segments JNT(J) and J + 1, respectively: G0 located at the origin of the joint coordinate system for the ATB segment JNT (J) G1 located on the local x-axis.	I > 0	Required

(Continued)

Field	Contents	Type	Default
	G2 located on the local y-axis.		
	G3 located on the local z-axis.		
	G4 located at the origin of the joint coordinate system for the ATB segment J + 1.		
	G5 located on the local x-axis.		
	G6 located on the local y-axis.		
	G7 located on the local z-axis.		
EID1-EID3 EID4-EID6	If EID1 through EID6, and PARAM,ATBSEGCREATE have been specified: MSC.Dytran will generate a Bulk Data file at time = 0. The grid points G0-G3 and G4-G7, at their initial position as specified in the ATB input file, will be written to the file. The files will also contain the following CBAR entries: For segment JNT(J): CBAR, EID1, PID-JNT(J), G0, G1, G2 CBAR, EID2, PID-JNT(J), G0, G2, G3 CBAR, EID3, PID-JNT(J), G0, G3, G1 For segment J+1: CBAR, EID4, PID-(J + 1), G4, G5, G6 CBAR, EID5, PID-(J + 1), G4, G6, G7 CBAR, EID6, PID-(J + 1), G4, G7, G4	I > 0	Blank

Remark:

1. All elements related to an ATB segment refer to the same material number. This material number is defined on the ATBSEG entry. If the material is defined to be rigid by means of a MATRIG entry, all elements can be easily connected to the contact ellipsoid of the ATB segment by means of an RCONREL entry referencing the MATRIG entry. In this way, all elements related to an ATB segment will move together with the ATB segment during the analyses and can be postprocessed.

ATBSEG

Interface to ATB Segments

Defines the position and orientation of the ATB segments. The position and orientation as specified on the G.2 and G.3 entries in the ATB input file will be overruled by the definitions given here.

This entry can be used to create a Bulk Data file containing elements and grid points to visualize the ATB segment, together with the contact ellipsoid and the joints it is connected by. See also ATBJNT and PARAM,ATBSEGCREATE.

Format and Example:

1	2	3	4	5	6	7	8	9	10
ATBSEG	ID	NAME	COVER	NUMELM	GSTART	ESTART	MID	PIDCOV	+CONT
ATBSEG	1	HEAD	YES	100	1000	1000	1000	1000	+CONT

+CONT	G0	G1	G2	G3	EID1	EID2	EID3	PIDCG	
+CONT	1010	1001	1002	1003	1001	1002	1003	1001	

Field	Contents	Type	Default
ID	Unique ATBSEG number.	I > 0	Required
NAME	Name of an ATB segment as given in the first field of a B.2 entry in the ATB input file.	C	Required
COVER	YES — If PARAM,ATBSEGCREATE has been specified, MSC.Dytran will generate a Bulk Data file containing grid points and elements located on the surface of the segment contact ellipsoid. The shape and position of the segment contact ellipsoid is defined on the B.2 entry in the ATB input file. See Remark 2.	C	NO
	NO — The covering is not performed.		
NUMELM	Maximum number of elements used for covering the ellipsoid.	I > 0	128
GSTART	Grid-point numbering for covering the ellipsoid starts at GSTART.	I > 0	1
ESTART	Element numbering for covering the ellipsoid starts at RESTART.	I > 0	1

(Continued)

Field	Contents	Type	Default
MID	All elements created by MSC.Dytran to visualize the ATB segment will have a rigid material (MATRIG) with MID as the material number. MID is used by both the elements covering the segment contact ellipsoid as well as by the CBAR elements used to visualize the segment coordinate system and joint coordinate systems (See ATBJNT).	I > 0	1
PIDCOV	All elements created by MSC.Dytran to cover the ATB segment contact ellipsoid will have PIDCOV as the property number.	I > 0	1
G0-G3	The grid points span the local coordinate system of the ATB segment: G0 located at the origin of the ATB segment. G1 located on the local x-axis. G2 located on the local y-axis. G3 located on the local z-axis. The above is used by MSC.Dytran to overwrite the initial position and orientation of the segments as specified in the ATB input file. See below (EID1-EID3) on how to generate the above grid points for an existing ATB input file.	I > 0	Required
EID1-EID3	If EID1, EID2, EID3, and PARAM,ATBSEGCREATE have been specified: MSC.Dytran will generate a Bulk Data file containing the grid points G0-G3 at the initial position as specified in the ATB input file. The file will also contain the three following CBAR entries: CBAR, EID1, PIDCG, G0, G1, G2 CBAR, EID2, PIDCG, G0, G2, G3 CBAR, EID3, PIDCG, G0, G3, G1	I > 0	Blank
PIDCG	Property number used by MSC.Dytran in generating the CBAR entries EID1 through EID3.	I > 0	1

(Continued)

Remarks:

1. All elements related to an ATB segment reference the same material number. This material number is defined on the ATBSEG entry. If the material is defined as rigid by means of a MATRIG entry, all elements can be easily connected to the contact ellipsoid of the ATB segment by means of an RCONREL entry referencing the MATRIG entry. In this way, all elements related to an ATB segment will move together with the ATB segment during the analysis and can be postprocessed. The elements can also be used in a CONTACT, ALE and/or COUPLING surface to define interaction between the ATB segment and other parts of the finite element model. The forces and moments acting on the elements are transferred to the ATB segment to which they are connected.

2. The MATRIG entry written to the file will have the inertia properties of the segment, as defined in the ATB input file.

BJOIN

Breakable Join

Defines (multiple) pairs of grid points of one-dimensional and/or shell elements to be joined during the analysis. When the failure criterion for a grid-point pair is satisfied, the grid-point pair is removed from the join and the grid points are computed as separate grid points from that moment on. The join ceases to exist when all pairs of the join have failed, after which all grid points of the join are treated as separate grid points.

Format and Example:

1	2	3	4	5	6	7	8	9	10
BJOIN	BID	SID	TOL	TYPE	CRIT	VALUE1	VALUE2	VALUE3	+CONT1
BJOIN	1	2	1.E-4	COMPO	FORCE	1.E3	1.E4	1.E3	+CONT1

+CONT1	VALUE4	VALUE5	VALUE6	EQUIV	MULTI				+CONT2
+CONT1	1.E5	1.E4	1.E2	YES	YES				+CONT2

+CONT2	VALUE7	VALUE8							
+CONT2									

Field	Contents			Type	Default
BID	BJOIN number.			I > 0	Required
SID	SET1 number.			I > 0	Required
TOL	Tolerance used in matching grid-point pairs.			R ≥ 0.	1.E-4
TYPE	Type of failure criterion.			C	FOMO
	FOMO	Constant force and/or moment.			
		CRIT	No meaning (ignored).	C	Blank
		VALUE1	Force at failure.	R ≥ 0.	1.E20
		VALUE2	Moment at failure.	R ≥ 0.	1.E20
	USER	User-defined failure.			
		CRIT	No meaning (ignored).	C	Blank
		VALUE1	Name of the user-defined criterion to be used in the EXBRK user subroutine.	C	Required
	TIME	Failure at a specified time.			
		CRIT	No meaning (ignored).	C	Blank
		VALUE1	Time of failure.	R ≥ 0.	1.E20

(Continued)

Field	Contents			Type	Default
	COMPO	Component failure at constant values.			
		CRIT is the criterion for failure.			
		FORCE	Failure on forces.	C	BOTH
		MOMENT	Failure on moments.		
		BOTH	Failure on force and moment.		
		VALUE1	x-Force at failure.	R ≥ 0.	1.E20
		VALUE2	y-Force at failure.	R ≥ 0.	1.E20
		VALUE3	z-Force at failure.	R ≥ 0.	1.E20
		VALUE4	x-Moment at failure.	R ≥ 0.	1.E20
		VALUE5	y-Moment at failure.	R ≥ 0.	1.E20
		VALUE6	z-Moment at failure.	R ≥ 0.	1.E20
	SPOTWELD	Spotweld-like failure.			
		CRIT	No meaning.		
		VALUE1	Failure force in tension.	R ≥ 0.0	No failure.
		VALUE2	Failure force in compression.	R ≥ 0.0	No failure.
		VALUE3	Failure force in shear.	R ≥ 0.0	No failure.
		VALUE4	Failure torque.	R ≥ 0.0	No failure.
		VALUE5	Failure bending moment.	R ≥ 0.0	No failure.
		VALUE6	Failure total force.	R ≥ 0.0	No failure.
		VALUE7	Failure total moment.	R ≥ 0.0	No failure.
		VALUE8	Failure time.	R ≥ 0.0	No failure.
	Note:	The failure forces in compression, shear, torque, and bending, are only used if the spotweld grid points are not coincident. EQUIV = NO must be specified to prevent an automatic equivalence of the grid-point positions.			

(Continued)

Field	Contents		Type	Default
EQUIV	Equivalence the positions of the grid points at time step zero.		C	YES
	YES	The positions of the two grid points are equivalenced as: $x_{bjoin} = \frac{1}{2}[x_{grid_1} + x_{grid_2}]$		
	NO	The positions of the two grid points are not equivalenced. The BJOIN behaves as a rigid body with the correct inertial properties until failure occurs.		
MULTI	Multiple breakable joins, where the grid points must be entered as a sequence of BJOIN pairs.		C	NO
	YES	The grid points are entered on the SET1 entry as a sequence of BJOIN pairs.		
	NO	The program creates BJOIN pairs for every two grid points entered on the SET1 entry, when the grid-point positions fall within the tolerance (TOL).		
	Independent of the setting of MULTI (either YES or NO), all BJOIN pairs that fall within the defined tolerance (TOL) are merged into one multiple breakable join.			

Remarks:

1. If the TYPE field is set to USER, the user subroutine must be present in the file referenced by the USERCODE FMS statement.
2. The breakable joins can only be used for grid points of one-dimensional and shell elements. Note that any grid point can be made into a one-dimensional grid-point type by connecting a spring to the grid point.

(Continued)

3. The constant force or constant moment failure criterion (TYPE = FOMO) is met once the following inequality is true:

   $$(F_{x1} - F_{x2})^2 + (F_{y1} - F_{y2})^2 + (F_{z1} - F_{z2})^2 > F_{max}^2$$

   In the above formula, F is either a force or a moment. F_{max} is the value defined in the VALUEn fields.

4. If component failure is requested (TYPE = COMPO), the comparison is performed for each component of the force and moment vector. Depending on the criterion-type definition, the forces, the moments, or both are taken into account to determine whether the join fails.

5. In component failure, note that if one of the determining failure component values is left blank, this component can never cause the join to fail.

6. The first entity that satisfies the criterion for failure will cause the join to fail.

7. The nondetermining components in component failure are automatically set to infinity. This means that when failure on force components is requested, the moment criteria are set to infinity. The same is true for the forces when moment component failure is requested.

8. The user-defined criterion name can be a maximum of eight characters long.

9. At the moment of failure, an informational message is written to the output file.

Defines a body force loading.

Format and Example

1	2	3	4	5	6	7	8	9	10
BODYFOR	BID	TYPE	TYPEV						+CONT1
BODYFOR	100	EULER							+CONT1

+CONT1	CID	SCALE	VALUE	N1	N2	N3			
+CONT1	5	TABLE	13	1.	0.	0.			

Field	Contents		Type	Default
BID	Unique body force number.		I > 0	Required
TYPE	Type of entity.		C	LAGRANGE
	LAGRANGE	Lagrangian type of grid point.		
	EULER	Euler type of element.		
	ELLIPS	Ellipsoid		
TYPEV	Name or ID of type of entity.		C or I	See Remark 1.
	TYPE	TYPEV		
	LAGRANGE	ELEM1D		
		SHTRIA		
		SHQUAD		
		MEMTRIA		
		LAGSOLID		
	EULER	EULHYDRO		
		EULSTREN		
		EULMM		
	ELLIPS	ID		
CID	Number of a CORDxxx entry.		I ≥ 0	0

(Continued)

Field	Contents		Type	Default
SCALE	Scale factor for the load.		C	CONSTANT
	CONSTANT	Constant scale factor.		
	TABLE	Tabular input for the scale factor.		
VALUE	Value or TABLE id for SCALE.		I or R	Required
N1,N2,N3	Components of a vector giving the load direction. At least one must be nonzero.		R	See Remark 2.

Remarks:

1. The default for entity TYPEV is all entities of TYPE.
2. By default the components are zero, but at least one of them should be nonzero.
3. Only one BODYFOR entry per type of entity TYPEV is allowed.

CBAR

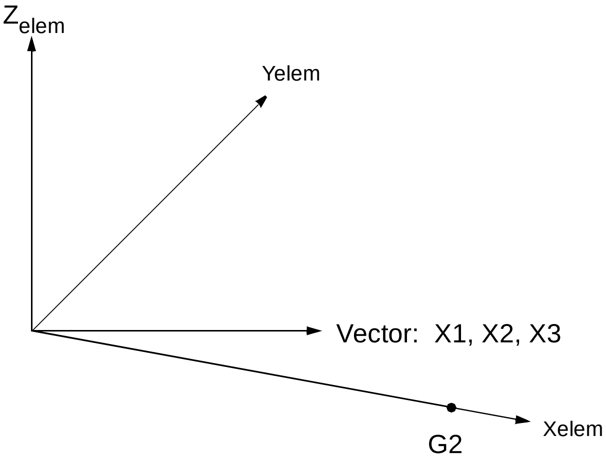
Bar Element Connection

Defines a simple beam element.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CBAR	EID	PID	G1	G2	X1, G3	X2	X3		
CBAR	2	39	7	3	3	13			

Field	Contents	Type	Default
EID	Unique element number.	I > 0	Required
PID	Number of a PBAR or PBEAM property entry.	I > 0	EID
G1, G2	Grid-point numbers at the ends of the beam. G1 must not be the same as G2.	I > 0	Required
G3	Grid-point number to specify the vector defining the local x-y plane for the element. G3 must not be colinear with G1 and G2.	I > 0	See Remark 2.
X1, X2, X3	Components of a vector at G1 in the basic coordinate system that lies in the element x-y plane.	R	See Remark 3.



(Continued)

Remarks:

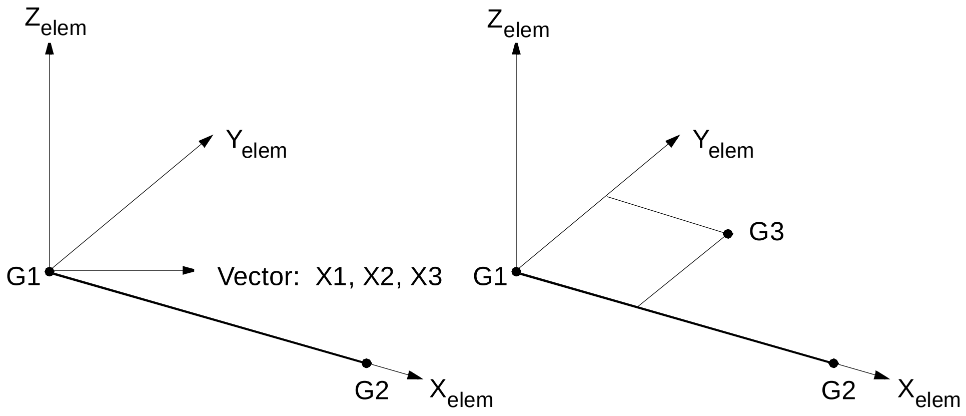
1. The element number must be unique with respect to all other element numbers.
2. The third grid point is used to specify a vector from G1 to G3. The local x-axis of the beam is in the direction of the beam from point G1 to G2. The local y-axis is perpendicular to the beam in the plane containing the vector from G1 to G3. The local z-axis is perpendicular to the local x and y-axes (see Section 2.3.6 on page 2-16).
3. If field 6 (X1, G3) is an integer, G3 is used to define the x-y plane. If field 6 (X1, G3) is real, X1, X2, and X3 are used.

Defines a beam element.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CBEAM	EID	PID	G1	G2	X1, G3	X2	X3		
CBEAM	2	39	7	3	13				

Field	Contents	Type	Default
EID	Unique element number.	I > 0	Required
PID	Number of a PBEAM or PBEAM1 property entry.	I > 0	EID
G1, G2	Grid-point numbers at the ends of the beam. G1 must not be the same as G2.	I > 0	Required
G3	Grid-point number to specify the vector defining the local x-y plane for the local element. G3 must not be collinear with G1 and G2.	I > 0	See Remark 2.
X1, X2, X3	Components of a vector at G1, in the basic coordinate system that lies in the element x-y plane.	R	See Remark 3.



Remarks:

1. The element number must be unique with respect to all other element numbers.

2. The third grid point is used to specify a vector from G1 to G3. The local x-axis of the beam is in the direction of the beam from point G1 to G2. The local y-axis is perpendicular to the beam in the plane containing the vector from G1 to G3. The local z-axis is perpendicular to the local x- and y-axes (See Section 2.3.6 on page 2-16).

3. If field 6 (X1, G3) is an integer, G3 is used to define the x-y plane. If field 6 (X1, G3) is real, X1, X2, and X3 are used.

CDAMP1

Damper Connection

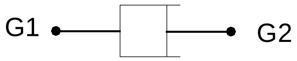
Defines a scalar damper element.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CDAMP1	EID	PID	G1	C1	G2	C2			+CONT1
CDAMP1	19	6	7	3	104	3			+CONT1

+CONT1	CORD	FOLLOW							
+CONT1	3	G1							

Field	Contents	Type	Default
EID	Unique element number.	I > 0	Required
PID	Number of a PDAMPn property entry.	I > 0	EID
G1,G2	Grid-point numbers at the ends of the damper. G1 must not be the same at G2. If either G1 or G2 are zero, the damper is connected to the ground.	I ≥ 0	0
C1,C2	Degree of freedom at G1 and G2 where the damper is connected.	1 ≤ I ≤ 6	Required



Field	Contents	Type	Default
CORD	Number of a coordinate system in which the degree of freedom (C1, C2) is defined.	I ≥ 0	0
FOLLOW	CORD direction vector follows the motion of the coordinate system as specified under CORD. G1 direction vector follows the motion of end point G1. G2 direction vector follows the motion of end point G2.	C	CORD

Remarks:

1. The element number must be unique with respect to all other element numbers.
2. The damper always acts in the direction given by C1 and C2 regardless of the relative positions of the grid points. CVISC defines a damper with an orientation that changes during the analysis.
3. Setting G1 or G2 to zero gives a grounded damper.

(Continued)

4. The damper can connect translational or rotational degrees of freedom.

5. The property entry PDAMP defines the damper characteristic.

6. If the degree of freedom is defined in a nonbasic coordinate system, the degrees of freedom G1 and G2 must be equal or one must be grounded.

7. The coordinate system CORD must be rectangular.

8. For fast rotating structures, it is advised to use a CORD3R or CORD4R to define the follow motion. A moving coordinate system CORD4R is updated according to the full-rotation equations, while a direction vector that rotates with G1 or G2 is updated using the Hughes-Winget relation. The Hughes-Winget relation becomes less accurate when the rotation angle per time step is very high.

CDAMP2

Linear Damper

Defines a linear scalar damper element where the damping coefficient is defined directly.

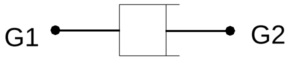
Format and Example:

1	2	3	4	5	6	7	8	9	10
CDAMP2	EID	B	G1	C1	G2	C2			+CONT1
CDAMP2	19	2.4 E3	7	3	14	3			+CONT1

+CONT1	CORD	FOLLOW							
+CONT1	3	G1							

Field	Contents	Type	Default
EID	Unique element number.	I > 0	Required
B	Damping coefficient. (Force/velocity).	R	0.0
G1, G2	Grid-point numbers at the end of the damper. G1 must not be the same as G2. If either G1 or G2 are zero, the damper is connected to the ground.	I ≥ 0	0
C1, C2	Degree of freedom at G1 and G2 where the damper is connected	1 ≤ I ≤ 6	Required

.



Field	Contents	Type	Default
CORD	Number of a coordinate system in which the degree of freedom (C1, C2) is defined.	I ≥ 0	0
FOLLOW	CORD direction vector follows the motion of the coordinate system as specified under CORD. G1 direction vector follows the motion of end point G1. G2 direction vector follows the motion of end point G2.	C	CORD

Remarks:

1. The element number must be unique with respect to all other element numbers.
2. The damper always acts in the direction given by C1 and C2, regardless of the relative positions of the grid points. CVISC defines a damper with an orientation that can change during the analysis.

(Continued)

3. Setting G1 or G2 to zero gives a grounded damper.

4. The damper can connect translational or rotational degrees of freedom.

5. CDAMP1 can also be used to define linear scalar dampers. When there are many dampers with the same damping coefficient, it is more efficient to use CDAMP1.

6. When the degree of freedom is defined in a nonbasic coordinate system, the degrees of freedom G1 and G2 must be equal or one must be grounded.

7. The coordinate system CORD must be rectangular.

8. For fast rotating structures it is advised to use a CORD3R or CORD4R to define the follow motion. A moving coordinate system CORD4R is updated according to the full-rotation equations, while a direction vector that rotates with G1 or G2 is updated using the Hughes-Winget relation. The Hughes-Winget relation becomes less accurate when the rotation angle per time step is very high.

CELAS1

Scalar-Spring Connection

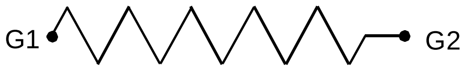
Defines a scalar-spring element.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CELAS1	EID	PID	G1	C1	G2	C2			+CONT1
CELAS1	2	6	6	2	8	1			+CONT1

+CONT1	CORD	FOLLOW							
+CONT1	3	G1							

Field	Contents	Type	Default
EID	Unique element number.	I > 0	Required
PID	Number of a PELAS property entry.	I > 0	EID
G1, G2	Grid-point number.	I ≥ 0	0
C1, C2	Component number.	0 ≤ I ≤ 6	0
CORD	Number of a coordinate system in which the degree of freedom (C1, C2) is defined.	I ≥ 0	0
FOLLOW	CORD direction vector follows the motion of the coordinate system as specified under CORD. G1 direction vector follows the motion of end point G1. G2 direction vector follows the motion of end point G2.	C	CORD



Remarks:

1. A grounded spring is defined by setting G1 or G2 to zero in which case the corresponding C1 or C2 is zero or blank. (A grounded grid point is a grid point where the displacement is constrained to zero.)

2. Element numbers must be unique with respect to all other element numbers.

3. The connection grid points G1 and G2 must be distinct.

(Continued)

4. If the degree of freedom is defined in a nonbasic coordinate system, the degrees of freedom G1 and G2 must be equal or one must be grounded.

5. The coordinate system CORD must be rectangular.

6. For fast rotating structures it is advised to use a CORD3R or CORD4R to define the follow motion. A moving coordinate system CORD4R is updated according to the full-rotation equations, while a direction vector that rotates with G1 or G2 is updated using the Hughes-Winget relation. The Hughes-Winget relation becomes less accurate when the rotation angle per time step is very high.

CELAS2

Scalar-Spring Property and Connection

Defines a scalar-spring element where the spring stiffness is defined directly.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CELAS2	EID	K	G1	C1	G2	C2			+CONT1
CELAS2	28	6.2+3	32	1	19	4			+CONT1

+CONT1	CORD	FOLLOW							
+CONT1	3	G1							

Field	Contents	Type	Default
EID	Unique element number.	I > 0	Required
K	The stiffness of the scalar spring.	R	0.
G1, G2	Grid-point number.	I ≥ 0	0
C1, C2	Component number.	0 ≤ I ≤ 6	0

Field	Contents	Type	Default
CORD	Number of a coordinate system in which the degree of freedom (C1, C2) is defined.	I ≥ 0	0
FOLLOW	CORD direction vector follows the motion of the coordinate system as specified under CORD. G1 direction vector follows the motion of end point G1. G2 direction vector follows the motion of end point G2.	C	CORD

Remarks:

1. A grounded spring is defined by:

   a. Setting G1 or G2 to zero in which case the corresponding C1 or C2 is zero or blank.

   b. Using a scalar point for G1 and/or G2 in which case the corresponding C1 and/or C2 is zero or blank. (A grounded grid point is a grid point where the displacement is constrained to zero.)

(Continued)

2. Element numbers must be unique with respect to all other element numbers.

3. This entry completely defines the element since no material or geometric properties are required.

4. The two connection points G1 and G2 must be distinct.

5. If the degree of freedom is defined in a nonbasic coordinate system, the degrees of freedom G1 and G2 must be equal or one must be grounded.

6. The coordinate system CORD must be rectangular.

7. For fast rotating structures it is advised to use a CORD3R or CORD4R to define the follow motion. A moving coordinate system CORD4R is updated according to the full-rotation equations, while a direction vector that rotates with G1 or G2 is updated using the Hughes-Winget relation. The Hughes-Winget relation becomes less accurate when the rotation angle per time step is very high.

8. If possible, use of the PELAS, CELAS1 entries is preferable. Many CELAS2 elements result in excessive input manipulation and storage.

CFACE

Face of an Element

Defines a face on an Eulerian or a Lagrangian element.

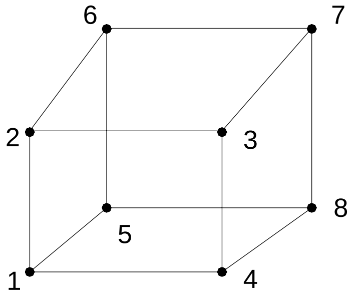
Format and Example:

1	2	3	4	5	6	7	8	9	10
CFACE	FID	SID	EID	FACE					
CFAC4E	37	100	1796	4					

Field	Contents	Type	Default
FID	Unique face number.	I > 0	Required
SID	Number of a set of faces to which the face belongs. It is referenced by a FLOW or SURFACE entry.	I > 0	Required
EID	Element number to which the face is attached.	I > 0	Required
FACE	The number of the element face that is to be used. See Remark 3.	1 ≤ I ≤ 6	Required

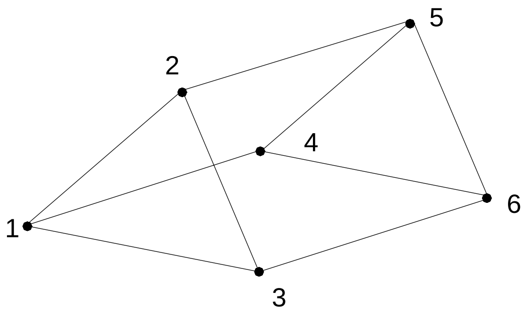
Remarks:

1. The face number FID must be unique with respect to all other face numbers.
2. The CSEG entry is also used to define faces in terms of the grid-point numbers. The CFACE1 entry is also used to define faces.
3. A negative face number indicates that the face normal direction is reversed.
4. The element-face numbers are as follows:

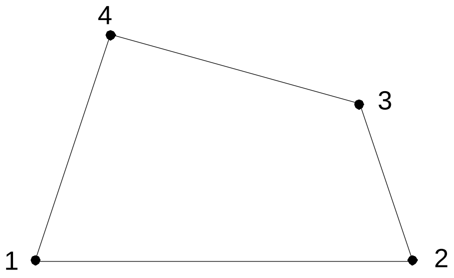


Face Number	Grid Points
1	1432
2	1265
3	1584
4	7856
5	7348
6	7623

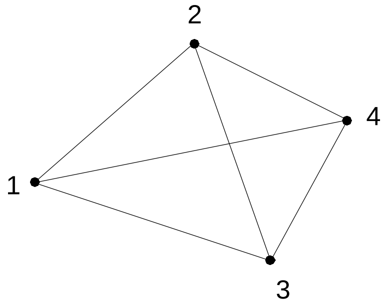
(Continued)



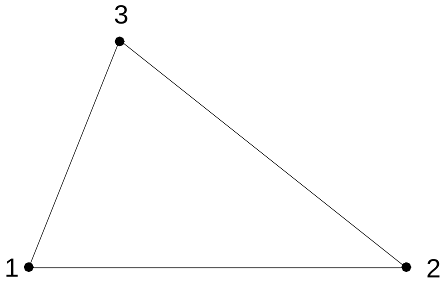
Face Number	Grid Points
1	132
2	1254
3	1463
4	645
5	6523



Face Number	Grid Points
1	1432
4	1234



Face Number	Grid Points
1	132
2	124
3	143
4	423



Face Number	Grid Points
1	132
4	123

CFACE1

Face of an Element

Defines a face on an element in terms of the element number and two grid points on the required face. This is particularly suitable for defining the faces on solid elements.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CFACE1	FID	SID	EID	G1	G3/G4				
CFACE1	497	100	2796	32	4162				

Field	Contents	Type	Default
FID	Unique face number.	I > 0	Required
SID	Number of a set of faces to which the face belongs. It is referenced by a FLOW or a SURFACE entry.	I > 0	Required
EID	Element number to which the face is attached.	I > 0	Required
G1	Number of a grid point connected to a corner of the face.	I > 0	Required
G3	Number of a grid point connected to a corner diagonally opposite to G1 on the same face of a CHEXA or CPENTA element. This applies to quadrilateral faces of CPENTA elements only. G3 must be omitted for a triangular face on a CPENTA element.	I > 0	Blank
G4	Number of the grid point of a CTETRA element that is not on the required face.	I > 0	Required

Remark:

1. A PLOAD4 entry with an absolute pressure of 9999. is automatically converted to a CFACE1 entry. This makes defining CFACE1 entries in preprocessors very easy. See also Section 3.2.6 on page 3-4.

CHEXA

Element with Eight Grid Points

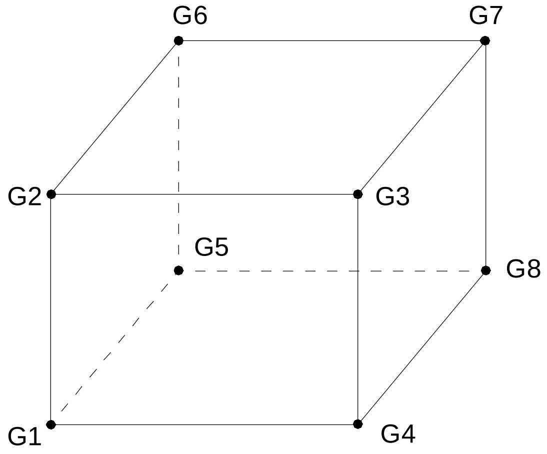
Defines an Eulerian or a Lagrangian element with eight corner grid points.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CHEXA	EID	PID	G1	G2	G3	G4	G5	G6	+CONT1
CHEXA	71	4	3	4	5	6	7	8	+CONT1

+CONT1	G7	G8							
+CONT1	9	10							

Field	Contents	Type	Default
EID	Unique element number.	I > 0	Required
PID	Number of a PSOLID or PEULERn property entry.	I > 0	EID
G1–G8	Grid-point numbers of the connected grid points. They must all be unique.	I > 0	Required



(Continued)

Remarks:

1. The element number must be unique with respect to all other element numbers.
2. Grid points G1 through G4 must be given in consecutive order about one quadrilateral face. G5 through G8 must be on the opposite face with G5 opposite G1, G6 opposite G2, etc.
3. Number according to the figure shown in this CHEXA entry description.
4. The property number references a PSOLID or a PEULERn entry. This determines whether the element is Lagrangian or Eulerian.
5. Only the first eight grid points on a CHEXA are used in MSC.Dytran. The excess is ignored.

CONM2

Concentrated Grid-Point Mass and/or Inertia

Defines a concentrated grid-point mass and/or inertia for Lagrangian elements.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CONM2	ID	G		M				I	
CONM2	7	9		.1				4.4E-3	

Field	Contents	Type	Default
ID	Unique CONM2 number.	I > 0	Required
G	Grid-point number.	I > 0	Required
M	Mass.	R	0.0
I	Inertia.	R	0.0

Remarks:

1. All grid points in the model must have mass associated with them, either by the properties of the elements attached to the grid point or by using a CONM2 entry.
2. When PARAM,CONM2OUT is set to NO, there will be no summary on the CONM2 entries defined. This means that the mass, momentum, and energy of the CONM2s are not added to the material and cycle summaries. Setting PARAM,CONM2OUT,NO saves memory and CPU time.
3. The CONM2 results cannot be output on time-history or archive files.

CONTACT

Contact Surface

Defines contact between Lagrangian grid points and elements.

The algorithm is based on the contact of “SLAVE NODES” with “MASTER FACES”.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CONTACT	CID	STYPE	MTYPE	SID	MID	FS	FK	EXP	+CONT1
CONTACT	7	SURF	SURF	3	7	0.0	0.0	0.0	+CONT1

+CONT1	VERSION	SIDE	SEARCH	ADAPT	THICK	GAP	DAMPING	WEIGHT	+CONT2
+CONT1	V4	BOTH	FULL	NO	1.0	0.0	YES	BOTH	+CONT2

+CONT2			PEN	PENV	MONVEL	FACT	MONDIS	MONDISV	+CONT3
+CONT2			DISTANCE	1.E20	1.1	0.1	FACTOR	2.0	+CONT3

+CONT3	TSTART	TEND	REVERSE	INITPEN	PENTOL	INIID	INITMON	SLVACT	+CONT4
+CONT3	0.0	1.E20	ON	ON	1.E20				+CONT4

+CONT4	DRWBEADF	CONTFORC							
+CONT4									

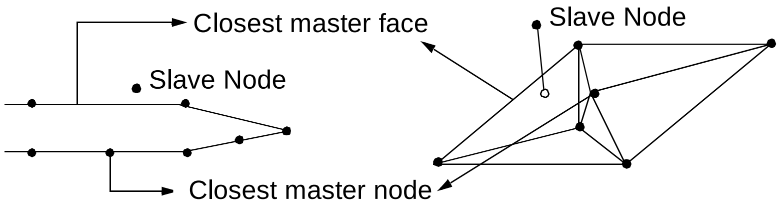
Field	Contents		Type	Default
CID	Unique contact number.		I > 0	Required
STYPE	Type of entity used to define the slave nodes.		C	SURF
	SURF	All nodes belonging to a SURFACE.		
	ELEM	All nodes belonging to a list of elements.		
	PROP	All nodes belonging to elements with certain property numbers.		
	MAT	All nodes belonging to elements with certain material numbers.		
	GRID	A list of grid points.		

(Continued)

Field	Contents		Type	Default
MTYPE	Type of entity used to define the master faces.		C	blank
	blank	All faces belonging to the slave SURFACE, or the faces belonging to the elements referenced by STYPE,SID. This option is only allowed for STYPE = SURF, ELEM, PROP, or MAT. The option results in a so-called “single surface contact”.		
	SURF	All faces belonging to a SURFACE.		
	ELEM	All faces belonging to a list of elements.		
	PROP	All faces belonging to elements with certain property IDs.		
	MAT	All faces belonging to elements with certain material IDs.		
SID	Number of a SURFACE entry if STYPE = SURF, or number of a SET1 entry if STYPE = ELEM, PROP, MAT, or GRID.		I > 0	Required
MID	Number of a SURFACE entry if MTYPE = SURF, or number of a SET1 entry if MTYPE = ELEM, PROP, MAT, or GRID.		I > 0	
FS	Static coefficient of friction (See Remark 2).		R	0.0
FK	Kinetic coefficient of friction (See Remark 2).		R	0.0
EXP	Exponential decay coefficient (See Remark 2).		R	0.0
VERSION	Version of the algorithm.		C	Required
	V2	Based on MSC/DYNA contact algorithm. Keeps tracks of only 1 simultaneous contact per slave node. Master-Slave contact can be adaptive. Single surface contact can not be adaptive. Search algorithm assumes that the master face is connected to the closest master node. This is not true in sharply folded regions and in cases where the neighboring master faces have very different sizes and aspect ratios:		

(Continued)

Field	Contents	Type	Default



V3 Obsolete (use V4 version).

V4 Keeps track of multiple simultaneous contacts per slave node.

Memory allocation is dynamic.

Both "single surface" as "master slave" contact can be adaptive.

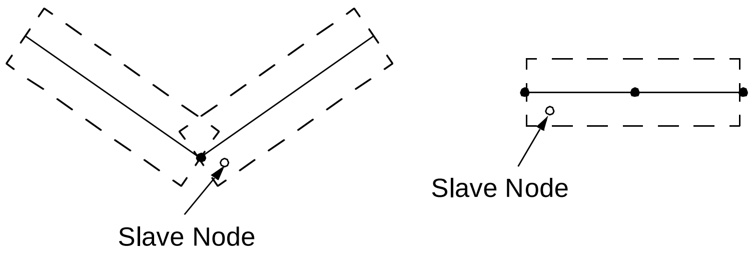
Search algorithm doesn't assume that the closest master face is connected to the closest master node.

Has a damping option to eliminate high frequency oscillations.

More options available on the CONTINI entries to initialize the contact states of inplane folded airbags.

Takes special care of reversed normals when neighboring master faces form a T-joint.

Takes special care of first penetrations of a slave node into a master face. The logic avoids applying high contact forces to slave nodes who enter the contact thickness of a master face from the side:



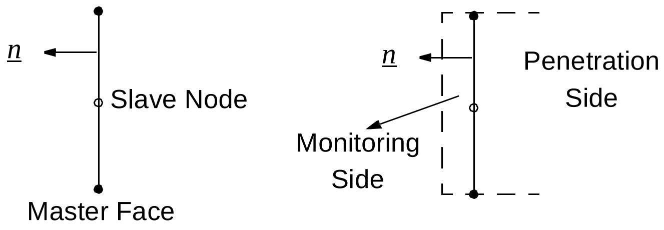
Note that initial penetrations are also treated as a first penetration.

An option is available to assign spring/damper characteristics to the contact force (CONTFORC).

(Continued)

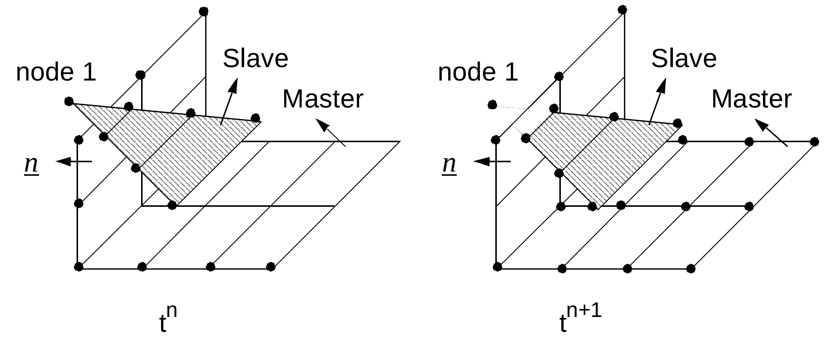
Field	Contents		Type	Default
V4 (Cont.)	BELT	Suited for modeling contact between a belt element and a rigid structure. Based on V2 algorithm. “Master slave” contact only. The contact logic doesn't apply a contact force, but applies an enforced displacement and velocity that keeps the slave nodes exactly on top of the master face. The slave node will not slide relative to the master face when the friction coefficient (FS) is set to 1E20.		
	BELT1	Identical to BELT algorithm, except that the slave nodes are initially repositioned on top of the closest master face. All slave nodes initially penetrated or within a distance of INITMON from a master face, are repositioned.		
	DRAWBEAD	Suited for modeling a drawbead. Based on V2 algorithm. STYPE must be GRID. The list of slave nodes must be ordered along the drawbead line. MTYPE must be SURF. The restraining force per unit drawbead length is specified in the field DRWBEADF.		

(Continued)

Field	Contents	Type	Default
SIDE	Defines which side will be the monitoring side of a master face. The opposite side of the master face will be the penetration side.	C	BOTH
	BOTH — The side from which a slave node approaches the master face will become the monitoring side.		
	TOP — The monitoring side will always be on the side of the master face that its normal is pointing at.		
	BOTTOM — The monitoring side will always be on the opposite side of the master face that its normal is pointing at.		
	The options TOP/BOTTOM are useful in the following cases:		
	1. When a slave node initially is located on the master face (see the picture below), the contact situation is uniquely defined, only if the TOP or BOTTOM side of the master surface is defined.		
	 n Slave Node Master Face n Penetration Side Monitoring Side		

(Continued)

Field	Contents	Type	Default
	2. When hooking of slave nodes on the wrong side of a master face might occur. This often is the case when the master face is at the edge of a shell element structure:		



→ no hooking of node 1 when SIDE = BOTTOM

→ hooking of node 1 when SIDE = BOTH

SEARCH	Defines the type of search algorithm. FULL — Regular search algorithm. SLIDE — Special option for in-plane folded air-bags. This option should be used with care in other applications. BPLANE — Improved search algorithm, where a back-up plane region alleviates the problem for SLIDE when a slave node has entered a dead region. See Remark 4 for a more detailed description of these methods.	C	FULL
ADAPT	Defines whether the master faces are (de)activated based on element failure. Slave nodes only check for contact with active master faces. NO — The contact is non-adaptive, and all the master faces are active during the whole analysis. YES — The contact is adaptive. The master faces are (de)activated according to the following logic:	C	NO

(Continued)

Field	Contents	Type	Default
ADAPT (Cont.)	Shell elements – At time zero all the master faces are active. Once an element fails, its corresponding master face will be deactivated. The contact will treat it as an actual hole. Lagrangian solids – At time zero only the Freefaces are active. All the internal faces will be deactivated. When an element fails, some of its faces might become Freefaces. These Freefaces will be activated. Once all the elements connected to a master face have failed, it will be deactivated for the remainder of the analyses. This logic allows for modeling of impact-penetration phenomena, and is sometimes called “eroding contact.” Note: (De)activation of slave nodes is selected on the SLVACT field.		
THICK	Shell thickness scale factor. See Remark 3.	R	Required for VERSION = V4
GAP	Artificial contact thickness. See Remark 3.	R	0.0
DAMPING	[YES/NO] - VERSION V4 only. Specifies if a high frequency damping is active or not. The damping force is based on the relative velocity of a slave node with respect to a master face. The damping is preferably turned on in all cases, except for RIGID-RIGID contact. In RIGID-RIGID contact it can result in a substantial loss of energy.	C	YES
WEIGHT	For contact versions V2 and V4, the contact force is multiplied by a mass-weighting factor. The following options are available: BOTH $M_{scale} = \dfrac{M_{master}M_{slave}}{M_{master} + M_{slave}}$ SLAVE $M_{scale} = M_{slave}$ MASTER $M_{scale} = M_{master}$ NONE $M_{scale} = 1.0$	C	BOTH

Field **Contents** **Type** **Default**

(Continued)

WEIGHT (Cont.)

Recommended usage:

SLAVE	MASTER	WEIGHT
deformable	deformable	BOTH
deformable	rigid	SLAVE
rigid	deformable	MASTER
rigid	rigid	NONE

PEN

Allowed penetration check. C No check

If the penetration depth exceeds a certain value it is assumed that the slave node is in a bad contact state. No contact force will be applied and the slave node will be taken out of the contact for the remainder of the calculation. This option is useful in the following applications:

1. In airbag analysis to prevent "locking" of the unfolding bag.
2. In crash analysis to prevent high contact forces in extremely folded regions that would require a much finer mesh without this option.

DISTANCE The allowed penetration depth is specified in PENV.

FACTOR The allowed penetration depth is equal to a factor times a characteristic length of the master faces. The factor is specified in PENV.

PENV

Value of the allowed penetration depth or value of the FACTOR to calculate the allowed penetration depth. R No check

MONVEL

The contact monitoring distance is increased by a value based on the relative velocity of a slave node and a master face. The increase is only used if the slave node is moving towards the master face, and is equal to: R 1.1

$$\text{MONVEL} * (\text{relative velocity}) * \text{DT}$$

(Continued)

Field	Contents	Type	Default
FACT	Scale factor for the contact forces. The default value for FACT works in most applications. When the slave nodes penetrate too much, the contact can me made stiffer by increasing the value of FACT. It is advised to limit the value of FACT to: Single Surface Contact:FACT = 1.0 Master-Slave Contact:FACT = 0.5 When a CONTFORC entry is defined for this contact, the value of FACT is not used. The contact force is based solely on the spring/damper characteristics as specified on the CONTFORC entry.	R > 0.0	0.1
MONDIS	Defines the fixed part of the monitoring distance. When the normal distance of a slave node to a master face becomes smaller than the monitoring distance the slave node will tag itself to the master face. The side from which the slave node is moving towards the master face becomes the monitoring region. The monitoring distance contains a fixed part and a velocity dependent part. See MONVEL for a description of the velocity dependent part. DISTANCE — The monitoring distance is specified in MONDISV. FACTOR — The monitoring distance is equal to a factor times a characteristic length of the master faces. The factor is specified in MONDISV.	C	FACTOR
MONDISV	Value of the monitoring distance or value of the FACTOR to calculate the monitoring distance.	R	2.0
TSTART	Time at which the contact is activated. This will overrule a possible definition on an ACTIVE entry.	R ≥ 0	0.0
TEND	Time at which the contact is deactivated. This will overrule a possible definition on an ACTIVE entry.	R ≥ 0	ENDTIME
REVERSE	[ON/OFF] Automatic reversal of master faces such that their normal point in the same direction.	C	ON

(Continued)

Field	Contents	Type	Default
INITPEN	[ON/OFF] Allowed initial penetration check. Each slave node is checked for an initial penetration, and if the initial penetration depth is within an allowed limit. If an initial penetration occurs, and the penetration depth falls within the allowed limit, a warning will be issued. If an initial penetration occurs, and the initial penetration depth is larger than the allowed value, the contact between the slave node and the master face will not be initialized. No warning will be issued.	C	ON
PENTOL	Tolerance for the allowed initial penetration check.	R	1.E20
INIID	ID of a set of CONTINI entries used to define the initial contact state. This option is useful for in-plane folded airbags.	I > 0	blank
INITMON	Fixed part of the monitoring distance used during the initialization. If not specified, the value of MONDIS is used.	R > 0.0	MONDIS

Field	Contents	Type	Default
SLVACT	Defines the method used to (de)activate the slave nodes.	C	See Remark 4.
	VERSION=V2 Applies only when ADAPT = YES. NO The slave nodes will be deactivated after all its connected elements have failed. YES The slave nodes will always be active.		
	VERSION=V4 Applies for both ADAPT = YES and ADAPT = NO.		
	METHOD1 Applies to all V4 contacts. METHOD2 Applies to all V4 contacts. METHOD3 Applies to all V4 contacts. METHOD4 Applies to all V4 contacts.		
	METHOD1A Applies to master-slave V4 contacts only. METHOD2A Applies to master-slave V4 contacts only. METHOD3A Applies to master-slave V4 contacts only. METHOD4A Applies to master-slave V4 contacts only.		
	See Remark 5 for a detailed description of these methods.		
DRWBEADF	Drawbead force per unit length.	R > 0.0	Required for VERSION = DRAWBEAD.
CONTFORC	ID of a CONTFORC entry.	I > 0	blank
	When specified, the contact force is not based on the Lagrangian multiplier method, but determined by spring/damper characteristics. The spring/damper characteristics are specified on a CONTFORC entry.		
	When the CONTFORC entry is specified, the value of FACT and DAMPING are not used.		

(Continued)

Remarks:

1. See also Section 2.6.3 on page 2-84.
2. The coefficient of friction is given by:

$$\mu = \mu_k + (\mu_s - \mu_k)e^{-\beta v}$$

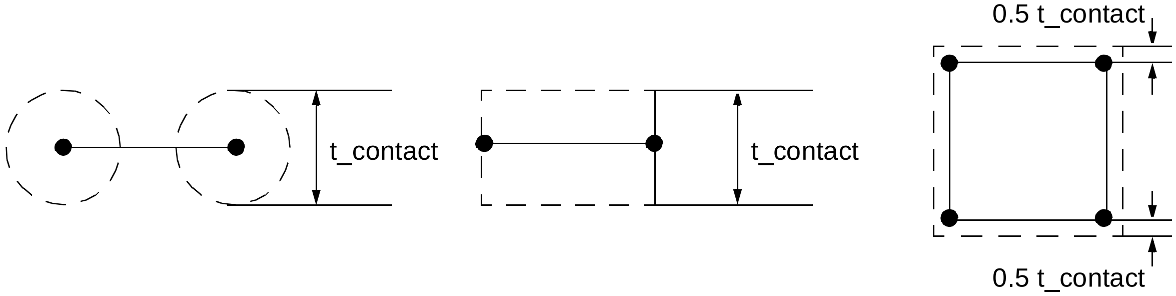
where μ_s = static coefficient of friction FS

μ_k = kinetic coefficient of friction FK

μ = exponential decay coefficient EXP

v = relative sliding velocity at the point of contact

3. When a non-zero value has been specified for THICK and/or GAP, a contact thickness will be assigned to both the slave nodes and the master faces:



Contact thickness for slave nodes

Contact thickness for master faces of shell elements

Contact thickness for master faces of Lagrangian solids

The contact thickness is equal to:

$$t\_contact = THICK * t\_shell + GAP$$

where t_contact = contact thickness

THICK = scale factor for shell thickness

t_shell = shell thickness

GAP = artificial contact thickness

(Continued)

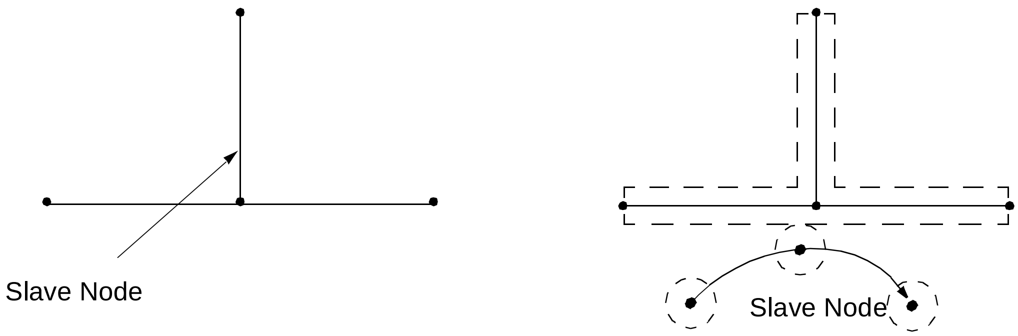
The shell thickness is zero for master faces of lagrangian solids.

The shell thickness for slave nodes is not calculated for STYPE = GRID.

For all other options, it is equal to the average thickness of all connected shell elements that are part of the slave "surface."

Contact will occur when the contact thickness of a slave node overlaps the contact thickness of a master face. This is the best physical contact representation of shell structures. There are also several other advantages to using a contact thickness:

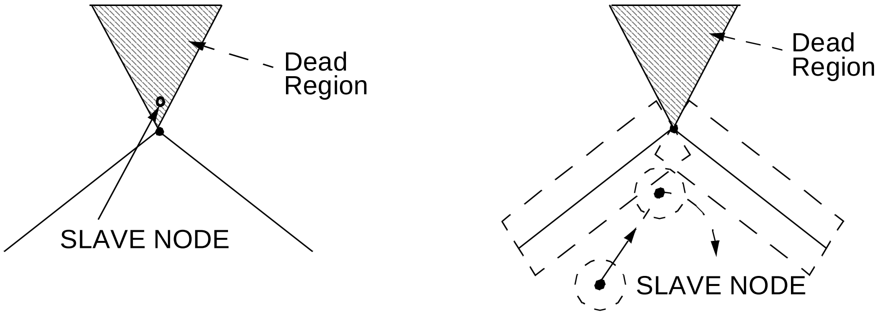
a. Use of a contact thickness prevents "hooking" in case of T-joints:



No contact thickness

With contact thickness

b. Use of a contact thickness prevents losing contacts in the "dead region" on the "penetrated side" of neighboring master faces. When a slave node enters the "dead region" between neighboring master faces it is not projected on either face, and the contact is lost:

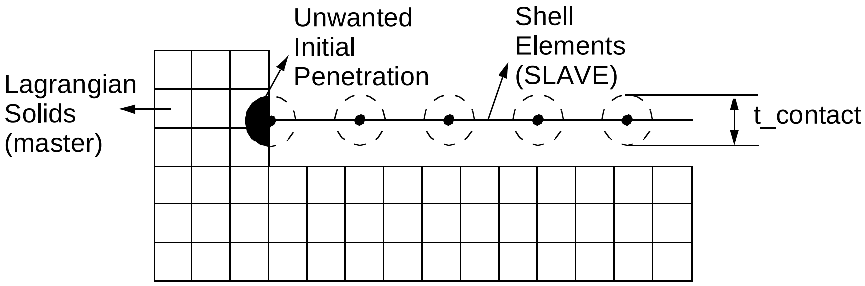


No contact thickness

With contact thickness

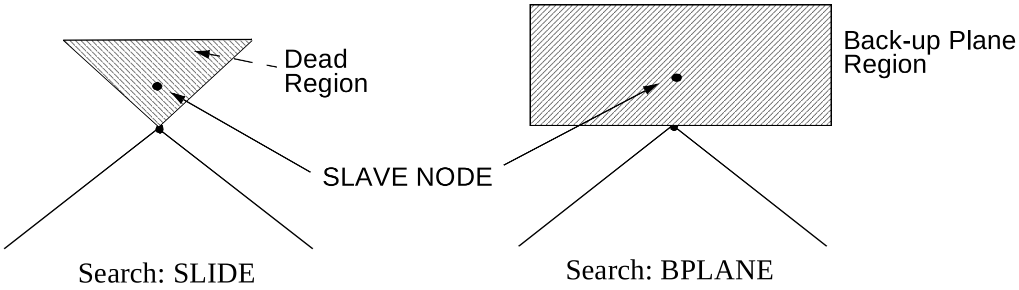
(Continued)

Using a contact thickness has the disadvantage that an unwanted initial penetration might occurs where the edge of shell elements meets a master surface. The following is a good example:



4. The search for the closest master face for a slave node is performed by a normal projection of the slave node on the faces of the master surface. For the search option FULL all faces of the master surface are taken into account. This is the most reliable option, but will take the most computational time.

   The SLIDE search option searches for the closest master face under the assumption that a slave node will only slide from the current master face to its neighbors during one time step. This search algorithm is much faster than the FULL option. However, this search option can cause problems for slave nodes that have entered the “dead-region” on the “penetrated side” between neigboring master faces, where it can not be projected on either face, and the contact is lost because of that.



   The BPLANE search option alleviates the problems for slave nodes that have entered the “dead-region” by creating automatically a plane perpendicular to the folding line between two master faces. Contact for slave nodes that reside in the back-up plane area is preserved and the nodes remain in the contact search algorithm. A force perpendicular to the BPLANE is applied to those nodes, since they are on the penetrated side of the master surface.

   Because the BPLANE search algorithm is very fast, it is the recommended contact search algorithm for airbag analysis.

5. A detailed description of the slave node (de)activation methods is given here. These methods are only available for VERSION = V4:

METHOD1 Nodes become active as slave once they reside on the outside of the mesh. In case of master slave contact, nodes on the master surface will not act as a slave.

Nodes will be deactivated as slave once all connected elements have failed.

METHOD2 Nodes become active as slave once they reside on the outside of the mesh. In case of master-slave contact, nodes on the master surface will not act as a slave.

Nodes will remain active as slave once all connected elements have failed.

METHOD3 Nodes are active as slave from the start of the calculation, independent of whether they reside on the inside or the outside of the mesh. In case of master-slave contact, nodes on the master surface will not act as a slave.

Nodes will be deactivated as slave once all connected elements have failed.

METHOD4 Nodes are active as slave from the start of the calculation, independent of whether they reside on the inside or the outside of the mesh. In case of master-slave contact, nodes on the master surface will not act as a slave.

Nodes will remain active as slave once all connected elements have failed.

METHOD1A For master-slave contact only. Nodes become active as slave once they reside on the outside of the mesh. Nodes on the master surface will also act as slave, once they reside on the outside of the mesh. This method is more suited for eroding master-slave contact than METHOD1.

Nodes will be deactivated as slave once all connected elements have failed.

METHOD2A For master-slave contact only. Nodes become active as slave once they reside on the outside of the mesh. Nodes on the master surface will also act as slave, once they reside on the outside of the mesh. This method is more suited for eroding master-slave contact than METHOD2.

Nodes will remain active as slave once all connected elements have failed.

METHOD3A For master-slave contact only. Nodes are active as slave from the start of the calculation, independent of whether they reside on the inside or the outside of the mesh. Nodes on the master surface will also act as slave, once they reside on the outside of the mesh. This method is more suited for eroding master-slave contact than METHOD3.

Nodes will be deactivated as slave once all connected elements have failed.

(Continued)

METHOD4A For master-slave contact only. Nodes are active as slave from the start of the calculation, independent of whether they reside on the inside or the outside of the mesh. Nodes on the master surface will also act as slave, once they reside on the outside of the mesh. This method is more suited for eroding master-slave contact than METHOD4.

Nodes will remain active as slave once all connected elements have failed.

Choosing the correct slave activity switch can be done by using the following flow schemes. The selection of a method depends on the desired results, and can be captured by three questions:

a. Only nodes on the outside of the mesh are active?

In most cases only the slave nodes on the outside need to be active.

In cases of high-velocity impact, it might be necessary to activate the internal slave nodes also. This will prevent missing contacts for slave nodes that move across the monitoring region of the master face during the time-step it is activated.

b. Deactivate slave nodes when all connected elements have failed?

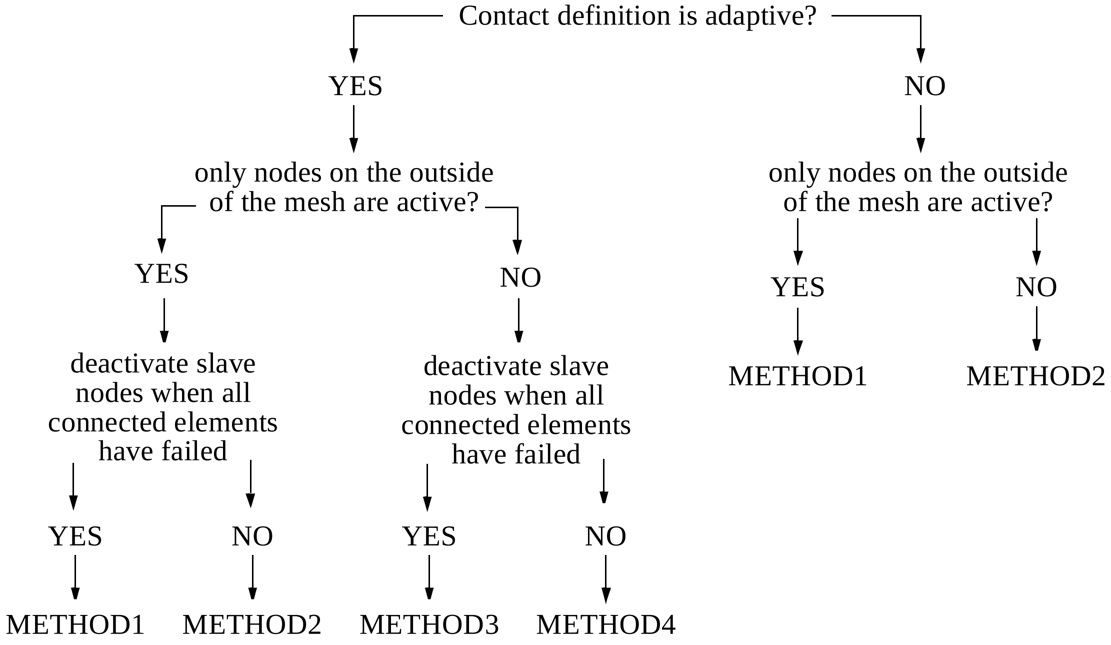
This determines whether slave nodes will remain active after all its connected elements have failed. This option only applies to an adaptive contact.

c. Nodes on the master surface will also act as slave?

When a master surface might fold onto itself this will prevent the master surface from penetrating itself. Therefore the master surface will behave as a single surface.

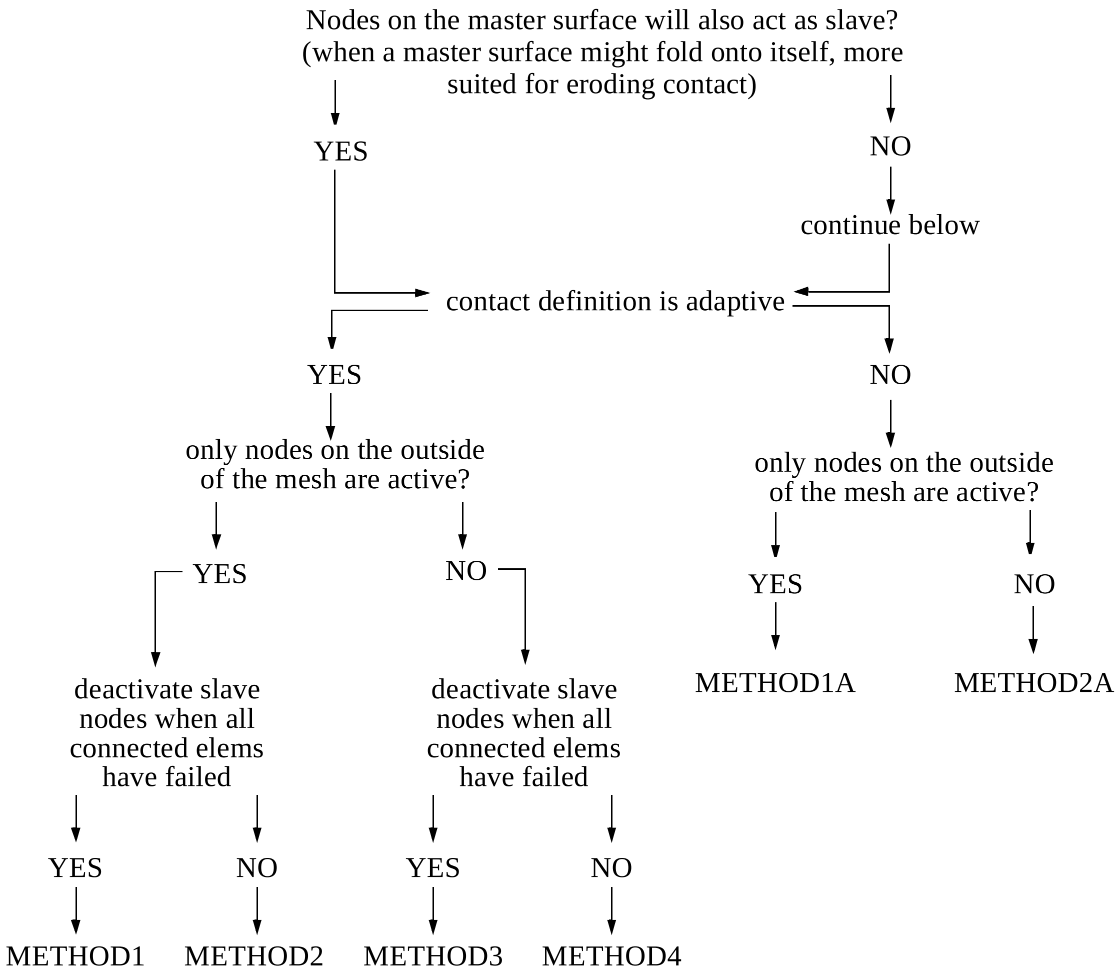
(Continued)

Flow scheme for a single surface contact:


Contact definition is adaptive?
YES
NO
only nodes on the outside of the mesh are active?
YES
NO
deactivate slave nodes when all connected elements have failed
YES
NO
METHOD1
METHOD2
deactivate slave nodes when all connected elements have failed
YES
NO
METHOD3
METHOD4
only nodes on the outside of the mesh are active?
YES
NO
METHOD1
METHOD2

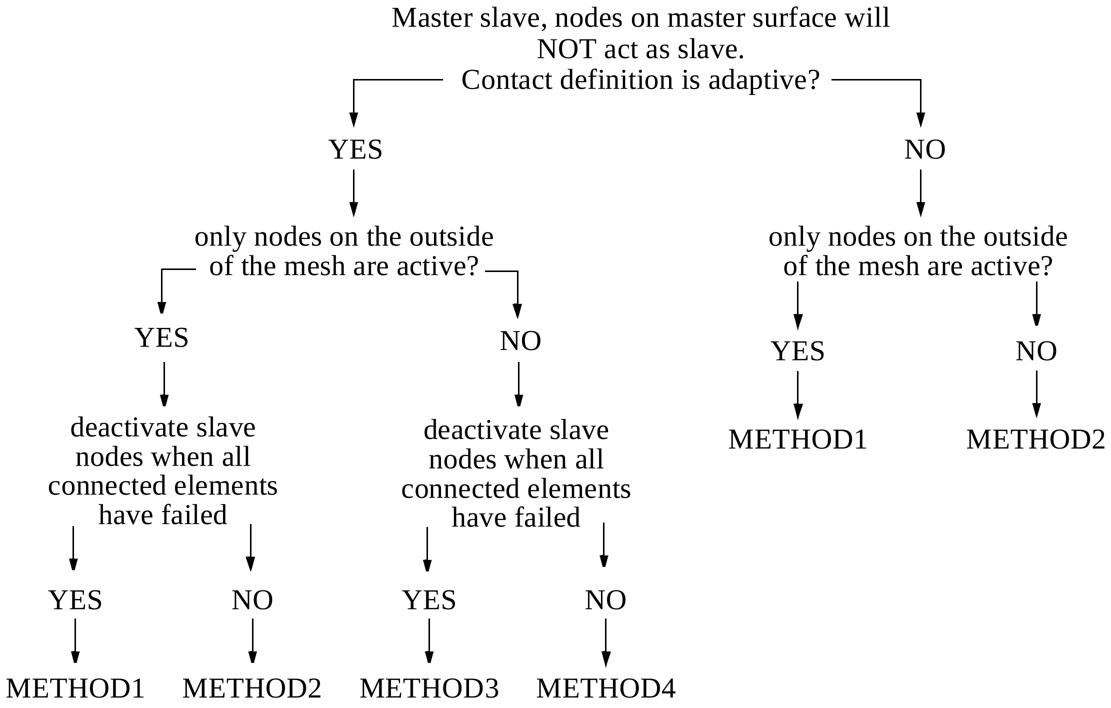
(Continued)

Flow scheme for a master slave contact:


Nodes on the master surface will also act as slave?
(when a master surface might fold onto itself, more suited for eroding contact)
YES
NO
continue below
contact definition is adaptive
YES
NO
only nodes on the outside of the mesh are active?
only nodes on the outside of the mesh are active?
YES
NO
YES
NO
deactivate slave nodes when all connected elems have failed
deactivate slave nodes when all connected elems have failed
METHOD1A
METHOD2A
YES
NO
YES
NO
METHOD1
METHOD2
METHOD3
METHOD4

(Continued)

Continuation:



CONTFORC

Contact Force Definition Using Force-Deflection Curves

The contact force is determined by force-deflection curves for loading and unloading. Damping can be specified either as a constant value or as a tabular function.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CONTFORC	CID	K	LOAD	UNLOAD	B-CONST	B-TABLE			
CONTFORC	9	1.E6				212			

Field	Contents	Type	Default
CID	Unique CONTFORC number, referenced from CONTACT entry.	I > 0	Required
K	Constant value for the contact stiffness. The contact force is calculated as: $F_{contact} = Kd$ where *d* is the penetration depth. The force acts in the direction normal to the master face. The same value will be used during loading and unloading.	R ≥ 0	See Remark 1.
LOAD	Number of a TABLED1 entry specifying the force versus penetration depth to be used when penetration increases. This is the loading phase.	I > 0	See Remark 1.
UNLOAD	Number of a TABLED1 entry specifying the force versus penetration depth to be used when penetration decreases. This is the unloading phase. By choosing a different unloading than loading curve, hysteresis can be modeled.	I > 0	Table number specified under LOAD.
B-CONST	Constant value of damper stiffness. The damper acts on the velocity difference between the slave node and the master face in the direction normal to the master face.	R ≥ 0	See Remark 2.
B-TABLE	Number of a TABLED1 entry specifying the damper stiffness. The damper acts on the velocity difference between the slave node and the master face in the direction normal to the master face.	I > 0	See Remark 2.

(Continued)

Remarks:

1. Either K or LOAD must be specified.
2. None, or just one of the options B-CONST, or B-TABLE must be specified.

CONTINI

Contact Initialization for In-Plane Folded Airbags

Defines the initial contact state between two SUBSURF entries. Used for contact initialization of in-plane folded airbags.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CONTINI	CID	INIID	SUBID1	SUBID2		LEVEL	SIDE	REVERSE	
CONTINI	1	79	53	54		1.0	BOTH	NO	

Field	Contents	Type	Default
CID	Unique number of a CONTINI entry.	I > 0	Required
INIID	Number of a set of CONTINI entries. INIID must be referenced from a CONTACT entry.	I > 0	Required
SUBID1	Number of a slave SUBSURF. The SUBSURF must be part of the slave SURFACE, referenced on the CONTACT entry. (In case of a single surface contact, it must be part of that SURFACE.)	I > 0	Required
SUBID2	Number of a master SUBSURF. The SUBSURF must be part of the master SURFACE, referenced on the CONTACT entry. (In case of a single surface contact, it must be part of that SURFACE.	I > 0	Required
LEVEL	Defines the LEVEL of a contact initialization.	R ≥ 0.0	1.0 See Remark 2.
SIDE	Defines the side of the contact that will be accepted. BOTH Contact from both sides is accepted TOP Only contact from the TOP side is accepted BOTTOM Only contact from the BOTTOM side is accepted	C	See Remark 1.
REVERSE	Defines if the reverse CONTINI must be generated: YES A reversed CONTINI will be generated with SUBID2 as the slave and SUBID1 as the master: CONTINI,--,INIID,SUBID2,SUBID1 NO A reversed CONTINI will not be generated.	C	NO

(Continued)

Remarks:

1. By default the SIDE is equal to the SIDE as specified on the CONTACT entry.

2. It is allowed to have multiple CONTINIs defined for a slave subsurface. If a grid point of a slave subsurface (SUBID1) finds a contact in more than one master subsurface, only the ones with the highest level will be accepted.

   For example, suppose a subsurf is used as slave in three CONTINI definitions:

   1) CONTINI,1,101,SUBID1,SUBID2,,LEVEL1

   2) CONTINI,2,101,SUBID1,SUBID3,,LEVEL2

   3) CONTINI,3,101,SUBID1,SUBID4,,LEVEL3

   When a slave node of SUBSURF,SUBID1 finds a contact in all three master SUBSURFs. The following logic will apply:

   LEVEL1=LEVEL2=LEVEL3 → all contacts are accepted

   LEVEL1>LEVEL2,LEVEL3 → only contact 1 is accepted

   LEVEL1=LEVEL2>LEVEL3 → only contact 1 and 2 are accepted

   LEVEL1,LEVEL2<LEVEL3 → only contact 3 and 2 are accepted

   LEVEL1<LEVEL2>LEVEL3 → only contact 2 is accepted

3. The options LEVEL, SIDE, and REVERSE are only valid for contact version V4.

CONTREL

Contact with Rigid Ellipsoids

Defines contact between rigid ellipsoids and Lagrangian grid points or rigid bodies.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CONTREL	CID	SIDC	TYPE	SID	ARF				
CONTREL	20	30	GRID	40					

Field	Contents	Type	Default
CID	Unique contact number.	I > 0	Required
SIDC	Number of a SETC entry giving a list of the names of rigid ellipsoids on which contact can occur.	I > 0	Required
TYPE	The type of entity that can contact the rigid ellipsoids.	C	Required
	GRID Grid points.		
	RIGID Rigid bodies.		
SID	The number of a SET1 entry listing the grid points or rigid bodies that can contact the rigid ellipsoids. (See also Remark 2.)	I > 0	Required
ARF	Artificial restoration factor. This is the factor by which penetrated grid points are moved back to the surface of the ellipsoids. A value of 0 indicates that they are not moved. A value of 1 indicates that they are moved all the way back to the surface of the ellipsoid.	0.0 ≤ R ≤ 1.0	See Remark 1.

Remarks:

1. For grid points attached to Lagrangian elements, the default for ARF is 1.0. For rigid bodies, the default is 0.1.

2. All types of rigid bodies, i.e., rigid surfaces, MATRIGs and RBE2-FULLRIGs can be defined in the SET1 entry. Rigid surfaces are referenced by their number: MATRIGs are referenced as MR<id> and RBE2-FULLRIGs as FR<id>, where id is the MATRIG or RBE2-FULLRIG number, respectively.

CORD1C

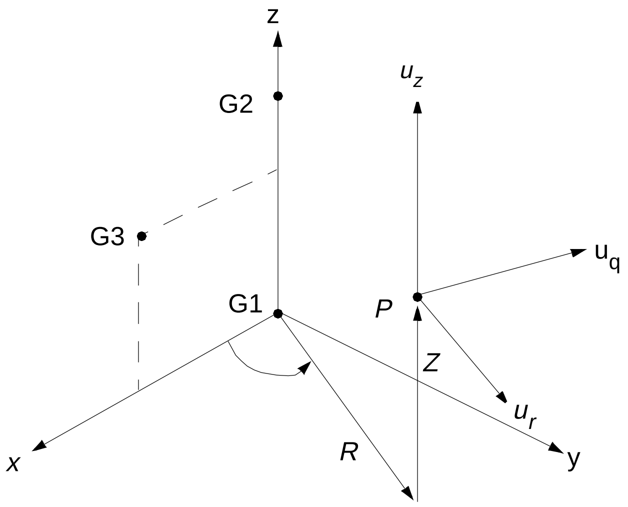
Cylindrical Coordinate System Definition, Form 1

Defines up to two cylindrical coordinate systems per entry by referencing three grid points that define a coordinate system. The grid points must be defined in a coordinate system other than the coordinate system that is being defined. The first grid point is the origin, the second lies on the z-axis, and the third lies in the plane of the azimuthal origin.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CORD1C	CID	G1	G2	G3	CID2	G4	G5	G6	
CORD1C	3	16	321	19					

Field	Contents	Type	Default
CID	Coordinate-system number.	I > 0	Required
G1, G2, G3	Grid-point numbers G1, G2, and G3. The grid-point numbers must be unique.	I > 0	Required
CID2	Optional second coordinate-system number.	I > 0	Blank
G4, G5, G6	Grid-point numbers G4, G5, and G6. The grid-point numbers must be unique.	I > 0	Blank



CORD1C Definition.

(Continued)

Remarks:

1. All coordinate-system numbers must be unique.
2. The three grid points G1, G2, and G3 must not be collinear.
3. The location of a grid point in the coordinate system is given by (R, θ, Z) where θ is measured in radians.
4. The velocity-component directions at P depend on the location of P as shown above by U_r, U_θ, and U_z, when the coordinate system is used in a motion prescription.
5. One or two coordinate systems may be defined on a single line.

CORD2C

Cylindrical Coordinate System Definition, Form 2

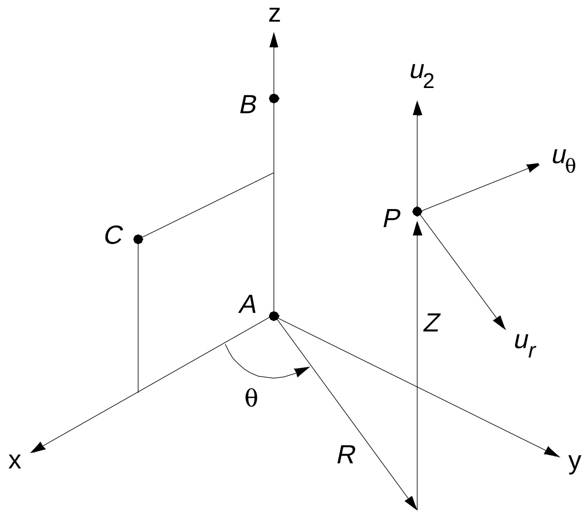
Defines a cylindrical coordinate system by referencing the coordinates of three grid points. The first point defines the origin, the second defines the direction of the z-axis, and the third lies in the plane of the azimuthal origin. The reference coordinate system must be independently defined.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CORD2C	CID	RID	A1	A2	A3	B1	B2	B3	+CONT1
CORD2C	3	17	–2.9	1.0	0.0	3.6	0.0	1.0	+CONT1

+CONT1	C1	C2	C3						
+CONT1	5.2	1.0	–2.9						

Field	Contents	Type	Default
CID	Coordinate-system number.	I > 0	Required
RID	Reference coordinate system that is defined independent of the new coordinate system.	I > 0	0
A1, A2, A3 B1, B2, B3 C1, C2, C3	Coordinates of three points in the coordinate system referenced by RID.	R	0.0



CORD2C Definition.

(Continued)

Remarks:

1. The continuation line must be present.
2. The three points (A1, A2, A3), (B1, B2, B3), and (C1, C2, C3) must be unique and must not be collinear.
3. All coordinate-system numbers must be unique.
4. The location of a grid point in the coordinate system is given by (R, θ, Z) where θ is measured in degrees.
5. The velocity-component directions at P depend on the location of P as shown above by U_r, U_θ, and U_z when the coordinate system is used in a motion prescription.
6. An RID of zero references the basic coordinate system.

CORD1R

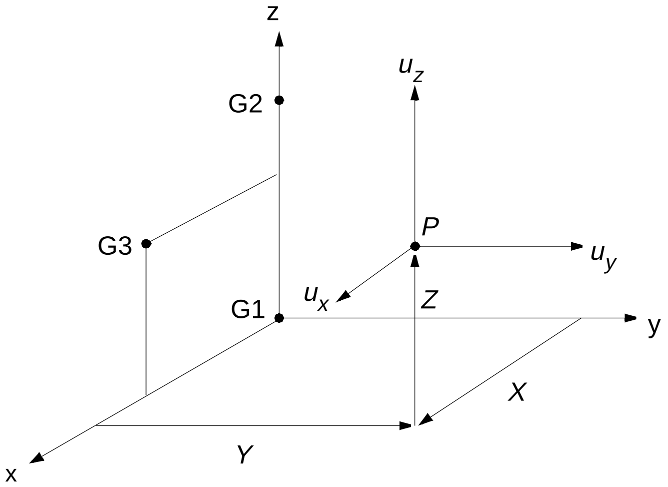
Rectangular Coordinate System Definition, Form 1

Defines up to two rectangular coordinate systems per entry by referencing three grid points that define a coordinate system. The grid points must be defined in a coordinate system other than the coordinate system that is being defined. The first grid point is the origin, the second lies on the z-axis, and the third lies in the x-z plane.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CORD1R	CID	G1	G2	G3	CID2	G4	G5	G6	
CORD1R	3	16	321	19					

Field	Contents	Type	Default
CID	Coordinate-system number.	I > 0	Required
G1, G2, G3	Grid-point numbers G1, G2, and G3. The grid points must be unique.	I > 0	Required
CID2	Optional second coordinate-system number	I > 0	Blank
G4, G5, G6	Grid-point numbers G4, G5, and G6. The grid-point numbers must be unique.	I > 0	Blank



CORD1R Definition.

(Continued)

Remarks:

1. All coordinate-system numbers must be unique.
2. The three grid points G1, G2, and G3 must not be collinear.
3. The location of a grid point in this coordinate system is given by (X, Y, Z).
4. The velocity-component directions at P depend on the location of P as shown above by U_x, U_y, and U_z, when the coordinate system is used in a motion prescription.
5. One or two coordinate systems may be defined on a single line.

CORD2R

Rectangular Coordinate System Definition, Form 2

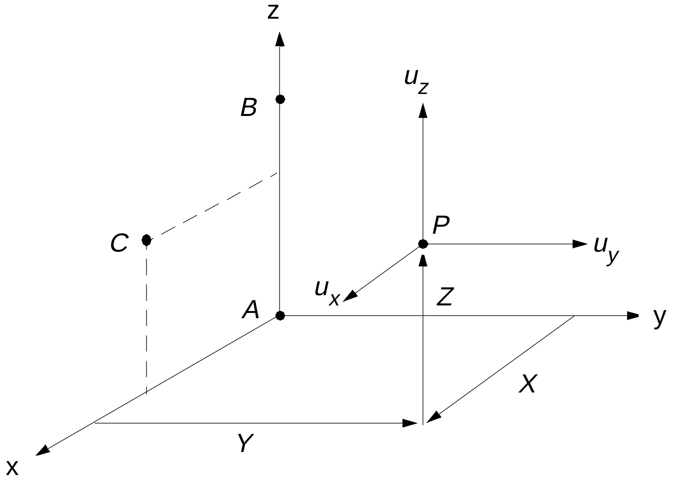
Defines a rectangular coordinate system by referencing the coordinates of three points. The first point defines the origin, the second defines the direction of the z-axis, and the third defines a vector that, with the z-axis, defines the x-z plane. The reference coordinate system must be independently defined.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CORD2R	CID	RID	A1	A2	A3	B1	B2	B3	+CONT1
CORD2R	3	17	–2.9	1.0	0.0	3.6	0.0	1.0	+CONT1

+CONT1	C1	C2	C3						
+CONT1	3.14	.1592	.653						

Field	Contents	Type	Default
CID	Coordinate-system number.	I > 0	Required
RID	Reference coordinate system that is defined independent of the new coordinate system.	I > 0	0
A1, A2, A3 B1, B2, B3 C1, C2, C3	Coordinate of three points in the coordinate system referenced by RID.	R	0.0



CORD2R Definition.

(Continued)

Remarks:

1. The continuation line must be present.
2. The three points (A1, A2, A3), (B1, B2, B3), and (C1, C2, C3) must be unique and must not be collinear.
3. All coordinate-system numbers must be unique.
4. The location of a grid point in this coordinate system is given by (X, Y, Z).
5. The velocity-component directions at P depend on the location of P as shown above by U_x, U_y, and U_z, when the coordinate system is used in a motion prescription.
6. An RID of zero references the basic coordinate system.

CORD3R

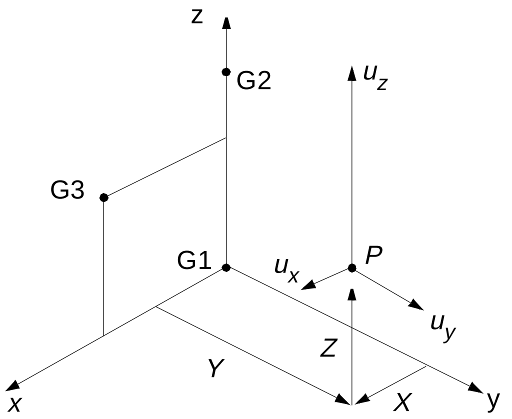
Moving Rectangular Coordinate System Definition, Form 1

Defines a rectangular coordinate system by referencing three grid points. The grid points must be defined in an independent coordinate system. The first grid point is the origin, the second lies on the z-axis, and the third lies in the x-z plane. The position and orientation of the coordinate system is updated as the grid points move.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CORD3R	CID	G1	G2	G3	CID	G1	G2	G3	
CORD3R	3	16	321	19					

Field	Contents	Type	Default
CID	Coordinate-system number.	I > 0	Required
G1, G2, G3	Grid-point numbers G1, G2, and G3 must be unique.	I > 0	Required



CORD3R Definition.

Remarks:

1. All coordinate-system numbers must be unique.
2. The three points G1, G2, G3 must not be collinear.
3. The location of a grid point (*P* in the figure) in this coordinate system is given by (*X*, *Y*, *Z*).
4. The displacement coordinate directions at P are shown above by u_x, u_y, and u_z.
5. One or two coordinate systems may be defined on a single line.
6. The orientation of the coordinate system is updated each time step based on the motion of the grid points.

CORD4R

Moving Rectangular Coordinate System Definition, Form 2

Defines a rectangular coordinate system by referencing the coordinates of three points. The first point defines the origin, the second defines the direction of the z-axis, and the third defines a vector that, with the z-axis, defines the x-z plane. The position and orientation of the coordinate system moves during the analysis by prescribed translation and rotation.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CORD4R	CID	RID	A1	A2	A3	B1	B2	B3	+CONT1
CORD4R	3	17	–2.9	1.0	0.0	3.6	0.0	1.0	+CONT1

+CONT1	C1	C2	C3						+CONT2
+CONT1	5.2	1.0	–2.9						+CONT2

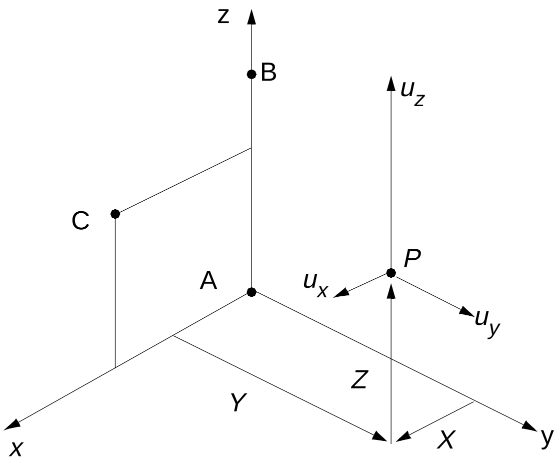
+CONT2	TTX	TTY	TTZ	TRX	TRY	TRZ			
+CONT2	33								

Field	Contents	Type	Default
CID	Coordinate-system number.	I > 0	Required
RID	Reference coordinate system that is defined independently of the new coordinate system.	I > 0	0
A1, A2, A3 B1, B2, B3 C1, C2, C3	Coordinates of three points in the basic coordinate system.	R	0.0
TTX, TTY, TTZ	Number of TABLED1 entries defining the velocity of the origin of the coordinate system in the x-, y-, z-directions of the basic coordinate system.	I > 0	Fixed
TRX, TRY, TRZ	Number of TABLED1 entries defining the angular velocity of the coordinate system about the x-, y-, z-axes of the basic coordinate system.	I > 0	Fixed

(Continued)

CORD4R

Moving Rectangular Coordinate System Definition, Form 2



CORD4R Definition.

Remarks:

1. The continuation line must be present.
2. The three points (A1, A2, A3), (B1, B2, B3), and (C1, C2, C3) must be unique and must not be collinear.
3. All coordinate-system numbers must be unique.
4. The location of a grid point (*P* in the figure) in this coordinate system is given by (*X*, *Y*, *Z*).
5. The displacement coordinate directions at *P* are shown by u_x, u_y, and u_z.

CORD1S

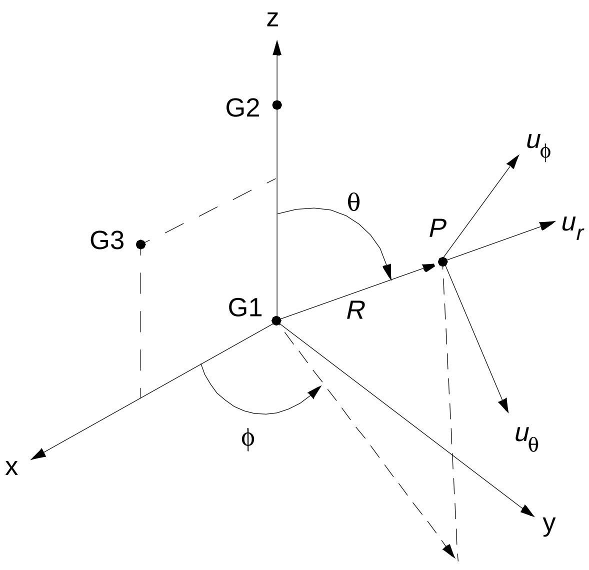
Spherical Coordinate System Definition, Form 1

Defines up to two spherical coordinate systems per entry by referencing three grid points that define a coordinate system. The grid points must be defined in an independent coordinate system. The first grid point is the origin. The second lies on the z-axis. The third lies in the plane of the azimuthal origin.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CORD1S	CID	G1	G2	G3	CID2	G4	G5	G6	
CORD1S	3	16	321	19					

Field	**Contents**	**Type**	**Default**
CID	Coordinate-system number.	I > 0	Required
G1, G2, G3	Grid-point numbers G1, G2, and G3. The grid points must be unique.	I > 0	Required
CID2	Optional second coordinate-system number.	I > 0	Blank
G4, G5, G6	Grid-point numbers G4, G5, and G6. The grid points must be unique.	I > 0	Blank



CORD1S Definition.

(Continued)

Remarks:

1. All coordinate system numbers must be unique.
2. The three grid points G1, G2, and G3 must not be collinear.
3. The location of a grid point in this coordinate system is given by (R, θ, ϕ) where θ and ϕ are measured in degrees.
4. The velocity-component directions at P depend on the location of P as shown above by U_r, U_θ, and U_ϕ, when the coordinate system is used in a motion prescription.
5. Grid points on the polar axis may not have their displacement directions defined in this coordinate system, since an ambiguity results.
6. One or two coordinate systems may be defined on a single line.

CORD2S

Spherical Coordinate System Definition, Form 2

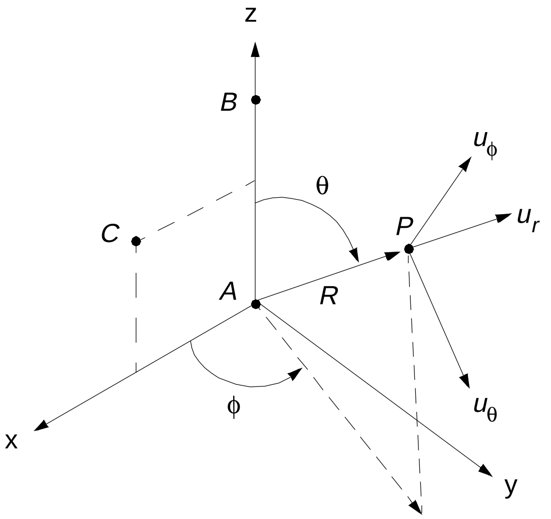
Defines a spherical coordinate system by referencing the coordinates of three points. The first point defines the origin, the second defines the direction of the z-axis, and the third lies in the plane of the azimuthal origin. The reference coordinate system must be independently defined.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CORD2S	CID	RID	A1	A2	A3	B1	B2	B3	+CONT1
CORD2S	3	17	–2.9	1.0	0.0	3.6	0.0	1.0	+CONT1

+CONT1	C1	C2	C3						
+CONT1	5.2	1.0	–2.9						

Field	Contents	Type	Default
CID	Coordinate-system number.	I > 0	Required
RID	Reference coordinate system that is defined independently of the new coordinate system.	I > 0	0
A1, A2, A3 B1, B2, B3 C1, C2, C3	Coordinates of three points in the coordinate system referenced by RID.	R	0.0



CORD2S Definition.

(Continued)

Remarks:

1. The continuation line must be present.
2. The three points (A1, A2, A3), (B1, B2, B3) and (C1, C2, C3) must be unique and must not be collinear.
3. All coordinate system numbers must be unique.
4. The location of a grid point in this coordinate system is given by (R, θ, ϕ) where θ and ϕ are measured in degrees.
5. The velocity-components directions at P depend on the location of P as shown above by U_r, U_θ, and U_ϕ, when the coordinate system is used in a motion prescription.
6. Grid points on the polar axis may not have their displacement directions defined in this coordinate system, since an ambiguity results.
7. An RID of zero references the basic coordinate system.

CORDROT

Corotational-Frame Definition

Defines the direction of corotational axes in a material.

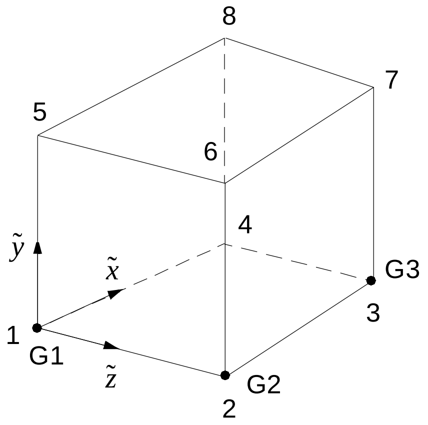
Format and Example:

1	2	3	4	5	6	7	8	9	10
CORDROT	CID	G1	G2	G3					
CORDROT	100	1	2	3					

Field	Contents	Type	Default
CID	Unique coordinate-system number referred to by a DMAT or DMATEL Bulk Data entry.	I > 0	Required
G1, G2, G3	Relative grid-point numbers of elements of DMAT and DMATEL referring to this entry defining the orientation of the corotational frame.	1 ≤ I ≤ 8	1,5,2

Remarks:

1. The DMAT and DMATEL entries can refer to this type of coordinate system.
2. G1 defines the origin, G2 lies on the corotational z-axis, and G3 lies in the corotational (X-Z) plane.



Element Corotational Frame According to the Example Given Above.

3. The orientation of the element corotational frame is updated according to the spin of the element.
4. If the fields G_1, G_2, G_3 are left blank, the default applies.

COUHTR

Heat Transfer Model to be Used with COUPLE Entry

Defines an heat transfer model suited for Euler Coupled analyses. The heat transfer model is defined as part of the coupling surface.

Format and Example:

1	2	3	4	5	6	7	8	9	10
COUHTR	CID	HTRID	SUBID	HTRTYPE	HTRTYPID	COEFF	COEFFV		
COUHTR	100	1	2	3					

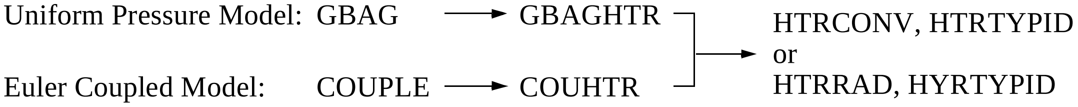
Field	Contents		Type	Default
CID	Unique number of a COUHTR entry.		I > 0	Required
HTRID	Number of a set of COUHTR entries HTRID must be referenced from a COUPLE entry.		I > 0	Required
SUBID	> 0	Number of a SUBSURF, which must be a part of the SURFACE referred to from the COUPLE entry.	I ≥ 0	0
	= 0	COUHTR definitions are used for the entire SURFACE referred to from the COUPLE entry.		
HTRTYPE	Defines the type of heat transfer.		C	
	HTRCONV	The HTRCONV logic is used to model heat transfer through convection in an air bag. The area of convection is defined by a subsurface (SUBID). The area of convection through which the energy is transported is equal to the area of the subsurface multiplied by COEFFV. A value of COEFFV = 1.0 will expose the complete subsurface area, while a value of COEFFV = 0.0 will result in no heat transfer through the subsurface.		

(Continued)

Field	Contents		Type	Default
	HTRRAD	The HTRRAD logic is used to model heat transfer through radiation in an air bag. The area of convection is defined by a subsurface (SUBID). The area of convection through which the energy is transported is equal to the area of the subsurface multiplied by COEFFV. A value of COEFFV = 1.0 will expose the complete subsurface area, while a value of COEFFV = 0.0 will result in no heat transfer through the subsurface.		
HTRTYPID	ID of the entry selected under HTRTYPE, e.g., HTRCONV, HTRTYPID.		I	Required
COEFF	Method of defining the area coefficient.		C	CONSTANT
	CONSTANT	The area coefficient is constant and specified on COEFFV.		
	TABLE	The area coefficient varies with time. COEFV is the number of a TABLED1 or TABLEEX entry giving the variation with time.		
COEFFV	The area coefficient or the number of a TABLED1 or TABLEEX entry depending on the COEFF entry.		0 < R < 1 or I > 0	1.0

Remarks:

1. The same HTRTYPE entry referenced from this COUHTR entry can be referenced by a GBAGHTR entry. This allows for setting up the exact same model for either a uniform pressure model or an Euler Coupled model:



Uniform Pressure Model: GBAG → GBAGHTR

Euler Coupled Model: COUPLE → COUHTR

→ HTRCONV, HTRTYPID
or
HTRRAD, HYRTYPID

   This makes it possible to set up the model using the switch from full gas dynamics to uniform pressure (GBAGCOU).

2. A mixture of multiple COUHTR with different HTRTYPEs is allowed.
3. For the same SUBSURF multiple different types of heat transfer may be defined.
4. A more detailed description can be found in Section 2.17 on page 2-127.

COUINFL

Inflator Model to be Used with COUPLE Entry

Defines an inflator model suited for Euler Coupled analyses. The inflator model is defined as part of the coupling surface.

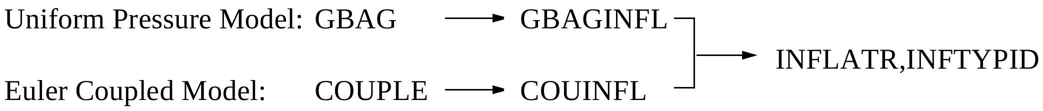
1	2	3	4	5	6	7	8	9	10
COUINFL	CID	INFID	SUBID	INFTYPE	INFTYPID	COEFF	COEFFV		
COUINFL	112	14	1204	INFLATR	80		0.012		

Field	Contents		Type	Default
CID	Unique number of a COUINFL entry.		I > 0	Required
INFID	Number of a set of COUINFL entries INFID must be referenced from a COUPLE entry.		I > 0	Required
SUBID	Number of a SUBSURF, which must be a part of the SURFACE referred to from the COUPLE entry.		I > 0	Required
INFTYPE	Defines the type of inflator.		C	
	INFLATR	The INFLATR logic is used to model inflators in an air bag. The inflator is defined by a subsurface (SUBID). The area of the hole through which the gas enters is equal to the area of the subsurface multiplied by COEFFV. A value of COEFFV = 1.0 will open up the complete subsurface area, while a value of COEFFV = 0.0 will result in a closed inflator area with no inflow.		
INFTYPID	ID of the entry selected under INFTYPE, for example, INFLATR,INFTYPID.		I	Required
COEFF	Method of defining the area coefficient.		C	CONSTANT
	CONSTANT	The area coefficient is constant and specified on COEFFV.		
	TABLE	The area coefficient varies with time. COEFV is the number of a TABLED1 or TABLEEX entry giving the variation with time.		
COEFFV	The area coefficient or the number of a TABLED1 or TABLEEX entry depending on the COEFF entry.		0.0 < R < 1.0 or I > 0	1.0

(Continued)

Remarks:

1. The same INFTYPE entry referenced from this COUINFL entry can be referenced by a GBAGINFL entry. This allows for setting up the exact same model for either a uniform pressure model or an Euler Coupled model:

   This makes it possible to set up the model using the switch from full gas dynamics to uniform pressure (GBAGCOU).

2. One couple entry can reference more than one COUINFL entry. This allows for modeling multiple inflators in an airbag module.

COUOPT

Coupling Options

Defines the interaction factor and a pressure load from the covered side acting on a (SUB)SURFACE.

Format and Example:

1	2	3	4	5	6	7	8	9	10
COUOPT	CID	OPTID	SUBID	FACTOR	FACTORV				+CONT1
COUOPT	1	80	42	CONSTANT	0				+CONT1

+CONT1	PLCOVER	PLCOVERV							
+CONT1	CONSTANT	1.E5							

Field	Contents		Type	Default
CID	Unique number of a COUOPT entry.		I > 0	Required
OPTID	Number of a set of COUOPT entries. OPTID must be referenced from a COUPLE entry.		I > 0	Required
SUBID	> 0	Number of a SUBSURF, which must be part of the SURFACE.	I ≥ 0	0
	= 0	COUOPT definitions used for the entire SURFACE.		
FACTOR	Method of defining the interaction FACTORV with which the Eulerian pressure acting on the SURFACE is multiplied.		C	CONSTANT
	CONSTANT	The FACTOR is constant and specified in FACTORV.		
FACTORV	The interaction factor.		R	1.
PLCOVER	Method of defining the pressure load exerted on the (SUB)SURFACE from the covered side. The pressure load is applied only when the Eulerian pressure is greater than zero.		C	CONSTANT
	CONSTANT	The PLCOVER is constant and specified in PLCOVERV.		
	TABLE	The PLCOVER varies with time. PLCOVERV is the number of a TABLED1 or TABLEEX entry giving the variation of the PLCOVER (y-value) with time (x-value).		
PLCOVERV	The pressure load or the number of a TABLED1 or TABLEEX entry depending on the PLCOVER entry.		R ≥ 0	0.

(Continued)

Remarks:

1. The effect of specifying an interaction FACTOR is similar to specifying a porosity coefficient on a COUPOR entry. The difference is that in this case the (SUB)SURFACE still acts as a wall boundary for the Eulerian material.

2. Applying a PLCOVER instead of applying a pressure load on the faces through either a PLOAD, PLOAD4, or DAREA entry gives the following differences:

   a. PLCOVER is applied only when there is a balancing Eulerian pressure greater than zero.

   b. Possible porosity as defined on a COUPOR entry is taken into account when applying the PLCOVER.

   c. With PARAM,PLCOVCUT you can define a cut-off time that is applied to PLCOVER.

3. The covered side of a SURFACE lies on the side where there is no Eulerian material. See Section 2.9 on page 2-101.

COUP1FL

Coupling Surface Failure

Defines the surrounding variables when a segment of a coupling surface will fail.

Format and Example:

1	2	3	4	5	6	7	8	9	10
COUP1FL	CFID	RHO	SIE	XVEL	YVEL	ZVEL			
COUP1FL	3	1.225	204082.	900.					

Field	Contents	Type	Default
CFID	Unique number of a COUP1FL entry.	I > 0	Required
RHO	Surrounding density.	R > 0.	See Remark 2.
SIE	Surrounding specific internal energy.	R	See Remark 2.
XVEL	Surrounding x-velocity.	R	See Remark 2.
YVEL	Surrounding y-velocity.	R	See Remark 2.
ZVEL	Surrounding z-velocity.	R	See Remark 2.

Remarks:

1. This entry can only be used in combination with PARAM,FASTCOUP, ,FAIL as well as PARAM,LIMITER,ROE and when failure of the Lagrangian structure on which the coupling surface lies is defined.
2. At least one of the surrounding variables should be defined. The default value of RHO will be equal to the reference RHO on the DMAT entry and the other variables (SIE, XVEL, YVEL and ZVEL) will by default be equal to zero.
3. The coupling surface should consist out CQUADs and/or CTRIAs.

Defines the interaction between two coupling surfaces.

Format and Example:

1	2	3	4	5	6	7	8	9	10
COUP1INT	CIID	CID1	CID2						
COUP1INT	33	2	5						

Field	Contents	Type	Default
CIID	Unique number of a COUP1INT entry.	I > 0	Required
CID1	Number of COUPLE1 entry 1.	I > 0	Required
CID2	Number of COUPLE1 entry 2.	I > 0	Required

Remarks:

1. This entry can only be used in combination with PARAM,FASTCOUP, ,FAIL as well as PARAM,LIMITER,ROE and when failure of the Lagrangian structure on which the coupling surface lies is defined.

2. The coupling surface should consist out CQUADs and/or CTRIAs.

COUPLE

General Euler-Lagrange Coupling Surface

Defines a coupling surface that acts as the interface between an Eulerian and a Lagrangian mesh.

Format and Example:

1	2	3	4	5	6	7	8	9	10
COUPLE	CID	SID	COVER	REVERSE	CHECK	PORID	OPTID	CTYPE	+CONT1
COUPLE	100	37							+CONT1

+CONT1	INFID	HTRID							
+CONT1									

Field	Contents		Type	Default
CID	Unique number of a COUPLE entry.		I > 0	Required
SID	Number of a SURFACE entry defining the coupling surface.		I > 0	Required
COVER	The processing strategy for Eulerian elements inside and outside of the coupling surface.		C	INSIDE
	INSIDE	The Eulerian elements inside the closed volume of the coupling surface are not processed.		
	OUTSIDE	The Eulerian elements outside the closed volume of the coupling surface are not processed.		
REVERSE	Autoreverse switch for coupling-surface segments.		C	ON
	ON	If necessary, the normals of the coupling-surface segments are automatically reversed so that they all point in the same general direction and give a positive closed volume.		
	OFF	The segment normals are not automatically reversed.		
CHECK	Checking switch for coupling-surface segments.		C	ON
	ON	The normals of the segments are checked to see whether they all point in the same general direction and give a positive closed volume.		
	OFF	The segment normals are not checked.		

(Continued)

Field	Contents	Type	Default
	When REVERSE is set to ON, CHECK is automatically set to ON.		
PORID	Number of a set of COUPOR entries, which define porosity for the SURFACE and SUBSURF entries.	I	No porosity.
OPTID	Number of a set of COUOPT entries, which define special options for the SURFACE and SUBSURF entries.	I	No special options.
CTYPE	Type of coupling surface. STANDARD Standard Euler-Lagrange coupling. AIRBAG Coupling for air-bag applications. It is equivalent to the standard coupling algorithm with the following exceptions that make the solution procedure more stable for air-bag applications: • Inflow through a porous (sub)surface will only occur when there is already some gas in the Eulerian element. • Almost empty Eulerian elements will be eliminated. The standard algorithm redistributes small masses to the most suited neighbor element.	C	STANDARD
INFID	Number of a set of COUINFL entries, that defines the one or more inflators on subsurface(s) of the coupling surface.		
HTRID	Number of a set of COUHTR entries, that defines the one or more heat transfer definitions on (sub)surface(s) of the coupling surface.		

Remarks:

1. All coupling surfaces must form a multifaceted closed volume. If necessary, additional segments must be specified to achieve this. This closed volume must intersect at least one Euler element.
2. All segments must be attached to the face of an element. Dummy elements must be used to define any additional segments that are required to create a closed volume.

(Continued)

3. The normals of all the segments that form the coupling surface must point in the same general direction and result in a positive closed volume. Setting the REVERSE field to ON ensures that this condition is satisfied, regardless of how the segments are defined initially.

4. The COVER field determines how Eulerian elements that are inside and outside of the coupling surface are processed. The default setting of INSIDE is appropriate for most problems, since in the majority of analyses, the Eulerian material flows around the outside of the coupling surface. Therefore, the Eulerian elements within the coupling surface are empty. However, for some specialized applications (such as air bag inflation), the Eulerian material is completely contained within the coupling surface. In these cases, COVER should be set to OUTSIDE.

5. To get the fast coupling algorithm use PARAM,FASTCOUP.

COUPLE1

Euler-Lagrange Coupling Surface

Defines a coupling surface that acts as the interface between an Eulerian and a Lagrangian mesh for the Roe solver for single hydrodynamic materials.

Format and Example:

1	2	3	4	5	6	7	8	9	10
COUPLE1	CID	SID	COVER	REVERSE	CHECK				+CONT1
COUPLE1	23	4							+CONT1

+CONT1	SET1ID	MESHID	TDEAC	COUP1FL					

Field	Contents		Type	Default
CID	Unique number of a COUPLE entry.		I > 0	Required
SID	Number of a SURFACE entry defining the coupling surface.		I > 0	Required
COVER	The processing strategy for Eulerian elements inside and outside of the coupling surface.		C	INSIDE
	INSIDE	The Eulerian elements inside the closed volume of the coupling surface are not processed.		
	OUTSIDE	The Eulerian elements outside the closed volume of the coupling surface are not processed.		
REVERSE	Autoreverse switch for coupling-surface segments.		C	ON
	ON	If necessary, the normals of the coupling-surface segments are automatically reversed so that they all point in the same general direction and give a positive closed volume.		
	OFF	The segment normals are not automatically reversed.		

(Continued)

Field	**Contents**	**Type**	**Default**
CHECK	Checking switch for coupling-surface segments. ON The normals of the segments are checked to see whether they all point in the same general direction and give a positive closed volume. OFF The segment normals are not checked. When REVERSE is set to ON, CHECK is automatically set to ON.	C	ON
SET1ID	The number of a SET1 entry, which defines the Eulerian region when multiple coupling surfaces are defined.	I > 0	See Remark 7.
MESHID	The number of a MESH entry, which defines the Eulerian region when multiple coupling surfaces are defined.	I > 0	See Remark 7.
TDEAC	Time of deactivation of the coupling surface and its Eulerian region.	R > 0	1.E20
COUP1FL	The number of a COUP1FL entry, which defines the surrounding variables for the coupling surface when its segments fail.	I > 0	See Remark 8.

Remarks:

1. All coupling surfaces must form a multifaceted closed volume. If necessary, additional segments must be specified to achieve this.

2. All segments must be attached to the face of an element. Dummy elements must be used to define any additional segments that are required to create a closed volume.

3. The normals of all the segments that form the coupling surface must point in the same general direction and result in a positive closed volume. Setting the REVERSE field to ON ensures that this condition is satisfied, regardless of how the segments are defined initially.

4. The COVER field determines how Eulerian elements that are inside and outside of the coupling surface are processed. The default setting of INSIDE is appropriate for most problems, since in the majority of analyses, the Eulerian material flows around the outside of the coupling surface. Therefore, the Eulerian elements within the coupling surface are empty. However, for some specialized applications (such as air bag inflation), the Eulerian material is completely contained within the coupling surface, and in these cases COVER should be set to OUTSIDE.

5. For the fast coupling algorithm use PARAM,FASTCOUP.

6. The COUPLE1 entry can only be used in combination with PARAM,LIMITER,ROE.

(Continued)

7. Multiple coupling surfaces can be used defining one Eulerian region belonging to each coupling surface by setting the SET1ID or the MESHID option. Only one of the two options can be set and will work only in combination with PARAM,FASTCOUP.

8. The COUP1FL option is only working in combination with PARAM,FASTCOUP, ,FAIL. If no number is given, the default values of the surrounding variables will be given; the RHO will be equal to the reference RHO on the DMAT entry and the other variables (SIE, XVEL, YVEL and ZVEL) will by default be equal to zero.

9. The ACTIVE entry will be ignored in case TDEAC is used in combination with PARAM,FASTCOUP.

COUPOR

Coupling Porosity

Defines porosity for SURFACE and SUBSURF entries used in general coupling.

Format and Example:

1	2	3	4	5	6	7	8	9	10
COUPOR	CID	PORID	SUBID	PORTYPE	PORTYPID	COEFF	COEFFV		
COUPOR	111	203	31	PORFLOW	75		0.2		

Field	Contents		Type	Default
CID	Unique number of a COUPOR entry.		I > 0	Required
PORID	Number of a set of COUPOR entries. PORID must be referenced from a COUPLE entry.		I > 0	Required
SUBID	> 0	Number of a SUBSURF, which must be a part of the SURFACE referred to from the COUPLE entry.	I ≥ 0	0
	= 0	COUPOR definitions used for the entire SURFACE referred to from the COUPLE entry.		
PORTYPE	Defines the type of porosity.		C	PORFLOW
	PORFLOW	The PORFLOW logic is used to model a flow boundary in the coupling surface. The flow boundary acts on the open area of the coupling (sub)surface (SUBID).The open area is equal to the area of the (sub)surface multiplied by COEFFV. A hole can be modeled when COEFFV is set to 1.0. A closed area results for COEFFV = 0.0. The characteristics of the flow are defined on a PORFLOW entry, with ID as defined on the PORTYPID.		

(Continued)

Field	Contents		Type	Default
	PORHOLE	The PORHOLE logic is used to model holes in an air bag. The hole is defined by a subsurface (SUBID). The open area of the hole is equal to the area of the (sub)surface multiplied by COEFFV. A value of COEFFV = 1.0 will open up the complete hole area, while a value of COEFFV = 0.0 will result in a closed hole. The velocity of the gas flow through the hole is based on the theory of one-dimensional gas flow through a small orifice. The velocity depends on the pressure difference. The characteristics for the flow are defined on a PORHOLE entry, with ID as defined on the PORTYPID.		
	PERMEAB	The PERMEAB logic is used to model permeable air-bag material. The permeable area can be defined for a subsurface (SUBID) or for the entire coupling surface. The velocity of the gas flow through the (sub)surface is defined as a linear or tabular function of the pressure difference between the gas inside the air bag and the environmental pressure. The function is specified on a PERMEAB entry, with ID as defined on the PORYPID. The area actually used for outflow is the subsurface area multiplied by the value of COEFFV.		

(Continued)

Field	Contents		Type	Default
	PORFGBG	The PORFGBG logic is used to model gas flow through a hole in the coupling surface connected to a GBAG. The hole is defined by a subsurface (SUBID). The open area of the hole is equal to the area of the (sub)surface multiplied by COEFFV. A value of COEFFV = 1.0 will open up the complete hole area, while a value of COEFFV = 0.0 will result in a closed hole. The velocity of the gas flow through the hole is based on the theory of one-dimensional gasflow through a small orifice. The velocity depends on the pressure difference. The characteristics for the flow are defined on a PORFGBG entry, with ID as defined on the PORTYPID.		
	PERMGBG	The PERMGBG logic is used to model gas flow through a permeable area in the coupling surface connected to a GBAG. The permeable area can be defined for a subsurface (SUBID) or for the entire coupling surface. The velocity of the gas flow through the (sub)surface is defined as a linear or tabular function of the pressure difference. This function is specified on a PERMGBG entry, with ID as defined on the PORYPID. The area actually used for outflow is the subsurface area multiplied by the value of COEFFV.		
PORTYPID	Number of a PORFLOW entry.		I > 0	Required
COEFF	Method of defining the porosity coefficient.		C	CONSTANT
	CONSTANT	The porosity coefficient is constant and specified on COEFFV.		
	TABLE	The porosity coefficient varies with time. COEFFV is the number of a TABLED1 or TABLEEX entry defining the variation with time.		
COEFFV	The porosity coefficient or the number of a TABLED1 or TABLEEX entry depending on the COEFF entry.		0 < R < 1. or I > 0	1.0

(Continued)

Remarks:

1. A mixture of multiple COUPORs with different PORTYPEs is allowed.
2. All options of PORTYPE except PORFLOW can also be referenced by a GBAGPOR. This makes it possible to setup the exact same model for either a uniform pressure model (GBAG → GBAGPOR) or an Eulerian model (COUPLE → COUPOR). It allows for setting up the model using a switch from full gas dynamics to uniform pressure (GBAGCOU).
3. The options PORFGBG and PERMGBG can be used to model airbags with different compartments.
4. The pressure, as defined by a PORFLOW or PORHOLE entry, is exerted on the Eulerian material. Similarly, the pressure in the connected GBAG, in case of a PORFGBG entry, is exerted on the Eulerian material. This pressure is applied over the open area only. The open area is equal to the area of the (sub)surface multiplied by COEFFV. The remaining closed area behaves as a wall boundary condition for the gas.

CPENTA

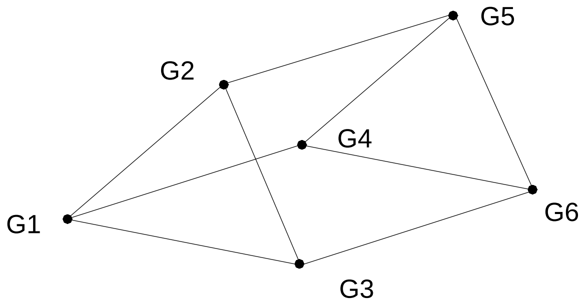
Solid Element with Six Grid Points

Defines a solid wedge element with six grid points.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CPENTA	EID	PID	G1	G2	G3	G4	G5	G6	
CPENTA	112	2	3	15	14	4	103	115	

Field	Contents	Type	Default
EID	Unique element number.	I > 0	Required
PID	Number of a PSOLID or PEULERn property entry.	I > 0	EID
G1–G6	Grid-point numbers of the connection points. They must all be unique.	I > 0	Required



Remarks:

1. The element number must be unique with respect to all other elements.
2. G1, G2, and G3 must define a triangular face. G4, G5, and G6 define the opposite face with G1 opposite G4; G2 opposite G5, etc.
3. The faces may be numbered either clockwise or counterclockwise.
4. The Lagrangian CPENTA element performs poorly compared with the CHEXA element. This element should only be used where the geometry demands it, and it should be located well away from any critical areas. Always use the CHEXA element if you can.
5. The property number references a PSOLID or a PEULERn entry. This determines whether the element is Lagrangian or Eulerian.

CQUAD4

Quadrilateral Element Connection

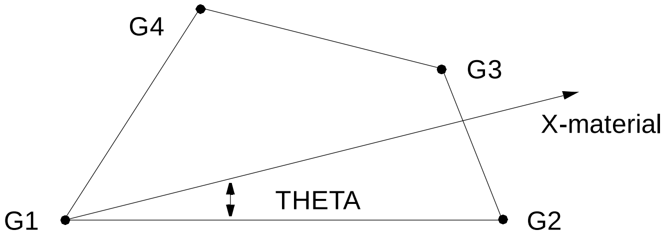
Defines a Lagrangian quadrilateral shell element.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CQUAD4	EID	PID	G1	G2	G3	G4	THETA		+CONT1
CQUAD4	111	203	31	74	75	32			+CONT1

+CONT1			T1	T2	T3	T4			
+CONT1									

Field	Contents	Type	Default
EID	Unique element number.	I > 0	Required
PID	Number of a PSHELLn property entry.	I > 0	EID
G1–G4	Grid-point numbers of the connection points. They must all be unique.	I > 0	Required
THETA	Material property orientation specification (Real or blank; or 0 ≤ Integer < 1,000,000). If real, it specifies the material property orientation angle in degrees. If integer, the orientation of the material x-axis is along the projection onto the plane of the element of the x-axis of the coordinate system specified by the integer value. The figure below gives the sign convention for THETA.	I or R	



Field	Contents	Type	Default
T1–T4	Thickness at the grid points G1 through G4.	R > 0.0	See Remark 4.

(Continued)

Remarks:

1. The element number must be unique with respect to all other elements.

2. Grid points G1 to G4 must be ordered consecutively around the perimeter of the element.

3. If a CQUAD4 element has a thickness of 9999. (set on the PSHELLn entry), it is not a shell element but it is converted to a CSEG entry. This conversion allows CSEGs to be easily defined using standard preprocessors. See Section 3.2.6 on page 3-4 for details.

4. If the four grid-point thicknesses are defined, the element thickness is the average of the defined thickness at the four grid points. If the thicknesses are not defined, the default thickness as specified on the PSHELLn entry is used.

5. The THETA entry is only used with orthotropic materials.

CROD

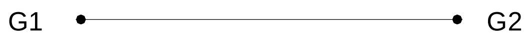
Rod Element Connection

Defines a tension-compression element.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CROD	EID	PID	G1	G2					
CROD	17	6	59	79					

Field	**Contents**	**Type**	**Default**
EID	Unique element number.	I > 0	Required
PID	Number of a PROD, PBELT or PWELD property entry.	I > 0	EID
G1, G2	Grid-point numbers of connected grid points.	I > 0	Required



Remarks:

1. Element numbers must be unique with respect to all other element numbers.
2. Only one rod element may be defined on a single line.

CSEG

Segment of a Contact Surface or Coupling Surface

Defines a segment with either three or four grid points.

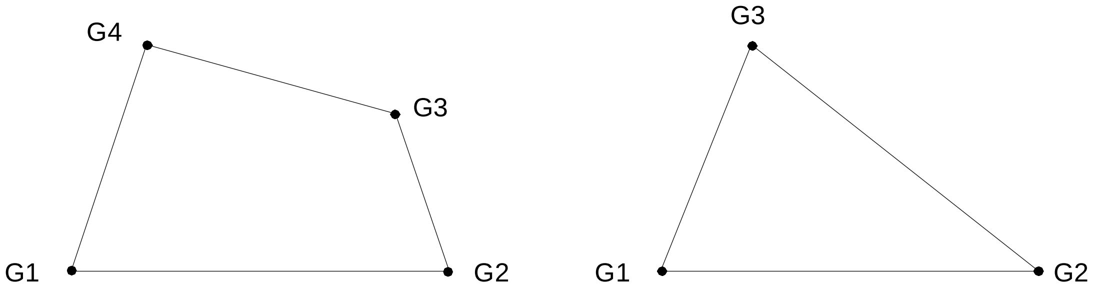
Format and Example:

1	2	3	4	5	6	7	8	9	10
CSEG	ID	SID	G1	G2	G3	G4			
CSEG	101	17	13	19	64	63			

Field	Contents	Type	Default
ID	Unique segment number.	I > 0	Required
SID	Number of the set of segments to which this CSEG belongs.	I > 0	Required
G1–G4	Grid-point numbers defining the connectivity of the segment. For triangular segments, G4 should be blank or zero. All the grid points must be unique.	I > 0	Required

Remarks:

1. The segment number must be unique with respect to all other segments.
2. Grid points G1 to G4 must be ordered consecutively around the perimeter of the element.



3. Segments can be automatically generated for shell and membrane elements, thereby saving the effort of creating several CSEG entries for contact surfaces and coupling with CQUAD4 and CTRIA3 elements. The elements for which segments are automatically generated are selected on SET1 entries referenced by the SURFACE entry.
4. To simplify the generation and checking of CSEG entries, CSEG entries can alternatively be defined using the CQUAD4 and CTRIA3 entries with a thickness of 9999. For details, see Section 3.2.6 on page 3-4.
5. Segments also can be defined using the CFACE and CFACE1 entries.

Defines a spring element.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CSPR	EID	PID	G1	G2					
CSPR	2	6	9	33					

Field	Contents	Type	Default
EID	Unique element number.	I > 0	Required
PID	Number of a PSPRn property entry.	I > 0	EID
G1, G2	Grid-point numbers at the ends of the spring. G1 must not be the same as G2.	I > 0	Required

Remarks:

1. The element number must be unique with respect to all other elements.

2. This entry defines a spring acting between two grid points. The force always acts in the direction of the line connecting the two grid points. The direction changes during the analysis as the grid points move.

3. The spring can have a linear or nonlinear force/deflection characteristic depending on the PSPRn entry it references.

Linear elastic with failure	(PSPR)
Nonlinear elastic	(PSPR1)
User-defined	(PSPREX)

4. CELAS1 and CELAS2 define springs with a fixed orientation.

CTETRA

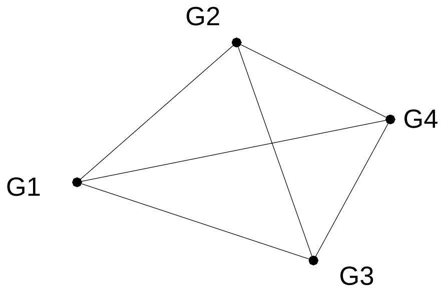
Solid Element with Four Grid Points

Defines a solid tetrahedral element with four grid points.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CTETRA	EID	PID	G1	G2	G3	G4			
CTETRA	112	2	3	15	14	4			

Field	Contents	Type	Default
EID	Unique element number.	I > 0	Required
PID	Number of a PSOLID or PEULERn property entry.	I > 0	EID
G1–G4	Grid-point numbers of the connection points. They must all be unique.	I > 0	Required



Remarks:

1. The element number must be unique with respect to all other element numbers.
2. The element can be numbered in any convenient order.
3. The Lagrangian CTETRA element performs poorly compared with the CHEXA element and should not be used unless absolutely necessary. It should be located well away from any area of interest.
4. The property number references a PSOLID or PEULERn entry. This determines whether the element is Lagrangian or Eulerian.

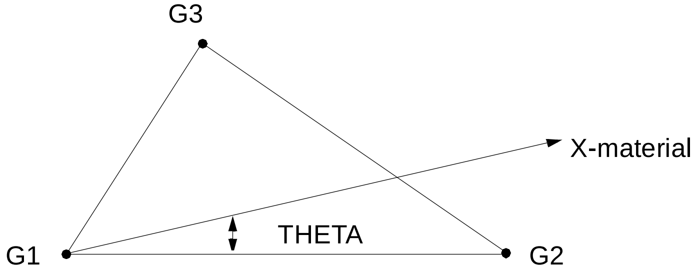
Defines a Lagrangian triangular shell or membrane element.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CTRIA3	EID	PID	G1	G2	G3	THETA			+CONT1
CTRIA3	111	203	31	74	75				+CONT1

+CONT1			T1	T2	T3				
+CONT1									

Field	Contents	Type	Default
EID	Unique element number.	I > 0	Required
PID	Number of a PSHELLn property entry.	I > 0	EID
G1–G3	Grid-point numbers of the connection points. They must all be unique.	I > 0	Required
THETA	Material property orientation specification (Real or blank; or 0 ≤ Integer < 1,000,000). If real, specifies the material property orientation angle in degrees. If integer, the orientation of the material x-axis is along the projection onto the plane of the element of the x-axis of the coordinate system specified by the integer value. The figure below gives the sign convention for THETA.	I or R	



Sign Convention for THETA.

T1–T3	Thickness at the grid points G1 through G3.	R > 0.0	See Remark 4.

(Continued)

Remarks:

1. The element number must be unique with respect to all other elements.
2. Grid points G1 to G3 must be ordered consecutively around the perimeter of the element.
3. If a CTRIA3 element has a thickness of 9999 (set on the PSHELLn entry), it is not a shell element but is converted to a CSEG entry. This conversion allows CSEGs to be easily defined using standard preprocessors. See Section 3.2.6 on page 3-4 for details.
4. If the three grid-point thicknesses are defined, the element thickness is the average of the defined thickness at the three grid points.

   If the thicknesses are not defined, the default thickness as specified on the PSHELLn entry is used.

Defines a viscous damper element.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CVISC	EID	PID	G1	G2					
CVISC	19	6	7	104					

Field	Contents	Type	Default
EID	Unique element number.	I > 0	Required
PID	Number of a PVISCn property entry.	I > 0	EID
G1, G2	Grid-point numbers at the ends of the damper. G1 must not be the same as G2.	I > 0	Required

Remarks:

1. The element number must be unique with respect to all other element numbers.
2. This entry defines a damper acting between two grid points. The force always acts in the direction of the line connecting the two grid points. The direction changes during the analysis as the grid points move.
3. The damper can have a linear or nonlinear force/velocity characteristic depending on the PVISCn entry it references.

Linear	(PVISC)
Nonlinear	(PVISC1)
User-defined	(PVISCEX)

4. CDAMP1 and CDAMP2 define dampers with a fixed orientation.

CYLINDER

Defines the Shape of a Cylinder

Cylindrical shape used in the initial condition definition on the TICEUL entry.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CYLINDER	VID		XC1	YC1	ZC1	XC2	YC2	ZC2	+CONT1
CYLINDER	4		0.	0.	0.	1.	1.	1.	+CONT1

+CONT1	RAD								
+CONT1	.5								

Field	Contents	Type	Default
VID	Unique cylinder number.	I > 0	Required
XC1, YC1, ZC1	Coordinates of point 1 (See Remark 1).	R	Required
XC2, YC2, ZC2	Coordinates of point 2 (See Remark 1).	R	Required
RAD	Radius of the cylinder.	R	Required

Remarks:

1. A cylinder is defined by the two end points of the cylinder axis and a radius.
2. Initial conditions are defined for the elements that are fully or partially inside the cylinder. See Section 2.8.4 on page 2-99.
3. See also TICEUL Bulk Data entry.

This entry is used in conjunction with a TLOAD entry and defines the location and direction of a concentrated load or enforced motion with a scale factor.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DAREA	LID	G	DIR	SCALE	G	DIR	SCALE		
DAREA	3	6	2	8.2	15	1			

Field	Contents	Type	Default
LID	Number of a set of loads.	I > 0	Required
G	Grid point or rigid body where the load is applied.	See Remark 5.	Required
DIR	Direction of the load. Enter 1, 2, or 3 to apply a loading in the x-, y-, or z-directions. Enter 4, 5, or 6 to apply loading about the x-, y-, or z-axes.	1 ≤ I ≤ 6	Required
SCALE	Scale factor for the load.	R	1.0

Remarks:

1. One or two loads can be defined on a line.
2. At time t, the load $F(t)$ is given by

$$F(t) = \text{SCALE} * T(t)$$

   where SCALE is the scale factor and T(t) is given by a table referenced from the TLOAD entry.
3. The load direction is defined in the basic coordinate system.
4. The direction of the load does not change during the analysis.
5. If G references a MATRIG, an RBE2-FULLRIG, or a RIGID surface, the load is applied to the center of the rigid body. If G references a MATRIG, G must be MR<id>, where id is the MATRIG number. If G references an RBE2-FULLRIG, G must be FR<id>, where id is the RBE2 number. If G references a RIGID surface, G is the RIGID surface number.
6. If the TYPE field on the TLOADn entry is 0, it defines a force or moment applied to a grid point. If the TYPE field is 2, it defines an enforced motion of the grid point. If the TYPE field is set to 12, it defines an enforced motion applied to the center of a rigid body. If the TYPE field is 13, it defines a force or moment applied to the center of a rigid body.

DETSPH

Spherical Detonation Wave

Defines the ignition point from which a spherical detonation wave travels, causing the reaction of high explosive materials.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DETSPH	DID	MID	X	Y	Z	VEL	TIME		
DETSPH	100	10	96.5	177.6	37.4	2379.	1.7E–6		

Field	Contents	Type	Default
DID	Unique detonation number.	I > 0	Required
MID	Material number.	I > 0	Required
X, Y, Z	Coordinates of the ignition point.	R	0.0
VEL	Velocity of the detonation wave.	R ≥ 0.0	0.0
TIME	Detonation time.	R ≥ 0.0	0.0

Remark:

1. An element detonates when a spherical detonation wave originating from the detonation point at the specified time reaches the element.

Defines a complete constitutive model as a combination of an equation of state, a shear model, a yield model, a failure model, a spall model (PMIN), and corotational frame.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DMAT	MID	RHO	EID	SID	YID	FID	PID	CID	+CONT1
DMAT	22	3000.	100	109	307	308	402		+CONT1

+CONT1	BULKL	BULKQ			BULKTYP				
+CONT1									

Field	Contents	Type	Default
MID	Unique material number.	I > 0	Required
RHO	Density.	R > 0.0	Required
EID	Number of an EOSxxx entry defining the pressure/density characteristic of the material.	I > 0	Required
SID	Number of a SHRxxx entry defining the shear properties of the material.	I ≥ 0	Hydrodynamic shear model.
YID	Number of a YLDxxx entry defining the yield model for the material.	I ≥ 0	Hydrodynamic yield model.
FID	Number of a FAILxxx entry defining the failure model for the material.	I ≥ 0	No failure.
PID	Number of a PMINC entry defining the spallation characteristics of the material.	I ≥ 0	See Remark 6.
CID	Number of a CORDROT entry. See Remark 7.	I ≥ 0	No corotational coordinate system.
BULKL	Linear bulk-viscosity coefficient.	R ≥ 0.0	0.0
BULKQ	Quadratic bulk-viscosity coefficient.	R ≥ 0.0	1.0
BULKTYP	Bulk viscosity type.	C	DYNA
	DYNA Standard DYNA3D model.		
	DYTRAN Enhanced DYNA model.		

(Continued)

Remarks:

1. This material model can be used with Lagrangian and Eulerian solid elements and membrane elements.
2. If YID is blank or zero, a hydrodynamic yield model is used.
3. For Eulerian elements, if the TYPE field on the PEULER entry is set to HYDRO, then YID is either blank or references a YLDHY entry, and SID is blank. Conversely, if the TYPE field is set to STRENGTH, a nonhydrodynamic yield model is specified.
4. This material is discussed in Section 2.5.3.1 on page 2-27.
5. Materials of shell elements need to be specified using the MAT1, MAT8, DMATEP, DYMAT24 or SHEETMAT entries.
6. If no PMINC entry is referenced, a minimum pressure of zero is assumed for the Eulerian elements, while spallation is prevented for the Lagrangian solid elements by assuming a minimum pressure of -1.E20. The PMINC entry will be ignored when a cavitation model through the EOSTAIT entry has been given.
7. The definition of a corotational coordinate system can only be used for Lagrangian solid elements. If no corotational coordinate system is specified, all stresses are in the basic coordinate system.
8. The failure model for Eulerian materials can be FAILEX or FAILMPS. For Lagrangian materials FAILMPS, FAILEX, FAILMES, and FAILSDT can be addressed.
9. When PARAM,PMINFAIL is also set and a failure model is defined, Lagrangian solid elements also fail on the defined spallation pressure.

DMATEL

Isotropic Elastic Material Properties

Defines the properties of an isotropic elastic material for Lagrangian solid and membrane elements.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DMATEL	MID	RHO	E	NU	G	K		CID	+CONT
DMATEL	11	7850.0	210.E9	0.3					+CONT

+CONT	CSCALE	DMPFAC	BULKTYP	BULKQ	BULKL				
+CONT									

Field	Contents	Type	Default
MID	Unique material number.	I > 0	Required
RHO	Density.	R > 0.0	Required
E	Young’s modulus.	R > 0.0	See Remark 1.
NU	Poisson’s modulus.	R > 0.0	See Remark 1.
G	Shear modulus.	R > 0.0	See Remark 1.
K	Bulk modulus.	R > 0.0	See Remark 1.
CID	Number of a CORDROT entry. See Remark 5.	I > 0	No corotational coordinate system
CSCALE	When this material model is used with MEMB shell elements, the compressive stresses in the principal directions will be scaled with this factor. CSCALE=0.0 results in a tension only material.	R ≥ 0.0	1.0 See Remark 5.
DMPFAC	When this material model is used with MEMB shell elements, damping is applied to the stresses: $\sigma_{ij} = DMPFAC \cdot E \cdot \varepsilon_{ij} \cdot dt_{elm}$	R ≥ 0.0	1.0 See Remark 5.
BULKTYP	For Lagrangian solids only, Bulk-viscosity model: DYNA Standard DYNA3D model. DYTRAN Enhanced DYNA model.	C	DYNA

(Continued)

Field	Contents	Type	Default
BULKQ	For Lagrangian solids only, Quadratic bulk-viscosity coefficient.	R ≥ 0.0	1.0
BULKL	For Lagrangian solids only, Linear bulk-viscosity coefficient.	R ≥ 0.0	0.0

Remarks:

1. Only two of the elastic constants E, Nu, G, and K should be defined.
2. The behavior of this material is discussed in Section 2.5.3.2 on page 2-27.
3. This material model can be used only with Lagrangian solid and membrane shell elements.
4. The definition of the corotational coordinate system can be used only for Lagrangian solid elements. If no corotational coordinate system is specified, all stresses are in the basic coordinate system.
5. For airbag modeling the following values of CSCALE and DMPFAC are suggested:

   CSCALE = 0.1
   DMPFAC= 0.05 to 0.20
6. The entry PMAXCUT, which was supported by MSC.Dytran V4.5 has become obsolete. A better methodology is now offered by scaling the compressive stresses in the principal directions, and using damping to suppress high-frequency oscillations (CSCALE and DMPFAC).

DMATEP

Elastoplastic Material Properties

Defines the properties of an isotropic-elastoplastic material for shell and beam elements.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DMATEP	MID	RHO	*E*	ν	*G*	*K*	YID	FID	
DMATEP	11	7850.0	210.E9	0.3			100	101	

Field	Contents	Type	Default
MID	Unique material number.	I > 0	Required
RHO	Density.	R > 0.0	Required
E	Young’s modulus.	R > 0.0	See Remark 1.
ν	Poisson’s ratio.	0.5 ≥ R ≥ 0.0	See Remark 1.
G	Shear modulus.	R ≥ 0.0	See Remark 1.
K	Bulk modulus.	R ≥ 0.0	See Remark 1.
YID	Number of a YLDxxx entry defining the yield model for the material. (See Remark 6.)	I ≥ 0	See Remark 5.
FID	Number of a FAILxxx entry defining the failure model for the material. (See Remark 7.)	I	No Failure

Remarks:

1. Only two of the elastic constants *E*, ν , *G*, or *K* should be defined.
2. The behavior of this material is discussed in Section 2.5.3.3 on page 2-28.
3. This material model can be used only with shell and beam elements.
4. If YID is 0 or blank, the material is elastic.
5. YID can refer to a YLDVM entry, in which case the material is elastoplastic with isotropic hardening, or for CQUADy and CTRIAz elements only, to a YLDJC entry to define a Johnson-Cook yield model.
6. If an elastoplastic material is specified for Belytschko-Schwer beams, a resultant plasticity model is used. The entire cross section yields at once.
7. The failure models that can be addressed by the DMATEP material definition are FAILMPS, FAILSSR, and FAILEX.

DMATOR

Orthotropic Elastic Material Properties

Defines an orthotropic elastic material for Lagrangian solid elements.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DMATOR	MID	RHO	OPTION		FILE			FID	+CONT1
DMATOR	9	7800E–9	ELMAT		MAT.DAT			1	+CONT1

+CONT1	EA	EB	EC	NUBA	NUCA	NUCB			+CONT2
+CONT1	200E3	175.E3	105.E3	0.3	0.25	0.29			+CONT2

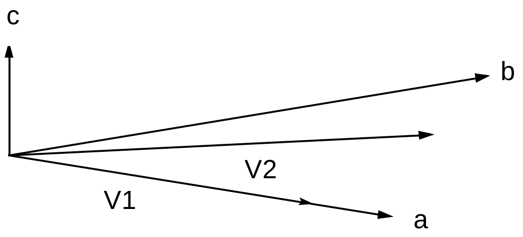
+CONT2	GAB	GBC	GCA						+CONT3
+CONT2	50E3	70E3	65.5E3						+CONT3

+CONT3	X1	Y1	Z1	X2	Y2	Z2			+CONT4
+CONT3									+CONT4

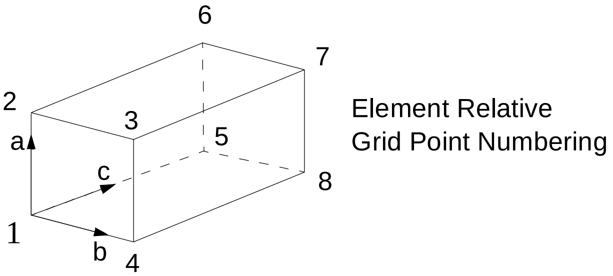
+CONT4			BULKTYP	BULKQ	BULKL				
+CONT4				1.2					

Field	Contents	Type	Default
MID	Unique material number.	I > 0	Required
RHO	Density.	R > 0.0	Required
OPTION	Material axes option used to determine how the local material axis system is defined.	C	ELEM

VECT Globally orthotropic with the material axes defined by two vectors V1 and V2, specified using the fields X1, Y1, Z1 and X2, Y2, Z2. The a-axis is defined by the first vector. The b- and c-axes are then defined as:



(Continued)

Field	Contents	Type	Default
	ELEM Globally orthotropic material with the material axes defined by element topology. The a, b, and c axes are defined as follows:  6 7 2 3 5 a c 8 1 b 4 Element Relative Grid Point Numbering Grid point 1 defines the origin, grid point 5 lies on the c-axis, and grid point 2 lies in the ac-plane.		
	ELMAT Orthotropic material properties and the material coordinate system are defined by the element. The material properties are read from a file (formatted). The filename is specified in the sixth field of the first line. Format of material properties file: Record# EID, DENSITY, DUMMY, DUMMY, DUMMY, E_a, E_b, E_c, G_{ab}, G_{bc}, G_{ca}, ν_{ab}, ν_{ac}, ν_{bc}, ν_{ba}, ν_{ca}, ν_{cb}		
	ELPROP Globally orthotropic material with the material axes defined by element topology (see also ELEM). The elasticity matrix is available per element.		
FILE	Material file name (OPTION = ELMAT only).	C	
FID	Failure model number.	I > 0	No Failure
EA, EB, EC	Young’s moduli in the a, b, and c directions.	R > 0.0	Required
NUBA, NUCA, NUCB	Poisson’s ratio among the a, b, and c material directions.	$0.0 \leq R \leq 1.0$	Required

(Continued)

Field	Contents		Type	Default
GAB, GBC, GCA	Shear moduli among the three material directions.		R > 0.0	Required
X1, Y1, Z1	Components of the vector V1 in the basic coordinate system.		R	0.0
X2, Y2, Z2	Components of the vector V2 in the basic coordinate system.		R	0.0
BULKTYP	Bulk viscosity type:		C	DYNA
	DYNA	Standard DYNA3D model.		
	DYTRAN	Enhanced DYNA model.		
BULKL	Linear bulk-viscosity coefficient.		R ≥ 0.0	0.0
BULKQ	Quadratic bulk-viscosity coefficient.		R ≥ 0.0	1.0

Remarks:

1. The continuation line with bulk-viscosity data can be omitted.
2. The behavior of this material is discussed in Section 2.5.3.4 on page 2-29.
3. This material model can be used only with Lagrangian solid elements.
4. The failure models that are addressed by an orthotropic (DMATOR) material definition are FAILEX, FAILEX1, FAILMES, FAILPRS, and FAILEST.
5. If FAILEX1, the extended user-defined failure, is used, set the OPTION to either ELMAT or ELPROP. The user-defined failure, FAILEX1, gives access to the material properties on an element basis.

DYMAT14

Soil and Crushable Foam Material Properties

Defines a nonlinear material for Lagrangian solid elements that crushes under hydrostatic loading and is elastic-plastic under deviatoric loading. Material failure can be included.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DYMAT14	MID	RHO	G	K	TABLE	TYPE	VALUE	CUTOFF	+CONT1
DYMAT14	3	0.01	5.	3.	111	CRUSH	-100.	PFRAC	+CONT1

+CONT1	A_0	A_1	A_2	YIELD		YSTYP			+CONT2
+CONT1	1.	0.	0.	YSURF		DYNA			+CONT2

+CONT2			BULKTYP	BULKQ	BULKL				
+CONT2			DYNA	1.4	0.05				

Field	Contents		Type	Default
MID	Unique material number.		I > 0	Required
RHO	Density.		R > 0.0	Required
G	Shear modulus.		R > 0.0	Required
K	Bulk modulus.		R > 0.0	Required
TABLE	Number of a TABLED1 entry giving the variation of pressure (y-value) with crush factor or volumetric strain (x-value).		I > 0	Required
TYPE	The type of data defined as the x value in the table:		C	CRUSH
	CRUSH	Crush factor (1 = relative volume).		
	STRAIN	Volumetric (true) strain.		
	See also Remark 3.			
VALUE	The value for the cut-off pressure.		R < 0.0	See Remark 4.
CUTOFF	Cut-off pressure.		C	PFRAC
	PFAIL	Pressure for total tensile failure.		
	PFRAC	Pressure for tensile failure.		
	PMIN	Minimum pressure.		

(Continued)

DYMAT14

Soil and Crushable Foam Material Properties

Field	Contents		Type	Default
A_0, A_1, A_2	Yield function constants.		R	0.0
YIELD	Surface description:		C	YSURF
	YSURF	The yield surface (see Remark 7.) is defined as a function of p and J_2.		
	YSTRESS	The yield surface is defined as a function of p and s_y.		
YSTYP	Type of YSURF Yield Surface description:		C	DYNA
	DYNA	DYNA definition.		
	DYTRAN	DYTRAN additional definition. (See Remark 7.)		
BULKTYP	Bulk-viscosity model:		C	DYNA
	DYNA	Standard DYNA3D model.		
BULKQ	Quadratic bulk-viscosity coefficient.		R ≥ 0.0	1.0
BULKL	Linear bulk-viscosity coefficient.		R ≥ 0.0	0.0

Remarks:

1. If BULKTYP, BULKQ, or BULKL are blank or zero the default values are used.
2. The continuation line with the bulk-viscosity data can be omitted.
3. The pressure-volume characteristic can either be defined in terms of the amount of crush, which is minus the engineering strain and is defined as $(1 - V/V_0)$, with V/V_0 as the relative volume; or in terms of the volumetric (true) strain which is defined as

$$\int_{t_0}^{t} \frac{dV}{V}$$

   or $ln\,(V/V_0)$. The crush must be between 0 and 1. The volumetric strain must always be negative.

(Continued)

4. If the field for the value of PFAIL/PFRAC/PMIN is left blank, then this value is calculated from the yield function defined by the constants A_0, A_1, and A_2. In case of a Mohr-Coulomb yield model, the cut-off pressure is calculated as the root of the pressure-yield stress curve. If the YSURF option is used, the cut-off pressure is calculated as the intersection point of the yield surface with the hydrostat (if only A_0 is nonzero, then the cut-off pressure is set to -100K, where K is the bulk modulus). The cut-off pressure must be negative.

5. Either a minimum pressure (PMIN) or a failure pressure (PFAIL or PFRAC) can be specified. In the first case, this corresponds to a tensile cutoff, where the pressure cannot fall below the minimum value. In the second case, if the pressure falls below the failure pressure the element fails and cannot carry tensile loading for the remainder of the analysis. Thus, the pressure can never become negative again. If PFAIL is used, the elements can physically fail, which means that the stresses are set to zero, but also the failure flag is used as in normal FAILxxx models. If PFRAC is used, only the stresses are set to zero.

6. This material can only be used with Lagrangian solid elements.

7. If the YSTRESS option is used, the yield stress is determined by a Mohr-Coulomb model:

$$\sigma_y = MIN(A_0 + A_1 p, A_2)$$

If the YSURF option is used, the yield surface in three-dimensional space is defined by $\Phi_s = 0$ where

$$\Phi_s = \frac{1}{2}s_{ij}s_{ij} - (B_0 + B_1 p + B_2 p^2) = J_2 - (B_0 + B_1 p + B_2 p^2)$$

where s_{ij} are the deviatoric stresses and J_2 is the second invariant of the stress deviation. The coefficients B_0, B_1, and B_2 can be related to the coefficients A_0, A_1, and A_2, which are defined on the DYMAT14 entry. The relation between the coefficients depends on the YSTYP field as shown below. If the YSTYP field is DYTRAN, then

$B_0 = A_0$

$B_1 = A_1$

$B_2 = A_2$

Thus, the yield stress (see Section 2.5.3.7 on page 2-39) is defined as

$$\sigma_Y = \sqrt{3(A_0 + A_1 p + A_2 p^2)}$$

(Continued)

If the YSTYP field is DYNA, then

$$B_0 = \frac{1}{3}A_0{}^2$$

$$B_1 = \frac{2}{3}A_0A_1$$

$$B_2 = \frac{1}{3}A_1{}^2$$

and A_2 is ignored.

Thus, the yield stress is defined as

$$\sigma_y = A_0 + A_1 p$$

8. The behavior of this material is described in Section 2.5.3.7 on page 2-39.

DYMAT24

Piecewise Linear Plasticity Material

Defines the properties of a nonlinear, plastic material with isotropic hardening where the stress-strain curve is piecewise linear for shell, beam and Lagrangian solid elements.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DYMAT24	MID	RHO	E	NU	TABLE	TYPE	TABY		+CONT1
DYMAT24	17	7850.	210.E9	0.3	39	ENG			+CONT1

+CONT1	YIELD	EH	EPSF	D	P	VOLF			+CONT2
+CONT1			0.37	40.5	5	1.E-6			+CONT2

+CONT2			BULKTYP	BULKQ	BULKL				
+CONT2			DYNA	1.4	0.05				

Field	Contents	Type	Default
MID	Unique material number.	I > 0	Required
RHO	Density.	R > 0.0	Required
E	Young’s modulus.	R > 0.0	Required
NU	Poisson’s ratio.	0.0 < R ≤ 0.5	Required
TABLE	Number of a TABLED1 entry giving the variation of effective stress (y-value) with effective strain (x-value).	I > 0	See Remark 3.
TYPE	The type of stress and strain defined in TABLE.	C	TRUE
	ENG Engineering stress and strain.		
	TRUE True stress and strain.		
	PLAST True stress and plastic strain.		
	PMOD Plastic modulus and plastic strain.		
TABY	Number of a TABLED1 entry giving the variation of the scale factor for the yield stress (y-value) with the strain rate (x-value). Strain rate effects can also be specified using the Cowper-Symonds relation. (See input parameters D and P.)	I > 0	See Remark 5.
YIELD	Yield stress.	R > 0.0	See Remark 5.

(Continued)

DYMAT24

Piecewise Linear Plasticity Material

Field	Contents	Type	Default
EH	Hardening modulus.	R > 0.0	See Remark 5.
EPSF	Plastic strain at failure.	R > 0.0	No failure
D	Factor “D” in the Cowper-Symonds rate enhancement equation.	R ≥ 0.0	See Remark 5.
P	Factor “P” in the Cowper-Symonds rate enhancement equation.	R ≥ 0.0	See Remark 5.
VOLF	If the volume of Lagrangian solid elements becomes less than VOLF, the element will fail.	R ≥ 0.0	1.E-12
BULKTYP	Bulk viscosity model.	C	DYNA
	DYNA Standard DYNA3D model.		
BULKQ	Quadratic bulk-viscosity coefficient.	R ≥ 0.0	1.0
BULKL	Linear bulk-viscosity coefficient.	R ≥ 0.0	0.0

Remarks:

1. If BULKTYP, BULKQ, or BULKL are blank or zero, the default values apply.
2. The continuation line with the bulk-viscosity data can be omitted.
3. If TABLE is blank or zero, a bilinear stress-strain curve is assumed. If TABLE has a value, it refers to a TABLED1 entry giving the stress-strain curve for the material.
4. If TABLE is defined, the value of YIELD should be left blank, since it is determined from the stress-strain curve.
5. If TABY is blank or zero, and D and P are blank or zero, the yield stress does not vary with strain rate. If TABY has a value, then it references a TABLED1 entry, which gives the variation with strain rate of the scale factor applied to the yield stress (D and P must be blank or zero).

   If TABY is blank or zero and D and P are defined, the enhancement of the yield stress with strain rate is calculated as

$$\frac{\sigma_d}{\sigma_y} = 1 + \left(\frac{\dot{\varepsilon}}{D}\right)^{1/P}$$

   where σ_d is the dynamic stress and σ_y is the static yield stress (YIELD) and $\dot{\varepsilon}$ is the equivalent plastic strain rate.

(Continued)

6. If TYPE is set to ENG or TRUE, Young’s modulus is calculated from the stress-strain curve. When Young’s modulus is specified together with TYPE set to ENG or TRUE, the calculated Young’s modulus will be substituted by the value specified.

7. The behavior of this material is described in Section 2.5.3.8 on page 2-48.

8. This material can only be used with Lagrangian solid, shell and beam elements.

DYMAT26

Orthotropic Crushable Material Model

Defines the properties of an orthotropic, crushable material model for Lagrangian solid elements.

Format and Example:

1	2	3	4	5	6	7	8	9	10
DYMAT26	MID	RHO	E	NU	YIELD	RELV	TYPE	OPTION	+CONT1
DYMAT26	5	1800.	180.E9	0.3	180.E6	0.1	CRUSH	VECT	+CONT1

+CONT1	TIDXX	TIDYY	TIDZZ	TIDXY	TIDYZ	TIDZX	TIDSR		+CONT2
+CONT1	10	11	12	13	14	15	16		+CONT2

+CONT2	EXX	EYY	EZZ	GXY	GYZ	GZX			+CONT3
+CONT2	60.E9	70.E9	60.E9	20.E9	10.E9	15.E9			+CONT3

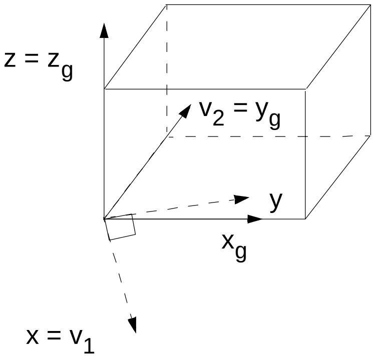
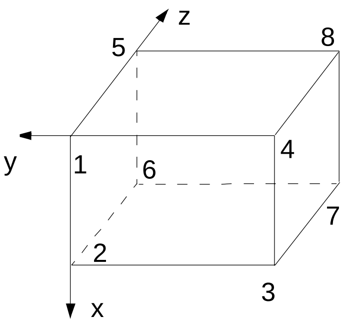
+CONT3			BULKTYP	BULKQ	BULKL				+CONT4
+CONT3			DYNA	1.4	0.05				+CONT4

+CONT4	NUYX	NUZX	NUZY						+CONT5
+CONT4	0.0	0.0	0.0						+CONT5

+CONT5	X1	Y1	Z1	X2	Y2	Z2			
+CONT5	0.	1.	1.	1.	1.	0.			

Field	Contents	Type	Default
MID	Unique material number.	I > 0	Required
RHO	Density.	R > 0.0	Required
E	Young's modulus for the fully compacted material.	R > 0.0	Required
NU	Poisson's ratio for the fully compacted material.	-1.0 < R < 0.5	Required
YIELD	Yield strength for fully compacted material.	R	Required
RELV	Relative volume at which the material is fully compacted.	0.0 < R < 1.0	Required
TYPE	The type of data defined as the x-value in the tables.	C	CRUSH
	CRUSH Crush factor (1-relative volume).		

(Continued)

Field	Contents		Type	Default
n	RELVOL	Relative volume V/V_0.		
OPTION	Material axes option used to determine how the local material axis system is defined.		C	ELEM
	VECT	Globally orthotropic with the material axes defined by two vectors V_1 and V_2, specified using the fields X1, Y1, Z1 and X2, Y2, Z2. The x-axis is defined by the vector V_1. The z-axis is defined as the cross product of V_1 and V_2. The y-axis is defined as the cross product of the z-axis and V_1.		
	 **Material Axes Defined by Two Vectors.**			
	ELEM	Global orthotropic material with the material axes defined by element topology. The x-, y-, and z-axis are defined in the following way:		
	 **Element Relative Grid Point Numbering.**			

(Continued)

Field	Contents	Type	Default
TIDXX	Number of a TABLED1 entry defining the variation of the (local) xx-stress (y-value) with relative volume or crush (x-value).	I > 0	Required
TIDYY	Number of a TABLED1 entry defining the variation of the (local) yy-stress (y-value) with relative volume or crush (x-value).	I > 0	Required
TIDZZ	Number of a TABLED1 entry defining the variation of the (local) zz-stress (y-value) with relative volume or crush (x-value).	I > 0	Required
TIDXY	Number of a TABLED1 entry defining the variation of the (local) xy-shear stress (y-value) with relative volume or crush (x-value).	I > 0	Required
TIDYZ	Number of a TABLED1 entry defining the variation of the (local) yz-shear stress (y-value) with relative volume or crush (x-value).	I > 0	Required
TIDZX	Number of a TABLED1 entry defining the variation of the (local) zx-shear stress (y-value) with relative volume or crush (x-value).	I > 0	Required
TIDSR	Number of an optional TABLED1 entry defining the variation of a yield factor (y-value) with the deviatoric strain rate (x-value).	I > 0	See Remark 7.
EXX	The elastic modulus in the (local) x-direction when the material expands.	R > 0.0	Required
EYY	The elastic modulus in the (local) y-direction when the material expands.	R > 0.0	Required
EZZ	The elastic modulus in the (local) z-direction when the material expands.	R > 0.0	Required
GXY	The shear modulus in the (local) xy-direction when the material expands.	R > 0.0	Required
GYZ	The shear modulus in the (local) yz-direction when the material expands.	R > 0.0	Required
GZX	The shear modulus in the (local) zx-direction when the material expands.	R > 0.0	Required
BULKTYP	Bulk-viscosity model.	C	DYNA
	DYNA Standard DYNA3D model.		
BULKQ	Quadratic bulk-viscosity coefficient.	R > 0.0	1.0

(Continued)

Field	Contents	Type	Default
BULKL	Linear bulk-viscosity coefficient.	R > 0.0	0.0
NUYX	The Poisson’s ratio between the (local) x- and y-axis when the material expands.	-1.0 < R < 1.0	0.0
NUZX	Poisson’s ratio between the (local) x- and z-axis when the material expands.	-1.0 < R < 1.0	0.0
NUZY	Poisson’s ratio between the (local) y- and z-axis when the material expands.	-1.0 < R < 1.0	0.0
X1, Y1, Z1	Components of the vector V_1 in the basic coordinate system.	R	0.0
X2, Y2, Z2	Components of the vector V_2 in the basic coordinate system.	R	0.0

Remarks:

1. If BULKTYP, BULKQ, or BULKL are blank or zero the default values are used.

2. If the initial Poisson’s ratios are not supplied, the default is set to zero. Therefore, the behavior of the material during compaction is uncoupled. This means that straining in the (local) x-direction will produce stresses only in the (local) x-direction, and not in the (local) y- or z-direction. The tables define the variation of the stress in a particular direction with the relative volume or the crush. The relative volume is defined as (current volume)/(initial volume) and varies from 1.0 (uncompressed) to 0.0 (zero volume). Crush is defined as one minus the relative volume and varies from 0.0 to 1.0. Since the tables should be defined with increasing x-values, it is convenient to use the default value for type, which is CRUSH. When defining the curves, care should be taken that the extrapolated values do not lead to negative yield stresses.

3. The elastic moduli (and the initial Poisson’s ratios only if they are supplied) vary linearly with the relative volume from their initial uncompacted values to the fully compacted ones.

4. When the material is fully compacted, its behavior becomes isotropic with an elastic perfectly plastic material characteristic.

5. This material can only be used with Lagrangian solid elements.

6. If the TIDSR option is used, you can supply a table including strain-rate effects. Strain rate is defined here as the Euclidean norm of the deviatoric strain-rate tensor, i.e., $\dot{e} = \sqrt{\dot{e}_{ij}^{dev}\dot{e}_{ij}^{dev}}$. The y-values in this table are factors with which the stresses in the other tables are multiplied to incorporate strain-rate effects.

7. The behavior of this material is described in Section 2.5.3.9 on page 2-49.

ENDDATA

Terminates the Input Data

Marks the end of the input file.

Format and Example:

1	2	3	4	5	6	7	8	9	10
ENDDATA									
ENDDATA									

Remarks:

1. Anything after the ENDDATA entry is ignored.
2. An ENDDATA entry in an INCLUDE file is ignored.

Defines an equation of state specified by a user subroutine.

Format and Example

1	2	3	4	5	6	7	8	9	10
EOSEX	EID	NAME							
EOSEX	12	WATER							

Field	Contents	Type	Default
EID	Unique equation of state number.	I > 0	Required
NAME	Name of the equation of state passed to the user subroutine.	C	blank

Remarks:

1. The subroutine EXEOS must be present in the file referenced by the USERCODE FMS statement.
2. See Section 3.15 on page 3-74 for a description of how to use user-written subroutines.
3. The equation of state name is passed to the EXEOS subroutine and can be used to identify the equation of state.
4. This entry can only be used for Lagrangian and Eulerian solids, but not in combination with the Roe solver, PARAM,LIMITER,ROE.

EOSGAM

Gamma Law Gas Equation of State

Defines the properties of a Gamma Law equation of state where the pressure p is defined as

$$p = (\gamma - 1)\rho e$$

where e = specific internal energy per unit mass

ρ = overall material density

γ = a constant

Format and Example:

1	2	3	4	5	6	7	8	9	10
EOSGAM	EID	GAMMA	R	CV	CP				
EOSGAM	35	1.4							

Field	Contents	Type	Default
EID	Unique equation of state number.	I > 0	Required
GAMMA	Constant γ.	R ≥ 0.	Required
R	Gas constant.	R > 0	See Remarks 2 and 3.
CV	Specific heat at constant volume.	R > 0	See Remarks 2 and 3.
CP	Specific heat at constant pressure.	R > 0	See Remarks 2 and 3.

Remark:

1. This equation of state is discussed in Section 2.5.3.9 on page 2-49.
2. The temperature of the gas will be calculated when one of the gas constants, R, CV or CP is specified. When you are not interested in the temperature, the constants can be omitted.
3. The Euler variable name for temperature is TEMPTURE.
4. Gamma, R, CV and CP have the following relationships:

$$\gamma = \frac{c_p}{c_v} \qquad R = c_p - c_v$$

EOSJWL

JWL Explosive Equation of State

Defines the properties of a JWL equation of state commonly used to calculate the pressure *p* of the detonation products of high explosives

$$p = A\left(1 - \frac{\omega\eta}{R_1}\right)e^{-R_1/\eta} + B\left(1 - \frac{\omega\eta}{R_2}\right)e^{-R_2/\eta} + \omega\eta\rho_o e \quad \mathrm{o}$$

where e = specific internal energy per unit mass

ρ_o = reference density

ρ = overall material density

η = ρ/ρ_o

A, B, R_1, and R_2 are constants.

Format and Example:

1	2	3	4	5	6	7	8	9	10
EOSJWL	EID	A	B	R1	R2	OMEGA			
EOSJWL	37	5.2E11	0.77E11	4.1	1.1	0.34			

Field	Contents	Type	Default
EID	Unique equation of state number.	I > 0	Required
A	Constant *A*.	R	0.0
B	Constant *B*.	R	0.0
R1	Constant R_1.	R	0.0
R2	Constant R_2.	R	0.0
OMEGA	Constant ω.	R	0.0

Remarks:

1. This equation of state can be used only with Eulerian elements.
2. A DETSPH entry must be used to specify the detonation model.
3. This equation of state is discussed in Section 2.5.6.4 on page 2-76.

EOSPOL

Polynomial Equation of State

Defines the properties of a polynomial equation of state where the pressure p is defined as follows:

In compression $(\mu > 0)$,

$$p = a_1\mu + a_2\mu^2 + a_3\mu^3 + (b_0 + b_1\mu + b_2\mu^2 + b_3\mu^3)\rho_0 e$$

In tension $(\mu \leq 0)$,

$$p = a_1\mu + (b_0 + b_1\mu)\rho_0 e$$

where μ = $h - 1$

η = ρ/ρ_0

ρ = overall material density

ρ_0 = reference density

e = specific internal energy per unit mass

Format and Example:

1	2	3	4	5	6	7	8	9	10
EOSPOL	EID	A1	A2	A3	B0	B1	B2	B3	+CONT1
EOSPOL	100	80.E6							+CONT1

+CONT1	HVL	VISC							
+CONT1	1.1								

Field	Contents	Type	Default
EID	Unique equation of state number.	I > 0	Required
A1	Coefficient a_1.	R	0.0
A2	Coefficient a_2.	R	0.0
A3	Coefficient a_3.	R	0.0
B0	Coefficient b_0.	R	0.0
B1	Coefficient b_1.	R	0.0
B2	Coefficient b_2.	R	0.0

(Continued)

Field	Contents	Type	Default
B3	Coefficient b_3.		R
HVL	Hydrodynamic volume limit.	R ≥ 1.0	1.1
VISC	Viscosity coefficient	R > 0.	No viscosity. See Remarks 4 and 5.

Remarks:

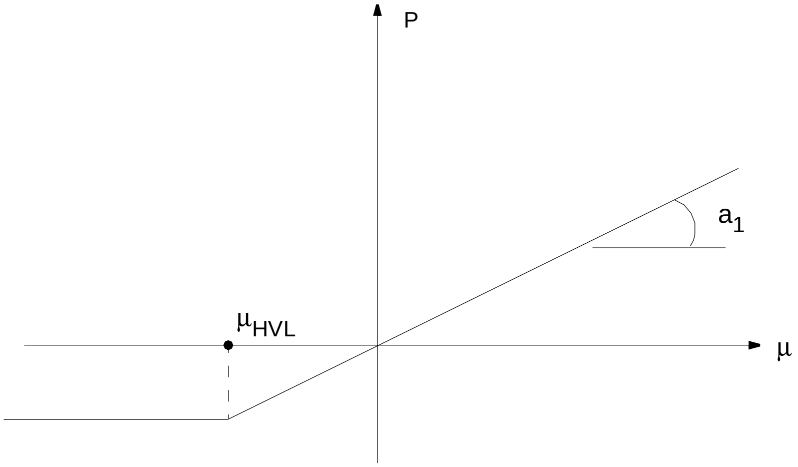
1. When the relative volume (ρ_0/ρ) exceeds HVL, the pressure is cut off to

   $$P_{HVL} = f(\mu_{HVL})$$

   with

   $$\mu_{HVL} = \frac{1}{HVL} - 1$$

   e.g., for $p = a_{1 \cdot \mu}$; the pressure behavior is as follows:



2. When the PARAM,HVLFAIL is set to YES, the elements where the relative volume (ρ_0/ρ) exceeds HVL fail completely. Their stress state is zero.
3. This equation of state is discussed in Section 2.5.6.2 on page 2-74.
4. Viscosity is only available in the Roe solver.
5. If possible use in coupled analysis the FASTCOUP coupling algorithm, because viscous fluxes are computed more accurately for fast coupling than for general coupling.

EOSTAIT

Tait Equation of State

Defines the properties of an equation of state based on the Tait model in combination with a cavitation model where the pressure p is defined as follows:

No cavitation $(\rho > \rho_c)$,

$$p = a_0 + a_1(\eta^{\gamma} - 1)$$

Cavitation $(\rho \leq \rho_c)$,

$$p = p_c$$

where η = ρ/ρ_0

ρ = overall material density

ρ_0 = reference density

ρ_c = critical density which produces the cavitation pressure p_c

Format and Example:

1	2	3	4	5	6	7	8	9	10
EOSTAIT	EID	A0	A1	GAMMA	RHOC	VISC			
EOSTAIT	3	1.E6	3.31E9	7.15	.9999578	.0001			

Field	Contents	Type	Default
EID	Unique equation of state number.	I > 0	Required
A0	Constant a_0.	R	0.0
A1	Constant a_1.	R	0.0
GAMMA	Constant η.	R > 0	1.0
RHOC	Constant ρ_c.	R	Required
VISC	Viscosity coefficient	R > 0	No viscosity See Remarks 4 and 5.

(Continued)

Remarks:

1. The pressure can not fall below the cavitation pressure $p_c = a_0 + a_1\left(\left(\frac{\rho_c}{\rho_0}\right)^{\gamma} - 1\right)$, although the density can continue to decrease below its critical value ρ_c.
2. The Tait equation of state can not be used in combination with a spallation model.
3. For a more detailed description, see Section 2.5.6.3 on page 2-75.
4. Viscosity is only available in the Roe solver.
5. If possible use in coupled analysis the FASTCOUP coupling algorithm, because viscous fluxes are computed more accurately for fast coupling than for general coupling.

FAILEST

Maximum Equivalent Stress and Minimum Time-Step Failure Model

Defines the properties of a failure model where total failure occurs when the equivalent stress exceeds the specified value and the element time step drops below the specified limit.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FAILEST	FID	MES	DT						
FAILEST	1	1.E9	1.E-9						

Field	Contents	Type	Default
FID	Unique failure model number.	I > 0	Required
MES	Maximum equivalent stress that causes failure on the deviatoric part of the stress tensor.	R	Required
DT	Minimum time step that causes total failure.	R	Required

Remarks:

1. This failure model is valid for Lagrangian solid (CHEXA) orthotropic materials. (See also the DMATOR entry.)
2. The FAILEST failure model is a two-stage failure. The first stage retains the hydrodynamic properties of the material. The second stage is reached when the global time step falls below the specified value. The element is then removed from the calculation.

FAILEX
User Failure Subroutine

Specifies that a user subroutine is being used to define the failure model.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FAILEX	FID								
FAILEX	200								

Field	Contents	Type	Default
FID	Unique failure model number.	I > 0	Required

Remarks:

1. The subroutine must be present in the file referenced by the USERCODE FMS statement.
2. See Section 3.15 on page 3-74 for a description of how to use user-written subroutines.

FAILEX1

Extended User Failure Subroutine

Specifies that a user subroutine is being used to define a failure model.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FAILEX1	FID								
FAILEX1	300								

Field	Contents	Type	Default
FID	Unique failure model number.	I > 0	Required

Remarks:

1. The subroutine must be present in the file referenced by the USERCODE FMS statement.
2. The failure model is available for orthotropic materials only. The FAILEX1 entry must be referenced on the DMATOR entry.
3. The failure model allows for an extensive description of the failure of composite materials in three-dimensional elements. It includes the possibility to have property degradation according to material damage.
4. See Section 3.15 on page 3-74 for a description of how to use user-written subroutines.

FAILMES

Maximum Equivalent Stress Failure Model

Defines the properties of a failure model where failure occurs when the equivalent stress exceeds the specified value.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FAILMES	FID	MES							
FAILMES	1	1.E9							

Field	Contents	Type	Default
FID	Unique failure model number.	I > 0	Required
MES	Maximum equivalent stress that causes failure.	R	Required

Remark:

1. This failure model is valid for Lagrangian solid element materials. (See also the DMAT and DMATOR entries.)

FAILMPS

Maximum Plastic Strain Failure Model

Defines the properties of a failure model where failure occurs when the equivalent plastic strain exceeds the specified value.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FAILMPS	FID	MPS							
FAILMPS	1	.15							

Field	Contents	Type	Default
FID	Unique failure model number.	I > 0	Required
MPS	Maximum plastic strain that causes failure.	R	Required

Remark:

1. This failure model is valid for Eulerian, shell (CQUAD4 and CTRIA3), Hughes-Liu beams, and Lagrangian solid element materials. (See also the DMAT and DMATEP entries.)

FAILPRS

Maximum Pressure Failure Model

Defines the properties of a failure model where failure occurs when the hydrodynamic pressure exceeds the specified value.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FAILPRS	FID	PRS							
FAILPRS	1	5.E8							

Field	Contents	Type	Default
FID	Unique failure model number.	I > 0	Required
PRS	Maximum pressure that causes failure.	R	Required

Remark:

1. This failure model is valid for Lagrangian solid element orthotropic materials. (See also the DMATOR entry.)

FAILSDT

Maximum Plastic Strain and Minimum Time-Step Failure Model

Defines the properties of a failure model where total failure occurs when the equivalent plastic strain exceeds the specified value and the element time step falls below the specified limit.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FAILSDT	FID	MPS	DT						
FAILSDT	1	.15	1.E-9						

Field	Contents	Type	Default
FID	Unique failure model number.	I > 0	Required
MPS	Maximum plastic strain that causes failure on the deviatoric part of the stress tensor.	R	Required
DT	Minimum time step that causes total failure.	R	Required

Remarks:

1. This failure model is valid for Lagrangian solid element materials. (See also the DMAT entry.)
2. The FAILSDT failure model is a two-stage failure. The first stage retains the hydrodynamic properties of the material. The second stage is reached when the global time step falls below the specified value. The element then is removed from the computation.

Defines the properties of a material for the boundaries of an Eulerian mesh.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FLOW	LID	SID	TYPE1	VALUE1	TYPE2	VALUE2	TYPE3	VALUE3	+CONT1
FLOW	120	122	XVEL	100.0					+CONT1

+CONT1	TYPE4	VALUE4							
+CONT1									

Field	Contents	Type	Default
LID	Number of a set of flow boundary conditions.	I > 0	Required
SID	Number of a set of segments, specified by CSEG or CFACE or CFACE1 entries, where the flow boundary is located.	I > 0	Required
TYPEi	The flow boundary property being defined. MATERIAL The material number. XVEL The material velocity in the x-direction. YVEL The material velocity in the y-direction. ZVEL The material velocity in the z-direction. PRESSURE The pressure of the material at the boundary. DENSITY The density of the material at inflow. SIE The specific internal energy at inflow. FLOW The type of flow boundary required.	C	
VALUEi	The value for the property defined. For TYPEi set to FLOW, the value is a character entry being either IN, OUT, or BOTH defining that the flow boundary is defined as an inflow, outflow, or possibly an in- or outflow boundary. The default is BOTH. VALUEi is required data only if one or more of the TYPEi entries are defined. The TYPEi entries are not required. Thus, a flow boundary by default allows for in- or outflow of the material adjacent to the boundary.	R or C	Required

(Continued)

Remarks:

1. LID must be referenced by a TLOAD1 entry.
2. Any material properties not specifically defined have the same value as the element with the flow boundary condition.
3. TLOAD entries referencing FLOW entries must have the TID field blank or zero.
4. In the case of multimaterial flow into or out of a multimaterial Euler mesh, the material flowing into or out of the mesh is assumed to be the same as in the elements adjacent to the boundary. If these boundary elements contain mixed (more than one) materials, the material flowing into or out of the mesh is assumed to be mixed in the same proportions. For this material flow, only velocity and pressure are prescribed. Both the density and specific internal energy of the mixed materials are assumed to be the same as those of the mixed materials in the element adjacent to the boundary.

Definition of default Eulerian flow boundary condition.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FLOWDEF	FID		TYPEM						+CONT1
FLOWDEF	25		HYDRO						+CONT1

+CONT1	TYPE1	VALUE1	TYPE2	VALUE2	-etc.-				
+CONT1	DENSITY	1000.							

Field	Contents	Type	Default
FID	Unique FLOWDEF number.	I > 0	Required
TYPEM	HYDRO, STRENGTH, MMHYDRO, or MMSTREN.	C	HYDRO
TYPEi	The flow boundary property being defined.	C	
	MATERIAL The material number.		
	XVEL The material velocity in the x-direction.		
	YVEL The material velocity in the y-direction.		
	ZVEL The material velocity in the z-direction.		
	PRESSURE The pressure of the material at the boundary.		
	DENSITY The density of the material at inflow.		
	SIE The specific internal energy at inflow.		
	FLOW The type of flow boundary required.		
VALUEi	The value for the property defined. For TYPEi set to flow the value is a character entry being either IN, OUT, or BOTH, defining that the flow boundary is defined as an inflow, outflow, or possibly an in- or outflow boundary. The default is BOTH. VALUEi is required data only if one or more of the TYPEi entries are defined. The TYPEi entries are not required. Thus, a flow boundary by default allows for in- or outflow of the material adjacent to the boundary.	R or C	Required

Remark:

1. If this entry is not specified, a default wall boundary condition is applied to all Eulerian free faces.

FLOWEX

User-Specified Flow Boundary

Defines a flow boundary specified by a user subroutine.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FLOWEX	LID	SID	NAME						
FLOWEX	150	300	PRES1						

Field	Contents	Type	Default
LID	Number of a set of flow boundary conditions.	I > 0	Required
SID	Number of a set of segments, specified by CSEG or CFACE entries, where the flow boundary is located.	I > 0	Required
NAME	Name of the flow boundary. (See also Remark 7.)	C	Required

Remarks:

1. LID must be referenced by a TLOAD1 entry.
2. The subroutine EXFLOW must be present in the file referenced by the USERCODE FMS statement. The EXFLOW user subroutine must be present in case single hydrodynamic materials, or materials with strength are used. For multimaterial problems, the EXFLOW2 subroutine must be used.
3. See Section 3.15 on page 3-74 for a description of how to use user-written subroutines.
4. TLOAD1 entries referencing FLOWEX entries must have the TID field blank or zero.
5. The flow boundary name is passed to the EXFLOW subroutine and can be used to identify the boundary.
6. The EXFLOW2 subroutine allows for the definition of any material to flow into the Eulerian mesh. The outflow can only be of materials present in the mesh.
7. There are two methodologies available to define an inflator model for an eulerian calculation:

   a. as a boundary condition for a subsurface on a coupling surface (see the COUPLE, COUPOR and INFLATR entries)

   b. as a FLOWEX boundary condition for an Euler face.

(Continued)

The second method can be activated by using a predefined name on the FLOWEX entry. The following name must be used:

INFLATR3 Inflator model, used for air bag calculations:

- The mass-flow rate must be input in TABLED1,1
- The temperature of the inflowing gas must be input in TABLED1,2
- The adiabatic constant of the gas [cp/cv] can be input by:

  PARAM,EXTRAS,GAMMA,value

  The default value is 1.4.
- The constant-volume specific heat of the gas can be input by:

  PARAM,EXTRAS,CV,value

  The default value is 743.
- The porosity coefficient of the eulerian faces can be input by:

  PARAM,EXTRAS,COEFFV,value

  The default value is 1.0.
  The area of the faces that will act as the inflow hole is equal to the uncovered part of the face area, multiplied by the value of COEFFV.

Note: The names INFLATOR and INFLATR2 are also allowed, but are previous versions of the inflator model, which have certain limitations.

FOAM1

Foam Material Properties

Defines the properties of an isotropic, crushable material where Poisson’s ratio is effectively zero.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FOAM1	MID	RHO	G	K	TABLE	TYPE			+CONT1
FOAM1	3	0.01		3.	111	CRUSH			+CONT1

+CONT1			BULKTYP	BULKQ	BULKL				
+CONT1			DYNA	1.4	0.05				

Field	Contents		Type	Default
MID	Unique material number.		I > 0	Required
RHO	Density.		R > 0	Required
G	Shear modulus.		R > 0	See Remark 3.
K	Bulk modulus.		R > 0	See Remark 3.
TABLE	Number of a TABLED1 entry defining the variation of stress (y-value) with crush factor or true strain (x-value).		I > 0	Required
TYPE	The type of data defined as the x-value in the table.		C	CRUSH
	CRUSH	Crush factor (1, relative volume).		
	STRAIN	True strain. See also Remark 4.		
BULKTYP	Bulk-viscosity model.		C	DYNA
	DYNA	Standard DYNA3D model.		
BULKQ	Quadratic bulk-viscosity coefficient.		R ≥ 0	1.0
BULKL	Linear bulk-viscosity coefficient.		R ≥ 0	0.0

Remarks:

1. If BULKTYP, BULKQ, or BULKL are blank or zero, the default values are used.
2. The continuation line with bulk-viscosity data can be omitted.
3. Poisson’s ratio for this model is effectively zero. Therefore, only one other elastic constant can be defined which can be G, the shear modulus, or K, the bulk modulus.

(Continued)

4. For this model, the stress-strain curve is independent of the experimental test performed to obtain the material data (uniaxial, shear, or volumetric). The most common test is the uniaxial compression test where the stress-strain characteristic can either be defined in terms of the amount of crush, which is minus the engineering strain, or in terms of the true strain. Since Poisson's ratio is effectively zero the amount of crush is defined as $\left(1-\frac{V}{V_0}\right)$, with $\frac{V}{V_0}$ as the relative volume, and the true strain is defined as

$$\int_{t_0}^{t} \frac{dV}{V}$$

   or $ln\left(\frac{V}{V_0}\right)$. The crush factor must be between 0 and 1. The true strain must always be negative and the stress positive (absolute value).

5. The yield surface in three-dimensional space is a sphere in principal stresses, and is defined by

$$\tau_{11}{}^2 + \tau_{22}{}^2 + \tau_{33}{}^2 = R_s{}^2$$

   where the radius of the sphere R_s depends on the strains as follows:

$$R_s = f(R_e)$$

   with

$$\varepsilon_{11}{}^2 + \varepsilon_{22}{}^2 + \varepsilon_{33}{}^2 = R_e{}^2$$

   and *f* is the function defined by the stress-strain table.

6. This material can only be used with Lagrangian solid elements.

7. The behavior of this material is described in Section 2.5.3.11 on page 2-58.

FOAM2

Foam Material Properties

Defines the properties of an isotropic, elastic foam material with user-specified hysteresis response for unloading, with strain rate dependency, and where Poisson's ratio is effectively zero.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FOAM2	MID	RHO	G	K	TABLE	TYPE	VALUE	CUTOFF	+CONT1
FOAM2	3	0.01		3.	111	CRUSH	-100.	SFRAC	+CONT1

+CONT1	TABY	ALPHA	UNLOAD						+CONT2
+CONT1	112	0.4	LINEAR						+CONT2

+CONT2			BULKTYP	BULKQ	BULKL				
+CONT2			DYNA	1.4	0.05				

Field	Contents	Type	Default
MID	Unique material number.	I > 0	Required
RHO	Density.	R > 0	Required
G	Shear modulus.	R > 0	See Remark 3.
K	Bulk modulus.	R > 0	See Remark 3.
TABLE	Number of a TABLED1 entry defining the variation of stress (y-value) with crush factor or true strain (x-value).	I > 0	Required
TYPE	The type of data defined as the x-value in the table: CRUSH Crush factor (=1–relative volume). STRAIN True strain. See also Remark 4.	C	CRUSH
VALUE	The value for cut-off stress.	R ≤ 0.0	-0.1 * Young's modulus.
CUTOFF	Cut-off stress. SFRAC Stress for tensile failure. SMIN Minimum stress.	C	SMIN
TABY	Number of a TABLED1 entry giving the variation of the scale factor for the stress (y-value) with the strain rate (x-value).	I > 0	See Remark 7.

(Continued)

Field	Contents		Type	Default
ALPHA	Energy dissipation factor.		0.0 ≤ R ≤ 1.0	Required
UNLOAD	Unloading option.		C	QDRATIC
	EXPTIAL	Unloading via exponential curve.		
	LINEAR	Unloading via piecewise linear curve.		
	QDRATIC	Unloading via quadratic curve.		
BULKTYP	Bulk-viscosity model:		C	DYNA
	DYNA	Standard DYNA3D model.		
BULKQ	Quadratic bulk-viscosity coefficient.		R ≥ 0	1.0
BULKL	Linear bulk-viscosity coefficient.		R ≥ 0	0.0

Remarks:

1. If BULKTYP, BULKQ, or BULKL are blank or zero, the default values are used.
2. The continuation line with bulk-viscosity data can be omitted.
3. Poisson's ratio for this model is effectively zero. Therefore, only one other elastic constant can be defined which can be G, the shear modulus, or K, the bulk modulus.
4. For this model, the stress-strain curve is independent of the experimental test performed to obtain the material data (uniaxial, shear, or volumetric). The most common test is the uniaxial compression test where the stress-strain characteristic can either be defined in terms of the amount of crush, which is minus the engineering strain, or in terms of the true strain. Since Poisson's ratio is effectively zero, the amount of crush is defined as $\left(1 - \frac{V}{V_0}\right)$, with $\frac{V}{V_0}$ as the relative volume, and the true strain is defined as

$$\int_{t_0}^{t} \frac{\mathrm{dV}}{V}$$

or $In\left(\frac{V}{V_0}\right)$. The crush factor must be between 0 and 1. The true strain must always be negative and the stress positive (absolute value).

(Continued)

5. The yield surface in three-dimensional space is a sphere in principal stresses, and is defined by

$$\tau_{11}^2 + \tau_{22}^2 + \tau_{33}^2 = R_s^2$$

where the radius of the sphere R_s depends on the strains and strain rates as follows:

$$R_s = f_1(R_e) f_2(R_r)$$

with

$$\varepsilon_{11}^2 + \varepsilon_{22}^2 + \varepsilon_{33}^2 = R_e^2$$

and

$$\dot{\varepsilon}_{11}{}^2 + \dot{\varepsilon}_{22}{}^2 + \dot{\varepsilon}_{33}{}^2 = R_r{}^2$$

and f_1 is the function supplied in the stress-strain table and f_2 (if defined) is the function supplied in the factor-strain rate table.

6. A minimum (SMIN) or failure (SFRAC) tensile stress can be defined. In the first case this corresponds to a tensile cut-off where the stress cannot fall below the minimum value. In the second case, if the stress falls below the failure stress the element fails and cannot carry tensile loading for the remainder of the analysis. Thus the stress can never become negative again.

7. If TABY is blank, the stress does not vary with strain rate. If TABY has a value, then it references to a TABLED1 entry, which gives the variation with strain rate of the scale factor applied to the stress.

8. The unloading behavior is piecewise linear (LINEAR), quadratic (QDRATIC) or exponential (EXPTIAL). The unloading curve is constructed such that the ratio of the dissipated energy (area between compressive loading and unloading curve) to total energy (area under the loading curve) is equal to the energy dissipation factor alpha. In the case of piecewise linear unloading, MSC.Dytran constructs an unloading curve whose segments are parallel to the supplied compression table, except for the first and last segments, which pass respectively through the origin and the point P on the compression curve where the unloading starts. In the case of quadratic unloading, MSC.Dytran constructs a quadratic curve starting in the origin and ending in point P. If the quadratic unloading curve falls below the strain axis, then the unloading stress is set to zero. In the case of exponential unloading, the unloading curve is constructed in a similarly to quadratic unloading except for the shape of the curve, which is created from an exponential function instead of a quadratic polynomial.

9. This material can only be used with Lagrangian solid elements.

10. The behavior of this material is described in Section 2.5.3.12 on page 2-58.

FORCE

Concentrated Load or Velocity

This entry is used in conjunction with a TLOADn entry and defines the location where the load or enforced motion acts, the direction in which it acts, and the scale factor.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FORCE	LID	G	CID	SCALE	N1	N2	N3		
FORCE	2	5	1	2.9		1.0			

Field	Contents	Type	Default
LID	Number of a set of loads.	I > 0	Required
G	Grid-point number or rigid body where the load is applied.	See Remark 4.	Required
CID	Number of a CORDxxx entry.	I ≥ 0	0
SCALE	Scale factor for the load.	R	1.0
N1, N2, N3	Components of a vector giving the load direction. At least one must be nonzero.	R	See Remark 6.

Remarks:

1. At time *t*, the load *F(t)* is given by

   $$F(t) = SCAL(E * N * T(t))$$

   where SCALE is the factor; *N* is the vector given by N1, N2, and N3; and *T(t)* is the value at *t* interpolated from the table referenced on the TLOADn entry.

2. Concentrated loads can also be defined on the DAREA entry.

3. LID must be referenced by a TLOADn entry.

4. If G references a MATRIG, an RBE2-FULLRIG, or a RIGID surface, the load is applied to the center of the rigid body. If G references a MATRIG, G must be MR<id>, where id is the MATRIG number. If G references an RBE2-FULLRIG, G must be FR<id>, where id is the RBE2 number. If G references a RIGID surface, G is the RIGID surface number.

5. If CID is specified, velocity prescriptions are processed in the local coordinate system referenced by CID. Only velocity prescriptions can be defined in the local coordinate system.

(Continued)

6. If a component field N1, N2, and/or N3 is left blank,

   Force prescription: The component of the force is equal to zero.

   Velocity prescription: The component of the velocity is not restrained.

7. If the TYPE field on the TLOADn entry is 0, it defines a force applied to a grid point. If the TYPE field is 2, it defines an enforced motion on the grid point. If the TYPE field is set to 12, it defines an enforced motion applied to the center of a rigid body, and if the TYPE field is 13, it defines a force applied to the center of a rigid body.

FORCE1

Follower Force, Form 1

This entry is used in conjunction with a TLOADn entry and defines a follower force with a direction that is determined by two grid points. FORCE1 applies to any type of grid point.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FORCE1	LID	G	SCALE	G1	G2				
FORCE1	2	5	2.9	16	13				

Field	Contents	Type	Default
LID	Number of a set of loads.	I > 0	Required
G	Grid-point number where the load is applied.	I > 0	Required
SCALE	Scale factor for the load.	R	1.0
G1, G2	Grid-point numbers. The direction of the load is a vector from G1 to G2. G1 must not be equal to G2.	I > 0	Required

Remarks:

1. At time *t*, the load $\underline{F}(t)$ is given by

$$\underline{F}(t) = SCALE * \underline{N} * T(t)$$

   where *SCALE* is the scale factor, $\underline{N}$ is the vector from G1 to G2, and $T(t)$ is the value at time *t* interpolated from the table referenced on the TLOADn entry.

2. LID must be referenced by a TLOADn entry.

3. The FORCE1 entry defines a follower force in that the direction of the force changes as the grid points G1 and G2 move during the analysis.

FORCE2

Follower Force, Form 2

This entry is used in conjunction with a TLOADn entry and defines a follower force with a direction that is determined by four grid points. FORCE2 can be applied to any type of grid point.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FORCE2	LID	G	SCALE	G1	G2	G3	G4		
FORCE2	2	5	2.9	16	13	17	18		

Field	Contents	Type	Default
LID	Number of a set of loads.	I > 0	Required
G	Grid-point number where the load is applied.	I > 0	Required
SCALE	Scale factor for the load.	R	1.0
G1-G4	Grid-point numbers. The load direction is determined by a vector product of the vectors from G1 to G2 and G3 to G4. (G1 must not be the same as G2, and G3 must not be the same as G4.)	I > 0	Required

Remarks:

1. At time *t*, the load $\underline{F}(t)$ is given by

   $$\underline{F}(t) = SCALE * \underline{N} * T(t)$$

   where *SCALE* is the scale factor, $\underline{N}$ is the vector product of the vectors from G1 to G2 and G3 to G4 respectively, and $T(t)$ is the value at time *t* interpolated from the table referenced by the TLOADn entry.

2. LID must be referenced by a TLOADn entry.

3. The FORCE2 entry defines a follower force in that the direction of the force changes as the grid points G1, G2, G3, and G4 move during the analysis.

FORCE3

Grid-Point Velocity Definition

Defines the velocity of a grid point in a local coordinate system or in a cascade of two local coordinate systems.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FORCE3	LID	G	CID1	SCALE1	N1	N2	N3		+CONT1
FORCE3	77	2		10.	1.	2.5			+CONT1

+CONT1	CID2	SCALE2	M1	M2	M3				
+CONT1									

Field	Contents	Type	Default
LID	Number of a set of loads.	I > 0	Required
G	Grid-point number.	I > 0	Required
CID1	Number of a coordinate system in which N1, N2, and N3 are defined.	I ≥ 0	0
SCALE1	Scale factor for the load.	R	1.0
N1, N2, N3	Components of a vector giving load direction.	R	See Remark 5.
CID2	Number of a coordinate system with respect to which coordinate system CID1 moves with an enforced motion equal to $\underline{M} * \text{SCALE2} * F(t)$.	I ≥ 0	0
SCALE2	Scale factor. For the enforced rigid-body motion of CID1.	R	1.0
M1, M2, M3	Components of a vector giving the enforced motion direction.	R	See Remark 5.

Remarks:

1. SCALE2 defines the enforced rigid-body motion of the coordinate system referenced by CID1 with respect to the coordinate system referenced by CID2.
2. This boundary condition can be used only to define the enforced velocities of grid points. Thus, the TYPE field in the TLOAD1 or TLOAD2 entry should be set to 2.
3. LID is referenced by a TLOAD entry.

(Continued)

4. If CIDx is specified, the velocity components are defined in the local coordinate directions, for example, if a cylindrical system is referenced, the velocity components define a radial, angular, and axial velocity.

5. If a component field N1, N2, N3, M1, M2, and/or M3 is left blank, that component of the velocity is not restrained.

6. The FORCE3 entry is valid for both Lagrangian as Eulerian gridpoints.

FORCEEX

User-Specified Enforced Motion at Grid Points

Defines enforced motion at grid points specified by a user subroutine.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FORCEEX	LID	NAME							+CONT1
FORCEEX	120	VEL7							+CONT1

+CONT1	G1	G2	G3	G4	THRU	G5	-etc.-		
+CONT1	100	319	728	429	THRU	457			

Field	Contents	Type	Default
LID	Number of a set of loads.	I > 0	Required
NAME	Constraint name passed to the user subroutine.	C	Required
Gi	Numbers of the grid points that are constrained. If the word “THRU” appears between two numbers, all the numbers in the range are included in the list. BY indicates the increment to be used within this range.	I > 0	Required

Remarks:

1. LID must be referenced by a TLOAD1 entry.
2. FORCEEX can only be used to specify enforced velocities for grid points. The TYPE field on the TLOAD1 entry must be set to two. The TID on the TLOAD1 entry must be set to zero or blank (no time variation).
3. The subroutine EXTVEL must be present in the file referenced by the USERCODE FMS statement.
4. See Section 3.15 on page 3-74 for a description of how to use user-written subroutines.
5. The constraint name is passed to the subroutine and can be used to identify the constraint.
6. A THRU specification, including the start and finish points in the range, must be on one line.
7. If the THRU specification is used, all the points in the sequence do not have to exist. Those that do not exist are ignored. The first point in the THRU specification must be a valid grid point. BY can be used to exclude grid points.
8. None of the fields in the list of grid points can be blank or zero, since this designation marks the end of the list.
9. Any number of continuation lines can be used to define the list of grid points.

GBAG

Gas-Bag Pressure Definition

Defines the pressure within an enclosed volume.

Format and Example:

1	2	3	4	5	6	7	8	9	10
GBAG	GID	SID	TRIGGER	TRIGGERV	PORID	INFID	HTRID		+CONT1
GBAG	101	37	TIME	0.0					+CONT1

+CONT1	CDEX	CDEXV	AEX	AEXV	CDLEAK	CDLEAKV	ALEAK	ALEAKV	+CONT2
+CONT1	TABLE	201	TABLE	202	TABLE	203	TABLE	204	+CONT2

+CONT2	FLGAS	FLGASV	TGAS	TGASV	VOLPOR	VOLPORV			+CONT3
+CONT2	TABLE	205	TABLE	206	TABLE	209			+CONT3

+CONT3	CPGAS	CPGASV	RGAS	PENV	PEX	REVERSE	CHECK	PINIT	+CONT4
+CONT3	CONSTANT	1.0							+CONT4

+CONT4	TINIT	TENV							+CONT5
+CONT4	7.								+CONT5

+CONT5	CONVEC	CONVECV	ACONVEC	ACONVECV					+CONT6
+CONT5									+CONT6

+CONT6	RADIAT	RADIATV	ARADIAT	ARADIATV	SBOLTZ				
+CONT6									

Field	Contents	Type	Default
GID	Unique gas-bag number.	I > 0	Required
SID	Number of a SURFACE entry defining the geometry of the gas bag.	I > 0	Required
TRIGGER	The time-dependent parameters are offset in time. TIME The offset is defined at TRIGGERV.	C	TIME
TRIGGERV	The value of the offset in time.	R	Required

(Continued)

Field	Contents	Type	Default
PORID	Number of a set of GBAGPOR entries, that defines the porosity (permeability) and holes for the gas-bag surface and/or subsurfaces.		
INFID	Number of a set of GBAGINFL entries, that defines the one or more inflators on subsurface(s) of the GBAG surface.		
HTRID	Number of a set of GBAGHTR entries, that defines the heat transfer definitions for the gas-bag surface and/or subsurfaces.		
CDEX	The variation of the discharge coefficient for the exhaust openings.	C	CONSTANT
	CONSTANT The discharge coefficient is constant and is specified in CDEXV.		
	TABLE The discharge coefficient varies with pressure. CDEXV is the number of a TABLED1 or TABLEEX entry giving the variation of the discharge coefficient (y-value) with the pressure (x-value).		
	TIME The area varies with time. CDEXV is the number of a TABLED1 or TABLEEX entry giving the variation of the discharge coefficient (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.		
CDEXV	The discharge coefficient or the number of a TABLED1 or TABLEEX entry, depending on the value of CDEX. Discharge coefficients must be between zero and one.	R or I > 0	1.0

(Continued)

Field	Contents	Type	Default
AEX	The variation of the total area of the exhaust openings.	C	CONSTANT
	CONSTANT The area is constant and is specified in AEXV.		
	TABLE The area varies with pressure. AEXV is the number of a TABLED1 or TABLEEX entry giving the variation of the area (y-value) with the pressure (x-value).		
	TIME The area varies with time. AEXV is the number of a TABLED1 or TABLEEX entry giving the variation of the area (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.		
AEXV	The total area of the exhaust openings or the number of a TABLED1 or TABLEEX entry, depending on the value of AEX.	R or I > 0	0.0
CDLEAK	The variation of the discharge coefficient for the permeability of the gas-bag fabric.	C	CONSTANT
	CONSTANT The discharge coefficient is constant and is specified in CDLEAKV.		
	TABLE The discharge coefficient varies with pressure. CDLEAKV is the number of a TABLED1 or TABLEEX entry giving the variation of discharge coefficient (y-value) with the pressure (x-value). The discharge coefficient must be between zero and one.		
	TIME The discharge coefficient varies with time. CDLEAKV is the number of a TABLED1 or TABLEEX entry giving the variation of the discharge coefficient (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.		
CDLEAKV	The discharge coefficient or the number of a TABLED1 or EXFUNC entry, depending on the value of CDLEAK.	R or I > 0	1.0

(Continued)

Field	Contents	Type	Default
ALEAK	The variation of the total leak area.	C	CONSTANT
	CONSTANT — The area is constant and is specified in ALEAKV.		
	TABLE — The area varies with pressure. ALEAKV is the number of a TABLED1 or TABLEEX entry giving the variation of the area (y-value) with the pressure (x-value).		
	TIME — The area varies with time. ALEAKV is the number of a TABLED1 or TABLEEX entry giving the variation of the area (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.		
ALEAKV	The total leak area or the number of a TABLED1 or TABLEEX entry, depending on the value of AEX.	R or I > 0	0.0
FLGAS	The variation of the total mass flux of the inflowing gas. The mass flux is in mass-per-unit time.	C	CONSTANT
	CONSTANT — The mass flux is constant and specified in FLGASV. Flow STARTS at the time specified on the TRIGGERV entry.		
	TABLE — The mass flux varies with time. FLGASV is the number of a TABLED1 or TABLEEX entry giving the variation of the mass flux (y-value) with time (x-value). The table is offset by the time specified on TRIGGERV entry.		
FLGASV	The mass flux or the number of a TABLED1 or TABLEEX entry, depending on the value of FLGAS.	R or I > 0	Required

(Continued)

Field	Contents	Type	Default
TGAS	The variation of the temperature of the inflowing gas.	C	CONSTANT
	CONSTANT The temperature is constant and specified in TGASV.		
	TABLE The temperature varies with time. TGASV is the number of a TABLED1 or TABLEEX entry giving the variation of the temperature (y-value) with the time (x-value). The table is offset by the time specified on the TRIGGERV entry.		
TGASV	The temperature of the inflowing gas or the number of a TABLED or TABLEEX entry depending on the value of TGAS.	R or I > 0	Required
VOLPOR	User-defined volumetric flow rate volume-per-unit time. See Remark 5.	C	CONSTANT
	CONSTANT The outflow rate is constant and specified in VOLPORV.		
	TABLE The outflow rate varies with pressure. VOLPORV is the number of a TABLED1 or TABLEEX entry giving the variation of the outflow rate (y-value) with the pressure (x-value).		
	TIME The outflow rate varies with time. VOLPORV is the number of a TABLED1 or TABLEEX entry giving the variation of the outflow rate (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.		
VOLPORV	The flow rate or the number of a TABLED1 or TABLEEX entry, depending on the value of VOLPOR.	R ≥ 0.0 or I > 0	0.0
CPGAS	The variation of the specific heat constant at constant pressure.	C	CONSTANT
	CONSTANT The specific heat is constant and specified in CPGASV.		
CPGASV	The specific heat of the gas.	R	Required
RGAS	Gas constant of the inflowing gas.	R	Required

(Continued)

Field	Contents		Type	Default
PENV	Environmental pressure surrounding the gas bag.		R	Required
PEX	There is only outflow from the gas bag if the pressure in the gas bag is greater than PEX.		R	PENV
REVERSE	Normal auto-reverse switch.		C	ON
	ON	The normals of the SURFACE are automatically reversed if necessary so that they point in the same direction and provide a positive volume.		
	OFF	The normals are not automatically reversed.		
CHECK	Normal checking switch.		C	ON
	ON	The normals of the SURFACE are checked to see if they all point in the same direction and provide a positive volume.		
	OFF	The normals are not checked.		
	If REVERSE is set to ON, CHECK is automatically set to ON.			
PINIT	Initial pressure inside the gas bag.		R	PENV
TINIT	Initial temperature inside the gas bag.		R	Required.
	See Remark 4.			
TENV	Environmental Temperature. The value is required when heat transfer is used.		R > 0	Required. See Remark 6.
CONVEC	The variation of the heat transfer coefficient for convection heat transfer.		C	CONSTANT
	CONSTANT	The heat transfer coefficient is constant and specified in CONVECV.		
	TABLE	The heat transfer coefficient varies with time. VONVECV is the number of a TABLED1 or TABLEEX entry giving the variation of the heat transfer coefficient (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.		

(Continued)

Field	Contents		Type	Default
CONVECV	The heat transfer coefficient or the number of a TABLED1 or TABLEEX entry, depending on value of CONVEC.		R or I > 0	0.0
ACONVEC	The variation of the total surface area to be used in the convective heat transfer equations. The area is calculated by multiplying the total area of the GBAG surface with the value of this coefficient.			
	CONSTANT	The area coefficient is constant and specified in ACONVECV.		
	TABLE	The area coefficient varies with time. ACONVECV is the number of a TABLED1 or TABLEEX entry giving the variation of the heat transfer coefficient (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.		
ACONVECV	The area coefficient of the number of a TABLED1 or TABLEEX entry, depending on value of AVONCEC.		R or I > 0	1.0
RADIAT	The variation of the gas emissivity coefficient for radiation heat transfer.		C	CONSTANT
	CONSTANT	The gas emissivity coefficient is constant and specified in RADIATV.		
	TABLE	The gas emissivity coefficient varies with time. RADIATV is the number of a TABLED1 or TABLEEX entry giving the variation of the gas emissivity coefficient (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.		
RADIATV	The gas emissivity coefficient or the number of a TABLED1 or TABLEEX entry, depending on value of RADIAT.		R or I > 0	0.0

(Continued)

Field	Contents	Type	Default
ARADIAT	The variation of the total surface area to be used in the radiation heat transfer equations. The area is calculated by multiplying the total area of the GBAG surface with the value of this coefficient. CONSTANT The area coefficient is constant and specified in ARADIATV. TABLE The area coefficient varies with time. ARADIATV is the number of a TABLED1 or TABLEEX entry giving the variation of the heat transfer coefficient (y-value) with the (x-value). The table is offset by the time specified on the TRIGGERV entry.	C	CONSTANT
ARADIATV	The area coefficient or the number of a TABLED1 or TABLEEX entry, depending on value of ARADIAT.	R or I > 0	1.0
SBOLTZ	Stefan-Boltzmann constant.	R	0.0

Remarks:

1. The SURFACE entry referenced by the SID field must form a closed volume.
2. The pressure in the gas bag is applied to all the faces of the SURFACE.
3. TABLEEX is valid also in all entries where TABLED1 is used.
4. TINIT is the temperature of the inflowing gas at *time* = 0. At *time* = 0, the mass of the gas inside the gas bag is calculated as

$$m = \frac{p_{init}V}{RT_{init}}$$

where, p_{init} the initial pressure, V the volume, R the gas constant, and T_{init} the initial gas temperature.

(Continued)

5. The flow through exhaust openings, leakage areas and user-specified outflow rate is accumulated. The volumetric porosity contributes to the outflow of gas as

$$\dot{m}_{out} = \rho * Q = \frac{p}{R * T} * Q$$

where: Q = volumetric flow rate

ρ = density inside the bag

P = pressure inside the bag

R = gas constant

T = temperature inside the bag

$\dot{m}_{out}$ = mass outflow rate

The value of Q can be specified as a constant, as a function of the pressure difference, or as a function of time. Negative values for the volumetric flow rate are not allowed, since this would mean inflow of outside air.

6. The heat-transfer rates are given by the following equations:

$$q_{conv} = hA_c(T - T_{env}) \qquad \text{CONVECTION}$$

$$(q_{rad} = eA_r(T^4 - T_{env}^4)) \qquad \text{RADIATION}$$

where h is the convection heat-transfer coefficient (CONVEC, CONVECV), e the gas emissivity coefficient (RADIAT, RADIATV), A_c the air-bag surface area for convective heat transfer, A_r the air-bag surface area for radiation, and T_{env} the environmental temperature.

GBAGC

Gas-Bag Connection

Connection between two gas bags.

Format and Example:

1	2	3	4	5	6	7	8	9	10
GBAGC	ID	GID1	GID2	TRIGGER	TRIGGERV	PRESTOL			+CONT1
GBAGC	100	11	12	TIME	0.0	0.0			+CONT1

+CONT1	CD12	CD12V	A12	A12V	CD21	CD21V	A21	A21V	
+CONT1	CONSTANT	0.8	CONSTANT	3.0	TABLE	12	TABLE	13	

Field	Contents		Type	Default
ID	Number of the GBAGC entry.		I > 0	Required
GID1	Number of a GBAG entry.		I > 0	Required
GID2	Number of a GBAG entry, different from GID1.		I > 0	Required
TRIGGER	The time-dependent parameters are offset in time.		C	TIME
	TIME	The offset is defined at TRIGGERV.		
TRIGGERV	The value of the offset in time.		R	Required
PRESTOL	If the pressure difference between the two gas bags is less than this value, no mass flow will occur. The value is specified as a percentage.		R ≥ 0.0	0.0
CD12	The variation of the discharge coefficient for the opening allowing flow from gas bag 1 into gas bag 2.		C	CONSTANT
	CONSTANT	The discharge coefficient is constant and is specified in CD12V.		
	TABLE	The discharge coefficient varies with pressure. CD12V is the number of a TABLED1 or TABLEEX entry giving the variation of the discharge coefficient (y-value) with the pressure (x-value).		
	TIME	The area varies with time. CD12V is the number of a TABLED1 or TABLEEX entry giving the variation of the discharge coefficient (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.		

(Continued)

Field	Contents	Type	Default
CD12V	The discharge coefficient or the number of a TABLED1 or TABLEEX entry depending on the value of CD12. Discharge coefficients must be between zero and one.	R or I > 0	1.0
A12	The variation of the total area of the opening that allows flow from gas bag 1 into gas bag 2.	C	CONSTANT
	CONSTANT The area is constant and specified in A12V.		
	TABLE The area varies with pressure. A12V is the number of a TABLED1 or TABLEEX entry giving the variation of the area (y-value) with the pressure (x-value).		
	TIME The area varies with time. A12V is the number of a TABLED1 or TABLEEX entry giving the variation of the area (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.		
A12V	The total area of the opening or the number of a TABLED1 or TABLEEX entry, depending on the value of A12.	R or I > 0	0.0
CD21	The variation of the discharge coefficient for the opening that allows flow from gas bag 2 into gas bag 1.	C	CONSTANT
	CONSTANT The discharge coefficient is constant and is specified in CD21V.		
	TABLE The discharge coefficient varies with pressure. CD21V is the number of a TABLED1 or TABLEEX entry giving the variation of discharge coefficient (y-value) with the pressure (x-value). The discharge coefficient must be between zero and one.		
	TIME The discharge coefficient varies with time. CD21V is the number of a TABLED1 or TABLEEX entry giving the variation of the discharge coefficient (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.		

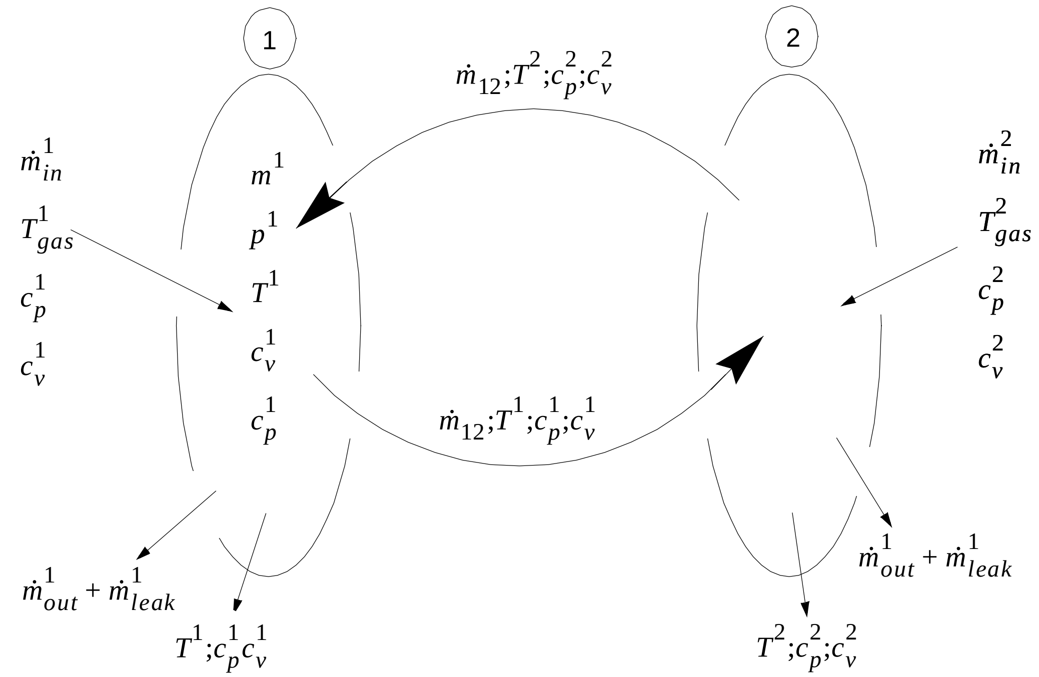
(Continued)

Field	Contents	Type	Default
CD21V	The discharge coefficient or the number of a TABLED1 or TABLEEX entry, depending on the value of CD21.	R or I > 0	1.0
A21	The variation of the total area of the opening that allows flow from gas bag 2 into gas bag 1. CONSTANT The area is constant and specified in A21V. TABLE The area varies with pressure. A21V is the number of a TABLED1 or TABLEEX entry giving the variation of the area (y-value) with the pressure (x-value). TIME The area varies with time. A21V is the number of a TABLED1 or TABLEEX entry giving the variation of the area (y-value) with time (x-value). The table is offset by the time specified on the TRIGGERV entry.	C	CONSTANT
A21V	The total area of the opening or the number of a TABLED1 or TABLEEX entry, depending on the value of A21.	R or I > 0	0.0

(Continued)

Remarks:

1. Both GBAGs are triggered before flow between the two gas bags begins.
2. The energy balance and mass flow is as shown below:



3. One GBAG can be referenced in multiple GBAGC entries.
4. For compartmented air bags, you can model each compartment as a separate gas bag and connect the gas bags using GBAGC entries.
5. The GBAGC entry is obsolete. It is preferable to model connections between GBAG entries using the GBAG → GBAGPOR → PORFGBG logic. See Section 2.17 on page 2-127 for more details.

GBAGCOU

General Coupling to Gas-Bag Switch

Defines a switch from full gas dynamics to uniform pressure formulation.

Format and Example:

1	2	3	4	5	6	7	8	9	10
GBAGCOU	ID	CID	GID	TSTART	PERCENT				
GBAGCOU	1	100	101	0.0	5				

Field	Contents	Type	Default
ID	Unique number of a GBAGCOU entry.	I > 0	Required
CID	Number of a COUPLE entry.	I > 0	Required
GID	Number of a GBAG entry.	I > 0	Required
TSTART	Time after which the coupling algorithm checks if a switch to the uniform pressure method is valid. It is valid when the following is true:	R > 0.	0.0

$$Max\left[\frac{(Pmax - Paverage)}{Paverage}, \frac{(Paverage–Pmin)}{Paverage}\right] < \frac{\text{PERCENT}}{100}$$

where Pmax = maximum Eulerian pressure exerted on the SURFAE

Pmin = minimum Eulerian pressure exerted on the SURFACE

Paverage = average Eulerian pressure exerted on the SURFAE

Field	Contents	Type	Default
PERCENT	Value used in validity check as defined above.	R > 0	5%

Remarks:

1. The SURFACE SID referenced by the COUPLE entry CID and by the GBAG entry GID must be equal.
2. All Eulerian and general coupling calculations are deactivated after transition from gas dynamics to uniform pressure.

GBAGHTR

Heat Transfer Model to be Used With GBAG Entry

Defines the heat-transfer model to be used with GBAG.

Format and Example:

1	2	3	4	5	6	7	8	9	10
GBAGHTR	CID	HTRID	SUBID	HTRTYPE	HTRTYPID	COEFF	COEFFV		
GBAGHTR	101	83		HTRCONV	2	TABLE	14		

Field	Contents		Type	Default
CID	Unique number of a GBAGHTR entry.		I > 0	Required
HTRID	Number of a set of GBAGHTR entries HTRID must be referenced from a GBAG entry.		I > 0	Required
SUBID	> 0	Number of a SUBSURF, which must be a part of the SURFACE referred to from the GBAG entry.	I ≥ 0	0
	= 0	GBAGHTR definitions are used for the entire SURFACE referred to from the GBAG entry.		
HTRTYPE	Defines the type of heat transfer.		C	
	HTRCONV	The HTRCONV logic is used to model heat transfer through convection in an air bag. The area of convection is defined by a subsurface (SUBID). The area of convection through which the energy is transported is equal to the area of the subsurface multiplied by COEFFV. A value of COEFFV = 1.0 will expose the complete subsurface area, while a value of COEFFV = 0.0 will result in no heat transfer through the subsurface.		

(Continued)

Field	Contents		Type	Default
	HTRRAD	The HTRRAD logic is used to model heat transfer through radiation in an air bag. The area of convection is defined by a subsurface (SUBID). The area of convection through which the energy is transported is equal to the area of the subsurface multiplied by COEFFV. A value of COEFFV = 1.0 will expose the complete subsurface area, while a value of COEFFV = 0.0 will result in no heat transfer through the subsurface.		
COEFF	Method of defining the area coefficient.		C	CONSTANT
	CONSTANT	The area coefficient is constant and specified on COEFFV.		
	TABLE	The area coefficient varies with time. COEFFV is the number of a TABLED1 or TABLEEX entry giving the variation with time.		
COEFFV	The area coefficient or the number of a TABLED1 or TABLEEX entry depending on the COEFF entry.		0 < R < 1 or I > 0	1.0

Remarks

1. A combination of multiple GBAGHTRs with different HTRTYPEs is allowed.
2. All options of HTRTYPE can also be referenced by a COUHTR. It allows for setting up the exact same model for either a uniform pressure model (GBAG to GBAGHTR) or an Eulerian model (COUPLE to COUHTR). It is then possible to set up the model using the switch from full gas dynamics to uniform pressure (GBAGCOU).
3. For the same SUBSURF multiple, different types of heat transfer may be defined.
4. A more detailed description can be found in Section 2.17 on page 2-127.

GBAGINFL

Inflator Model to be Used With GBAG Entry

Defines an inflator model suited for airbag analyses using the uniform pressure approach (GBAG). The inflator model is defined as part of the GBAG surface.

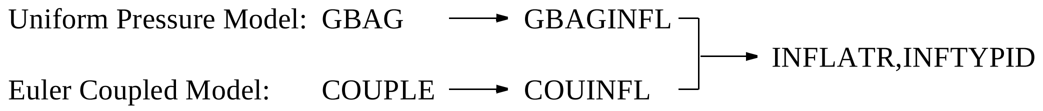
1	2	3	4	5	6	7	8	9	10
GBAGINFL	CID	INFID	SUBID	INFTYPE	INFTYPID	COEFF	COEFFV		
GBAGINFL	201	1	120	INFLATR	11		0.012		

Field	**Contents**	**Type**	**Default**
CID	Unique number of a GBAGINFL entry.	I > 0	Required
INFID	Number of a set of GBAGINFL entries NFID must be referenced from a GBAG entry.	I > 0	Required
SUBID	Number of a SUBSURF, which must be a part of the SURFACE referred to from the GBAG entry.	I>0	Required
INFTYPE	Defines the type of inflator.	C	
	INFLATR The INFLATR logic is used to model inflators in an air bag. The inflator is defined by a subsurface (SUBID). The area of the hole through which the gas enters is equal to the area of the subsurface multiplied by COEFFV. A value of COEFFV = 1.0 will open up the complete subsurface area, while a value of COEFFV = 0.0 will result in a closed inflator area with no inflow.		
INFTYPID	Number of the entry selected under INFTYPE, for example, INFLATR,INFTYPID.	I	Required
COEFF	Method of defining the area coefficient.	C	CONSTANT
	CONSTANT The area coefficient is constant and specified on COEFFV.		
	TABLE The area coefficient varies with time. COEFFV is the number of a TABLED1 or TABLEEX entry giving the variation with time.		
COEFFV	The area coefficient or the number of a TABLED1 or TABLEEX entry depending on the COEFF entry.	0 < R < 1 or I > 0	1.0

(Continued)

Remarks:

1. The same INFTYPE entry referenced from this GBAGINFL entry can be referenced by a COUINFL entry. This allows for setting up the exact same model for either a uniform pressure model or an Euler Coupled model:



   This makes it possible to set up the model using the switch from full gas dynamics to uniform pressure (GBAGCOU).

2. One GBAG entry can reference more than one COUINFL entry. This allows for modeling of multiple inflators in an airbag module.

GBAGPOR

Gasbag Porosity

Defines the porosity model to be used with GBAG.

Format and Example:

1	2	3	4	5	6	7	8	9	10
GBAGPOR	CID	PORID	SUBID	PORTYPE	PORTYPID	COEFF	COEFFV		
GBAGPOR	7	100	365	PERMEAB	63		0.99		

Field	Contents		Type	Default
CID	Unique number of a GBAGPOR entry.		I > 0	Required
PORID	Number of a set of GBAGPOR entries. PORID must be referenced from a GBAG entry.		I > 0	Required
SUBID	> 0	Number of a SUBSURF, which must be a part of the SURFACE referred to from the GBAG entry.	I ≥ 0	0
	= 0	GBAGPOR definitions are used for the entire SURFACE referred to from the GBAG entry.		
PORTYPE	Defines the type of porosity.			
	PORHOLE	The PORHOLE logic is used to model holes in an air bag. The hole is defined by a subsurface (SUBID). The open area of the hole is equal to the area of the (sub)surface multiplied by COEFFV. A value of COEFFV = 1.0 will open up the complete hole area, while a value of COEFFV = 0.0 will result in a closed hole. The velocity of the gas flow through the hole is based on the theory of one-dimensional gas flow through a small orifice. The velocity depends on the pressure difference. The characteristics for the flow are defined on a PORHOLE entry, with ID as defined on the PORTYPID.		

(Continued)

Field	Contents		Type	Default
	PERMEAB	The PERMEAB logic is used to model permeable air-bag material. The permeable area can be defined for a subsurface (SUBID) or for the entire coupling surface. The velocity of the gas flow through the (sub)surface is defined as a linear or tabular function of the pressure difference between the gas inside the air bag and the environmental pressure. The function is specified on a PERMEAB entry, with ID as defined on the PORYPID. The area actually used for outflow is the subsurface area multiplied by the value of COEFFV.		
	PORFGBG	The PORFGBG logic is used to model gas flow through a hole in the coupling surface connected to a GBAG. The hole is defined by a subsurface (SUBID). The open area of the hole is equal to the area of the (sub)surface multiplied by COEFFV. A value of COEFFV = 1.0 will open up the complete hole area, while a value of COEFFV = 0.0 will result in a closed hole. The velocity of the gas flow through the hole is based on the theory of one-dimensional gasflow through a small orifice. The velocity depends on the pressure difference. The characteristics for the flow are defined on a PORFGBG entry, with ID as defined on the PORTYPID.		
	PERMGBG	The PERMGBG logic is used to model gas flow through a permeable area in the coupling surface connected to a GBAG. The permeable area can be defined for a subsurface (SUBID) or for the entire coupling surface. The velocity of the gas flow through the (sub)surface is defined as a linear or tabular function of the pressure difference. This function is specified on a PERMGBG entry, with ID as defined on the PORYPID. The area actually used for outflow is the subsurface area multiplied by the value of COEFFV.		

(Continued)

GBAGPOR
Gasbag Porosity

Field	Contents		Type	Default
COEFF	Method of defining the porosity coefficient.		C	CONSTANT
	CONSTANT	The porosity coefficient is constant and specified on COEFFV.		
	TABLE	The porosity coefficient varies with time. COEFV is the number of a TABLED1 or TABLEEX entry giving the variation with time.		
COEFFV	The porosity coefficient or the number of a TABLED1 or TABLEEX entry depending on the COEFF entry.		0.0 < R < 1.0 or I > 0	

Remarks:

1. The combination of multiple GBAGPORs with different PORTYPEs is allowed.
2. All options of PORTYPE can also be referenced by a COUPOR. It allows for setting up the exact same model for either a uniform pressure model (GBAG to GBAGPOR) or an Eulerian model (COUPLE to COUPOR). It is then possible to set up the model using the switch from full gas dynamics to uniform pressure (GBAGCOU).
3. The options PORFGBG and PERMGBG can be used to model air bags with different compartments.

GRAV

Gravity Field

Defines a gravity acceleration field.

Format and Example:

1	2	3	4	5	6	7	8	9	10
GRAV	LID		SCALE	NX	NY	NZ			
GRAV	4		–2.0	0.	1.				

Field	Contents	Type	Default
LID	Number of a set of loads.	I > 0	Required.
SCALE	Gravity scale factor.	R ≥ 0.	1.0
NX, NY, NZ	Components of gravity vector. At least one component must be nonzero.	R ≥ 0.	0.0

Remarks:

1. The gravity acceleration $g(t)$ is defined as

$$g(t) = T(t) * \text{SCALE} * \underline{N}$$

where SCALE is the gravity scale factor; *N* is the vector defined by NX, NY, and NZ; and $T(t)$ is the value interpolated at time *t* from the table referenced by the TLOADn entry.

2. LID must be referenced by a TLOADn entry.
3. The type field on the TLOADn entry must be set to zero.
4. One gravitational field can be defined per problem.
5. The gravitational accelerations are applied to all masses in the problem.

GRDSET

Grid-Point Default

Defines default options for the GRID entries.

Format and Example:

1	2	3	4	5	6	7	8	9	10
GRDSET		CP					PS		
GRDSET							3456		

Field	Contents	Type	Default
CP	Number of a coordinate system in which the location of the grid point is defined.	I ≥ 0	0
PS	Single-point constraints associated with the grid point. This should be an integer of any of the digits 1 through 6.	I > 0	0

Remarks:

1. Any GRID entry with a blank value of PS is set to the value given on this entry. Note that the constraints on the GRID and GRDSET entries are not cumulative, i.e., if there is a GRDSET entry with constraint code 34 and a GRID entry with constraint code 2, the grid point is constrained only in direction 2.

2. There can only be one GRDSET entry in the input data.

Defines the location of a geometric grid point in the model and its constraints.

Format and Example:

1	2	3	4	5	6	7	8	9	10
GRID	ID	CP	X	Y	Z		PS		
GRID	2		1.0	–2.0	3.0		316		

Field	Contents	Type	Default
ID	Unique grid-point number.	I > 0	Required
CP	Number of a coordinate system in which the location of the grid point is defined.	I ≥ 0	See Remark 2.
X, Y, Z	Location of the grid point.	R	0.0
PS	Permanent single-point constraints associated with the grid point. This must be an integer made up of the digits 1 through 6 with no embedded blanks.	I > 0	See Remark 2.

Remarks:

1. All grid-point numbers must be unique.
2. If CP or PS is blank or zero, the value given on the GRDSET entry is used.
3. Grid points can also be constrained using the SPC and SPC1 entries.
4. The values of X, Y and Z depend on the type of the coordinate system CP. Their meaning in each type of coordinate system is listed in the following table.

Type	X	Y	Z
Rectangular	X	Y	Z
Cylindrical	R	θ	Z
Spherical	R	θ	ϕ

θ and ϕ are measured in degrees.

GROFFS

Grid-Point Offset

Defines a grid-point offset in the local coordinate system.

Format and Example:

1	2	3	4	5	6	7	8	9	10
GROFFS	ID	SID	XOFF	YOFF	ZOFF				
GROFFS	32	2	8.E-4	0.75	0.0				

Field	Contents	Type	Default
ID	Unique grid-point offset number.	I > 0	Required
SID	Number of a SET1 entry containing a list of grid points.	I > 0	Required
XOFF, YOFF, ZOFF	Components of a vector in a local coordinate system defining the grid-point offset.	R	0.0

HGSUPPR

Hourglass Suppression Method

Defines the hourglass suppression method and the corresponding hourglass damping coefficients.

Format and Example:

1	2	3	4	5	6	7	8	9	10
HGSUPPR	HID	PROP	PID	HGTYPE	HGCMEM	HGCWRP	HGCTWS	HGCSOL	+CONT1
HGSUPPR	1	SHELL	100	FBV	0.1	0.1	0.1		+CONT1

+CONT1	RBRCOR	VALUE							
+CONT1	YES	10000							

Field	Contents	Type	Default
HID	Hourglass suppression definition number.	I > 0	Required
PROP	Property type.	C	See Remark 1.
PID	Property number.	I > 0	See Remark 1.
HGTYPE	Hourglass suppression method.	C	See Remark 2.
HGCMEM	Membrane damping coefficient.	$0.0 \leq R \leq 0.15$	0.1
HGCWRP	Warping damping coefficient.	$0.0 \leq R \leq 0.15$	0.1
HGCTWS	Twisting damping coefficient.	$0.0 \leq R \leq 0.15$	0.1
HGCSOL	Solid damping coefficient.	$0.0 \leq R \leq 0.15$	0.1
RBRCOR	Rigid body rotation correction: NO No rigid-body rotation correction is applied to hourglass resisting forces. YES Rigid-body rotation correction is applied to hourglass resisting forces. See also Remark 3.	C	NO
VALUE	Number of steps.	R > 0.0	See Remark 3.

(Continued)

Remarks:

1. The property type definition and the property number are required. Since property numbers are unique within a certain class of element types, the property type and the property number uniquely define to what elements the hourglass suppression method and coefficients apply. The following property types are valid entries:

BAR	For bar elements.
BEAM	For beam elements.
BELT	For belt elements.
COMP	For composite shell elements.
DAMP	For damper elements.
ELAS	For spring elements.
EULER	For Eulerian elements.
ROD	For rod elements.
SHELL	For shell elements.
SOLID	For solid Lagrangian elements.

   It must be noted however, that only shell CQUAD4 and Lagrangian CHEXA and CPENTA elements can suffer from undesired hourglass modes. All HGSUPPR entries referring to other types of elements will therefore be ignored.

2. There are three types of hourglass suppression methods available in MSC.Dytran. These are standard DYNA viscous (DYNA) hourglass damping, the Flanagan-Belytschko Stiffness (FBS) hourglass damping, and the Flanagan-Belytschko Viscous (FBV) hourglass damping.

   Lagrangian solid elements can address DYNA and FBS suppression, shell elements can address DYNA and FBV suppression. The default for the Lagrangian solid elements is FBS. The default for the shell elements is FBV.

   The default hourglass suppression method can be globally changed by the PARAM,HGTYPE.

3. The rigid-body rotation correction on the hourglass forces is only necessary in cases where shell elements undergo a large rigid-body rotation. If the RBRCOR field is set to YES, and the VALUE field is left blank, the correction will be applied during each time step. If the VALUE field is set to a number, the rotation correction will be applied only when the rigid-body rotation would result in a rotation of the element over 90° in less than VALUE time steps. Usually, if the rigid-body rotation correction is necessary, 10000 is a good value. This option saves some CPU time.

   The RBRCOR option applies to the Key-Hoff shell formulation only; for all other element types and formulations, the option is ignored.

(Continued)

4. The membrane, warping and twisting coefficients apply to shell elements only; for all other element types, the data is ignored. The solid damping coefficient applies to solid Lagrangian elements only; for all other element types, the data is ignored.

   The default value of the damping coefficients can be globally changed by PARAM,HGCOEFF.

5. See Section 2.5.10.2 on page 2-80 for details on hourglass suppression.

HTRCONV

Air Bag Convection

Defines the heat transfer through convection for a COUPLE and/or GBAG (sub)surface.

Convection is heat transfer from the air bag to the environment through the air bag surface.

Format and Example:

1	2	3	4	5	6	7	8	9	10
HTRCONV	HTRID	HTRCF-C	HTRCF-T	TENV					
HTRCONV	8		14	293.0					

Field	Contents	Type	Default
HTRID	Unique number of a HTRCONV entry.	I > 0	Required
HTRCF-C	Constant heat transfer convection coefficient.	R > 0	See Remark 3.
HTRCF-T	The heat transfer convection coefficient is a tabular function of time. The number given here is the number of a TABLED1 or TABLEEX entry.	I > 0	See Remark 3.
TENV	Environmental temperature.	R > 0	Required

Remarks:

1. The HTRCONV entry can be referenced from a COUHTR and/or GBAGHTR entry.
2. When used with Euler, the entry can only be used with the single material hydrodynamic Euler solver using an EOSGAM (ideal gas) equation of state. Two of the four gasconstants (γ, R, c_v and/or c_p) have to be defined on the EOSGAM entry.
3. Either HTRCF-C or HTRCF-T must be specified.
4. Energy will only transfer out of the air bag if the temperature inside the air bag is higher than the environmental temperature.
5. A more detailed description of heat transfer by convection can be found in Section 2.19 on page 2-133.

HTRRAD

Air Bag Radiation

Defines the heat transfer through radiation for a COUPLE and/or GBAG (sub)surface.

Radiation is heat transfer from the air bag to the environment through the air bag surface.

Format and Example:

1	2	3	4	5	6	7	8	9	10
HTRDAD	HTRID	GASEMI-C	GASEMI-T	TENV	SBOLTZ				
HTRRAD	2	0.15		293.0	5.676E-8				

Field	Contents	Type	Default
HTRID	Unique number of a HTRRAD entry.	I > 0	Required
GASEMI-C	Constant gas emissivity.	R > 0	See Remark 3.
GASEMI-T	The gas emissivity is a tabular function of time. The number given here is the number of a TABLED1 or TABLEEX entry.	I > 0	See Remark 3.
TENV	Environmental temperature.	R > 0	Required
SBOLTZ	Stephan-Boltzman constant.	R > 0	Required

Remarks:

1. The HTRRAD entry can be referenced from a COUHTR and/or GBAGHTR entry.
2. When used with Euler, the entry can only be used with the single material hydrodynamic Euler solver using an EOSGAM (ideal gas) equation of state. Two of the four gasconstants (γ, R, c_v and/or c_p) have to be defined on the EOSGAM entry.
3. Either GASEMI-C or GASEMI-T must be specified.
4. Energy will only transfer out of the air bag if the temperature inside the air bag is higher than the environmental temperature.
5. A more detailed description of heat radiation can be found in Section 2.19 on page 2-133.

INCLUDE

Starts Reading a New Input File

Switches reading of the input data to another file. Once that file is read, processing reverts back to the original file immediately after the INCLUDE statement.

Format and Example:

1	2	3	4	5	6	7	8	9	10
INCLUDE filename									
INCLUDE BULK.DAT									

Field	Contents	Type	Default
filename	Name of the new input filename to be used. The name must be appropriate for the machine that is executing MSC.Dytran.	C	No new file.

Remarks:

1. The file must be present in the area where MSC.Dytran is executing.
2. A comma cannot be used to separate the fields.
3. BEGIN BULK and ENDDATA can be present in the INCLUDE file.

INFLATR

Airbag Inflator Model

Defines the inflator characteristics of a couple and/or GBAG subsurface.

1	2	3	4	5	6	7	8	9	10
INFLATR	INFLID	MASFLR-T	TEMP-T	TEMP-C	γ	c_v	R	c_p	
INFLATR	5	100		907.0		283.0			

Field	Contents	Type	Default
INFLID	Unique number of an INFLATR entry.	I > 0	Required
MASFLR-T	Table number of a TABLED1 or TABLEEX entry specifying the massflow-rate as a function of time.	I > 0	Required
TEMP-T	Table number of a TABLED1 or TABLEEX entry specifying the temperature of the inflowing gas as a function of time.	I > 0	See Remark 3.
TEMP-C	Constant value of the temperature of the inflowing gas constant.	R > 0	See Remark 3.
γ	Ratio of specific heat constants.	R > 0	See Remark 4.
c_v	Specific heat at constant volume.	R > 0	See Remark 4.
R	Gas constant.	R > 0	See Remark 4.
c_p	Specific heat at constant pressure.	R > 0	See Remark 4.

Remarks:

1. The INFLATR entry can be referenced from a COUINFL and/or GBAGINFL entry.
2. When used in an Euler coupled analysis, the entry can only be used with the single material hydrodynamic Euler solver using an EOSGAM (ideal gas) equation of state.
3. Either TEMP-C or TEMP-T must be specified.
4. Specify only two of the four gas constants. They are related as:

$$\gamma = \frac{c_p}{c_v} \qquad R = c_p - c_v$$

JOIN

Hinge-Type Join of 6 DOF Grid Points with 3 DOF Grid Points

Defines a hinge-type join of Lagrangian elements with 6 DOF grid points (e.g., CQUAD4, CBAR, etc.) to Lagrangian elements with 3 DOF grid points (CHEXA).

Format and Example:

1	2	3	4	5	6	7	8	9	10
JOIN	ID	SID	TOL						
JOIN	1	100	1.E–6						

Field	Contents	Type	Default
ID	Unique JOIN number.	I > 0	Required
SID	Number of a SET1 entry containing the list of grid points to be joined.	I > 0	Required
TOL	Tolerance for joining the grid points. Grid points that have mutual distance within this tolerance are joined.	R ≥ 0	5.10^{-4}

Remarks:

1. Grid points with the same number of degrees of freedom (DOF) can be equivalenced in the pre-processing phase.
2. The JOIN gives rise to a hinge connection. A stiff connection can be achieved by using KJOIN.

KJOIN

Kinematic Join of 6 DOF Grid Points with 3 DOF Grid Points

Defines the joining of Lagrangian elements with 6 DOF grid points (e.g., CQUAD4, CBAR, etc.) to Lagrangian elements with 3 DOF grid points (CHEXA).

Format and Example:

1	2	3	4	5	6	7	8	9	10
KJOIN	ID	SID	TOL	INTERFERE	STIFFNESS				
KJOIN	ID	150	1.E–5	STRONG	0.5				

Field	Contents	Type	Default
ID	Unique KJOIN number.	I > 0	Required
SID	Number of a SET1 entry containing the list of grid points to be joined.	I > 0	Required
TOL	Tolerance for joining the grid points. Grid points with mutual distance that is within the tolerance are joined.	R > 0.0	5.E–4
INTERFERE	Defines whether the rotation present at a 6 DOF grid point interferes with the rotation from the kinematic constraint (STRONG or NONE).	C	STRONG
STIFFNESS	Defines the relative stiffness of the kinematic join.	R	0.0

Remarks:

1. To change the stiffness of the join, the STIFFNESS field can be defined.

2. Stiffness is increased by setting INTERFERE to none.

3. The kinematic join acts as a locally inserted stiff element.

4. The STIFFNESS field defines a relative stiffness where the value should be in the interval $(-1/2, 1/2)$. Values less than zero reduce the stiffness, and values greater than zero increase the stiffness.

5. Geometric aspects are automatically taken into account.

6. In cases where the set of grid points for the KJOIN is to large to fit in one SET1 entry, you can define multiple SET1 entries, all with the same set number. The SET1 entries that have the same set number are automatically merged into one set.

7. You can define a hinge connection by using the JOIN entry.

MAT1

Material Property Definition, Form 1

Defines the material properties for linear, isotropic materials.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MAT1	MID	E	G	NU	RHO				
MAT1	17	3.+7		0.33	4.28				

Field	Contents	Type	Default
MID	Unique material number.	I > 0	Required
E	Young's modulus E.	R ≥ 0.	See Remark 2.
G	Shear modulus G.	R ≥ 0	See Remark 2.
NU	Poisson's ratio ν.	0. < R ≤ 0.5	See Remark 4.
RHO	Mass density ρ.	R > 0	Required

Remarks:

1. The material number must be unique for all MAT1 and MAT8 entries.

2. The following rules apply when E, G, or ν are blank:

   a. E and G may not both be blank.

   b. If ν and E, or ν and G, are both blank, then both are set to 0.0.

   c. If only one of E, G, or ν is blank, it will be computed from the equation:

$$E = 2(1 + \nu)G$$

3. Implausible data on one or more MAT1 entries results in a User Warning Message. Implausible data is defined as any of the following $E < 0.0$ or $G < 0.0$ and, $\nu > 0.5$ or $\nu < 0.0$.

4. It is strongly recommended that only two of the values be specified on the MAT1 entry.

MAT8

Orthotropic Elastic Material Properties

Defines the properties for an orthotropic material for shell elements.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MAT8	MID	E1	E2	N12	G12	G1,Z	G2,Z	RHO	
MAT8	171	30.+6	1.+6	0.3	2.+6	3.+6	1.5+6	0.056	

Field	Contents	Type	Default
MID	Unique material number.	I > 0	Required
E1	Modulus of elasticity in longitudinal direction (also defined as fiber direction or one-direction).	R > 0.0	Required
E2	Modulus of elasticity in lateral direction (also defined as matrix direction or two-direction).	R > 0.0	Required
N12	Poisson's ratio ($\varepsilon_2/\varepsilon_1$ for uniaxial loading in one-direction). Note that $\nu_{21} = \varepsilon_1/\varepsilon_2$ for uniaxial loading in two-direction is related to ν_{12}, E_1, E_2 by the relation $\nu_{12} E_2 = \nu_{21} E_1$.	R > 0.0	Required
G12	In-plane shear modulus.	R > 0.0	Required
G1, Z	Transverse shear modulus for shear in 1-Z plane (default implies G1,Z = G12).	R > 0.0	Blank
G2, Z	Transverse shear modulus for shear in 2-Z plane (default implies G2,Z = G12).	R > 0.0	Blank
RHO	Mass density.	R > 0.0	Required

Remarks:

1. An approximate value for G1,Z and G2,Z is the in-plane shear modulus G12. If test data is not available to accurately determine G1,Z and G2,Z if the material and transverse shear calculations are deemed essential, the value of G12 may be supplied for G1,Z and G2,Z. The MSC.Nastran defaults for G_1,Z and G_2,Z are infinite if left blank. MSC.Dytran assumes the transverse shear moduli to be equal to G12.

2. Excess data as defined in the MSC.Nastran MAT8 continuation lines is ignored. Equivalent entries can be defined in the MAT8A Bulk Data entry.

3. This material model can only be referenced from a PCOMP entry.

MAT8A

Orthotropic Failure Material Properties

Defines the failure properties for an orthotropic material for shell elements.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MAT8A	MID	FT	NV	S	ALPHA	TRSFAIL	F12		+CONT1
MAT8A	7	COMBINAT		100.					+CONT1

+CONT1	XT	XC	YT	YC	PFD	VALUE	PFDST		+CONT2
+CONT1	200	150	100	110.	STEPS	200			+CONT2

+CONT2	FBTEN	FBCOM	MXTEN	MXCOM	MXSHR				+CONT3
+CONT2	CHANG	STRSS	MODTSAI	MODTSAI	STRSS				+CONT3

+CONT3									+CONT4
+CONT3									+CONT4

+CONT4	PRDFT	PRDFC	PRDMT	PRDMC	PRDSH				
+CONT4					0011				

Field	Contents		Type	Default
MID	Unique material number.		I	See Remark 1.
FT	Failure theory to be used to test whether the element layer fails.		C	Blank
	Blank	No failure.		
	HILL	Tsai-Hill theory.		
	TSAI	Tsai-Wu theory.		
	MODTSAI	Modified Tsai-Wu theory.		
	STRSS	Maximum stress.		
	CHANG	Chang-Chang theory.		
	USER	User-defined model.		
	COMBINAT	Combination.	C	See Remark 2.
	HASHIN	Hashin theory.		

(Continued)

Field	Contents		Type	Default
NV	Number of additional history variables for a user model. See Remark 7.		0 < I < 10	0
S	Failure stress for in-plane shear.		R > 0.0	See Remark 3.
ALPHA	Nonlinear shear coefficient. See Remark 4.		R ≥ 0.0	0.
TRSFAIL	Transverse shear failure.		C	SUBL
	ELEM	Failure if element fails.		
	SUBL	Failure if sublayer fails.		
F12	Interaction term in Tsai-Wu theory.		R	0.
XT, XC	Tensile compressive failure stress in the large structural direction.		R > 0.0	See Remark 3.
YT, YC	Tensile compressive failure stress in the lateral direction.		R > 0.0	See Remark 3.
PFD	Post-failure degradation model. See Remark 6.		C	STEPS
	STEPS	Degrade stresses by time steps.		
	TIME	Degrade stresses by time.		
	VELOC	Degrade stresses by velocity.		
VALUE	Depending on PFD, VALUE gives the number of time steps, time interval, or propagation velocity.		I or R	100
PFDST	Post-failure degradation start.			
	INDV	Stresses are degraded per distinct failure mode.	C	INDV
	ALL	Stresses are degraded if all elastic constants are zero.		
	See Remark 8.			
FBTEN, FBCON, MXTEN, MXCOM, MXSHR	Failure modes in fiber, matrix direction, and theory failure. Enter values if FT = COMBINAT.		C	See Remark 5.
PRDFT	Property degradation due to fiber-tension failure.		I	1111
PRDFC	Property degradation due to fiber-compression failure.		I	1110
PRDMT	Property degradation due to matrix-tension failure.		I	0111

(Continued)

Field	Contents	Type	Default
PRDMC	Property degradation due to matrix-compression failure.	I	0110
PRDSH	Property degradation due to in-plane shear failure.	I	0001

Remarks:

1. The material number must refer to a MAT8 material definition.
2. If a failure theory is selected other than USER or COMBINAT, the theory defines the following failure modes:

CHANG	Fiber tension, matrix tension/compression.
HILL	All modes.
TSAI	All modes.
MODTSAI	Matrix tension/compression.
STRSS	All modes.
HASHIN	Fiber tension/compression. Matrix tension/compression.

   For an element to fail completely, both fiber and matrix in all sublayers must fail.
3. This material model can only be referenced from a PCOMP entry.
4. Failure stresses are required if a failure theory is selected.
5. ALPHA is used for all failure theories to define a nonlinear stress-strain relation.
6. The individual failure modes are defined according to the corresponding mode in the theory as listed under FT. To be relevant, the theory must define the failure mode (see Remark 2). You must enter data if FT is set to COMBINAT.
7. The property degradation rules due to the various failure modes are listed in the following table.

Material Constant	Failure Mode				
	Fiber Tens	Fiber Comp	Matrix Tens	Matrix Comp	Shear
E1	X	X			
E2	X	X	X	X	
V12	X	X	X	X	
G12	X		X		X

(Continued)

The Poisson's ratio Nu21 is treated the same as Nu12.

To override the default model, an integer value is defined as a packed word in the following order:

$$(E_1)\ (E_2)\ (\text{Nu }12)\ (G_{12})$$

1 denotes property degradation.

0 denotes no degradation.

The last five fields of the MAT8A Bulk Data entry are input for the user to specify the degradation behavior for each mode of failure.

8. User variables for sublayers are used on restart and archive output. Refer to them as USRnLxx where n is the user ID and xx is the sublayer number (see Section [3.7.2]). User variables are available in the subroutine EXCOMP.

9. The PFD entry indicates how the stresses are degraded to zero. The PFDST indicates when the stresses start to degrade.

   Using ALL means that degradation starts when all material constants (E_1, E_2, Nu12, G_{12}) are degraded to zero as specified by the FT entry and the property degradation rules. Note that property degradation means that the stress increments are zero but that the stresses degrade according to PFD.

   INDV means that stress degradation starts for the fiber stress if $E_1 = 0$, for matrix stress if $E_2 = 0$, and for shear stress if $G_{12} = 0$.

10. Any failure theory introduces five additional sublayer variables. The PFDST entry introduces three additional variables. The number of user variables is defined by NV.

MATRIG

Rigid-Body Properties

Defines the properties of a rigid body.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MATRIG	MID	RHO	E	NU	MASS	XC	YC	ZC	+CONT1
MATRIG	7	7850.	210.E9	0.3	750	0.0	7.0	–3.0	+CONT1

+CONT1	IXX	IXY	IXZ	IYY	IYZ	IZZ	CID		+CONT2
+CONT1	17.0	13.2	14.3	20.9	15.7	10.0	12		+CONT2

+CONT2	VX	VY	VZ	WX	WY	WZ			
+CONT2			13.3						

Field	Contents	Type	Default
MID	Unique material number.	I > 0	Required
RHO	Density.	R > 0	1.0
E	Young’s modulus.	R > 0	1.0
NU	Poisson’s ratio.	0.0 ≤ R < 0.5	0.2
MASS	Mass of the rigid body.	R > 0.0	See Remark 2.
XC,YC,ZC	x, y, and z coordinates of the center of gravity.	R	See Remark 7.
IXX, IXY, IXZ, IYY IYZ, IZZ	Inertia tensor of the rigid body about the center of gravity.	R	See Remark 7.
CID	Number of a coordinate system in which the inertia tensor is defined.	I > 0	See Remark 8.
VX, VY, VZ	Initial translational velocity of the center of gravity in the basic coordinate system.	R	0.0
WX, WY, WZ	Initial rotational velocities of the rigid body about the center of gravity in the basic coordinate system.	R	0.0

(Continued)

Remarks:

1. All coordinates are defined in the basic coordinate system.
2. If MASS is blank or zero, the mass will be calculated from the density and the geometry of the mesh defining the rigid body.
3. The continuation lines are not required.
4. The MATRIG definition is used instead of a DYMATn definition and is referenced by properties PSOLIDn, PSHELLn, PBAR, and PBEAMn. Different properties can refer to the same MATRIG entry forming one rigid body. The MATRMERG or MATRMRG1 option (see PARAM,MATRM(E)RG(1)) can be used for merging different MATRIG and RBE2-FULLRIG definitions into one single rigid body.
5. By using PARAM,RBE2INFO,GRIDON, the grid points of the MATRIG will be listed in the output file.
6. If the fields VX, VY, VZ, WX, WY, and WZ are blank, then the initial conditions of the rigid body are calculated from the initial velocities on the TIC and TIC1 entries referring to grid points attached to the rigid body. The net initial conditions are the average of those for all the grid points attached to the rigid body.

   If the initial conditions are set using the VX, VY, VZ, WX, WY, and WZ fields, the TIC and TIC1 entries referring to grid points attached to the rigid body are ignored.
7. If the inertia tensor or the coordinates of the center of gravity are undefined, then they will be computed from the density or mass and the geometry of the mesh defining the rigid body.
8. The inertia tensor can only be defined in a local rectangular coordinate system. If the entry for a local coordinate system is left blank, then the inertia tensor is defined in the global coordinate system.
9. The behavior of rigid bodies in discussed in Section 2.3.5 on page 2-12.

MESH

Mesh Generator

Defines a mesh.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MESH	MID	TYPE							+CONT1
MESH	1	BOX							+CONT1

+CONT1	X0	Y0	Z0	DX	DY	DZ			+CONT2
+CONT1	0.	0.	0.	5.	5.	100.			+CONT2

+CONT2	NX	NY	NZ		NSTGP	NSTEL	PROP	PID	
+CONT2	10	10	400		1001	5001	EULER	1	

Field	Contents	Type	Default
MID	Unique MESH number.	I > 0	Required
TYPE	Type of mesh generation.	C	Required
	BOX Rectangular mesh aligned with the basic coordinate system will be created, filled with HEXA elements.		
X0,Y0,Z0	Coordinates of point of origin.	R	Required
DX,DY,DZ	Width of box in different directions.	R	Required
NX,NY,NZ	Number of elements in the different directions.	I > 0	Required
NSTGP	Starting grid-point number.	I > 0	See Remark 2.
NSTEL	Starting element number.	I > 0	See Remark 2.
PROP	Property type.	C	Required
PID	Property number.	I > 0	Required

Remarks:

1. Currently only type BOX is available.
2. When the starting grid point and/or element number is left blank the maximum used number + 1 is used as a default value.
3. The PID should refer to an existing property id, which can handle the property type given by PROP.
4. The PROP option depends on the type of mesh generation. For type BOX property types SOLID and EULER are available.

MOMENT

Concentrated Moment or Enforced Motion

This entry is used in conjunction with a TLOADn entry and defines the location where the moment or enforced motion acts as well as the direction and scale factor.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MOMENT	LID	G		SCALE	N1	N2	N3		
MOMENT	2	5		2.9		1.0			

Field	Contents	Type	Default
LID	Number of a set of loads.	I > 0	Required
G	Grid-point number or rigid body where the load is applied.	See Remark 5.	Required
SCALE	Scale factor for the moment.	R	1.0
N1, N2, N3	Components of a vector giving the direction of the moment. At least one must be nonzero.	R	See Remark 4.

Remarks:

1. At time *t*, the moment *M(t)* is given by

$$M(t) = \text{SCALE} * N * T(t)$$

   where SCALE is the scale factor; *N* is the vector given by N1, N2, and N3; and $T(t)$ is the value at time *t* interpolated from the table referenced on the TLOADn entry.

2. Moments can also be defined on the DAREA entry.

3. LID must be referenced by a TLOADn entry.

4. If a component field N1, N2, and/or N3 is left blank:

   Moment prescription – The component of the moment is equal to zero.

   Velocity prescription – The component of the angular velocity is not restrained.

5. If G references a MATRIG, an RBE2-FULLRIG, or a RIGID surface, the load is applied to the center of the rigid body. If G references a MATRIG, G must be MR<id>, where id is the MATRIG number. If G references an RBE2-FULLRIG, G must be FR<id>, where id is the RBE2 number. If G references a RIGID surface, G is the RIGID surface number.

6. If the TYPE field on the TLOADn entry is 0, this defines a moment applied to a grid point. If the TYPE field is 2, it defines an enforced motion on the grid point. If the TYPE field is 12, it defines an enforced motion applied to the center of the rigid body, and if the TYPE field is 13, it defines a moment applied to the center of a rigid body.

MOMENT1

Follower Moment, Form 1

This entry is used in conjunction with a TLOADn entry and defines a follower moment with direction that is determined by two grid points. MOMENT1 can be applied to any type of grid point.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MOMENT1	LID	G	SCALE	G1	G2				
MOMENT1	2	5	2.9	16	13				

Field	Contents	Type	Default
LID	Number of a set of loads.	I > 0	Required
G	Grid-point number where the moment is applied.	I > 0	Required
SCALE	Scale factor for the moment.	R	1.0
G1, G2	Grid-point numbers. The direction of the moment is a vector from G1 to G2. G1 must not be the same as G2.	I > 0	Required

Remarks:

1. At time *t*, the moment $\underline{M}(t)$ is given by

$$\underline{M}(t) = \text{SCALE} * \underline{N} * T(t)$$

   where *SCALE* is the scale factor, $\underline{N}$ is the vector from G1 to G2, and $T(t)$ is the value at time *t* interpolated from the table referenced by the TLOADn entry.

2. LID must be referenced by a TLOADn entry.

3. The MOMENT1 entry defines a follower moment in that the direction of the moment changes as the grid points G1 and G2 move during the analysis.

MOMENT2

Follower Moment, Form 2

This entry is used in conjunction with a TLOADn entry and defines a follower moment with direction that is determined by four grid points. MOMENT2 can be applied to any type of grid point.

Format and Example:

1	2	3	4	5	6	7	8	9	10
MOMENT2	LID	G	SCALE	G1	G2	G3	G4		
MOMENT2	2	5	2.9	16	13	17	18		

Field	Contents	Type	Default
LID	Number of a set of loads.	I > 0	Required
G	Grid-point number where the moment is applied.	I > 0	Required
SCALE	Scale factor for the moment.	R	1.0
G1-G4	Grid-point numbers. The moment direction is determined by a vector product of the vectors G1 to G2 and G3 to G4. (G1 must not be the same as G2, and G3 must not be the same as G4.)	I > 0	Required

Remarks:

1. At time *t*, the moment $\underline{M}(t)$ is given by

$$\underline{M}(t) = \mathrm{SCALE} * \underline{N} * T(t)$$

where *SCALE* is the scale factor, $\underline{N}$ is the vector product of the vectors from G1 to G2 and G3 to G4 respectively, and $T(t)$ is the value at time t interpolated from the table referenced by the TLOADn entry.

2. LID must be referenced by a TLOADn entry.

3. The MOMENT2 entry defines a follower moment in that the direction of the moment changes as the grid points G1, G2, G3, and G4 move during the analysis.

NASINIT

MSC.Nastran Initialization

Definition of the logistics of an MSC.Dytran prestress run.

Format and Example:

1	2	3	4	5	6	7	8	9	10
NASINIT	STEPS	DAMP	TNOACT	FACTOR					
NASINIT	1000	YES	1.E–02	0.01					

Field	Contents	Type	Default
STEPS	Number of steps used to set the grid-point displacement.	I > 0	1
DAMP	Request for additional relaxation phase after displacement phase (Yes/No).	C	No
TNOACT	End time of relaxation phase.	R > 0	1.E20
FACTOR	Viscous-damping factor.	R > 0	0.001

Remarks:

1. The time step is constant during the displacement phase and is defined by PARAM,INISTEP.
2. Damping is optional and is not always necessary.
3. The deformed geometry grid-point data is written out after the displacement phase, if no damping is requested, or after the relaxation phase, when the DAMP field is set to YES. The same applies to the solution file. (See also the SOLUOUT and BULKOUT FMS statements.)
4. The displacements from an MSC.Nastran solution are imposed by an enforced velocity field calculated from the displacements and control parameters.
5. All boundary conditions and loads defined are deactivated during the displacement phase and are activated after the displacement phase ends.
6. Note that although the deformed geometry after the displacement phase is exactly the same as the MSC.Nastran geometry, the actual stress state may differ due to differences in MSC.Dytran and MSC.Nastran element formulations.
7. Lagrangian CHEXA, CQUAD4, and CTRIA3 elements can be initialized, but the shell membranes cannot.
8. For prestressing rotating structures, it is recommended that a centrifugal force field be used, rather than a rotational velocity field. In the actual transient dynamic analysis, the centrifugal force field can be replaced by a rotational velocity field with consistent boundary conditions.
9. Make the problem setup for the final transient analysis consistent with the prestress analysis.

Defines the values for parameters used during the solution.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PARAM	NAME	V1							
PARAM	REZTOL	0.1							

Field	Contents	Type	Default
NAME	Parameter name.	C	Required
V1	Value associated with NAME.	I, R, C	See Section 4.7 on page 4-437.

Remarks:

1. A list of the parameters that can be set, along with the parameter names and values, is given in Section 4.7 on page 4-437.

2. PARAM entries do not necessarily have to be located in the Bulk Data Section.

3. PARAM values can be redefined during restarts.

PBAR

Simple Beam Property

Defines the properties of a simple beam (bar) that is used to create bar elements via the CBAR entry.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PBAR	PID	MID	A	I1	I2	J			
PBAR	39	6	2.9		5.97				

Field	Contents	Type	Default
PID	Unique property number.	I > 0	Required
MID	Material number.	I > 0	Required
A	Area of bar cross section.	R > 0	Required
I1, I2	Area moments of inertia.	R ≥ 0	Required
J	Torsional constant.	R ≥ 0	0.0

Remarks:

1. I_1 is the moment of inertia about the element z-axis, I_{zz}.
   I_2 is the moment of inertia about the element y-axis, I_{yy}.
2. This element is solved as a Belytschko-Schwer beam.

Defines the properties of the CBAR and CBEAM element.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PBEAM	PID	MID	A(A)	I1(A)	I2(A)		J(A)		+CONT1
PBEAM	7	14	3.6	24.9	24.9		22.6		+CONT1

+CONT1									+CONT2
+CONT1									+CONT2

+CONT2		X/XB	A(B)	I1(B)	I2(B)		J(B)		
+CONT2		1.0	3.6	24.9	24.9		22.6		

Field	**Contents**	**Type**	**Default**
PID	Unique property number.	I > 0	Required
MID	Material number.	I > 0	PID
A(A)	Area of the beam cross section at end A of the beam.	R > 0.	Required
I1(A)	Area moment of inertia about the beam-element’s z-axis at end A of the beam.	R > 0.	Required
I2(A)	Area moment of inertia about the beam-element’s y-axis at end A of the beam.	R > 0.	Required
J(A)	Torsion constant at end A of the beam.	R	0.0
X/XB	For MSC.NASTRAN, this is the distance along the beam from end A divided by the length of the beam. The properties are defined at several positions along the beam’s length. For MSC.Dytran, all the intermediate positions are ignored. The only relevant data occurs when X/XB is 1.0. corresponding to end B of the beam.	R	Required
A(B)	Area of the cross section at end B of the beam.	R > 0.	Required
I1(B)	Area moment of inertia about the beam-element’s z-axis at end B of the beam.	R > 0.	Required
I2(B)	Area moment of inertia about the beam-element’s y-axis at end B of the beam.	R > 0.	Required
J(B)	Torsion constant at end B of the beam.	R	0.0

(Continued)

Remarks:

1. This entry is an alternative to the PBAR entry and defines exactly the same element and properties. It is more complicated to use than PBAR and has no advantages. PBEAM is retained for compatibility with MSC.NASTRAN and those modeling packages that write PBEAM entries. Use the PBAR entry if you can.

2. A Belytschko-Schwer beam is used with a shear factor of 0.83333. The plastic moduli are assumed to be those for a rectangular section

$$Zp_y = \sqrt{0.75 * A * I2}$$

$$Zp_z = \sqrt{0.75 * A * I1}$$

   To specify values of Z_p for other sections, use the PBEAM1 entry.

3. For more complex beam properties, use the PBEAM1 entry.

4. Note the following:

$$I_1 = I_{zz}$$

$$I_2 = I_{yy}$$

$$J = I_{xx}$$

PBEAM1

Beam Properties (Hughes-Liu)

Defines complex beam properties that cannot be defined using the PBAR or PBEAM entries. The following entries are for Hughes-Liu beam elements only.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PBEAM1	PID	MID	FORM	QUAD	NUMB	SHFACT	SECT		+CONT1
PBEAM1	1	7	HUGHES	GAUSS	3	0.9	RECT		+CONT1

+CONT1	V1	V2	V3	V4					
+CONT1	30.1	30.1	10.0	10.0					

Field	Contents	Type	Default
PID	Unique property number.	I > 0	Required
MID	Material number.	I > 0	PID
FORM	Element formulation.	C	Required
	HUGHES Hughes-Liu.		
QUAD	Type of quadrature.	C	GAUSS
	GAUSS Gauss quadrature.		
	LOBATTO Lobatto quadrature.		
NUMB	The number of integration points for Hughes-Liu beams. For Gauss integration, the following can be specified:	I > 0	3
	1 1 point (rod element).		
	2 2×2 points (4-point circle, if tubular).		
	3 3×3 points (9-point circle, if tubular).		
	4 4×4 points (16-point circle, if tubular).		
	At present only 3×3 points are available with the Lobatto quadrature.		
SHFACT	Shear factor for the section.	R	0.83333

(Continued)

Field	Contents	Type	Default
SECT	Type of section.	C	RECT
	RECT Rectangular cross section.		
	TUBE Tubular cross section.		

Field	Contents	Type	Default
V1–V4	Geometric properties of the beam. The data in these fields depends on the beam formulation and the type of cross section. For Hughes formulations with rectangular cross sections. V1 The thickness in the element y direction at grid point 1. V2 The thickness in the element y direction at grid point 2. V3 The thickness in the element z direction at grid point 1. V4 The thickness in the element z direction at grid point 2. For Hughes formulations with tubular cross sections: V1 The outer diameter at grid point 1. V2 The outer diameter at grid point 2. V3 The inner diameter at grid point 1. V4 The inner diameter at grid point 2.	R	Required

Remark:

1. Only the entries that are relevant for the Hughes-Liu beam definition are listed here. PBEAM1 entries that apply to Belytschko-Schwer beams are provided in the PBEAM1 description that follows.

Defines complex beam properties that cannot be defined using the PBAR or PBEAM entries. These entries are to be used only for Belytschko-Schwer elements.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PBEAM1	PID	MID	FORM			SHFACT	SECT		+CONT1
PBEAM1	1	7	BELY			0.9	RECT		+CONT1

+CONT1	A	I1	I2	J	ZPZ	ZPY			+CONT2
+CONT1									+CONT2

+CONT2	CS1	CS2	CS3	CS4	CS5				
+CONT2									

Field	Contents	Type	Default
PID	Unique property number.	I > 0	Required
MID	Material number.	I > 0	PID
FORM	Element formulation.	C	Required
	BELY Belytschko-Schwer.		
SHFACT	Shear factor for the section.	R	0.83333
SECT	Type of section. See Remark 4.	C	RECT
A	Area of the section.	R	Blank
I1	The moment of inertia about the element z-axis.	R	Blank
I2	The moment of inertia about the element y-axis.	R > 0.	Blank
J	The torsional stiffness of the section.	R ≥ 0.	Blank
ZPZ	Plastic modulus *Zp* about the element z-axis.	R > 0.	Blank
ZPY	Plastic modulus *Zp* about the element y-axis.	R > 0.	Blank
CSi	Geometrical definition of the cross section. The data in these fields depends on the type of the section.	R ≥ 0.	See Remark 4.

(Continued)

Remarks:

1. Only the entries that are relevant for Belytschko-Schwer beam definition are listed. PBEAM1 entries that apply to Hughes-Liu beams appear earlier in this PBEAM1 discussion.

2. Note the following:

$$I1 = I_{zz}$$

$$I2 = I_{yy}$$

$$J = I_{xx}$$

3. The cross-sectional properties are calculated as follows:

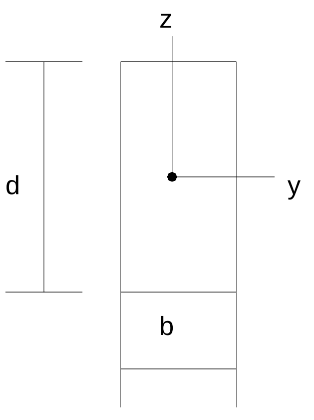
   a. If the geometry is defined in the fields CSi, the values of A, I1, I2, J, ZPZ and ZPY are calculated automatically.

   b. If a value is defined in the fields A, I1, I2, J, ZPZ, ZPY, these values override the values as calculated in step a.

   c. All values of CSi for a particular cross section (see Remark 4) must be entered for the geometry to be defined. If not all values of CSi are supplied, then values for A, I1, I2 and J are required, and ZPZ, ZPY will have a default value of 1.E20.

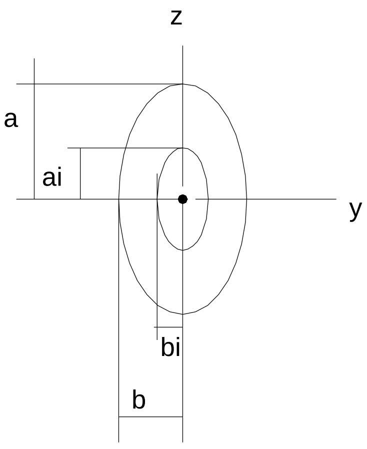
4. The geometrical definitions for the various cross sections are defined in the element coordinate system as follows:

SECT = RECT



CS1 = b
CS2 = d

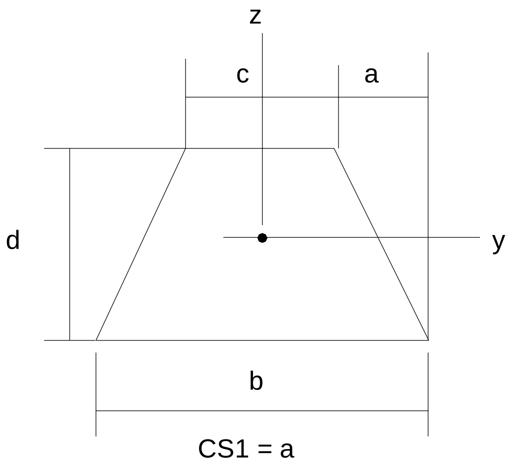
SECT = TUBE



CS1 = b
CS2 = a
CS3 = bi
CS4 = ai

(Continued)

SECT = TRAPZ



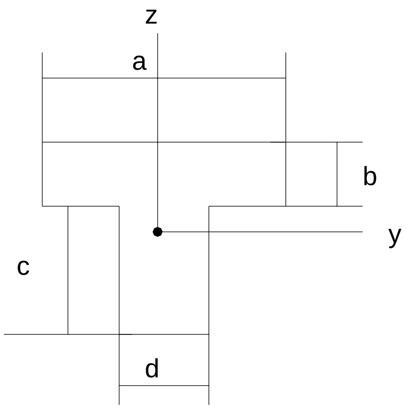
CS1 = a

CS2 = b

CS3 = c

CD4 = d

SECT = TSECT



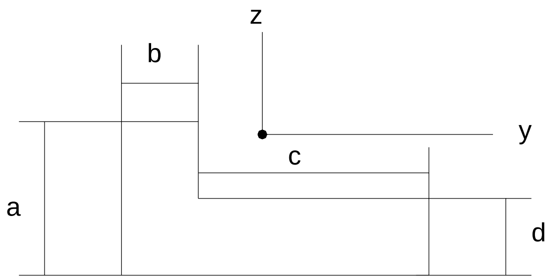
CS1 = a

CS2 = b

CS3 = d

CD4 = c

SECT = LSECT



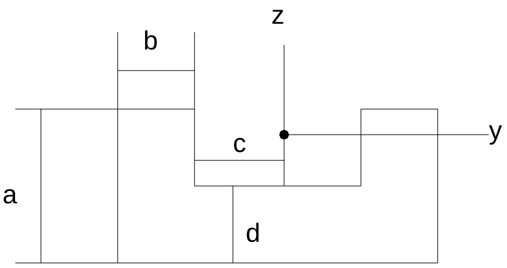
CS1 = a

CS2 = b

CS3 = c

CD4 = d

SECT = USECT



CS1 = a

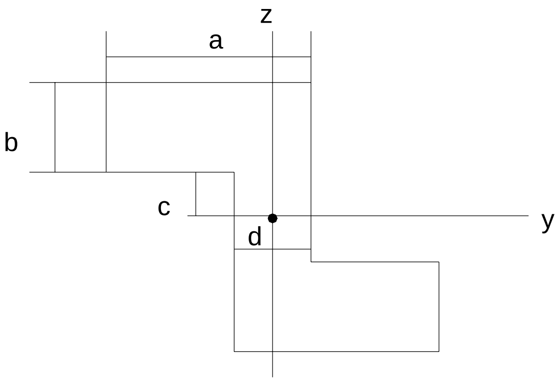
CS2 = b

CS3 = c

CD4 = d

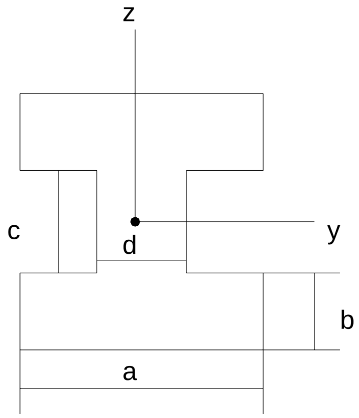
(Continued)

SECT = ZSECT



CS1 = a
CS2 = b
CS3 = c
CD4 = d

SECT = ISECT



CS1 = a
CS2 = b
CS3 = c
CD4 = d

PBELT

Belt Property

Defines the properties of a belt element referenced by a CROD entry.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PBELT	PID	LOAD	UNLOAD	DENSITY	DAMP1	DAMP2	SLACK	PRESTRESS	
PBELT	9	12	12	2.E–5	0.1	0.1			

Field	Contents	Type	Default
PID	Unique belt-property number.	I > 0	Required
LOAD	Number of a TABLED1 defining the force as a function of strain during loading. The strain at time *n* is specified as engineering strain: strain(n) = (length(n) – length (0)) / (length (0))	I > 0	Required
UNLOAD	Number of a TABLED1 defining the force as a function of strain during unloading. The strain at time *n* is specified as engineering strain: strain(n) = (length(n) – length (0)) / (length (0))	I > 0	Required
DENSITY	Density of the belt elements as mass per unit length.	R > 0.0	Required
DAMP1	A damping force is added to the internal force of the belt elements to damp out high frequency oscillations. The damping force is equal to: $F_{damp} = \text{DAMP1} * (mass) * (dvel)/(dt)$ where F_{damp} = damping force DAMP1 = damping coefficient *mass* = mass of belt element *dvel* = velocity of elongation *dt* = time step	R > 0.0	0.1
DAMP2	The damping force is limited to: $\text{DAMP2} * F_{belt}$ where F_{belt} is the internal force in the belt element.	R > 0.0	0.1

(Continued)

Field	Contents	Type	Default
SLACK	Number of a TABLED1 defining the slack as a function of time. The slack must be specified as engineering strain and will be subtracted from the element strain at time *n* as: strain(n) = strain(n) – SLACK(n) The force in the element will be zero until the element strain exceeds the slack.	I > 0	Blank
PRESTRESS	Number of a TABLED1 defining a prestress strain as a function of time. The prestress strain must be specified as engineering strain and will be added to the element strain at time *n* as: strain(n) = strain(n) + PRESTRESS(n)	I > 0	Blank

Remarks:

1. The loading and unloading curves must start at (0.0, 0.0).
2. During loading, the loading curve is applied to determine the force in the belt element. At unloading, the unloading curve is shifted along the strain axis until it intersects the loading curve at the point from which unloading commences. The unloading table will be applied for unloading and reloading, until the strain again exceeds the intersection point. Upon further loading, the loading table will be applied. For subsequent unloading, the sequence will be repeated.
3. Belt elements are tension-only elements.
4. Instantaneous slack of an element can also be initialized per element using the TICEL entry with the keyword SLACK and a corresponding VALUE.
5. Belt elements are discussed Section 2.13 on page 2-113.

PCOMP

Layered Composite Element Property

Defines the properties of a multi-ply laminate composite material.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PCOMP	PID							LAM	+CONT1
PCOMP	181								+CONT1

+CONT1	MID1	T1	THETA1		MID2	T2	THETA2		+CONT2
+CONT1	171	0.056	0.				45.		+CONT2

+CONT2	MID3	T3	THETA3		MID4	T4	THETA4		
+CONT2			–45.				90.		

Field	Contents		Type	Default
PID	Unique property number.		I ≥ 0	Required
LAM	Symmetric lamination option.		C	Blank
	Blank	Enter all plies.		
	SYM	Describe only plies on one side of the element center line. (See Remark 3.)		
MIDi	Material number of the various plies. Identify the plies by sequentially numbering them starting from 1 at the bottom layer. The MIDs must refer to a MAT1, MAT8, DMATEP or DYMATzy entry.		I ≥ 0	See Remark 1.
Ti	Thickness of ply i.		R ≥ 0	See Remark 1.
THETAi	Orientation angle of the longitudinal direction of each ply with the material axis of the element. (If the material angle on the element connection entry is 0.0, the material axis and side 1-2 of the element coincide.) The plies are numbered sequentially starting with 1 at the bottom layer. (The bottom layer is defined as the surface with the largest negative z-value in the element coordinate system.)		R	0.0

(Continued)

Remarks:

1. The default under MID2, MID3, . . ., is the last-defined material, in this case MID1; for T2, T3, . . ., all these thicknesses are equal to T1.

2. At least one of the three values (MIDi, Ti, THETAi) must be present for a ply to exist. The minimum number of plies is one.

3. The symmetric laminate option is currently not available.

4. The thickness of the element is the sum of the ply thicknesses regardless of the values on the CTRIA3 or CQUAD4 Bulk Data entries.

PCOMPA

Additional Data for Layered Composite Element Property

Defines additional properties of a multi-ply laminate composite material.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PCOMPA	PID	FORM	SHFACT	REF	STRDEF	DT1D	STRNOUT		
PCOMPA	10								

Field	Contents		Type	Default
PID	Property number referring to a PCOMP property number.		I ≥ 0	Required
FORM	Element formulation.		C	See Remark 1.
SHFACT	Shear correction factor.		R	0.83333
REF	Reference surface.		C	MID
	TOP	Reference surface is the top surface.		
	MID	Reference surface is the central surface.		
	BOT	Reference surface is the bottom surface.		
STRDEF	Definition in stress-strain output.		C	FIBER
	FIBER	Stresses defined in the fiber and matrix directions.		
	ELEMENT	Stresses defined in the element coordinate system.		
DT1D	Time step skip for one-dimensional failure modes.		C	NO
	YES	Skip one-dimensional failure modes.		
	NO	Normal time-step calculation.		
	See Remark 2.			
STRNOUT	Strain output option.		C	YES
	YES	Total strain is calculated.		
	NO	No strain is stored in memory.		
	See Remark 3.			

(Continued)

Remarks:

1. For CQUAD4 elements, the default formulation is Key-Hoff. For CTRIA3 elements, the default formulation is C0-TRIA. See also Section 2.15 on page 2-118 on application sensitive defaults.
2. If the failure mode is such that fiber and shear strength or matrix and shear strength are lost in all layers, the element is not included in the time-step calculation. If the element fails completely, the element is omitted from the time-step calculations, irrespective of the value entered in this field.
3. If the STRNOUT field is NO, the strain cannot be output.

PDAMP

Linear Damper Properties

Defines the properties of a linear viscous damper.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PDAMP	PID	C							
PDAMP	7	0.01							

Field	Contents	Type	Default
PID	Unique property number.	I > 0	Required
C	Damping constant (force/velocity or moment/ velocity).	R	0.0

Remarks:

1. This entry defines a linear viscous damper.
2. For a discussion of the various types of damper elements, see Section 2.3.9 on page 2-19.

PELAS

Elastic Spring Property

Defines the stiffness coefficient, the damping coefficient, and the stress coefficient of an elastic spring.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PELAS	PID	K			PID	K			
PELAS	7	4.29			27	2.17			

Field	Contents	Type	Default
PID	Property number.	I ≥ 0	Required
K	Spring stiffness.	R	0.

Remarks:

1. Be cautious when using negative spring-stiffness values because values are defined directly on some of the CELASn entry types.
2. One or two elastic spring properties may be defined on a single entry.
3. For a discussion of the various types of spring elements, see Section 2.3.8 on page 2-17.

PELAS1

Nonlinear Elastic Spring Property

Defines the properties of nonlinear, elastic springs.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PELAS1	PID	TABLE							
PELAS1	5	25							

Field	Contents	Type	Default
PID	Unique property number.	I > 0	Required
TABLE	Number of a TABLExx entry defining the variation of force/moment (y-value) with displacement/ rotation (x-value).	I > 0	Required

Remarks:

1. The values in the table are either force and displacement or moment and rotation, depending on whether the spring connects translational or rotational degrees of freedom.
2. The values in the table are interpolated to determine the force/moment for a particular displacement/rotation.
3. For a discussion of the various types of spring elements, see Section 2.3.8 on page 2-17.

PELASEX

User-Defined Spring Properties

Defines the properties for CELASn scalar spring elements used with user-written spring subroutines.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PELASEX	PID	V1	V2	V3	V4	V5	V6	V7	
PELASEX	27	39.6	100.E6						

Field	**Contents**	**Type**	**Default**
PID	Unique property number.	I ≥ 0	Required
V1–V7	User values.	R	0.0

Remarks:

1. The seven user values are passed to the user subroutine EXELAS.
2. MSC.Dytran does no checking on the user values.
3. For a discussion of the various types of spring elements, see Section 2.3.8 on page 2-17. For a discussion of user-written subroutines, see Section 3.15 on page 3-74.

PERMEAB

Air Bag Permeability

Defines the permeability of a couple and/or GBAG (sub)surface.

Permeability is the velocity of gasflow through a (sub)surface and is defined as a linear or tabular function of the pressure difference over the (sub)surface.

1	2	3	4	5	6	7	8	9	10
PERMEAB	PID	PERM-C	PERM-T	FLOW	PENV	RHOENV	SIEENV		

Field	**Contents**	**Type**	**Default**
PID	Unique number of a PERMEAB entry.	I > 0	Required
PERM-C	Permeability is a linear function of the pressure difference.	R > 0	See Remark 3.
	permeability = PERM – C∗abs (Pinside – PENV)		
	For Pinside > PENV: outflow		
	For Pinside < PENV: inflow		
PERM-T	Permeability is a tabular function of the pressure difference:	I > 0	See Remark 3.
	table contains: permeability versus |Pinside – PENV|		
	For Pinside > PENV: outflow		
	For Pinside < PENV: inflow		
FLOW	Defines the allowed directions of the flow.	C	BOTH
	BOTH In- and outflow are allowed.		
	IN Only inflow allowed.		
	OUT Only outflow allowed.		
PENV	Environmental pressure.	R > 0	Required
RHOENV	Environmental density.	R > 0	Required
SIEENV	Environmental specific internal energy.	R > 0	Required

(Continued)

Remarks:

1. The PERMEAB entry can be referenced from a COUPOR and/or GBAGPOR entry.
2. When used with Euler, the entry can only be used with the single material hydrodynamic Euler solver using an EOSGAM (ideal gas) equation of state.
3. Either PERM-C or PERM-T must be specified.
4. The values for the environment p_{env} (PENV), ρ_{env} (RHOENV), e_{env} (SIEENV) must be defined consistent with an ideal-gas equation of state:

   $$p_{env} = (\gamma_{env} - 1)\rho_{env}e_{env}$$

   The γ_{env} is calculated by MSC.Dytran, and is used when inflow occurs. Inflow occurs when $p_{env} > p_{inside}$.

Defines a permeable area of a couple and/or GBAG (sub)surface, connected to another GBAG.

The velocity of the gas flow through the (sub)surface is defined as a linear or tabular function of the pressure difference.

1	2	3	4	5	6	7	8	9	10
PERMGBG	FID	PERM-C	PERM-T	FLOW	GBID				

Field	Contents	Type	Default
FID	Unique number of a PERMGBG entry. It can be referenced from either a GBAGPOR to model the flow between GBAGs, or from a COUPOR to model the flow between an Eulerian air bag and a GBAG.	I > 0	Required
PERM-C	Permeability is a linear function of the pressure difference. permeability = PERM-C∗abs (Pinside – Pgbid) The gas flow is from the higher to the lower pressure.	R > 0	See Remark 3.
PERM-T	Permeability is a tabular function of the pressure difference. table contains: permeability versus |Pinside – Pgbid| The gas flow is from the higher to the lower pressure.	I > 0	See Remark 3.
FLOW	Defines the allowed directions of the flow. BOTH In- and outflow are allowed. IN Only inflow allowed into the GBAG or the coupling surface that references this entry. OUT Only outflow allowed into the GBAG or the coupling surface that references this entry.	C	BOTH
GBID	Number of a GBAG entry. This GBAG is the one that is connected to the GBAG or coupling surface that references this entry.	R > 0	Required

(Continued)

Remarks:

1. The PERMGBG entry can be referenced from a COUPOR and/or GBAGPOR entry.
2. When used with Euler, the entry can only be used with the single material hydrodynamic Euler solver, using an EOSGAM (ideal gas) equation of state.
3. Either PERM-C or PERM-T must be specified.

PEULER

Eulerian Element Properties

Defines the properties of Eulerian elements.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PEULER	PID	MID	TYPE						
PEULER	100	25							

Field	**Contents**	**Type**	**Default**
PID	Unique property number.	I > 0	Required
MID	Number of a DMATxxx entry defining the constitutive model.	I ≥ 0	Required
TYPE	The type of Eulerian material being used.	C	HYDRO
	HYDRO Hydrodynamic material with no shear strength + void.		
	STRENGTH Structural material with shear strength + void.		
	MMHYDRO Multimaterial hydrodynamic material with no shear strength + void.		
	MMSTREN Structural multimaterial with shear strength + void.		

Remarks:

1. Make the property number unique with respect to all other property numbers.
2. The elements that reference this property use the Eulerian formulation.
3. If TYPE is set to HYDRO, only one material number for all the Eulerian elements of TYPE is used and a hydrodynamic yield model is chosen.
4. If TYPE is set to STRENGTH, only one material number for all the Eulerian elements of TYPE is used and a nonhydrodynamic yield model is chosen.
5. If TYPE is set to MMHYDRO, different material numbers for all Eulerian elements of TYPE are used and a hydrodynamic behavior is chosen for each material.
6. If TYPE is set to MMSTREN, different material numbers for all Eulerian elements of TYPE are used and a yield model is chosen for each material.
7. In a multimaterial Euler calculation, the options MMSTREN and MMHYDRO can not be mixed; they are mutually exclusive.
8. If the material number is blank or zero, the corresponding elements are void.
9. Initial conditions are defined on the TICEL Bulk Data entry.

PEULER1

Eulerian Element Properties

Eulerian element properties. The initial conditions of these elements are defined in geometric regions.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PEULER1	PID		TYPE	SID					
PEULER1	100		HYDRO	300					

Field	Contents		Type	Default
PID	Unique property number.		I > 0	Required
TYPE	The type of Eulerian material(s) being used.		C	HYDRO
	HYDRO	Hydrodynamic material + void.		
	STRENGTH	Structural material with shear strength + void.		
	MMHYDRO	Multimaterial hydrodynamic + void.		
	MMSTREN	Structural multimaterial with shear strength + void.		
SID	Number of a TICEUL entry specifying the materials and geometric grouping criteria.		I > 0	Required

Remarks:

1. Remarks 1 through 6 of PEULER apply also here.
2. Initial conditions and/or material assignments are defined on the TICEUL Bulk Data entry.

PLOAD

Pressure Loads on the Face of Structural Elements

Defines a pressure load on a triangular or quadrilateral shell or membrane element or on the face of a Lagrangian solid element.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PLOAD	LID	SCALE	G1	G2	G3	G4			
PLOAD	1	-4.0	16	32	11				

Field	Contents	Type	Default
LID	Load set number.	I > 0	Required
SCALE	Scale factor for the pressure.	R	1.0
G1–G4	Grid-point numbers defining either a triangular or quadrilateral surface to which the pressure is applied. For a triangular surface, G4 is blank or zero.	I > 0	Required

Remarks:

1. For quadrilateral surfaces, order the grid points G1 through G4 around the perimeter of the surface, and number them clockwise or counterclockwise.
2. The direction of positive pressure is calculated according to the right-hand rule using the sequence of grid points. See Section 2.3.7 on page 2-17.
3. Reference LID from a TLOADn entry.
4. The pressure $p(t)$ at time *t* is given by

$$p(t) = \text{SCALE} * T(t)$$

where SCALE is the scale factor and $T(t)$ is the value interpolated from the function or table given on the TLOADn entry at time t.

PLOAD4

Pressure Loads on the Face of Structural Elements

Defines a load on a face of a CHEXA, CPENTA, CTETRA, CTRIA3, or CQUAD4 element.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PLOAD4	LID	EID	P1				G1	G3/G4	
PLOAD4	2	1106	10.0				48	123	

Field	Contents	Type	Default
LID	Load set number.	I > 0	Required
EID	Element number.	I > 0	Required
P1	Load per unit surface (pressure) on the face of the element.	R	Required
G1	Number of a grid point connected to a corner of the face. Required data for solid element only (integer or blank).	I > 0	Required
G3	Number of a grid point connected to a corner diagonally opposite to G1 on the same face of a CHEXA or CPENTA element. Required data for quadrilateral faces of CHEXA and CPENTA elements only (integer or blank). G3 is omitted for a triangular surface on a CPENTA element.	I > 0	Required
G4	Number of the CTETRA grid point located at the corner not on the face being loaded. This is required data and is used for CTETRA elements only.	I > 0	Required

Remarks:

1. For solid (CHEXA, CPENTA, CTETRA) elements, the direction of positive pressure is inwards.
2. For plate elements, (CQUAD4, CTRIA3) the direction of positive pressure is the direction of the positive normal determined by applying the right-hand rule to the sequence of the element grid-point connectivity.
3. G1 and G3 are ignored for CTRIA3 and CQUAD4 elements.
4. For the triangular faces of CPENTA elements, G1 is a corner grid-point number that is on the face being loaded, and the G3 or G4 field is left blank. For the faces of CTETRA elements, G1 is a corner grid-point number that is on the face being loaded, and G4 is a corner grid-point number that is not on the face being loaded. Since a CTETRA has only four corner grid points, this grid point G4 is unique and different for each of the four faces of a CTETRA element.

(Continued)

5. If the pressure is 9999., a pressure load is not applied. Instead, it is translated to a CFACE1 entry. This makes it easy to generate CFACE1 entries using a standard preprocessor. See Section 3.2.6 on page 3-4 for details. The LID field is converted to the number of the set of faces.

6. Reference LID by a TLOADn Bulk Data entry.

7. The pressure $p(t)$ at time *t* is given by:

$$p(t) = \text{SCALE} * T(t)$$

   where SCALE is the scale factor and $T(t)$ is the value interpolated from the function or table given on the TLOADn entry at time *t*.

PLOADEX

User-Defined Pressure Load

Defines a pressure load of which the magnitude is specified by a user subroutine.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PLOADEX	LID	NAME	G1	G2	G3	G4			
PLOADEX	100	SIDE	221	222	122	121			

Field	Contents	Type	Default
LID	Number of a set of loads.	I > 0	Required
NAME	Name of the set of pressure loads.	C	Required
G1–G4	Grid-point numbers defining either a triangular or quadrilateral surface to which pressure is applied. For a triangular surface, G4 should be zero or blank.	I ≥ 0	Required

Remarks:

1. Reference LID by a TLOAD1 entry.
2. The subroutine EXPLD must be present in the file referenced by the USERCODE FMS statement.
3. See Section 3.15 on page 3-74 for a description of user-written subroutines.

Defines a spallation model where the minimum pressure is constant.

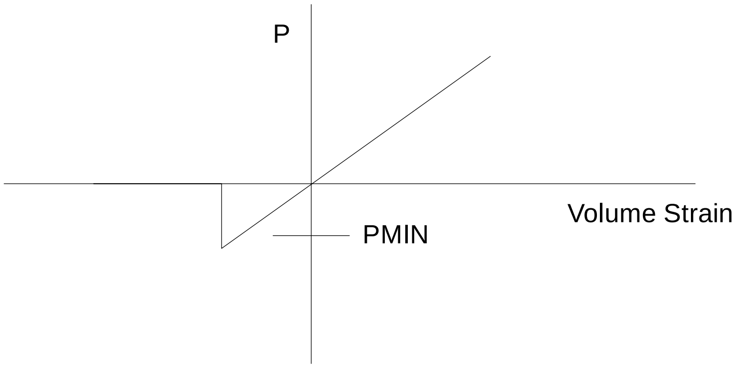
Format and Example:

1	2	3	4	5	6	7	8	9	10
PMINC	PID	VALUE							
PMINC	220	-370.							

Field	Contents	Type	Default
PID	Unique PMINC number.	I > 0	Required
VALUE	The value of the minimum pressure.	R ≤ 0.0	See Remark 2.

Remarks:

1. If the pressure in an element falls below the minimum pressure, the element spalls. The pressure and yield stress are set to zero.



2. The default for the minimum pressure for Lagrangian solids is -1.E20. For Eulerian elements, the default is 0.0.

PORFGBG

Flow Between Two Air Bags Through a Hole

Defines a hole in a couple and/or GBAG (sub)surface, connected to another GBAG.

The velocity of the gasflow through the hole is based on the theory of one-dimensional gas flow through a small orifice, and depends on the pressure difference.

1	2	3	4	5	6	7	8	9	10
PORFGBG	FID			FLOW	GBID				

Field	Contents		Type	Default
FID	Unique number of a PORFGBG entry. It can be referenced from either a GBAGPOR to model the flow between GBAGs, or from a COUPOR to model the flow between an Eulerian air bag and a GBAG.		I > 0	Required
FLOW	Defines the allowed directions of the flow.		C	BOTH
	BOTH	In- and outflow are allowed.		
	IN	Only inflow allowed into the GBAG or the coupling surface that references this entry.		
	OUT	Only outflow allowed into the GBAG or the coupling surface that references this entry.		
GBID	Number of a GBAG entry. This GBAG is the one that is connected to the GBAG or coupling surface that references this entry.		I > 0	Required

Remarks:

1. The PORFGBG entry can be referenced from a COUPOR and/or GBAGPOR entry,
2. When used with Euler, this entry can only be used with the single material hydrodynamic Euler solver, using an EOSGAM (ideal gas) equation of state.

PORFLOW

Porous Flow Boundary

Defines the material properties for the in- or outflow of an Eulerian mesh through a porous (SUB)SURFACE.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PORFLOW	FID		TYPE1	VALUE1	TYPE2	VALUE2	TYPE3	VALUE3	+CONT
PORFLOW	120		XVEL	100.0					+CONT1

+CONT1	TYPE4	VALUE4	-etc.-						
+CONT1									

Field	Contents	Type	Default
FID	Unique number of a PORFLOW entry.	I > 0	Required
TYPEi	The properties on the flow boundary.	C	Required

MATERIAL	Material number.
XVEL	Velocity in the x-direction.
YVEL	Velocity in the y-direction.
ZVEL	Velocity in the z-direction.
PRESSURE	Pressure.
DENSITY	Density.
SIE	Specific internal energy.
FLOW	The type of flow boundary required.
METHOD	The method used for the material transport.

Field	Contents	Type	Default
VALUEi	The value of the property specified in the TYPE field.	R or C	Required
	For TYPEi set to FLOW, the value is a character entry: either IN, OUT or BOTH, indicating that the flow boundary is defined as an inflow, outflow, or possibly an in- or outflow boundary. The default is BOTH.		See Remark 4.
	For TYPEi set to METHOD, the value is a character entry: either VELOCITY or PRESSURE, indicating that the material transport is based on the velocity method or the pressure method. The default is VELOCITY.		See Remark 4.

(Continued)

Remarks:

1. Reference FID by a COUPOR entry.
2. Any material properties not specifically defined have the same value as the element that the (SUB)SURFACE segment is intersecting.
3. The SURFACE can be only a general coupling surface (see the COUPLE entry).
4. The different methods used to calculate the material transport through a porous (sub)surface are described in Section 2.9 on page 2-101.
5. This entry can only be used with the single material hydrodynamic Euler solver.
6. METHOD=VELOCITY is valid for all equation of state models.
   METHOD=PRESSURE is valid for EOSGAM (ideal gas) only.
7. Alternative methods are available to define holes and permeable sections in an air bag. See the entries: COUPOR, GBAGPOR, PORHOLE, PERMEAB, PORFGBG and PERMGBG.

PORHOLE

Holes in Air Bag Surface

Defines a hole in a couple and/or GBAG (sub)surface.

The velocity of the gasflow through the hole is based on the theory of one-dimensional gas flow through a small orifice, and depends on the pressure difference.

1	2	3	4	5	6	7	8	9	10
PORHOLE	FID			FLOW	PENV	RHOENV	SIEENV		

Field	Contents		Type	Default
PID	Unique number of a PORHOLE entry.		I > 0	Required
FLOW	Defines the allowed directions of the flow.		C	BOTH
	BOTH	In- and outflow are allowed.		
	IN	Only inflow allowed.		
	OUT	Only outflow allowed.		
PENV	Environmental pressure.		R > 0	Required
RHOENV	Environmental density.		R > 0	Required
SIEENV	Environmental specific internal energy.		R > 0	Required

Remarks:

1. The PORHOLE entry can be referenced from a COUPOR and/or GBAGPOR entry.
2. When used with Euler, this entry can only be used with the single material hydrodynamic Euler solver, using an EOSGAM (ideal gas) equation of state.
3. The values for the environment p_{env} (PENV), ρ_{env} (RHOENV), e_{env} (SIEENV) must be defined consistent with an ideal-gas equation of state:

$$p_{env} = (\gamma_{env} - 1)\rho_{env}e_{env}$$

   The γ_{env} is calculated by MSC.Dytran, and is used when inflow occurs. Inflow occurs when $p_{env} > p_{inside}$.

PROD

Rod Property

Defines the properties of a rod that is referenced by the CROD entry.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PROD	PID	MID	A						
PROD	17	23	42.6						

Field	Contents	Type	Default
PID	Property number.	I > 0	Required
MID	Material number.	I > 0	Required
A	Cross-sectional area of the rod.	R > 0.	Required

Remark:

1. All PROD entries must have unique property numbers.

PSHELL

Shell-Element Properties

Defines the properties of shell elements.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PSHELL	PID	MID	T						
PSHELL	203	204	0.01						

Field	Contents	Type	Default
PID	Unique property number.	I > 0	Required
MID	Material number. See Remark 4.	I ≥ 0	PID
T	Default value for element thickness.	R ≥ 0.0	See Remark 5.

Remarks:

1. The property number must be unique with respect to all other properties.
2. Shells of constant thickness with three-point Gauss integration and a shear-correction factor of 0.83333 are assumed. For CQUAD4 elements, the formulation is Key-Hoff and for CTRIA3 elements the formulation is C0-Triangle (C0-TRIA).
3. If the thickness is set to 9999., all the elements with this property ID are not treated as CQUAD4 and CTRIA3 elements but are converted to CSEG entries. This allows CSEGs to be defined easily using standard preprocessors. See Section 3.2.6 on page 3-4 for details.
4. Material entries that can be referenced by shell elements defined on the PSHELLn entry can be found in Section 2.5.3 on page 2-27.
5. If the thickness is set to blank or 0.0, the thickness on the CTRIA3 and CQUAD4 must be defined.
6. See also Section 2.15 on page 2-118 on application-sensitive defaults.

PSHELL1

Shell-Element Properties

Defines the properties of Lagrangian shell elements that are much more complicated than the shell elements defined using the PSHELL entry.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PSHELL1	PID	MID	FORM	QUAD	NUMB	SHFACT	REF		+CONT1
PSHELL1	7	2	BLT	GAUSS	5	0.9	MID		+CONT1

+CONT1	T1	T2	T3	T4	TRANSHR	SHRLCK	ADDRES	LENVEC	
+CONT1	10.0	10.0	10.0	10.0					

Field	Contents		Type	Default
PID	Unique property number.		I > 0	Required
MID	Material number. See Remark 2.		I ≥ 0	PID
FORM	Shell formulation.		C	See Remark 3.
	HUGHES	Hughes-Liu.		
	BLT	Belytschko-Lin-Tsay.		
	KEYHOFF	Key-Hoff.		
	C0-TRIA	C0 triangle.		
	MEMB	Membrane element (no bending).		
	DUMMY	Dummy element.		
QUAD	Type of quadrature.		C	GAUSS
	GAUSS	Gauss quadrature.		
	LOBATTO	Lobatto quadrature.		
NUMB	The number of integration points through the thickness. For Gauss and Lobatto quadrature:		I > 0	3
	1	1 point (membrane element)		
	2	2 point		
	3	3 point		
	4	4 point		
	5	5 point		

(Continued)

Field	Contents		Type	Default
SHFACT	Shear factor.		R	0.83333
REF	Reference surface:		C	MID
	TOP	Reference surface is the top surface.		
	MID	Reference surface is the central surface.		
	BOT	Reference surface is the bottom surface.		
T1...T4	Element thickness at the grid points.		R ≥ 0.0	See Remark 8.
TRANSHR	Method of transverse-shear calculation.		C	See Remark 10.
	LINEAR	Linear transverse shear.		
	CONSTANT	Constant transverse shear.		
	CONAPX	Approximated constant transverse shear.		
SHRLCK	Shear-lock avoidance.		C	See Remark 10.
	AVOID	Avoid shear lockup.		
	NOAVOID	No avoid.		
ADDRES	Stores grid-point addresses in memory.		C	See Remark 10.
	SAV	Save addresses.		
	NOSAVE	Do not save.		
LENVEC	Vector length.		I	See Remark 10.

Remarks:

1. Shells of constant thickness with three-point Gauss integration are more easily defined using the PSHELL entry.

2. Material entries that can be referenced by shell elements can be found in Section 2.5.3 on page 2-27.

3. For CQUAD4 elements, the default formulation is KEYHOFF. For CTRIA3 elements, the default formulation is CO-TRIA. See also Section 2.15 on page 2-118 for application sensitive defaults.

4. Make the property number unique with respect to all other properties.

(Continued)

5. If the thickness T is set to 9999., all elements with this property number are not treated as CQUAD4 and CTRIA3 elements but are converted to CSEG entries. This conversion allows CSEGs to be defined easily using standard preprocessors. See Section 3.2.6 on page 3-4 for details.

6. Membrane elements can only be triangular and must reference a DMAT or DMATEL material entry. In case the HUGHES shell formulation is used, only an elastic material can be referred to.

7. Dummy elements are used to define rigid bodies or to achieve a closed volume when defining coupling surfaces. Do not use them to create CSEG entries.

8. If the thickness is set to blank or 0.0, the thickness is defined on the CTRIA3 and CQUAD4 entry.

9. Specifying QUAD and NUMB is not necessary if FORM is MEMB.

10. The following defaults apply:

	BLT	HUGHES	KEYHOFF
TRANSHR	NOT AVAILABLE	NOT AVAILABLE	LINEAR
SHRLCK	NOAVOID	NOT AVAILABLE	AVOID
ADDRES	SAVE	NOT AVAILABLE	SAVE
LENVEC	LENVEC	NOT AVAILABLE	LENVEC

11. When shell elements undergo large twisting, the linear transverse shear calculations must be used (TRANSHR). It increases accuracy at the expense of more computer time.

12. The default vector length for vector machines is used whenever LENVEC is not defined. Increasing the vector length is usually more efficient, but requires more memory. In some problems a recurrence in the force update may inhibit vectorization on vector machines. Decreasing the vector length may avoid the recurrence. Examine the problem output for information on this recurrence.

PSOLID

Lagrangian Solid-Element Properties

Defines the properties of Lagrangian solid elements.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PSOLID	PID	MID							
PSOLID	2	100							

Field	Contents	Type	Default
PID	Unique property number.	I > 0	Required
MID	Material number. See Remark 2.	I > 0	PID

Remarks:

1. The property number must be unique with respect to all other property numbers.
2. Material entries that can be referenced by Lagrangian solid elements are given in Section 2.5.3 on page 2-27.
3. Single-point integration is assumed.
4. Use the PEULER entry to define the properties of the Eulerian elements.

PSPR

Linear-Elastic Spring Properties

Defines the properties for a linear-elastic spring with failure.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PSPR	PID	K	FAILMTF	FAILMCF					
PSPR	8	20.0E3							

Field	Contents	Type	Default
PID	Unique property number.	I > 0	Required
K	Elastic stiffness (force/displacement).	R > 0	Required
FAILMTF	Tensile failure force.	R > 0	No failure
FAILMCF	Compressive failure force.	R > 0	No failure

Remarks:

1. This entry defines a linear-elastic spring with failure. PSPR1 can be used to define nonlinear springs.
2. The behavior of this spring is discussed in Section 2.3.8 on page 2-17.

PSPR1

Nonlinear-Elastic Spring Properties

Defines the properties for a nonlinear spring where the stiffness varies with displacement.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PSPR1	PID	TABLE							
PSPR1	8	15							

Field	Contents	Type	Default
PID	Unique property number.	I > 0	Required
TABLE	Number of a TABLExx entry defining the variation of force (y-value) with displacement (x-value).	I > 0	Required

Remarks:

1. The values in the table are interpolated to determine the force for a particular displacement.
2. The behavior of this spring is discussed in Section 2.3.8 on page 2-17.

PSPREX

User-Defined Spring Properties

Defines the properties for CSPR spring elements that are used with user-written spring subroutines.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PSPREX	PID	V1	V2	V3	V4	V5	V6	V7	
PSPREX	27	39.6	100.E6						

Field	Contents	Type	Default
PID	Unique property number.	I > 0	Required
V1–V7	User values.	R	0.0

Remarks:

1. The seven user values are passed to the user subroutine EXSPR.
2. MSC.Dytran does no checking of the user values.
3. For a discussion of the various types of spring elements, see Section 2.3.8 on page 2-17. For a discussion of how to use user-written subroutines, see Section 3.15 on page 3-74.

PVISC

Linear-Damper Properties

Defines the properties of a linear viscous damper.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PVISC	PID	C			PID	C			
PVISC	7	0.01							

Field	Contents	Type	Default
PID	Unique property number.	I > 0	Required
C	Damping constant.	R	0.0

Remarks:

1. This entry defines a linear viscous damper. PVISC1 can be used to define nonlinear dampers.
2. The behavior of this type of damper is discussed in Section 2.3.9 on page 2-19.

PVISC1

Nonlinear Damper Properties

Defines the properties of a nonlinear damper where the damping constant varies with the velocity.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PVISC1	PID	TABLE							
PVISC1	8	236							

Field	Contents	Type	Default
PID	Unique property number.	I > 0	Required
TABLE	Number of a TABLExx entry defining the variation of the force (y-value) with velocity (x-value).	I > 0	Required

Remarks:

1. This entry defines the properties of a nonlinear damper. Use the PVISC entry to define linear dampers.
2. The values in the table are interpolated to get the force for a particular velocity.
3. The behavior of this damper is discussed in Section 2.3.9 on page 2-19.

PVISCEX

User-Defined Damper Properties

Defines the properties for CVISC damper elements used with user-written subroutines.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PVISCEX	PID	V1	V2	V3	V4	V5	V6	V7	
PVISCEX	27	39.6	100.E6						

Field	Contents	Type	Default
PID	Unique property number.	I > 0	Required
V1–V7	User values.	R	0.0

Remarks:

1. The seven user values are passed to the user subroutine EXVISC.
2. MSC.Dytran does no checking on the user-supplied values.
3. For a discussion of the various types of damper elements, see Section 2.3.9 on page 2-19. For a discussion of user-written subroutines, see Section 3.15 on page 3-74.

PWELD

Spotweld Property

Defines the properties of a spotweld connection between two grid points. It is referenced by the CROD entry.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PWELD	PID	FAILTENS	FAILCOMP	FAILSHEA	FAILTORQ	FAILBEND	FAILTOTF	FAILTOTM	+CONT1
PWELD	101	1.E5							+CONT1

+CONT1	FAILTIME								
+CONT1									

Field	Contents	Type	Default
PID	Property number.	I > 0	Required
FAILTENS	Failure force in tension.	R ≥ 0.0	No failure
FAILCOMP	Failure force in compression.	R ≥ 0.0	No failure
FAILSHEA	Failure force in shear.	R ≥ 0.0	No failure
FAILTORQ	Failure torque.	R ≥ 0.0	No failure
FAILBEND	Failure bending moment.	R ≥ 0.0	No failure
FAILTOTF	Failure total force.	R ≥ 0.0	No failure
FAILTOTM	Failure total moment.	R ≥ 0.0	No failure
FAILTIME	Failure based on time.	R ≥ 0.0	No failure

Remarks:

1. A spotweld is treated as a rigid body with its inertia properties calculated by lumping the properties of the end points.

   A set of spotwelds and/or BJOINs connected to each other is treated as one rigid body.

   Lumping of the initial positions and velocities:

   a. The lumped rigid-body mass is not zero:

      The initial positions and velocities are lumped using mass-weighting.

      If a grid point has zero mass, it’s initial position and velocity will be ignored.

(Continued)

b. The lumped rigid-body mass is zero:

The initial positions and velocities are lumped by averaging.

Boundary conditions allocated to the grid points will be combined, if possible.

When failure of a spotweld that is connected to other spotweld(s) and/or BJOINs occurs, the rigid-body lumped properties and boundary conditions are redefined.

2. If the end points of a spotweld coincide, the direction vector can not be determined. As a result, no components of tension, compression, shear, torque, and bending can be calculated. The total force or moment will be used instead to check for failure against the specified failure criteria:

   a. The total force acting on the spotweld will be checked against:

      FAILTENS
      FAILCOMP
      FAILSHEA
      FAILTOTF

   b. The total moment acting on the spotweld will be checked against:

      FAILTORQ
      FAILBEND
      FAILTOTM

   The spotweld will fail if one of the above criteria is satisfied.

3. All failure modes are checked simultaneously.

4. An overview of the generated spotwelds and BJOINs can be requested. See PARAM,INFO-BJOIN.

5. You have access to the results of the spotweld elements by requesting for results of the corresponding CROD elements.

   The variables are only calculated for spotwelds with a failure criterion. They are described as follows:

   FAIL Failure time.

   XFORCE Tension/compression force in the spotweld.

   YFORCE Shear force in the spotweld in direction of shear vector at end point 1.

   ZFORCE Shear force in the spotweld in direction of shear vector at end point 2.

   XMOMENT Torque in the spotweld.

   YMOMENT Bending moment in the spotweld in direction of bending moment vector at end point 1.

(Continued)

ZMOMENT Bending moment in the spotweld in direction of bending moment vector at end point 2.

FIBL1 Mode of failure: 0 = Not failed.

1 = Failed on TAILTENS.

2 = Failed on FAILCOMP.

3 = Failed on FAILSHEA.

4 = Failed on FAILTORQ.

5 = Failed on FAILBEND.

6 = Failed on FAILTOTF.

7 = Failed on FAILTOTM.

8 = Failed on FAILTIME.

RBC3

Rigid-Body Constraint

Defines a three-point constraint on a RIGID surface, a MATRIG, or RBE2-FULLRIG rigid body.

Format and Example:

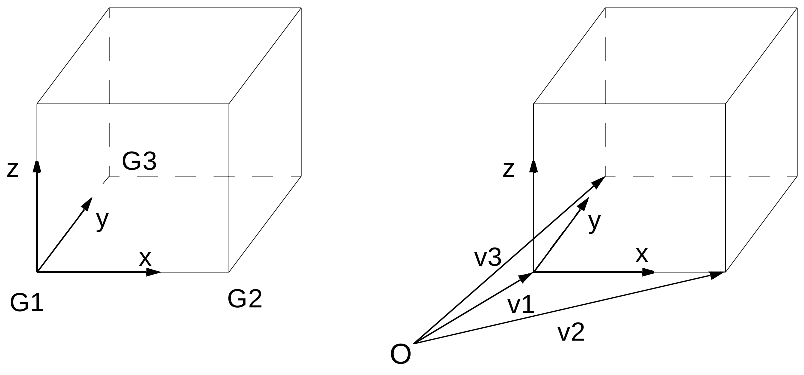
1	2	3	4	5	6	7	8	9	10
RBC3	RID	MID	C	G1	G2	G3			+CONT1
RBC3	3	MR5	12	26	23	27			+CONT1

+CONT1	X1	X2	X3	Y1	Y2	Y3			+CONT2
+CONT1									+CONT2

+CONT2	Z1	Z2	Z3						
+CONT2									

Field	Contents	Type	Default
RID	Unique rigid-body constraint number.	I > 0	Required
MID	Number, MR<Number>, or FR<Number>, where a number refers to a RIGID surface, MR<Number> refers to a MATRIG, and FR<Number> refers to an RBE2-FULLRIG entry.	C or I	Required
C	Component number of local coordinate (any unique combination of the digits 1 through 6 with no embedded blanks).	I > 0	See Remark 3.
G1, G2, G3	Grid-point numbers defining the RBC3 coordinate system.	I > 0	See Remark 1.
X1, X2, X3 Y1, Y2, Y3 Z1, Z2, Z3	Coordinates of three points defining the RBC3 coordinate system.	R	See Remark 1.

(Continued)



RBC3 Coordinate System.

Remarks:

1. If G1, G2, and G3 are specified, then the RBC3 coordinate system is determined by the grid points. The position vectors for G1, G2, and G3 will be denoted by v1, v,2 and v3, respectively. If G1, G2 and G3 are not specified, then the coordinate system is either specified by the vectors v1 = (X1, Y1, Z1), v2 = (X2, Y2, Z2) and v3 = (X3, Y3, Z3) if X1 through Z3 are specified, or by the vectors v1 = (0, 0, 0), v2 (1, 0, 0), and v3 = (0, 1, 0), by default. The local x-axis is the normalized vector v2-v1. The local z-axis is the normalized cross product of the vectors v2-v1 and v3-v1 and is thus perpendicular to the plane spanned by these vectors. The local y-axis is the cross product of the local z- and x-axis.

2. The grid points G1, G2, and G3 must be unique. Also, the vectors (X1, X2, X3), (Y1, Y2, Y3), and (Z1, Z2, Z3) must be unique.

3. The translational and rotational constraints are applied to the center of gravity of the rigid body in the local coordinate system.

RBE2

Rigid-Body Element

Defines a set of grid points that form a rigid element.

Format and Example:

1	2	3	4	5	6	7	8	9	10
RBE2	EID	G1	CM	G2	G3	G4	G5	G6	+CONT
RBE2	9	8	12	10	12	14	15	16	+CONT
+CONT	G7	G8	THRU	G10	-etc.-				
+CONT	20	25	THRU	32					

Field	Contents	Type	Default
EID	Number of the rigid-body element.	I > 0	Required
G1..Gn	Grid-point numbers with degrees of freedom that are specified by CM are coupled.	I > 0	Required
CM	Component numbers of the grid points that are coupled. These are in the basic coordinate system. The components are indicated by any of the digits 1, 2, 3, 4, 5, or 6 with no embedded blanks. Combinations are allowed, e.g., 12, 123. In case the rigid element should behave as a full rigid body, CM should read FULLRIG.	See Remark 7.	Required

Remarks:

1. The element number should be unique with respect to all other rigid-element numbers.
2. The RBE2 definition allows particular degrees of freedom of a set of grid points to be coupled so that the grid points always move the same amount. The motion of the set of grid points is the weighted average of the motion of all the grid points for the degrees of freedom coupled through the RBE2 definition.
3. The component numbers refer to the basic coordinate system.
4. Loads, initial velocities, or constraints should be applied to the first (master) grid point. They are then applied to the coupled degrees of freedom for all the grid points defined on the RBE2 entry.
5. Both rotational and translational degrees of freedom can be coupled.
6. Grid points associated with rigid surfaces cannot be part of an RBE2 grid-point list.

(Continued)

7. Instead of defining coupled components, it is possible to define the RBE2 entry as a single rigid body by using the FULLRIG option. The geometric properties of the rigid body are calculated from the geometry and the mass of the grid points.

8. Grid points referred to by the JOIN entry cannot be part of an RBE2 grid-point list.

9. It is possible to merge an RBE2 entry with a MATRIG entry by using the FULLRIG option and PARAM,MATRMERG or PARAM,MATRMRG1. A normal RBE2 entry (with constraint) however cannot be merged with a MATRIG entry or an RBE2-FULLRIG entry.

10. By using PARAM,CFULLRIG, all 123456 constraints on a normal RBE2 will be automatically converted to the FULLRIG option.

11. By using PARAM,RBE2INFO,GRIDON, the gridpoints of the RBE2 will be listed in the output file.

12. Section 2.3.5 on page 2-12 for a description of the use of RBE2.

RBHINGE

Rigid Body Hinge

Defines a hinge between a rigid body and a deformable structure on the common six degrees of freedom.

Format and Example:

1	2	3	4	5	6	7	8	9	10
RBHINGE	RID	MID	C	G1	G2	THRU	G3	BY	+CONT1
RBHINGE	1	14	456	1	10	THRU	100	BY	+CONT1

+CONT1	G5	-etc.-							
+CONT1	2								

Field	Contents	Type	Default
RID	Unique rigid body hinge number.	I > 0	Required
MID	Number, MR<Number>, or FR<Number>, where a number refers to a RIGID surface, MR<Number> refers to a MATRIG, and FR<Number> refers to an RBE2-FULLRIG entry.	C or I	Required
C	Component number of rotation which is defined as a hinge (any unique combination of the digits 4, 5 or 6).	I > 0	456
Gi	Grid point numbers. THRU indicates a range of grid points. BY is the increment within this range.	I > 0	Required

Remarks:

1. When grid points are part of a rigid body and a deformable structure, they transfer rotational momentum if they posses six degrees of freedom. This is the case when they are connected to a beam/bar, tria or quad shell element. Using the RBHINGE entry specifies rotational degrees of freedom that can be defined as behaving as a hinge.

   RBHINGE is not needed for grid points that have only three degrees of freedom, since the hinge is defined by default.

2. The component number refers to the global coordinate system.

RCONN

Rigid Connection

Defines a rigid connection between the different parts of Lagrangian meshes (tied surfaces).

Format and Example:

1	2	3	4	5	6	7	8	9	10
RCONN	CID	STYPE	MTYPE	SID	MID	OPTION			+CONT1
RCONN	7	GRID	SURF	3	7	NORMAL			+CONT1

+CONT1									+CONT2
+CONT1									+CONT2

+CONT2	CLSGAP	GAPDIS	GAPDISV						
+CONT2									

Field	Contents	Type	Default
CID	Unique rigid-connection number.	I > 0	Required
STYPE	Type of entity used to define the slave surface. SURF A SURFACE entry is used to select the faces of the elements on the slave surface SID is the number of the SURFACE entry. See Remark 2. GRID Grid points will be tied to the master surface. SID then refers to a SET1 entry containing the list of grid points to be used. See Remarks 3 and 4.	C	SURF
MTYPE	Type of entity used to define the master surface. SURF A SURFACE entry is used to select the faces of the elements on the master surface. MID is the number of the SURFACE entry.	C	SURF
SID	The number of a slave SURFACE entry or the number of a SET1 entry containing the list of grid points.	I > 0	Required
MID	The number of a master SURFACE entry.	I > 0	Required

(Continued)

Field	Contents		Type	Default
OPTION	Only used if discrete grid points are tied to a surface (STYPE is equal to GRID).		C	NORMAL
	NORMAL	The grid points are tied to the master surface. See Remark 3.		
	SHELL	The grid points are attached to the edge of shell or beam elements, which are tied to the shell surface. See Remark 4.		
CLSGAP	Switch to automatically close any gaps that are present between the master-slave surface.		C	No
	YES	Gaps are automatically closed.		
	NO	Gaps are not closed. See Remark 6.		
GAPDIS	Defines the tolerance used in the search for a master face. If the distance between a slave point and a master face falls within this tolerance, the master face is accepted. If not, the search for a correct master face continues.		C	DISTANCE
	FACTOR	The tolerance has the length of: (GAPDISV) ∗ (Minimum side of faces in slave surface). See Remark 9.		
	DISTANCE	The tolerance has the length as specified at GAPDISV.		
GAPDISV	The value of the gap tolerance or a factor to calculate this tolerance depending on the value of GAPDIS.		R	1.E20

Remarks:

1. The RCONN entry can be used to define three types of connection as described in Section 2.6.5 on page 2-90.

2. Two Surfaces Tied Together.

   Define slave and master segments representing the two surfaces to be tied together. There should not be a gap between the two sets of segments. The two surfaces are tied together during the analysis.

(Continued)

3. Grid Points Tied to a Surface.

   If STYPE is set to GRID and OPTION is set to NORMAL, the slave entities comprise discrete grid points that are tied to the master surface during the analysis. The grid points must lie on the surface.

4. Shell Edge Tied to a Shell Surface.

   If STYPE is set to GRID and OPTION is set to SHELL, the edges of shell or beams elements can be tied to the faces of other shells. The grid points attached to the edge of the shells/beams must be selected as the slave grid points. The shell surface to which they are tied must be selected as the master surface. The two sets will then be tied together throughout the analysis. All degrees of freedom will be coupled.

5. The CLSGAP entry enables you to define two different meshes that are not coincident over the master/slave interface. If the option is set to YES, the slave surface becomes coincident (according to projections) with the master surface.

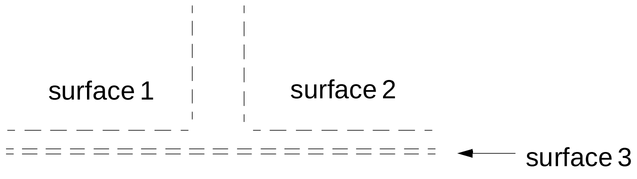
6. The search method of the contact algorithm is used to find the closest master face. The tolerance defined with the GAPDIS/GAPDISV fields is similar to the monitoring distance defined on the CONTACT entry with the MONDIS/MONDISV fields.

7. The use of the gap closing CLSGAP can cause an element to collapse. This may happen if the GAPDISV tolerance is set to a value greater than the length of the side of an element.

8. When a solid and a shell mesh are tied together, the rotational degrees of freedom of the shell grid points are not coupled.

9. If STYPE is set to GRID, the option FACTOR in the GAPDIS field is not allowed.

10. Avoid the following situation when using the RCONN entry:



RCONN1: surface 1 as slave of surface 2

RCONN2: surface 1 as slave of surface 3

In this situation, the corner point of surface 1 will have two masters to follow. Therefore, the mass and the force of the corner point will be lumped twice.

RCONREL

Rigid Connection with Rigid Ellipsoids

Defines a connection between a rigid ellipsoid and Lagrangian grid points or rigid bodies.

Format and Example:

1	2	3	4	5	6	7	8	9	10
RCONREL	RID	SIDC	TYPE	SID					
RCONREL	20	30	GRID	40					

Field	Contents	Type	Default
RID	Unique number of an RCONREL entry.	I > 0	Required
SIDC	Number of a SETC entry giving the name of the rigid ellipsoid to which entities are connected. See Remark 1.	I > 0	Required
TYPE	The type of entities that are connected to the rigid ellipsoid. GRID Grid points. RIGID Rigid surface, RBE2-FULLRIG, and MATRIG.	C	Required
SID	The number of a SET1 entry listing the grid points or rigid surfaces that are connected to the rigid ellipsoid. In case a MATRIG or an RBE2-FULLRIG entry is connected to the rigid ellipsoid, SID refers to a SET1 entry listing MR<id> or FR<id>, where id refers to a MATRIG or an RBE2-FULLRIG entry, respectively.	I > 0	Required

Remark:

1. The SETC entry can only contain the name of one ellipsoid.

RELEX

External Definition of a Rigid Ellipsoid

Defines a rigid ellipsoid whose properties and motion are defined by either ATB or MADYMO.

Format and Example:

1	2	3	4	5	6	7	8	9	10
RELEX	NAME	PROG	LABEL						
RELEX	HEAD	MADYMO	110						

Field	Contents		Type	Default
NAME	This name is used within the MSC.Dytran input file to define the interactions between the external ellipsoid and MSC.Dytran grid points and rigid bodies. This name is also used in the output requests. When coupled to ATB**:** The name must correspond to the name of the ATB segment. When coupled to MADYMO: The name doesn’t need to correspond to the name of the ellipsoid in the MADYMO input file.		C	Required
PROG	Name of the external program.		C	Required
	MADYMO	MSC.Dytran runs coupled with MADYMO V5.1.1.		
	ATB	MSC.Dytran runs coupled with ATB.	C	Required
LABEL	Identification label when running coupled to MADYMO. The label is not used by DYTRAN, it is only received from MADYMO to identify the ellipsoid. The value must be equal to the value used in the MADYMO input file: • Force models. • Coupling. • Ellipsoids. • LABEL system ellipsoid var1...varN. • End ellipsoids. • End coupling. • End force models.		I	Required

(Continued)

Remarks:

1. This entry should only be used when MSC.Dytran is used with MADYMO or ATB.
2. Rigid ellipsoids can be defined directly within MSC.Dytran using the RELLIPS entry.
3. RELEX and RELLIPS entries can not be mixed in the same model. A mixture of MADYMO and ATB ellipsoids is not allowed.
4. For ATB, only the segment contact ellipsoid can be used. The name of the contact ellipsoid is equal to the name of the segment, as specified on the first field of the B.2 entry in the ATB input file.
5. See Appendix D for instructions on how to use ATB.

RELLIPS

Rigid Ellipsoid

Defines an analytical rigid ellipsoid.

Format and Example:

1	2	3	4	5	6	7	8	9	10
RELLIPS	NAME	A	B	C	MASS	XCG	YCG	ZCG	+CONT1
RELLIPS	10	0.1	10.0	10.0	0.1	0.	0.	0.	+CONT1

+CONT1	XL	YL	ZL	XS	YS	ZS			+CONT2
+CONT1	0.	0.	1.	1.	0.	0.			+CONT2

+CONT2	VX	VY	VZ	WA	WB	WC			
+CONT2	–0.1								

Field	Contents	Type	Default
NAME	Ellipsoid name.	C	Required
A, B, C	Size of the ellipsoid in the a-, b-, and c-directions (a > b > c).	R > 0	Required
MASS	Mass of the ellipsoid.	R > 0	Required
XCG, YCG, ZCG	Coordinates of the geometric center of the ellipsoid (the geometric center of the ellipsoid coincides with the center of gravity).	R	0.0
XL, YL, ZL	Vector defining the orientation of the longest axis of the ellipsoid.	R	0.0
XS, YS, ZS	Vector defining the orientation of the shortest axis of the ellipsoid.	R	0.0
VX, VY, VZ	Initial translational velocities of the center of the ellipsoid in the x-, y-, and z-directions.	R	0.0
WA, WB, WC	Initial rotational velocities of the ellipsoid in the a-, b-, and c-directions.	R	0.0

Remark:

1. RELEX and RELLIPS entries cannot be mixed in the same model.

RFORCE

Rotational Force Field

Defines loading due to a centrifugal acceleration field.

Format and Example:

1	2	3	4	5	6	7	8	9	10
RFORCE	LID	G		SCALE	NX	NY	NZ		
RFORCE	29	2		37.6	1.0	2.0	0.		

Field	Contents	Type	Default
LID	Number of a set of loads.	I > 0	Required
G	Grid-point number on the axis of rotation.	I > 0	Required
SCALE	Scale factor for rotational velocity. See Remark 6.	R	1.0
NX, NY, NZ	Components of the rotational-direction vector. At least one component must be nonzero. The vector (NX, NY, NZ) acts at grid point G.	R	0.

Remarks:

1. The rotational velocity is calculated as

$$W(t) = T(t) * \text{SCALE} * N$$

where SCALE is the scale factor, N the directional vector (NX, NT, NZ), and $T(t)$ the value at time t interpolated from the table or function referenced by the TLOADn entry.

2. LID must be referenced by a TLOADn entry.

3. The type field on the TLOADn entry must be set to zero.

4. Only one centrifugal force field can be defined in the problem.

5. Centrifugal forces act on all Lagrangian structural elements and rigid surfaces.

6. The rotation is input in revolutions per unit time.

RIGID

Rigid Surface

Defines a rigid surface.

Format and Example:

1	2	3	4	5	6	7	8	9	10
RIGID	RID	SID	MASS		XCG	YCG	ZCG		+CONT1
RIGID	25	32	527.		117.6	339.4	21.0		+CONT1

+CONT1		VX	VY	VZ	CID	WX	WY	WZ	+CONT2
+CONT1									+CONT2

+CONT2		IXX	IXY	IXZ	IYY	IYZ	IZZ		
+CONT2		4495.			4495.		4495.		

Field	Contents	Type	Default
RID	Unique rigid-surface number.	I > 0	Required
SID	Number of a SURFACE entry defining the shape of the rigid surface.	I > 0	Required
MASS	Mass of the rigid body.	R > 0	Required
XCG,YCG, ZCG	Coordinates of the center of gravity of the rigid body.	R	Required
VX, VY, VZ	Initial translational velocities of the center of mass in the basic coordinate system.	R	0.0
CID	Number of a CORD2R entry.	I	0
WX, WY, WZ	Initial rotational velocities, relative to a coordinate system with its origin at the center of gravity, and its axes aligned with the coordinate system CID.	R	0.0
IXX, IXY, IXZ IYY, IYZ, IZZ	Moments of inertia, relative to a coordinate system with its origin at the center of gravity, and its axes aligned with the coordinate system CID.	R	See Remark 2.

Remarks:

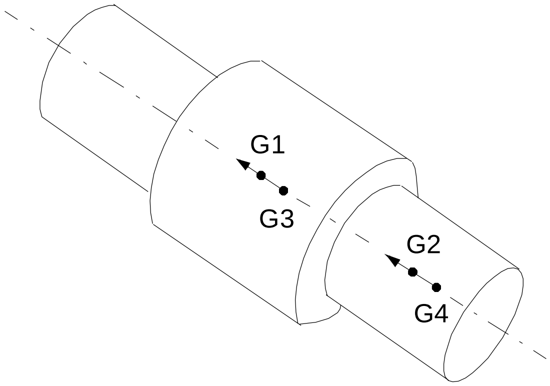
1. A CID of zero references the basic coordinate system.
2. The default for IXX, IYY, and IZZ is 1.E10; the default for IXY, IXZ, and IYZ is zero.
3. The mass of the rigid surface is distributed to the grid points on the surface.

Defines a cylindrical joint between grid points on two rigid bodies.

Format and Example:

1	2	3	4	5	6	7	8	9	10
RJCYL	ID	STIFF	G1	G2	G3	G4			
RJCYL	9	1.0	47	173	53	269			

Field	Contents	Type	Default
ID	Unique joint number.	I > 0	Required
STIFF	Relative stiffness of the joint.	R	1.0
G1-G4	Grid-point numbers defining the joint connectivity.	I > 0	Required



Remarks:

1. The geometry of the joint changes during the analysis as the grid points move.
2. G1 and G3 are grid points belonging to the first rigid body; G2 and G4 are grid points belonging to the second rigid body.
3. The vector from G1 to G3 determines the axis of sliding. Spring forces are calculated between G1 and G2 and between G3 and G4 to keep all four points on the axis of sliding.
4. If the initial position of grid points G2 and/or G4 is off the axis of sliding a force in the joint is initialized.
5. The absolute stiffness of the rigid body joints is calculated automatically by MSC.Dytran. The stiffness of the joints is taken such that a stable solution is guaranteed. This stiffness calculation takes into account the fact that a rigid body can be constrained by more than one joint.

(Continued)

6. The absolute stiffness of the rigid-body joints is multiplied by a factor defined on PARAM,RJSTIFF. By default, RJSTIFF = 1.0. This parameter can be used to increase or decrease the stiffness of the joints. Care must be taken by using this parameter because too high a value might lead to an unstable calculation.

7. Although the joint is designed for usage with rigid bodies, it is allowed to use finite-element grid points.

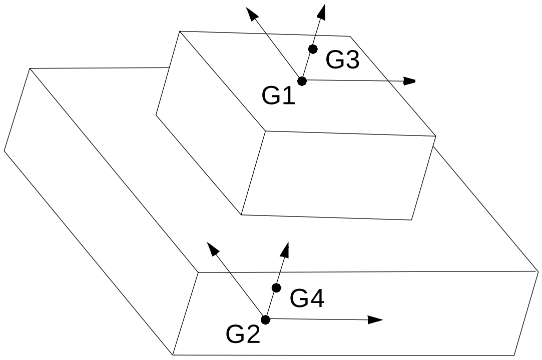
8. RJCYL can be applied to rigid bodies defined by the RIGID entry as well as to rigid bodies defined by the MATRIG or RBE2-FULLRIG entries.

Defines a planar joint between grid points on two rigid bodies.

Format and Example:

1	2	3	4	5	6	7	8	9	10
RJPLA	ID	STIFF	G1	G2	G3	G4			
RJPLA	9	1.0	47	173	53	269			

Field	Contents	Type	Default
ID	Unique joint number.	I > 0	Required
STIFF	Relative stiffness of the joint.	R	1.0
G1-G4	Grid-point numbers defining the joint connectivity.	I > 0	Required



Remarks:

1. The geometry of the joint changes during the analysis as the grid points move.
2. G1 and G3 are grid points belonging to the first rigid body; G2 and G4 are grid points belonging to the second rigid body.
3. The vector from G1 to G3 defines the normal to the plane on which the two bodies can slide relative to each other. Spring forces are calculated between G1 and G2 and between G3 and G4 to keep all four points in the plane of sliding.
4. If the initial position of grid points G2 and/or G4 is off the normal to the plane of sliding, a force in the joint is initialized.
5. The absolute stiffness of the rigid-body joints is calculated automatically by MSC.Dytran. The stiffness of the joints is taken such that a stable solution is guaranteed. This stiffness calculation takes into account the fact that a rigid body can be constrained by more than one joint.

(Continued)

6. The absolute stiffness of the rigid-body joints is multiplied by a factor defined on PARAM,RJSTIFF. By default, RJSTIFF = 1.0. This parameter can be used to increase or decrease the stiffness of the joints. Care must be taken by using this parameter because too high a value might lead to an unstable calculation.

7. Although the joint is designed for usage with rigid bodies, it is allowable to use finite-element grid points.

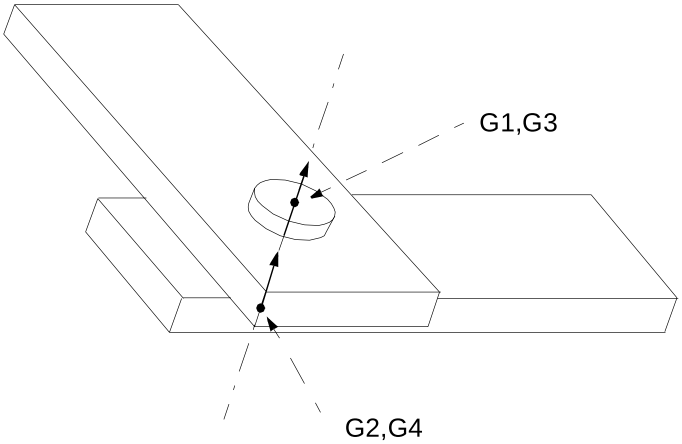
8. RJPLA can be applied to rigid bodies defined by the RIGID entry as well as to rigid bodies defined by the MATRIG or RBE2-FULLRIG entries.

Defines a revolute joint (hinge) between grid points on two rigid bodies.

Format and Example:

1	2	3	4	5	6	7	8	9	10
RJREV	ID	STIFF	G1	G2	G3	G4			
RJREV	9	1.0	47	173	53	269			

Field	Contents	Type	Default
ID	Unique joint number.	I > 0	Required
STIFF	Relative stiffness of the joint.	R	1.0
G1-G4	Grid point numbers defining the joint connectivity.	I > 0	Required



Remarks:

1. The geometry of the joint changes during the analysis as the grid points move.
2. G1 and G3 are grid points belonging to the first rigid body; G2 and G4 are grid points belonging to the second rigid body. G1 and G2 should be coincident, and G3 and G4 should be coincident.
3. The vector from G1 to G3 determines the axis about which the two bodies can rotate. Spring forces are calculated between G1 and G2 and between G3 and G4 to keep all four points on the axis of rotation.
4. The absolute stiffness of the rigid-body joints is calculated automatically by MSC.Dytran. The stiffness of the joints is calculated so that a stable solution is guaranteed. This stiffness calculation takes into account the fact that a rigid body can be constrained by more than one joint.

(Continued)

5. The absolute stiffness of the rigid-body joints is multiplied by a factor defined on PARAM,RJSTIFF. By default, RJSTIFF = 1.0. This parameter can be used to increase or decrease the stiffness of the joints. Care must be taken by using this parameter because too high a value might lead to an unstable calculation.

6. Although the joint is designed for usage with rigid bodies, it is allowed to use finite-element grid points.

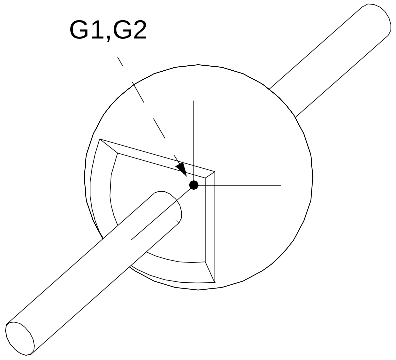
7. RJREV can be applied to rigid bodies defined by the RIGID entry as well as to rigid bodies defined by the MATRIG or RBE2-FULLRIG entries.

Defines a spherical (ball) joint between grid points on two rigid bodies.

Format and Example:

1	2	3	4	5	6	7	8	9	10
RJSPH	ID	STIFF	G1	G2					
RJSPH	9	1.0	47	173					

Field	Contents	Type	Default
ID	Unique joint number.	I > 0	Required
STIFF	Relative stiffness of the joint.	R	1.0
G1-G2	Grid-point numbers defining the joint connectivity.	I > 0	Required



Remarks:

1. The geometry of the joint changes during the analysis as the grid points move.
2. G1 belongs to the first rigid body, G2 belongs to the second rigid body. G1 and G2 should be coincident. Spring forces are calculated between G1 and G2 so that the two bodies can rotate about the joint.
3. The absolute stiffness of the rigid-body joints is calculated automatically by MSC.Dytran. The stiffness of the joints is taken such that a stable solution is guaranteed. This stiffness calculation takes into account the fact that a rigid body can be constrained by more than one joint.
4. The absolute stiffness of the rigid-body joints is multiplied by a factor defined on PARAM,RJSTIFF. By default, RJSTIFF = 1.0. This parameter can be used to increase or decrease the stiffness of the joints. Care must be taken by using this parameter because too high a value might lead to an unstable calculation.

(Continued)

5. Although the joint is designed for usage with rigid bodies, it is allowable to use finite-element grid points.

6. RJSPH can be applied to rigid bodies defined by the RIGID entry as well as to rigid bodies defined by the MATRIG or RBE2-FULLRIG entries.

RJTRA

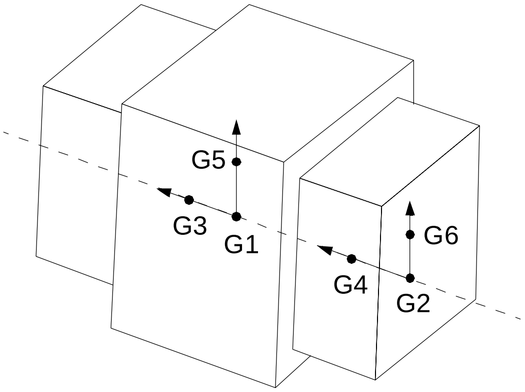
Translational-Joint Constraint Between Rigid Bodies

Defines a translational joint, which allows relative sliding but no rotation, between grid points on two rigid bodies.

Format and Example:

1	2	3	4	5	6	7	8	9	10
RJTRA	ID	STIFF	G1	G2	G3	G4	G5	G6	
RJTRA	9	1.0	47	173	53	269	17	87	

Field	Contents	Type	Default
ID	Unique joint number.	I > 0	Required
STIFF	Relative stiffness of the joint.	R	1.0
G1-G6	Grid-point numbers defining the joint connectivity.	I > 0	Required



Remarks:

1. The geometry of the joint changes during the analysis as the grid points move.
2. G1, G3, and G5 are grid points belonging to the first rigid body; G2, G4, and G6 are points belonging to the second rigid body.
3. The vector from G1 to G3 determines the axis along which the two bodies can slide relative to each other. The vectors from G1 to G5 and from G2 to G6 are perpendicular to the axis of sliding. Spring forces are calculated between G1 and G2, between G3 and G4, and between G5 and G6 to keep the first four grid points on the axis of sliding and the other two grid points on a vector that is parallel to the axis of sliding.

(Continued)

4. If the initial position of grid points G2 and/or G4 is off the axis of sliding a force in the joint is initialized. If the initial vector from G5 to G6 is not parallel to the vector from G1 to G3, a force in the joint is initialized.

5. The absolute stiffness of the rigid-body joints is calculated automatically by MSC.Dytran. The stiffness of the joints is calculated so that a stable solution is guaranteed. This stiffness calculation takes into account the fact that a rigid body can be constrained by more than one joint.

6. The absolute stiffness of the rigid body joints is multiplied by a factor defined on PARAM,RJSTIFF. By default, RJSTIFF = 1.0. This parameter can be used to increase or decrease the stiffness of the joints. Care must be taken by using this parameter because too high a value might lead to an unstable calculation.

7. The grid points used in the definition of the joint do not have to be rigid-body joints, but may also be finite-element grid points.

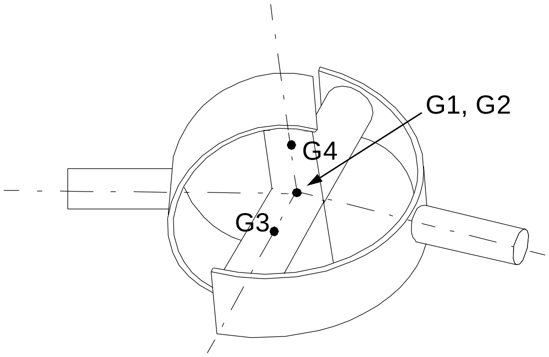
8. RJTRA can be applied to rigid bodies defined by the RIGID entry as well as to rigid bodies defined by the MATRIG or RBE2-FULLRIG entries.

Defines a universal joint between grid points on two rigid bodies.

Format and Example:

1	2	3	4	5	6	7	8	9	10
RJUNI	ID	STIFF	G1	G2	G3	G4			
RJUNI	9	1.0	47	173	53	269			

Field	Contents	Type	Default
ID	Unique joint number.	I > 0	Required
STIFF	Relative stiffness of the joint.	R	1.0
G1-G4	Grid-point numbers defining the joint connectivity.	I > 0	Required



Remarks:

1. The geometry of the joint changes during the analysis as the grid points move.

2. G1 and G3 are grid points belonging to the first rigid body; G2 and G4 are grid points belonging to the second rigid body. G1 and G2 should be coincident, while G3 and G4 cannot be coincident.

3. G3 and G4 define the orientation of the rotation of the joint, as shown in the figure above. Spring forces are calculated between G1 and G2 as in the spherical joint and between G3 and G4, based on the Pythagorean theorem.

4. The absolute stiffness of the rigid-body joints is calculated automatically by MSC.Dytran. The stiffness of the joints is taken such that a stable solution is guaranteed. This stiffness calculation takes into account the fact that a rigid body can be constrained by more than one joint.

(Continued)

5. The absolute stiffness of the rigid body joints is multiplied by a factor defined on PARAM,RJSTIFF. By default, RJSTIFF = 1.0. This parameter can be used to increase or decrease the stiffness of the joints. Care must be taken by using this parameter because too high a value might lead to an unstable calculation.

6. Although the joint is designed for usage with rigid bodies, it is allowable to use finite-element grid points.

7. RJUNI can be applied to rigid bodies defined by the RIGID entry as well as to rigid bodies defined by the MATRIG or RBE2-FULLRIG entries.

RPLEX

External Definition of a Rigid Plane

Defines a rigid plane whose properties and motion are defined by MADYMO.

Format and Example:

1	2	3	4	5	6	7	8	9	10
RPLEX	NAME	PROG	LABEL						
RPLEX	FLOOR	MADYMO	110						

Field	Contents	Type	Default
NAME	This name is used in the output request only. The name does not need to correspond to the name of the plane in the MADYMO input file.	C	Required
PROG	Name of the external program. MADYMO MSC.Dytran runs coupled with MADYMO V5.3.	C	Required
LABEL	Identification label when running coupled to MADYMO. The label is not used by DYTRAN, it is only received from MADYMO to identify the rigid planes. The value must be equal to the value used in the MADYMO input file: • FORCE MODELS • COUPLING • PLANES • LABEL system ellipsoid var1...varN • END PLANES • END COUPLING • END FORCE MODELS	I	Required

(Continued)

Remarks:

1. This entry should only be used when MSC.Dytran is used with MADYMO.
2. All planes attached to an ellipsoid in ATB are automatically visualized when the ellipsoid is asked for in an output request.
3. The mesh density of the plane can be changed by using "PARAM,MESHPLN,x", where x is the mesh density. See also PARAM,MESHPLN on page 4-498.
4. Rigid ellipsoids can be defined directly within MSC.Dytran using the RELLIPS entry.
5. RELEX and RELLIPS entries can not be mixed in the same model. A mixture of MADYMO and ATB ellipsoids is not allowed.
6. For ATB, only the segment contact ellipsoid can be used. The name of the contact ellipsoid is equal to the name of the segment, as specified on the first field of the B.2 entry in the ATB input file.
7. See Appendix D for instructions on how to use ATB.

Defines a nearly incompressible hyperelastic material for Lagrangian solid elements.

Format and Example:

1	2	3	4	5	6	7	8	9	10
RUBBER1	MID	RHO	A	B	NU				+CONT1
RUBBER1	3	1000.	0.34	0.27	0.495				+CONT1

+CONT1			BULKTYP	BULKQ	BULKL				
+CONT1									

Field	Contents	Type	Default
MID	Unique material number.	I > 0	Required
RHO	Density.	R > 0.0	Required
A	Strain-energy density function constant.	R	Required
B	Strain-energy density function constant.	R	Required
NU	Poisson’s ratio.	0.0 ≤ R < 0.5	Required
BULKTYP	Bulk-viscosity model.	C	DYNA
	DYNA Standard DYNA3D model.		
BULKQ	Quadratic bulk-viscosity coefficient.	R ≥ 0.0	1.0
BULKL	Linear bulk-viscosity coefficient.	R ≥ 0.0	0.0

Remarks:

1. The continuation line with the bulk-viscosity data can be omitted.
2. The constitutive behavior of this material is defined as a total stress/total strain relationship. The nonlinear elastic material response is formulated by a strain-energy density function for large-strain components rather than by Hooke’s law.

   The strain-energy density function is formulated according to the Mooney-Rivlin model and is defined as

$$W(I_1, I_2, I_3) = A(I_1 - 3) + B(I_2 - 3) + C\left[\frac{1}{I_3^2} - 1\right] + D(I_3 - 1)^2$$

(Continued)

The constants C and D are defined as

$$C = \frac{1}{2}A + B$$

$$D = \frac{A(5\nu - 2) + B(11\nu - 5)}{2(1 - 2\nu)}$$

where *A*, *B* and ν are input parameters.

I_1, I_2 and I_3 are strain invariants in terms of stretches.

For a rubber-like material the shear modulus G will be much less than the bulk modulus K. As a result, Poisson's ratio ν will be nearly equal to one half.

3. This material can only be used with Lagrangian solid elements.
4. The behavior of this material is discussed in more detail in Section 2.5.3.10 on page 2-49.

SECTION

Cross Section

Defines a cross section of the model for force output.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SECTION	SID	GID	EID						
SECTION	101	5	8						

Field	Contents	Type	Default
SID	Unique cross-section number.	I > 0	Required
GID	The number of a SET1 entry containing a list of grid points that define the cross section.	I > 0	Required
EID	The number of a SET1 entry containing a list of elements that define the cross section.	I > 0	Required

Remarks:

1. The cross sections for which output is required are referenced in a SET command in Case Control Section. The SET entry is referenced by the CSECS Case Control command.
2. The cross section is defined as a consecutive sequence of elements extending across the model. In addition, a consecutive sequence of grid points attached to one side of the elements must be defined. The GID field is required together with EID, the list of elements.
3. For compatibility with MSC/DYNA, the method of specifying three EIDs (i.e., one for one-dimensional elements, one for plate elements and one for hexahedral elements) is retained.
4. Cross sections cannot be defined for Eulerian models.

SET1

Set of Numbers

Defines a set of grid points, elements, etc., for use by other entries (e.g., WALL, SURFACE).

Format and Example:

1	2	3	4	5	6	7	8	9	10
SET1	ID	N1	N2	N3	N4	N5	N6	N7	+CONT1
SET1	101	7	17	32	45	8	9	22	+CONT1

+CONT1	N8	N9	THRU	N10	BY	N11	-etc.-		
+CONT1	107	221	THRU	229	BY	3			

Field	Contents	Type	Default
ID	Number of the set.	I > 0	Required
N1, N2, . . .	Numbers of the items in the set. If the word THRU appears between two numbers, all the numbers in the range are included in the list. BY indicates the increment within this THRU specification.	I > 0	Required

Remarks:

1. Use as many continuation lines as necessary.

2. If the THRU specification is used, all the items in the sequence between the beginning and the end of the range do not have to exist. Items that do not exist are ignored. BY can be used as an increment to exclude grid points.

3. SET1 Bulk Data entries with the same number are merged into one set.

SETC

List of Names

Defines a list of names (character strings) for use by other entries.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SETC	ID	V1	V2	V3	V4	-etc.-			
SETC	100	HUB	RIM	HEAD	CHEST				

Field	Contents	Type	Default
ID	Unique SETC number.	I > 0	Required
Vi	Character strings.	C	Required

Remarks:

1. Use as many continuations as required to define the complete list of names. A blank field terminates the list.

2. The SETC entry may be referred to from outside the Bulk Data Section.

SETTING

Application-Sensitive Defaults

Defines application-sensitive defaults for element formulation, element options, hourglass control and material behavior.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SETTING	SID	TYPE	PROP1	PID1	PROP2	PID2	PROP3	PID3	+CONT1
SETTING	100	CRASH	PCOMP	101	SHELL	102	SOLID	103	+CONT1

+CONT1	PROP4	PID4	PROP5	PID5	-etc.-				
+CONT1	SOLID	104	PCOMP						

Field	Contents		Type	Default
SID	Setting number.		I > 0	Required
TYPE	Application type.		C	STANDARD See Remark 1.
	STANDARD	Standard defaults.		
	CRASH	Defaults designed for crash simulations.		
	SHEETMETAL	Defaults designed for sheet metal forming analysis.		
	SPINNING	Defaults designed for fast rotating structures.		
	FAST	Defaults for fast, but not necessarily the most accurate, solution.		
	VERSION2	Defaults from MSC.Dytran V2.3.		
PROPi	Property type.		C	See Remark 2.
PIDi	Property number.		I > 0	See Remark 2.

(Continued)

Remarks:

1. The application-sensitive defaults are set according to the specification in the TYPE field. If no application type is specified, the setting will be STANDARD. The default settings concern the element formulation, element formulation options, hourglass control, material-plasticity calculation method, and strain dependency of the thickness of shell elements. See Section 2.15 on page 2-118 for a more detailed explanation on the application-sensitive defaults.

2. If no property type and property number are supplied, the setting is done for all properties in the model. If the property type and the property number are defined, the setting will apply to the elements that have the specified property. As such it is possible to define a global application setting and have a different setting for certain properties in the model.

3. See Section 2.15 on page 2-118 for more details on application-sensitive defaults.

SGAUGE

Surface Gauge

Defines a gauge on a surface for output purpose.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SGAUGE	SGID	G1	G2	G3	G4				
SGAUGE	2	12	23	24	11				

Field	Contents	Type	Default
SGID	Unique SGAUGE number.	I > 0	Required
G1-G4	Grid point number defining the connectivity of the surface gauge. For triangular gauges, G4 should be blank. All the gridpoints must be unique.	I > 0	Required

Remarks:

1. SGAUGE can only be defined on USA surface.
2. The surface gauges for which output is required are referenced in a SET command in the Case Control Section. The SET command is referenced by the SGAUGES Case Control command.

Defines the properties of an anisotropic plastic material for Lagrangian shell elements.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SHEETMAT	MID	RHO	EXX	EYY	EZZ	GXY	GYZ	GXZ	+CONT1
SHEETMAT	1	2.7E–6	72E6						+CONT1

+CONT1	NUXY	NUYZ	NUXZ	ELASTIC		XMAT	YMAT	ZMAT	+CONT2
+CONT1	0.33			ISO		1.0	0.0	0.0	+CONT2

+CONT2	a	b	c	n	k	m			+CONT3
+CONT2	0.0	570E3	0.017	0.359	0.014	0.389			+CONT3

+CONT3	TYPEYLD	R0	R45	R90					+CONT4
+CONT3	PLANANI	0.73	0.51	0.69					+CONT4

+CONT4	TYPEHRD								+CONT5
+CONT4	ISO								+CONT5

+CONT5	C1	C2	C3	C4	C5				+CONT6
+CONT5	0.244	-0.195	0.857	3.439	-11.92				+CONT6

+CONT6		D2	D3	D4	D5				
+CONT6		-0.417	-1.567	-4.849	-6.061				

Field	**Contents**	**Type**	**Default**
MID	Unique material number.	I > 0	Required
RHO	Mass density.	R > 0.0	Required
EXX, EYY, EZZ	Young's moduli in the X, Y and Z-direction (also defined as rolling, transverse and through-the-thickness directions, respectively).	R > 0.0	See Remark 2.
GXY	In-plane shear modulus.	R > 0.0	See Remark 2.

(Continued)

Field	Contents		Type	Default
GYZ, GXZ	Transverse shear moduli for shear in the YZ and XZ planes, respectively.		R > 0.0	See Remark 2.
NUXY, NUYZ, NUXZ	Poisson's ratios (coupled strain ratios in the XY, YZ and XZ directions, respectively).		R ≥ 0.0	See Remark 2.
ELASTIC	Type of elasticity.		C	ISO
	ISO	ISOtropic material.		See Remark 3.
	PLANISO	PLANar ISOtropic material.		
XMAT, YMAT, ZMAT	Vector indicating the rolling direction of the material.		R	(0., 0., 0.) See Remark 4.
a	Power-law stress constant.		R ≥ 0.0	Required See Remark 5.
b	Power-law hardening parameter.		R ≥ 0.0	0.0
c	Power-law strain offset.		R ≥ 0.0	0.0
n	Power-law, strain-hardening exponent.		R ≥ 0.0	1.0
k	Power-law, strain-rate sensitivity constant.		R ≥ 0.0	0.0 See Remark 6.
m	Power-law, strain-rate exponent.		R ≥ 0.0	1.0
TYPEYLD	Type of yielding criterion.		C	ISO
	ISO	ISOtropic yielding (von Mises).		See Remark 7.
	NORMANI	NORMal ANIsotropic yielding.		
	PLANANI	PLANar ANIsotropic yielding.		
R0, R45, R90	Anisotropic yielding parameters (Lankford parameters) defined in 0, 45, and 90 degrees with respect to the rolling direction.		R > 0.0	See Remark 8.
TYPEHRD	Type of hardening rule.		C	ISO
	ISO	ISOtropic hardening.		
	NORMANI	NORMal ANIsotropic hardening.		
C1-C5	Engineering coefficients in limit function for e2 > 0.		R	C1 = 1.0 See Remark 9.
D2-D5	Engineering coefficients in limit function for e2 < 0.		R	0.0 See Remark 9.

(Continued)

Remarks:

1. SHEETMAT materials may only be referenced by PSHELL and PSHELL1 entries.

2. The necessary number or combination of elasticity constants depends on the field ELASTIC. If ELASTIC = ISO then only EXX and NUXY (or GXY) must be defined. For ELASTIC = PLANISO, only EXX (or EYY), EZZ, NUXY (or GXY), NUXZ (or NUYZ), and GXZ (or GYZ) must be defined.

3. The field ELASTIC provides you with an input check on the consistency of the elasticity constants. Planar isotropic material is equivalent to transversely isotropic material, which means that the through-the-thickness (elastic) properties may differ from the in-plane isotropic (elastic) properties.

4. Due to anisotropic behavior, the rolling direction must be specified. The projection of the vector (XMAT, YMAT, ZMAT) on the surface of each element is used to determine the angle between the element and the material coordinate system. This angle can be overwritten using the THETA field on the CQUAD4 and CTRIA3 entries. Both the constitutive law and the output of variables are applied with respect to this material coordinate system (see Remark 10).

5. For a description of the anisotropic-plastic model including the power-law yield function, see Section 2.5.3.6. The power-law stress constant, a, is not necessarily the initial yield stress: the value of a is allowed to be equal to zero if the value of the hardening parameter, b, and the strain offset, c, are unequal to zero.

6. Strain-rate dependence is not accounted for by default.

7. The field TYPEYLD provides you with an input check on the consistency of the anisotropy parameters. Normal anisotropic material is equivalent to transversely anisotropic or planar isotropic material which means that the through-the-thickness yielding properties may differ from the in-plane, isotropic, yielding properties. Planar anisotropic material is characterized by three orthogonal axes of anisotropy (in rolling, transverse and through-the-thickness direction), about which the yielding properties have twofold symmetry.

8. The necessary number of anisotropic-yielding parameters depends on the field TYPEYLD. For TYPEYLD = ISO, all fields for R0, R45, and R90 can be left blank because the default corresponds to von Mises yielding (R0 = R45 = R90 = 1.0). For TYPEYLD = NORMANI, only R0 must be defined while the other two fields can be left blank due to their equality. The input of all three anisotropy parameters is needed for TYPEYLD = PLANANI.

9. C1 through C5 and D2 through D5 do not affect the material behavior but are used to fit the lower bound of experimental results for diffuse and localized necking represented by two polynomial lines:

   $$FLD\ (e_2) = C_1 + C_2\, e_2 + C_3\, e_2^{\,2} + C_4\, e_2^{\,3} + C_5\, e_2^{\,4} \quad \text{for } e_2 > 0$$

   $$FLD\ (e_2) = C_1 + D_2\, e_2 + D_3\, e_2^{\,2} + D_4\, e_2^{\,3} + D_5\, e_2^{\,4} \text{ for } e_2 < 0$$

(Continued)

10. The output of variables related to SHEETMAT is defined with respect to the material coordinate system (see Remark 4). There are a number of specific output variables useful for this material:

Element Variables

Q1, Q2	Direction cosines/sines between the element coordinate system and the material coordinate system.

Sublayer Variables

TXX	Stress - XX component.
TYY	Stress - YY component.
TXY	Stress - XY component.
TYZ	Stress - YZ component.
TZX	Stress - ZX component.
EFFST	Effective Stress.
EFFPL	Effective Plastic Strain.
YLDRAD	Radius of Yield Surface.
EPSXX	Strain - XX component.
EPSYY	Strain - YY component.
EPSXY	Strain - XY component.
EPSZZ	Strain - ZZ component.
EPZZ	Plastic Strain – ZZ component.
EPSMX	Strain - Major Principal Strain.
EPSMN	Strain - Minor Principal Strain.
FLP	Forming-Limit Parameter.

SHREL

Elastic Shear Model

Defines an elastic shear model with a constant shear modulus.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SHREL	SID	G							
SHREL	250	80.E6							

Field	Contents	Type	Default
SID	Unique shear model number referenced from a DMAT entry.	I > 0	Required
G	Shear-modulus value.	R	0.0

Remark:

1. Shear model numbers must be unique.

SHRLVE

Isotropic Linear Viscoelastic Shear Model

Defines an isotropic linear viscoelastic shear model where the mechanical analog is a spring, a dashpot, and a Maxwell element connected in parallel.

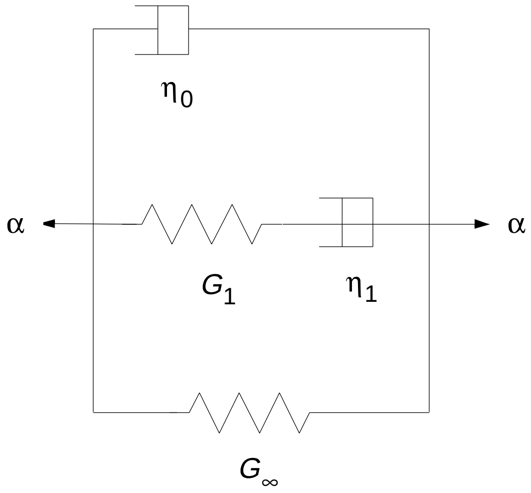
Format and Example:

1	2	3	4	5	6	7	8	9	10
SHRLVE	SID	G_0	G_∞	β	η_0				
SHRLVE	250	8.E7	1.E7	0.1					

Field	Contents	Type	Default
SID	Unique shear model number referenced from a DMAT entry.	I > 0	Required
G_0	Short-time shear modulus.	R	0.0
G_∞	Long-time shear modulus.	R	0.0
β	Decay constant.	R	0.0
η_0	Shear viscosity constant.	R	0.0

Remarks:

1. Shear model numbers must be unique.
2. The spring-damper analog of this model is



$$\text{with} \begin{cases} \beta = \dfrac{G_1}{\eta_1} \\ G_1 = G_0 + G_\infty \end{cases}$$

(Continued)

3. The deviatoric stress is given by

$$\sigma'_{ij}(t) = 2G_\infty \varepsilon'_{ij}(t) + 2\int_0^t G(t-\tau) + \frac{\partial \varepsilon'_{ij}(\tau)}{\partial \tau} d\tau + 2\eta_0 \frac{\partial \varepsilon'_{ij}(t)}{\partial t}$$

   with the relaxation function

$$G(t-\tau) = (G_0 - G_\infty)e^{-\beta(t-\tau)}$$

   The above equation for the deviatoric stress is the integral form of the differential equation

$$\dot{\sigma}'_{ij} + \beta\sigma'_{ij} = 2\eta_0\ddot{\varepsilon} + (2\eta_0\beta + 2G_0)\dot{\varepsilon} + 2G_\infty\beta\varepsilon$$

   A special case is $\eta_0 = G_\infty = 0$, for which is often written

$$\dot{\varepsilon} = \dot{\varepsilon}_{elastic} + \dot{\varepsilon}_{viscous} = \frac{\dot{\sigma}'_{ij}}{2G_0} + \frac{\beta}{2G_0}\sigma'_{ij}$$

   This shear model is further described in Section 2.5.4.2 on page 2-61.

4. A yield model cannot be used in combination with this shear model.

5. The element formulation for this material is in a corotational frame. The default CORDROT definition is G1 = 1, G2 = 5, G3 = 2. (See also DMAT and CORDROT entries.)

SPC

Single-Point Constraint

Defines sets of single-point constraints.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SPC	SID	G	C		G	C			
SPC	2	32	436		5	1			

Field	Contents	Type	Default
SID	Number of single-point constraint sets.	I > 0	Required
G	Grid-point number.	I > 0	Required
C	Component number of global coordinate (any unique combination of the digits 1 through 6 with no embedded blanks). Combinations are allowed, e.g., 23, 156.	I > 0	Required

Remarks:

1. SPC degrees of freedom may also be specified as permanent constraints on the GRID entry.
2. Continuation lines are not allowed.
3. Select single-point constraints in the Case Control Section (SPC = SID) to be used by MSC.Dytran.
4. A single-point constraint is treated as a zero-velocity boundary condition. For this reason, make SPCs consistent with other velocity boundary conditions and velocity initial conditions.

SPC1

Single-Point Constraint

Defines a single-point constraint for a set of grid points.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SPC1	SID	C	G1	G2	G3	G4	G5	G6	+CONT1
SPC1	3	2	1	3	10	9	6	5	+CONT1

+CONT1	G7	G8	THRU	G10	BY	G11	-etc.-		
+CONT1	2	8	THRU	24	BY	3			

Field	Contents	Type	Default
SID	Number of a single-point constraint.	I > 0	Required
C	Component number of global coordinate (any unique combination of the digits 1 through 6 with no embedded blanks). Combinations are allowed, e.g., 12, 456.	I > 0	Required
Gi	Grid-point numbers. THRU indicates a range of grid points. BY is the increment within this range.	I > 0	Required

Remarks:

1. As many continuation lines as desired may appear.
2. SPC degrees of freedom may be redundantly specified as permanent constraints on the GRID entry.
3. If the THRU specification is used, grid points in the sequence between the beginning and the end of the range are not required. Grid points that do not exist are ignored. BY can be used to exclude grid points within this range.
4. Single-point constraints must be selected in the Case Control Section (SPC = SID) if they are to be used by MSC.Dytran.
5. None of the fields in the list of grid points can be blank or zero, since this marks the end of the list.
6. A single-point constraint is treated as a zero-velocity boundary condition. For this reason, make SPCs consistent with other velocity boundary conditions and velocity initial conditions.

SPC2

Single-Point Constraint

Rotational boundary constraint on grid points.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SPC2	SID	G		TYPE1	VALUE1	NX	NY	NZ	+CONT1

+CONT1	TYPE2	VALUE2							+CONT2
+CONT1	CONSTANT	0.							+CONT2

+CONT2	G1	G2	THRU	G3	BY	G5	-etc-		
+CONT2	10	13	THRU	56	BY	4			

Field	Contents		Type	Default
SID	Number of a single-point constraint.		I > 0	Required
G	Grid-point number of a point on the rotation axis.		I > 0	Required
TYPE1	Defines the type of rotational constraint.		C	Required
	CONSTANT	The rotational (angular) velocity is constant at VALUE1 times the length of the rotation vector.		
	TABLE	The rotational (angular) velocity varies with time as the interpolated value in TABLED1 with number VALUE1, times the rotation vector magnitude.		
VALUE1	Value depending on TYPE1.		R or I > 0	Required
NX, NY, NZ	Rotation vector.		R	Required
TYPE2	Defines the type of radial constraint.		C	Required
	CONSTANT	The radial velocity is constant at VALUE2 where VALUE2 must be zero.		
	FREE	The radial velocity is free and determined by the forces in the direction of the radius. The VALUE2 entry is ignored.		
VALUE2	Value depending on TYPE2.		R	Required

(Continued)

Field	Contents	Type	Default
Gi	Grid-point numbers. THRU indicates a range of grid points. BY is the increment to be used within this range. (G2 < G3)	I > 0	Required

Remarks:

1. The angular velocity is specified in radians per unit time.
2. The SPC3 entry is valid for both Lagrangian as Eulerian gridpoints.
3. If the TYPE2 field is set to FREE, the referenced grid points move in a radial direction according to the acceleration caused by forces in the radial direction.
4. You can use as many continuation lines as required.
5. If the THRU specification is used, grid points in the sequence between the beginning and the end of the range are not required to exist. Grid points that do not exist are ignored.
6. Select the rotational constraints in the Case Control Section (SPC = SID) if they are to be used by MSC.Dytran.
7. None of the fields in the list of grid points can be blank or zero, since this marks the end of the list.
8. Both Lagrangian and Eulerian grid points can have a rotational constraint. In the case of Eulerian grid points, this results in a moving Eulerian mesh.
9. For 6 DOF grid points, the angular velocities are also constrained consistent with the defined velocity field.
10. The velocity in axial direction is constrained to zero.

SPC3

Single-Point Constraint

Defines a single-point constraint in a local coordinate system or a cascade of two local coordinate systems.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SPC3	SID	CID1	C1	CID2	C2				+CONT1
SPC3	1	5	12						+CONT1

+CONT1	G1	G2	THRU	G3	BY	G4	-etc.-		
+CONT1	5	6	THRU	18	BY	3			

Field	Contents	Type	Default
SID	Number of a single-point constraint.	I > 0	Required
CID1	Number of the primary coordinate system.	I > 0	See Remark 7.
C1	Constraint with respect to CID1.	I > 0	See Remark 7.
CID2	Number of the secondary coordinate system.	I > 0	See Remark 7.
C2	Constraint motion of primary coordinate system CID1 with respect to CID2.	I > 0	See Remark 7.
Gi	Grid-point numbers. THRU indicates a range of grid points. BY is the increment to be used within this range.	I > 0	Required

Remarks:

1. CID1 references the primary system. In this system the grid point constraints are applied. The CID2 system defines a secondary system that constrains the motion of the primary system and the grid points defined on the entry.
2. The SPC3 entry is valid for both Lagrangian as Eulerian gridpoints.
3. As many continuation lines as desired may appear.
4. If the THRU specification is used, grid points in the sequence between the beginning and the end of the range are not required. Grid points that do not exist are ignored. (See Remark 3 of SPC1.)
5. Select the single-point constraint in the Case Control Section (SPC = SID) if it is to be used by MSC.Dytran.

(Continued)

6. None of the fields in the list of grid points can be blank or zero, since this marks the end of the list.
7. If CID1 or CID2 is blank, the basic system is used. If C1 is blank, no constraints are applied in the primary system. If C2 is blank, no constraints are applied in the primary system with respect to the secondary system.
8. If CID1, CID2, and C2 are left blank, the constraint acts as defined by an SPC1 entry.
9. If a component references an angular velocity, the units are radians per unit time.
10. A single-point constraint is treated as a zero velocity boundary condition. For this reason, make SPCs consistent with other velocity boundary conditions and velocity initial conditions.

SPHERE

Defines the Shape of a Sphere

Spherical shape used in the initial condition definition on the TICEUL entry.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SPHERE	VID		X	Y	Z	RADIUS			
SPHERE	100		1.	1.	1.	.5			

Field	Contents	Type	Default
VID	Number of the sphere.	I > 0	Required
X, Y, Z	Coordinates of the center of the sphere.	R	0.0
RADIUS	Radius of the sphere.	R > 0	Required

SUBSURF

Multifaceted Subsurface

Defines a multifaceted subsurface for contact and coupling interfaces.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SUBSURF	SSID	SID	TYPE1	SID1	TYPE2	SID2	TYPE3	SID3	+CONT1
SUBSURF	100	100	ELEM	10	PROP	20	SEG	30	+CONT1

+CONT1	TYPE4	SID4	-etc.-						
+CONT1	MAT	100							

Field	Contents	Type	Default
SSID	Unique subsurface number.	I > 0	Required
SID	Number of a SURFACE entry of which these segments are a subsurface.	I > 0	Required
TYPEi	The type of entity used to define the subsurface.	C	Required
	SEG A set of segments defined using CSEG, CFACE, or CFACE1 entries. SIDi is the set number of the segments.		
	ELEM A set of segments attached to shell and/or membrane elements and selected by the element number. SIDi is the number of a SET1 entry containing a list of the element numbers to be used.		
	PROP A set of segments attached to shell and/or membrane elements and selected by the property number. SIDi is the number of a SET1 entry containing a list of the property numbers to be used.		
	MAT A set of segments attached to shell and/or membrane elements and selected by material number. SIDi is the number of a SET1 entry containing a list of the material numbers to be used.		

(Continued)

Field	Contents	Type	Default
SIDi	The number of a set of CSEG, CFACE, or CFACE1 entries or the number of a SET1 entry, depending on the value of TYPEi.	I > 0	Required

Remarks:

1. You can use as many continuation lines as necessary to define all of the segments in the surface.
2. CSEGs are defined indirectly using CQUAD4 and/or CTRIA3 elements with a 9999. thickness. CFACE1 are entries defined indirectly using PLOAD4 entries with a 9999. pressure. This allows CSEG and CFACE1 entries to be easily defined using standard preprocessors that can generate CQUAD4, CTRIA3, and PLOAD4 entries.
3. The subsurface SSID can be referenced from the following entries:

   SURFACE To define a surface that has the same segments as this subsurface.

   CONTINI To define the initial contact between Lagrangian surfaces. The surface SID must then be used in a CONTACT entry.

   COUPOR To define the porosity of a coupling surface. The surface SID must then be used in a COUPLE entry.

   COUOPT To define the options used in a coupling surface. The surface SID must then be used in a COUPLE entry.

Defines a multifaceted surface for contact and coupling interfaces as well as rigid-surface geometry.

Format and Example:

1	2	3	4	5	6	7	8	9	10
SURFACE	SID		TYPE1	SID1	TYPE2	SID2	TYPE3	SID3	+CONT1
SURFACE	100		ELEM	10	PROP	20	SEG	30	+CONT1

+CONT1	TYPE4	SID4	TYPE5	SID5	-etc.-				
+CONT1	MAT	100	SUB	200					

Field	Contents	Type	Default
SID	Unique surface number.	I > 0	Required
TYPEi	The type of entity used to define the surface.	C	Required

SEG — A set of segments defined using CSEG, CFACE, or CFACE1 entries. SIDi is the set number of the segments.

ELEM — A set of segments attached to shell and/or membrane elements and selected by element number. SIDi is the number of a SET1 entry containing a list of the element numbers to be used.

PROP — A set of segments attached to shell and/or membrane elements and selected by property number. SIDi is the number of a SET1 entry containing a list of the property numbers to be used.

MAT — A set of segments attached to shell and/or membrane elements and selected by material number. SIDi is the number of a SET1 entry containing a list of the material numbers to be used.

SUB — A set of segments defined by a SUBSURF entry. SIDi is the number of the SUBSURF entry.

(Continued)

Field	Contents	Type	Default
SIDi	The number of a set of CSEG, CFACE, or CFACE1 entries, the number of a SET1 entry or the number of a SUBSURF entry depending on the value of TYPEi.	I > 0	Required

Remarks:

1. You can use as many continuation lines as necessary to define all of the segments in the surface.

2. CSEGs are defined indirectly using CQUAD4 and/or CTRIA3 elements with a 9999. thickness. CFACE1 are entries defined indirectly using PLOAD4 entries with a 9999. pressure. This allows CSEG and CFACE1 entries to be easily defined using standard preprocessors that can generate CQUAD4, CTRIA3, and PLOAD4 entries.

TABLED1

Table

Defines a tabular function.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TABLED1	ID								+CONT1
TABLED1	32								+CONT1

+CONT1	X1	Y1	X2	Y2	X3	Y3	X4	Y4	+CONT2
+CONT1	-3.0	6.9	2.0	5.6	3.0	5.6	XSMALL	ENDVAL	+CONT2

+CONT2	X5	Y5	X6	Y6	X7	Y7	X8		
+CONT2	XLARGE	EXTRAP	XOFFSET	.05	YOFFSET	.04	ENDT		

Field	Contents	Type	Default
ID	Unique table number.	I > 0	Required
Xi, Yi	Tabular entries. Special entries for Xi, Yi are given in Remark 6.	R or C	0.0

Remarks:

1. The values of Xi must be in ascending or descending order but not both.
2. At least two entries must be present.
3. The end of the table is marked by the characters “ENDT” in the field following the last table entry or by a blank field.
4. The table is used according to

   $$y = f(x)$$

   where x is input to the table and y is output. Linear interpolation is used within the table to determine y. Outside the table, the last entry for y is taken.
5. Instead of a numerical value for a y entry, the keyword FREE can be entered. The value of FREE in the table can be used together with constraints and loading to switch these on and off. FREE means that the constraint or loading is not active during the time interval for which the FREE entry is defined.

(Continued)

6. Special entries can be given for Xi,Yi to specify:

   - Extrapolation outside x-range or not.
   - Offset for the x- and y-axis.
   - Scale factor for the x- and y-axis:

Xi	Yi	Meaning
XSCALE	value	x-axis of table will be multiplied by the specified value.
YSCALE	value	y-axis of table will be multiplied by the specified value.
XSCALE	value	the x-axis of table will be multiplied with the specified value.
YSCALE	value	the y-axis of table will be multiplied with the specified value.

TABLEEX

User-Defined Function

Specifies that a user routine is being used to define an arbitrary function.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TABLEEX	ID	NAME							
TABLEEX	2	MYTABLE							

Field	Contents	Type	Default
ID	Unique table number.	I > 0	Required
NAME	Name of the function (no longer than 16 characters).	C	None

Remarks:

1. The subroutine EXFUNC must be present in the file referenced by the USERCODE FMS statement.
2. See Section 3.15 on page 3-74 for a description of how to use user-written subroutines.
3. Since tables and user-defined functions belong to the same group, the table numbers must be unique.

TIC

Transient Initial Velocities of Grid Points

Defines the initial velocities of Lagrangian grid points at the beginning of the analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TIC	SID	G	C		V0				
TIC	1	3	2		-13.3				

Field	Contents	Type	Default
SID	Set number.	I > 0	Required
G	Grid-point number to be initialized.	I > 0	Required
C	Component number (a digit 1 through 6).	1 ≤ I ≤ 6	Required
V0	Initial velocity value.	R	0.0

Remarks:

1. Initial conditions for grid points that are not specified on TICn or TICGP entries are assumed to be zero.
2. Initial conditions to be used by MSC.Dytran must be selected in the Case Control Section (TIC = SID).
3. Only Lagrangian grid points can have initial conditions specified by the TIC Bulk Data entry.
4. Only one TIC entry per grid point is allowed. When more than one velocity component needs to be initialized, TICGP offers a more general way of initializing grid-point velocities.

TIC1

Transient Initial Velocities of Grid Points

Defines the initial velocities of Lagrangian grid points at the beginning of the analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TIC1	SID	C		V0	G1	G2	G3	G4	+CONT1
TIC1	3	2		3	10	9	6	5	+CONT1

+CONT1	G5	G6	THRU	G7	BY	G8	-etc.-		
+CONT1	2	8	THRU	17	BY	3			

Field	Contents	Type	Default
SID	Set number.	I > 0	Required
C	Component number (a digit 1 through 6).	1 ≤ I ≤ 6	Required
V0	Initial velocity value.	R	0.0
G1, G2, ...	Grid-point numbers to be initialized. If the word THRU appears between two numbers, all the grid points in the range are initialized. BY indicates an increment within this range.	I > 0	Required

Remarks:

1. Initial conditions for grid points that are not specified on TICn or TICGP entries are assumed to be zero.

2. Only one TIC1 entry per grid point is allowed. When more than one velocity component needs to be initialized, TICGP offers a more general way of initializing grid-point velocities.

3. If the THRU specification is used, all grid points in the sequence between the beginning and the end of the range do not have to exist. Grid points that do not exist are ignored. The first grid point in the THRU specification must be a valid grid point. BY enables grid points to be ignored in this range.

4. None of the fields in the list of grid points can be blank or zero since this marks the end of the list.

5. The initial conditions to be used by MSC.Dytran must be selected in the Case Control Section (TIC = SID).

6. Only Lagrangian grid points can have initial conditions specified by the TIC1 Bulk Data entry.

TIC2

Transient Initial Velocities of Grid Points

Defines the initial velocities of grid points consistent with a rotational field.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TIC2	SID	G		SCALE	NX	NY	NZ		+CONT1
TIC2	3	1		10.	0.1	0.2	0.3		+CONT1

+CONT1	G1	G2	THRU	G3	BY	G4	-etc.-		
+CONT1	1	2	THRU	10000	BY	23			

Field	Contents	Type	Default
SID	Number of a set of loads.	I > 0	Required
G	Number of a grid point on the axis of rotation.	I > 0	Required
SCALE	Scale factor for the rotational velocity.	R	1.0
NX, NY, NZ	Components of the rotation direction vector. The vector acts at point G.	R	Required
G1, G2, ...	Grid points to be initialized. THRU indicates a range of grid points. BY is the increment to be used within this range.	I > 0	Required

Remarks:

1. The rotational velocity *w* is calculated as

$$w = \text{SCALE} * \underline{N}$$

   where SCALE is the scale factor and $\underline{N}$ is the vector defined by NX, NY, NZ.

2. Any number of TIC2 entries can be used.
3. The rotational velocity is defined in radians per unit time.
4. For 6 DOF grid points, the angular velocities are initialized also.
5. Initial conditions for grid points that are not specified on TICn or TICGP entries are assumed to be zero.

(Continued)

6. If the THRU specification is used, all grid points in the sequence between the beginning and the end of the range do not have to exist. Grid points that do not exist are ignored. The first grid point in the THRU specification must be a valid grid point. BY enables grid points to be ignored in this range.

7. None of the fields in the list of grid points can be blank or zero, since this marks the end of the list.

8. The initial conditions to be used by MSC.Dytran must be selected in the Case Control Section (TIC = SID).

TICEEX

User-Defined Transient Initial Conditions of Elements

Defines the initial values of element variables at the beginning of the analysis by a user-written subroutine.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TICEEX	SID	SETID	NAME						
TICEEX	2	20	INEL1						

Field	Contents	Type	Default
SID	Set number.	I > 0	Required
SETID	Number of a SET1 entry defining the elements to be initialized.	I > 0	Required
NAME	Initial condition name passed to the user-written subroutine.	C	None

Remarks:

1. The subroutine EXINIT must be present and referenced in the input file by the USERCODE FMS statement.
2. See Section 3.15 on page 3-74 for a description of how to use user-written subroutines.
3. Initial conditions to be used by MSC.Dytran must be selected in the Case Control Section (TIC = SID).

TICEL

Transient Initial Conditions of Elements

Defines the initial values of element variables at the beginning of the analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TICEL	SID	SETID	NAME1	VALUE1	NAME2	VALUE2	-etc.-		
TICEL	3	40	DENSITY	100.	SIE	1.E5			

Field	Contents	Type	Default
SID	Set number.	I > 0	Required
SETID	Number of a SET1 entry defining the elements to be initialized.	I > 0	Required
NAMEi	Element variable to be initialized. See Section 3.7.2.	C	Required
VALUEi	Value of the variable.	R	Required

Remarks:

1. Initial conditions for elements that are not specified on TICEL entries are assumed to be zero except density, which is set to the reference density.
2. Initial conditions to be used by MSC.Dytran must be selected in the Case Control Section (TIC = SID).
3. As many continuation lines as required can be used to specify all the variables being initialized. A blank field terminates the list.
4. Element variables for Eulerian elements can be initialized with a TICEL or a TICEUL entry. The TICEL entry initializes a set of elements, while the TICEUL entry initializes either a set of elements or geometrical regions (sphere, cylinder,...). When a Euler element is part of both a TICEL and a TICEUL entry, the TICEL entry takes precedence, and overrules the TICEUL initialization for the element.

TICEUL

Transient Initial Conditions of Eulerian Regions

Defines the initial value sets for Eulerian regions. The Eulerian regions are defined by geometric shapes.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TICEUL	SID								+CONT1
TICEUL	300								+CONT1

+CONT1	TYPE1	VID1	MID1	TSID1	LEVEL1				+CONT2
+CONT1	SPHERE	400	100	3	4.0				+CONT2

+CONT2	TYPE2	VID2	MID2	TSID2	LEVEL2				+CONT3
+CONT2	ELEM	500	200	4	2.1				+CONT3

+CONT3	TYPEi	VIDi	MIDi	TSIDi	LEVELi	-etc.-			
+CONT3	CYLINDER	300	300	5	1.0				

Field	Contents	Type	Default
SID	Unique TICEUL number referenced from a PEULER1 entry.	I > 0	Required
TYPEi	The type of Eulerian region. SPHERE Region inside a sphere. CYLINDER Region inside a cylinder. ELEM Region defined by element list.	C	Required
VIDi	Number of a geometric entity, a SET1 number, or number of a SURFACE entry.	I > 0	Required
MIDi	Number of a DMAT entry defining this material.	I > 0	Required
TSIDi	Unique TICVAL number containing a list of initial values for this material.	I > 0	Required
LEVELi	Level indicator for this material and initial values.	R	0.0

(Continued)

Remarks:

1. When the material number is left blank or zero, the Eulerian elements inside the region will be void.

2. All level indicators LEVELi must have different values. The level indicator can be negative.

3. See also the parameter MICRO for the accuracy of the initial value generation.

4. See Section 2.8.4 on page 2-992 for instructions on how to use the geometric shapes on the TICEUL entry for arbitrary initial value generation in Eulerian regions.

5. Element variables for Eulerian elements can be initialized with a TICEL or a TICEUL entry. The TICEL entry initializes a set of elements, while the TICEUL entry initializes either a set of elements or geometrical regions (sphere, cylinder,...). When a Euler element is part of both a TICEL and a TICEUL entry, the TICEL entry takes precedence and overrules the TICEUL initialization for the element.

TICGEX

User-Specified Transient Initial Conditions of Grid Points

Defines the initial values of grid-point variables at the beginning of the analysis by a user-written subroutine.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TICGEX	SID	SETID	NAME						
TICGEX	4	40	INGP3						

Field	Contents	Type	Default
SID	Set number.	I > 0	Required
SETID	Number of a SET1 entry defining the grid points to be initialized.	I > 0	Required
NAME	Initial condition name passed to the user-written subroutine.	C	None

Remarks:

1. The subroutine EXINIT must be present in the input file, and it must be referenced by the USERCODE FMS statement.
2. See Section 3.15 on page 3-74 for a description of how to use user-written subroutines.
3. Initial conditions must be selected in the Case Control Section (TIC = SID) to be used by MSC.Dytran.

TICGP

Transient Initial Conditions for Grid Points

Defines the initial conditions of grid points at the beginning of the analysis.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TICGP	SID	SETID	NAME1	VALUE1	NAME2	VALUE2	-etc.-		
TICGP	3	30	PMASS	100.0	YVEL	30.0			

Field	Contents	Type	Default
SID	Transient initial condition set number.	I > 0	Required
SETID	Number of a SET1 entry listing the grid points to be initialized.	I > 0	Required
NAMEi	Grid-point variable to be initialized (see Section 3.9.2 on page 3-24) or CID1, CID2 (see Remark 4).	C	Required
VALUEi	Value of the grid point variable, or number of coordinate system CID1, CID2 (see Remark 4).	I or R	Required

Remarks:

1. Initial conditions for grid-point components that are not specified on TICn or TIGGP entries are assumed to be zero.

2. Select initial conditions to be used by MSC.Dytran in the Case Control Section (TIC = SID).

3. Use as many continuation lines as required to specify all the variables being initialized. A blank field terminates the list.

4. The NAMEi on the TICGP entry can also be CID1 or CID2. In that case, VALUEi denotes the number of a defined coordinate system. Velocities are initialized according to the type of defined coordinate system. If coordinate systems are used, the velocity components must follow the CID definition immediately. All other variables must be defined before the first CID definition. Only for Lagrangian grid points the velocity can be defined in a local coordinate system.

   For example:

   ```
   TICGP, 1, 1,PMASS,10.,CID1,1,YVEL,10.
   ```

5. All velocity components defined and preceding a coordinate system reference are overruled by the definition following the coordinate system reference.

TICVAL

Transient Initial Condition Set

Defines the initial values of an Eulerian geometric region.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TICVAL	TSID		NAME1	VALUE1	NAME2	VALUE2	NAME3	VALUE3	+CONT1
TICVAL	3		DENSITY	100.	YVEL	25.	SIE	3.7	+CONT1

+CONT1	NAMEi	VALUEi	-etc.-						
+CONT1	XVEL	3.5							

Field	Contents	Type	Default
TSID	Unique TICVAL number referenced from a TICEUL entry.	I > 0	Required
NAMEi	Variable to be initialized. See Section 3.9.2 on page 3-24.	C	Required
VALUEi	Value of the variable.	R	Required

Remarks:

1. Initial conditions for geometric regions that are not specified on TICVAL entries are assumed to be zero except density, which is set to the reference density.
2. As many continuation lines as required can be used to specify all the variables to be initialized. A blank field terminates the list.

Defines a transient dynamic load, enforced motion, or an Eulerian boundary condition.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TLOAD1	SID	LID		TYPE	TID				
TLOAD1	5	7			13				

Field	Contents		Type	Default
SID	Load number.		I > 0	Required
LID	Number of a set of loads (DAREA, FORCEn, RFORCE, MOMENT, GRAV, PLOADn, FLOW, FLOWEX, MOMENTn that defines the loading type, position, and value.		I > 0	Required
TYPE	Nature of the dynamic excitation.		I	0
	0	Force on a grid point.		
		Pressure on a Lagrangian element.		
		GRAV applied to model.		
		RFORCE applied to model.		
		ATBACC applied to ATB segments.		
	2	Velocity of a Lagrangian or ALE (Eulerian) grid point.		
	4	FLOW boundary condition.		
	12	Velocity of a rigid body.		
	13	Force on a rigid body.		
TID	Number of a TABLED1 or TABLEEX entry defining the variation of load with time or by means of a user routine. If blank or zero, the loads do not vary with time.		I ≥ 0	No time variation.

Remarks:

1. See the FORCEn, MOMENTn, DAREA, PLOADn, GRAV, RFORCE, ATBACC, FLOW, FORCEEX, and FLOWEX, entries for a description of how the loading or motion is calculated.

2. There can be one or more TLOAD1 entries in a set.

(Continued)

3. Transient loads to be used by MSC.Dytran must be selected in the Case Control Section (TLOAD = SID).

4. TID must be blank if it references a FLOW or FLOWEX entry.

5. If TYPE is 0, the LID field can reference any of the entries: FORCEn, MOMENTn, GRAV, RFORCE, DAREA, or PLOADn and will apply the appropriate type of load.

   If TYPE is 2, the LID field can only reference DAREA, FORCE, MOMENT, FORCE3, or FORCEEX entries and will apply enforced velocity to the specified grid points.

   If TYPE is 4, the LID field can only reference FLOW or FLOWEX entries and will apply a flow boundary condition to the specified Eulerian faces.

   If TYPE is 12, the LID field can only reference the DAREA, FORCE, or MOMENT entries and will apply an enforced velocity to the center of the specified rigid body.

   If TYPE is 13, the LID field can only reference the FORCE or MOMENT entries and applies a force or moment to the center of the specified rigid body.

6. If more than one velocity boundary condition (TYPE = 2) is applied to a grid point, the boundary conditions can only be merged when the boundary conditions are consistently defined.

TLOAD2

Transient Dynamic Load, Form 2

Defines a transient dynamic load or enforced motion of the following form:

$$Y(t) = 0 \text{ for } \bar{t} < 0 \text{ or } \bar{t} > T_2 - T_1$$

$$Y(t) = A\bar{t}^{-B} e^{c\bar{t}} \cos\left(2\pi F t + \frac{\pi P}{180}\right) \text{ for } 0 \le \bar{t} \le T_2 - T_1$$

where $\bar{t} = t - T_1$, and t is the analysis time.

Format and Example:

1	2	3	4	5	6	7	8	9	10
TLOAD2	SID	LID		TYPE	T1	T2	F	P	+CONT1
TLOAD2	5	7		2	0.	10.E–3	1000.	90.	+CONT1

+CONT1	C	B							
+CONT1	.0	2.							

Field	Contents	Type	Default
SID	Set number.	I > 0	Required
LID	Number of a set of loads (DAREA, FORCEn, MOMENTn, PLOAD, GRAV, RFORCE) that defines the loading type, position, and scale factor A.	I > 0	Required
TYPE	Nature of the dynamic excitation.	I	0
	0 Force on a grid point. Pressure on a Lagrangian element. GRAV applied to model. RFORCE applied to model. ATBACC applied to ATB segments. 2 Velocity of a Lagrangian grid point. 12 Velocity of a rigid body. 13 Force on a rigid body.		
T1	Time constant.	R	0.0
T2	Time constant (T2 > T1).	R	0.0

(Continued)

TLOAD2

Transient Dynamic Load, Form 2

Field	Contents	Type	Default
F	Frequency in cycles per unit time.	R ≥ 0.0	0.0
P	Phase angle in degrees.	R	0.0
C	Exponential coefficient.	R	0.0
B	Growth coefficient.	R	0.0

Remarks:

1. See the FORCEn, MOMENTn, DAREA, PLOADn, GRAV, RFORCE, ATBACC, FLOW, FORCEEX, and FLOWEX entries for a description of how the loading or motion is calculated.

2. There can be one or more TLOAD1 and TLOAD2 entries in a set.

3. Select transient loads to be used by MSC.Dytran in the Case Control Section (TLOAD = SID).

4. If TYPE is 0, the LID field can reference any of the entries: FORCEn, MOMENTn, DAREA, PLOAD, GRAV, or RFORCE and applies the appropriate type of load.

   If TYPE is 2 or 3, the LID field can only reference DAREA, FORCE, MOMENT, or FORCEEX entries and applies enforced velocity or acceleration to the specified grid points.

   If TYPE is 12, the LID field can only reference the DAREA, FORCE, or MOMENT entries and applies an enforced motion to the center of gravity of the specified rigid bodies.

   If TYPE is 13, the LID field can only reference the FORCE or MOMENT entries and applies a force or moment to the center of the specified rigid body.

5. If more than one velocity boundary condition (TYPE = 2) is applied to a grid point, the boundary conditions are constant velocity boundary conditions and are consistently defined.

USA

Interface Between MSC.Dytran and USA

Defines the interface between MSC.Dytran and USA.

Format and Example

1	2	3	4	5	6	7	8	9	10
USA	ID	SID	SET1ID						
USA	1	2	44						

Field	Contents	Type	Default
ID	Unique number of an USA entry.	I > 0	Required
SID	Number of a SURFACE entry defining the USA surface.	I > 0	Remark 3.
SET1ID	Number of a SET1 entry defining the USA 1D element type of gridpoints.	I > 0	Remark 3.

Remarks:

1. Only one USA surface is allowed.
2. By defining an USA surface the run will stop after cycle zero and writes out a restart file and a preprocessing file for USA (See Appendix G). How to use the interface of USA in combination with MSC.Dytran see Section 2.22 on page 2-137.
3. A surface and/or set of 1D element type of gridpoints must be defined.

VISCDMP

Dynamic Relaxation

Defines the dynamic relaxation for the various types of Lagrangian elements, rigid bodies, and ellipsoids.

Format and Example:

1	2	3	4	5	6	7	8	9	10
VISCDMP	SOLST	SOLEND		SOLV1					+CONT1
VISCDMP									+CONT1

+CONT1	SHST	SHEND		SHV1					+CONT2
+CONT1									+CONT2

+CONT2	MEMST	MEMEND		MEMV1		MEMV2			+CONT3
+CONT2									+CONT3

+CONT3	EL1DST	EL1DEND		EL1DV1					+CONT4
+CONT3	0.	10.E–3		0.01					+CONT4

+CONT4	RIGST	RIGEND		RIGV1					+CONT5
+CONT4	0.	10.E–3		0.05					+CONT5

+CONT5	ELLST	ELLEND		ELLV1					
+CONT5	0.	10.E–3		0.06					

Field	Contents	Type	Default
SOLST	Start time for solid-element dynamic relaxation.	R ≥ 0	0.0
SOLEND	End time for solid-element dynamic relaxation.	R ≥ 0	1.E20
SOLV1	Dynamic relaxation factor for grid points of solid elements.	R ≥ 0	0.0
SHST	Start time for shell-element dynamic relaxation.	R ≥ 0	0.0
SHEND	End time for shell-element dynamic relaxation.	R ≥ 0	1.E20
SHV1	Dynamic relaxation factor for grid points of shell elements.	R ≥ 0	0.0
MEMST	Start time for membrane-element dynamic relaxation.	R ≥ 0	0.0

(Continued)

Field	Contents	Type	Default
MEMEND	End time for membrane-element dynamic relaxation.	R ≥ 0	1.E20
MEMV1	Dynamic relaxation factor for grid points of membrane elements.	R ≥ 0	0.0
MEMV2	Dynamic relaxation factor for membrane-element stiffness.	R ≥ 0	0.0
EL1DST	Start time for one-dimensional element dynamic relaxation.	R ≥ 0	0.0
EL1DEND	End time for one-dimensional element dynamic relaxation.	R ≥ 0	1.E20
EL1DV1	Dynamic relaxation factor for grid points of one-dimensional elements.	R Š≥ 0	0.0
RIGST	Start time for rigid-body dynamic relaxation.	R ≥ 0	0.0
RIGEND	End time for rigid-body dynamic relaxation.	R ≥ 0	1.E20
RIGV1	Dynamic relaxation factor for the rigid-body masses.	R ≥ 0	0.0
ELLST	Start time for ellipsoid dynamic relaxation.	R ≥ 0	0.0
ELLEND	End time for ellipsoid dynamic relaxation.	R ≥ 0	1.E20
ELLV1	Dynamic relaxation factor for the ellipsoid masses.	R ≥ 0	0.0

Remarks:

1. A dynamic relaxation factor defined for a certain element type applies to all elements of that type present in the problem.
2. See also Section 2.12 on page 2-109 for general information on dynamic relaxation in MSC.Dytran.

WALL

Lagrangian Rigid Wall

Defines a rigid plane through which specified Lagrangian gridpoints cannot penetrate.

Format and Example:

1	2	3	4	5	6	7	8	9	10
WALL	ID	XP	YP	ZP	NX	NY	NZ	SET	+CONT1
WALL	17						1.0	21	

+CONT1	METHOD	FS	FK	EXP					
	PENALTY	0.2							

Field	Contents		Type	Default
ID	Unique rigid-wall number.		I > 0	Required
XP, YP, ZP	Coordinates of the origin of the wall.		R	0.0
NX, NY, NZ	A vector normal to the wall pointing towards the model.		R	0.0
SET	Number of a SET1 entry listing the points that can not penetrate the wall.		I > 0	Required
METHOD	Algorithm for contact processing.		C	PENALTY
	PENALTY	penalty method, allowing for extra boundary conditions, friction and output.		
	KINMATIC	kinematic method, only included for compatibility reasons with older versions. This method allows no extra boundary conditions, no friction and no output.		
FS	Static coefficient of friction (see Remark 5).		R ≥ 0	0.0
FK	Kinetic coefficient of friction (see Remark 5).		R ≥ 0	0.0
EXP	Exponential decay coefficient (see Remark 5).		R ≥ 0	0.0

(Continued)

Remarks:

1. A rigid plane of infinite size is generated that the gridpoints can not penetrate. The plane is fixed in space.
2. The gridpoints can slide on the wall and separate from it.
3. A (moving) rigid plane of finite size can be modeled by using a rigid surface and a master-slave contact.
4. For the wall definition using penalty method, output can be requested by referencing it in a SET command in the Case Control Section. The keywords for output are WALLS and WALLOUT.
5. The coefficient of friction is given by:

$$\mu = \mu_k + (\mu_s - \mu_k)e^{-\beta v}$$

where μ_s = static coefficient of friction FS

μ_k = kinetic coefficient of friction FK

β = exponential decay coefficient EXP

v = relative sliding velocity at the point of contact

WALLET

Barrier for Eulerian Transport

Defines a barrier for transport in an Eulerian mesh.

Format and Example:

1	2	3	4	5	6	7	8	9	10
WALLET	WID	SID							
WALLET	100	20							

Field	Contents	Type	Default
WID	Unique wall number.	I > 0	Required
SID	Number of a set of CSEG, CFACE, and CFACE1 entries that define the element faces that are barriers to Eulerian transport.	I > 0	Required

Remarks:

1. Material cannot pass through any of the faces referenced by the SID field.
2. Barriers can be modeled on the outside as well as the inside of an Eulerian mesh.
3. See Section 2.8.3 on page 2-98 for a more detailed description of the use of Eulerian barriers.

YLDEX

User-Specified Yield Model

Specifies that a user subroutine is being used to define the yield model.

Format and Example:

1	2	3	4	5	6	7	8	9	10
YLDEX	YID								
YLDEX	200								

Field	Contents	Type	Default
YID	Unique yield model number.	I > 0	Required

Remarks:

1. The subroutine must be present in the file referenced by the USERCODE FMS statement.
2. See Section 3.15 on page 3-74 for a description of how to use user-written subroutines.
3. This yield model is applicable only for Lagrangian solid elements and Eulerian elements with shear strength.

YLDHY

Hydrodynamic Yield Model

Defines a yield model with zero yield stress.

Format and Example:

1	2	3	4	5	6	7	8	9	10
YLDHY	YID								
YLDHY	200								

Field	Contents	Type	Default
YID	Unique yield-model number referenced from a DMAT entry.	I > 0	Required

Remark:

1. This yield model should be used for fluids that have no shear strength.

Defines a Johnson-Cook yield model where the yield stress is a function of effective plastic strain, strain rate, and temperature.

Format and Example:

1	2	3	4	5	6	7	8	9	10
YLDJC	YID	A	B	n	C	m	EPS0	CP	+CONT1
YLDJC	100	200E6	50.E6	0.1	.15	.95	1.	285.	+CONT1

+CONT1	TMELT	TROOM							
+CONT1	1500.	273.							

Field	Contents	Type	Default
YID	Unique yield-model number referenced from a DMAT or DMATEP entry.	I > 0	Required
A	Static yield stress.	R ≥ 0.0	Required
B	Hardening parameter.	R	0.0
n	Hardening exponent.	R	1.0
C	Strain-rate parameter.	R	0.0
m	Temperature exponent.	R	1.0
EPS0	Reference strain rate.	R > 0.0	1.0
CP	Specific heat.	R > 0.0	1.E20
TMELT	Melt temperature.	R	1.E20
TROOM	Room temperature.	R	293.0

Remarks:

1. This yield model is described in Section 2.5.5.3 on page 2-73.

(Continued)

The yield stress is computed from

$$\sigma_y = (A + B\varepsilon_p^n)\left(1 + C\ ln\left(\frac{\dot{\varepsilon}}{\dot{\varepsilon}_0}\right)\right)(1 - T^{*m})$$

where ε_p = effective plastic strain

T^* = $\dfrac{(T - T_r)}{(T_m - T_r)}$

$\dot{\varepsilon}$ = effective strain rate

$\dot{\varepsilon}_0$ = reference strain rate

T = temperature

T_r = room temperature

T_m = melt temperature

and *A*, *B*, *n*, *C*, and *m* are constants.

2. The reference strain rate is per unit time.

Defines a Mohr-Coulomb yield model.

Format and Example:

1	2	3	4	5	6	7	8	9	10
YLDMC	YID	Y1	Y2	Y3					
YLDMC	1	10.E5	20.E5	1.E4					

Field	Contents	Type	Default
YID	Unique yield-model number referenced from: • DMAT for Eulerian elements with shear strength. • DMATEP for shell elements.	I > 0	Required
Y1, Y2, Y3	Yield parameters.	R	Required

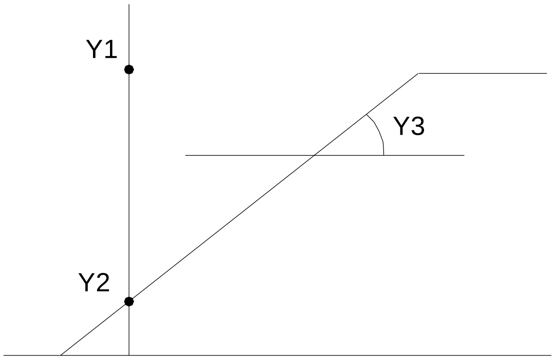
Remarks:

1. For a description of the yield models, see Section 2.5.5 on page 2-68.

   The yield stress depends on the pressure as

   $$\sigma_y = MIN(\mathrm{Y1}, (\mathrm{Y2} + \mathrm{Y3} * P))$$

   where Y_1, Y_2, Y_3 are constants and P is the pressure.



2. This yield model is applicable only for Eulerian materials with shear strength.

YLDVM

von Mises Yield Model

Defines a bilinear or piecewise-linear yield model with isotropic hardening, using the von Mises yield criterion.

Format and Example:

1	2	3	4	5	6	7	8	9	10
YLDVM	YID	YIELD	EH						+CONT1
YLDVM	32	250.E6	2000.E6						+CONT1

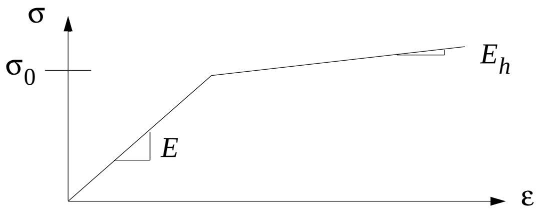
+CONT1	TABLE	TYPE	TABY	D	P				
+CONT1									

Field	Contents		Type	Default
YID	Unique yield-model number.		I > 0	Required
YIELD	Yield stress.		R	Required
EH	Hardening modulus.		R	0.0
TABLE	Number of a TABLED1 entry giving the variation of effective stress (y-value) with effective strain (x-value).		I > 0	See Remark 5.
TYPE	The type of stress and strain defined in TABLED1.		C	TRUE
	ENG	Engineering stress and strain.		
	TRUE	True stress and strain.		
	PLAST	True stress and plastic strain.		
	PMOD	Plastic modulus and true stress.		
TABY	Number of TABLED1 entry giving the variation of the scale factor for the yield stress (y-value) with the strain rate (x-value). Strain-rate effects are also specified using the Cowper-Symonds relation (see input parameters D and P).		I > 0	See Remark 7.
D	Factor D in the Cowper-Symonds rate enhancement equation.		R ≥ 0	See Remark 7.
P	Factor P in the Cowper-Symonds rate enhancement equation.		R ≥ 0	See Remark 7.

(Continued)

Remarks:

1. A bilinear stress-strain characteristic is used by specifying YIELD and EH:



where the yield stress σ_y is given by

$$\sigma_y = \sigma_0 + \frac{E E_h}{E - E_h}\varepsilon_p$$

where σ_0 = yield stress specified in the YIELD field

E = Youngs modulus

E_h = hardening modulus specified in the EH field

ε_p = equivalent plastic strain

σ_y = yield stress

2. A piecewise linear, stress-strain characteristic is used by specifying TABLE and TYPE (beams and shells only)

$$\sigma_{ij} = [(\sigma_i - \sigma_{i-1})(\varepsilon - \varepsilon_{i-1})/(\varepsilon_i - \varepsilon_{i-1})] + \sigma_{i-1}$$

The stress-strain characteristic used internally in MSC.Dytran is in terms of true stress and equivalent plastic strain. However, for convenience, the stress-strain characteristic can be input in any of the following ways (see Section 2.5.5.2 on page 2-68):

True stress/true strain	(TYPE = TRUE)
Engineering stress/engineering strain	(TYPE = ENG)
True stress/plastic strain	(TYPE = PLAST)
Plastic modulus/true stress	(TYPE = PMOD)

3. For a description of all of the yield models, see Section 2.5.5 on page 2-68.

(Continued)

4. With Lagrangian and Eulerian solid elements, only an elastic perfectly plastic yield model is currently used. Only the YIELD field is used.

5. If TABLE is blank or zero, a bilinear stress-strain curve is assumed. If TABLE has a value, it refers to a TABLED1 entry giving the stress-strain curve for the material.

6. If TABLE is defined, the value of YIELD is left blank, since it is determined from the stress-strain curve.

7. If TABY is blank or zero and D and P are blank or zero, the yield stress does not vary with strain rate. If TABY has a value, then it references a TABLED1 entry, which gives the variation of the scale factor applied to the yield stress with strain rate. (D and P must be blank or zero.)

   If TABY is blank or zero and D and P are defined, the enhancement of the yield stress with strain rate is calculated as

$$\frac{\sigma_d}{\sigma_y} = 1 + \left(\frac{\dot{\varepsilon}_P}{D}\right)^{1/p}$$

   Where σ_d is the dynamic stress, σ_y is the static yield stress (YIELD), and $\dot{\varepsilon}_P$ is the equivalent plastic strain rate.

8. If TYPE is PLAST or PMOD, Young’s modulus must be defined. If TYPE is ENG or TRUE and Young’s modulus is defined it will override the value calculated from the stress-strain curve. See Section 2.5.5.2 on page 2-68 for more details.

9. Note that for values exceeding the maximum x-value of either of the TABLED1 entries (see TABLE and TABY fields), linear extrapolation is used based upon the last two points specified in the TABLED1.

4.7 Parameter Options

4.7.1 Introduction

The PARAM entry in the Bulk Data Section of the input file is used to change a number of the values that control the analysis. This section describes all possible options and values that can be set with the PARAM entry.

All the options are set to default values if no PARAM entry with that option is present in the input file. You therefore only need to include a PARAM entry if you want to change one of the defaults. Normally, the default values work well and need not be altered.

4.7.2 Summary

The following PARAM options are available:

Contact Control

LIMCUB	Contact cube-sort algorithm.

Coupling Subcycling

COSUBCYC	Growth of subcycling in coupling.
COSUBMAX	Subcycle limit.

Blending Control

DELCLUMP	Delete clump factor.
FBLEND	Blend fraction.

Time-Step Control

INISTEP	Initial time step.
MAXSTEP	Maximum time step.
MINSTEP	Minimum time step.
STEPFCT	Time-step scale factor.

Limits

FMULTI	Dimensioning of multimaterial overflow array.
MICRO	Microzoning parameter.
RHOCUT	Global minimum density for Eulerian elements.
ROHYDRO	Minimum density for hydrodynamic, single-material Eulerian elements.
ROMULTI	Minimum density for multimaterial Eulerian elements.
ROSTR	Minimum density for single material Eulerian elements with strength.
SNDLIM	Minimum value of speed of sound.
VELCUT	Velocity cutoff.
VELMAX	Maximum velocity for Eulerian and Lagrangian elements.

Restart Control

RSTDROP	Type of elements dropped when restarting.

ALE Motion Control

ALEITR	Number of ALE grid iterations.
ALETOL	Tolerance at ALE interface.
ALEVER	ALE Volume Computation Method.

Hourglass Suppression Control

HGCMEM	Shell membrane hourglass damping coefficient.
HGCOEFF	Hourglass damping coefficient.
HGCSOL	Solid hourglass damping coefficient.
HGCTWS	Shell twisting hourglass damping coefficient.
HGCWRP	Shell warping hourglass damping coefficient.
HGSHELL	Shell hourglass suppression method.
HGSOLID	Solid hourglass suppression method.
HGTYPE	Global hourglass suppression method.

Miscellaneous

CFULLRIG	Converts 123456 constraints to FULLRIG for RBE2s.
CONTACT	Sets the contact defaults equal to MSC/DYNA defaults.
EFMETH	Eulerian face generation method.
EULTRAN	Switch for the multimaterial Euler transport scheme.
EXTRAS	Extra input constants.
FSMETH	Face-Seg method.
FASTCOUP	Fast coupling algorithm.
GEOCHECK	Defines geometry consistency check.
LIMITER	Defines type of scheme used in the Euler solver.
MATRMERG	Merges MATRIG and RBE2-FULLRIG assemblies.
MATRMRG1	Merges MATRIG and RBE2-FULLRIG assemblies.
PARALLEL	Defines use of parallel processing capability.
PLCOVCUT	Defines time that PLCOVER will be cut.
RBE2INFO	Lists MATRIG and RBE2 grids in the output file.
RJSTIFF	Rigid-joint stiffness.
RKSCHEME	Defines the type of time scheme used in the Euler solver.
VARACTIV	Activation or deactivation of grid-point, element, or face variables.

Material Parameter Control

BULKL	Linear bulk-viscosity coefficient.
BULKQ	Quadratic bulk-viscosity coefficient.
BULKTYP	Bulk-viscosity type.
HVLFAIL	Switches failure at hydrodynamic volume limit.
PMINFAIL	Switches failure at spall limit.

Shell Options

SHELMSYS	Shell-element system definition.
SHPLAST	Type of plane-stress plasticity for shells.
SHTHICK	Shell-thickness modification option.
SHELLFORM	Sets the default shell formulation.

Element Subcycling

ELSUBCHK	Efficiency check on element subcycle groups.
ELSUBCYC	Element subcycling.
ELSUBDAC	Deactivation of element subcycling.
ELSUBMAX	Defines maximum number of groups in element subcycling.
ELSUBRGP	Redefinition of element subcycling groups.
ELSUBRRG	Redefinition of element subcycling groups at restart.

Dynamic Relaxation

VDAMP	Defines dynamic-relaxation parameter.

ATB Positioning

ATBSEGCREATE	Creates grids and elements for ATBSEG.
ATB-H-OUTPUT	Writes ATB output to MSC.Dytran time-history files.

Output Control

ATBAOUT	Frequency of output written to main outputfile of ATB.
ATBTOUT	Frequency of output written to the time-history files of ATB.
AUTHINFO	Licensing information control.
CONM2OUT	CONM2 summary output.
ERRUSR	Error message redefinition.
FAILOUT	Failed element output parameter.
IEEE	IEEE binary data output format.

IGNFRCER	Ignores warnings.
INFO-BJOIN	Lists the generated BJOINs and spotwelds.
MESHELL	Mesh density for covering rigid ellipsoids.
MESHPLN	Mesh density for covering of rigid planes.
NASIGN	Echo ignored data entries.
SLELM	Store shell sublayer variables.
STRNOUT	Shell sublayer strain output.
SHSTRDEF	Composite shell stress and strain output.

Prestressing Analysis

INITFILE	Defines method of initialization from a solution file.
INITNAS	Defines the type of displacement initialization file.

4.7.3 PARAM Descriptions

The parameter options are listed in alphabetical order. The entry and the examples are shown in free format, although small or large fixed-format entries can also be used.

The default value indicates the value that is used if no PARAM entry with the option is present. The type column indicates the type of data you must supply. This can be I (Integer), R (Real), or C (Character). In addition, a range of permissible values may also be indicated for example, I > 0 means that you must supply an integer that is greater than zero.

ALEITR

Number of ALE Mesh Iterations

Defines the number of mesh iterations to be used to move the free ALE grid points.

Format and Example	Default
PARAM,ALEITR,value	1
PARAM,ALEITR,3	

Option	Meaning	Type
value	Number of mesh iterations.	1 < I < 6

Remarks:

1. In most applications, one iteration should be sufficient. If not, the number of iterations can be increased to a maximum of six.

2. Less mesh iterations reduce the computational effort.

ALETOL

Tolerance at ALE Interface

Defines the tolerance in matching Eulerian and Lagrangian grid points in the ALE-interface surface.

Format and Example	Default
PARAM,ALETOL,value	1.E–4
PARAM,ALETOL,1.E–4	

Option	Meaning	Type
value	Tolerance.	R > 0

Remark:

1. Grid points in the ALE interface with coordinates that fall within the tolerance are recognized to be an ALE-interface pair.

4

ALEVER

ALE Volume Computation Method

Defines the method to be used in the element-volume computation in ALE meshes.

Format and Example	Default
PARAM,ALEVER,option	V2.1
PARAM,ALEVER,V2.2	

Option	Meaning	Type
V2.1	V2.1 method uses a fast approximation scheme for the element-volume computation.	C
V2.2	V2.2 method gives the exact element volume.	

Remark:

1. The V2.2 option should be used in problems where the pressure levels are expected to be low. The V2.1 method is faster and consumes less CPU time, but it can lead to spurious pressure levels in a low pressure level calculation.

ATB-H-OUTPUT

Write ATB Output to MSC.Dytran Time-History Files

An MSC.Dytran time-history file will be created containing the output as requested in the ATB input file on cards H.1 to H.11.

Format and Example

PARAM,ATB-H-OUTPUT,[YES/NO]

PARAM,ATB-H-OUTPUT,NO

Option	Meaning		Type	Default
YES/NO	YES	The MSC.Dytran time-history files will be created.	C	YES
	NO	The MSC.Dytran time-history files will not be created.		

ATBAOUT

Output Frequency to Main Output File of ATB

Defines the frequency at which output is written to the main output file of ATB..

Format and Example	Default
PARAM,ATBAOUT,value	10.0E-3
PARAM,ATBAOUT,5.0E-3	

Option	Meaning	Type
value	Every multiple of ATBAOUT seconds, the main output file of ATB will be updated.	R > 0.0

Remark:

1. Only active when field 3 on the A5 card of the ATB input file is set to a value of -1.
2. Controls the frequency of the output of segment acceleration, velocity and displacement, joint forces and moments.

ATBTOUT

Output Frequency to Time-History Files of ATB

Defines the frequency at which output is written to the time-history files of ATB..

Format and Example	**Default**
PARAM,ATBTOUT,value	1.0E-3
PARAM,ATBTOUT,1.0E-4	

Option	**Meaning**	**Type**
value	Every multiple of ATBTOUT seconds, the time-history files of ATB will be updated.	R > 0.0

Remark:

1. Only active when field 26 on the A5 card of the ATB input file is set to a value of –1.
2. Controls the frequency of all output requested on the H-cards, and of the tabular time-histories that are controlled by field 18 on the A5 card of the ATB input file.

ATBSEGCREATE

Create Grids and Elements for ATBSEG

A Bulk Data file will be created containing grid points and elements visualizing the initial position and orientation of the coordinate systems of the ATB segment and its joints.

Format and Example

PARAM,ATBSEGCREATE,[YES/NO],NAME,LENGTH1,LENGTH2

PARAM,ATBSEGCREATE,YES,HYBRID-III,0.1,0.05

Option	Meaning		Type	Default
YES/NO	YES	If EID1 through EID3 on the ATBSEG entry and/or EID1 through EID6 on the ATBJNT entry are defined, MSC.Dytran will extract the initial positions from the ATB input file for the grid points G0 through G3 from the ATBSEG entry and/or for the grid points G0 through G6 from the ATBJNT entry. Bulk Data entries as specified on the ATBSEG and ATBJNT entries will be written to the file with name ATB_<NAME>.DAT, where NAME is equal to the name specified on this PARAM entry.		
	NO	The specifications for EID1 through EID3 on the ATBSEG entry and/or EID1 through EID6 on the ATBJNT entry are ignored. No Bulk Data file will be created.		
NAME	Name given to the Bulk Data file.		C	Required
LENGTH1	Specifies the length of the axes spanned by the grid points that represent the local coordinate systems of the segments.		R > 0.0	0.1
LENGTH2	Specifies the length of the axes spanned by the grid points that represent the local coordinate systems of the joints.		R > 0.0	0.05

AUTHINFO

Licensing Information Control

Defines the amount of information FLEXlm licensing writes to the output file.

Format and Example	Default
PARAM,AUTHINFO,value	1
PARAM,AUTHINFO,9	

Option	Meaning	Type
AUTHINFO	The amount of licensing information that FLEXlm writes to the output file. A value of 1 provides the minimum amount of licensing information, while a value of 9 provides the maximum amount of information.	I > 0

Remark:

1. You can use this parameter to obtain extra licensing information in case you experience a FLEXlm licensing problem. Under normal circumstances, where FLEXlm licensing does not cause any problems, there is no need to use this parameter.

BULKL

Linear Bulk Viscosity Coefficient

Defines the default value of the linear bulk viscosity coefficient.

Format and Example	Default
PARAM,BULKL,value	0.0
PARAM,BULKL,0.1	

Option	Meaning	Type
value	Value of the linear coefficient in the bulk viscosity equation.	R ≥ 0.0

Remarks:

1. The default value works well for the majority of problems.
2. The value defined on this entry is used as the default whenever BULKL is blank on the DMATxx material entries.
3. When BULKL is specified on a material definition entry, the default value is overridden for that specific material.
4. See Section 2.5.10.1 on page 2-79 for details on bulk viscosity.

BULKTYP

Bulk Viscosity Type

Defines the default type of bulk viscosity.

Format and Example	**Default**
PARAM,BULKTYP,option	DYNA
PARAM,BULKTYP,DYNA	

Option	**Meaning**	**Type**
DYNA	Standard DYNA3D model.	C
DYTRAN	Enhanced DYNA model.	

Remark:

1. See Section 2.5.10.1 on page 2-79 for details on bulk viscosity.

BULKQ

Quadratic Bulk Viscosity Coefficient

Defines the default value of the quadratic bulk viscosity coefficient.

Format and Example	Default
PARAM,BULKQ,value	1.0
PARAM,BULKQ,1.6	

Option	Meaning	Type
value	Value of the quadratic coefficient in the bulk viscosity equation.	R ≥ 0.0

Remarks:

1. The default value works well in the majority of situations.
2. The value defined on this entry is used as the default whenever BULKQ is blank on the DMATxx material entries.
3. When BULKQ is specified on a material definition entry, the default value is overridden for that specific material.
4. See Section 2.5.10.1 on page 2-79 for details on bulk viscosity.

CFULLRIG

Converts 123456 Constraints to FULLRIG on RBE2 Entries

Converts all 123456 constraints to the FULLRIG option on all RBE2 entries.

Format and Example	Default
PARAM,CFULLRIG,value	YES
PARAM,CFULLRIG,NO	

Option	Meaning	Type
YES	123456 constraints are converted to FULLRIG.	C
NO	123456 constraints are not converted to FULLRIG.	

CONM2OUT

CONM2 Summary Output

Determines if a summary of concentrated masses and their energy and momentum is written to the output file.

Format and Example	Default
PARAM,CONM2OUT,option	YES
PARAM,CONM2OUT,NO	

Option	Meaning	Type
NO	No information about concentrated masses is written to the cycle and material summaries on the output file.	C
YES	A complete summary of concentrated masses including the associated mass, momentum, and energy is written to the output file.	C

Remark:

1. When PARAM,CONM2OUT is set to NO, there will be no summary of the concentrated mass. This means that the mass, momentum, and energy of the concentrated masses, is not added to the material and cycle summaries. Setting PARAM,CONM2OUT,NO saves memory and CPU time.

CONTACT

Sets Defaults for CONTACT

Defines certain defaults for the contact definitions.

Format and Example	Type
PARAM,CONTACT,option,value1,value2,...	Blank
PARAM,CONTACT,VERSION,V4	

Option	Meaning		Type
VERSION, [V2, V3, V4, BELT, BELT1] [DRAWBEAD]	Defines the default version.		C, Required
THICK,value	Defines the default value for THICK.		C, R
GAP,value	Defines the default value for GAP.		C, R > 0
LIMITS, [XMIN,XMAX, YMIN,YMAX, ZMIN,ZMAX]	Definition of a three dimensional contact region where contact in the analysis model will take place. Significant CPU time savings can be achieved when used in adaptive contact.		
	XMIN	Lower limit in x-direction where main contact will occur.	-1.E20
	XMAX	Upper limit in x-direction where main contact will occur.	1.E20
	YMIN	Lower limit in y-direction where main contact will occur.	-1.E20
	YMAX	Upper limit in y-direction where main contact will occur.	1.E20
	ZMIN	Lower limit in z-direction where main contact will occur.	-1.E20
	ZMAX	Upper limit in z-direction where main contact will occur.	1.E20
DAMPING, [YES/NO]	Defines the default value for DAMPING.		C, C
DAMPFOR, [YES/NO]	Defines whether the non-contact forces acting on the grid points need to be taken into account in the contact damping of the V4 contact algorithm. This option will only be used if DAMPING = YES. This option will prevent large penetrations that might occur when the forces acting on the grid points tend to push them into the contact surface. This happens, for example, in airbag analyses, where a large pressure exists inside the bag.		C,C

Option	Meaning	Type
DYNA	The following parameters of the contact definition will get default values consistent with MSC/DYNA. THICK 1.0 GAP 0.0 PEN FACTOR PENV 0.4	C
INFO, [G1, G2, ...]	Information on the contact state of grid points G1, G2, ... will be printed to ASCII files, named CNT... This information can be useful in debugging models with contacts.	C,I
CONTINI, [INITIAL, ALLWAYS]	Defines how the CONTINI logic is used during the analysis. INITIAL: The CONTINI logic is used only during the analysis initialization. Note that for SEARCH=SLIDE the slave nodes will lose their contact once they slide off a previous page and not onto a neighbor face. This is mostly a correct behavior for deploying airbags, except in very complicated folding patterns. For very complicated folding patterns it is advised to use PARAM,CONTACT,CONTINI, ALLWAYS. ALLWAYS: For SEARCH=SLIDE: When a slave node slides off a previous face, and not onto a neighbor face, new faces will be checked, using the CONTINI logic. For SEARCH=FULL: New face are always checked using the CONTINI logic. This setting requires more memory and also more CPU-time will be spend on the contact logic.	C,C

Controls the growth of the subcycling interval in Euler/Lagrange coupling.

Format and Example	**Default**
PARAM,COSUBCYC,value	1

Option	**Meaning**	**Type**
value	Maximum growth of the subcycling interval.	I > 0

Remarks:

1. The subcycling algorithm automatically estimates the number of subcycles to be used. This is updated throughout the calculation. This parameter controls how much the number of subcycles can grow. For example, COSUBCYC is set to 1, and the current number of time steps between updates of the coupling geometry is 4. If MSC.Dytran estimates that the subcycling interval should be 7, the subcycling interval is increased by 1 until a value of 7 is reached.

2. There is no control on the amount by which the subcycling interval can decrease.

COSUBMAX

Subcycle Limit in Euler/Lagrange Coupling

Defines the maximum number of subcycles that can occur in Euler/Lagrange coupling. During a subcycle, the geometry of the coupling surface is not updated.

Format and Example	Default
PARAM,COSUBMAX,value	1
PARAM,COSUBMAX,10	

Option	Meaning	Type
value	The maximum number of time steps between updating the coupling surface geometry in the coupling calculations.	I > 0

Remarks:

1. Updating the coupling geometry takes a lot of CPU time. Subcycling gives substantial savings in CPU time for coupled calculations.
2. The smaller the value of this parameter, the greater the accuracy of the analysis and the greater the cost. Conversely, larger values offer significant CPU savings, but very large values give incorrect results.
3. If the geometry of the coupling surface is changing rapidly, smaller values of PARAM,COSUBMAX should be used.

DELCLUMP

Delete Clump Fraction

Material in Eulerian elements of a clump with

$$\text{fvunc} < \text{DELCLUMP} * \text{fblend}$$

is eliminated. This prevents small clumps from determining the time step and prevents the leakage of small masses to isolated regions.

Format and Example	Default
PARAM,DELCLUMP,value	0.5
PARAM,DELCUMP,0.1	

Option	Meaning	Type
value	The value of DELCLUMP.	R ≥ 0.0

Remark:

1. See also parameters FBLEND and FVUNC.

ELSUBCHK

Interval for Efficiency Checks on Element Subcycle Groups

Defines the number of steps between two checks on the efficiency of the element subcycle groups. This efficiency can degrade due to changes in the stable time steps. Based on the values of PARAM,ELSUBDAC, and PARAM,ELSUBRGP, the subcycling algorithm could be deactivated or the element groups can be redefined.

Format and Example	**Default**
PARAM,ELSUBCHK,value	999999999
PARAM,ELSUBCHK,100	(only initial check)

Option	**Meaning**	**Type**
value	Number of steps between two efficiency checks.	I > 0

Remark:

1. The parameters ELSUBDAC and ELSUBRGP influence the results of the efficiency checks.

ELSUBCYC

Element Subcycling

Defines the element types that use element subcycling.

Format and Example	Default
PARAM,ELSUBCYC,option	NONE
PARAM,ELSUBCYC,ALL	

Option	Meaning	Type
ALL	Subcycling activated for all element types.	C
NONE	Subcycling deactivated for all element types.	C
Any combination of the following:		
ELEM1D	Subcycling activated for 1-D elements.	C
SHTRIA	Subcycling activated for triangular shells.	C
SHQUAD	Subcycling activated for quadrilateral shells.	C
MEMTRIA	Subcycling activated for triangular membranes.	C
LAGSOLID	Subcycling activated for Lagrangian solids.	C
EULHYDRO	Subcycling activated for single material hydrodynamic Eulerian elements.	C
EULSTRENGTH	Subcycling activated for single material Eulerian elements with strength.	C
MULTIEULHYDRO	Subcycling activated for multimaterial Eulerian elements.	C

(Continued)

Remarks:

1. If subcycling is activated for a specific element type, the elements are grouped according to the following table:

Element Type Name	Level of Grouping for Element Subcycling
ELEM1D	1
SHTRIA	1
SHQUAD	1
MEMTRIA	2
LAGSOLID	2
EULHYDRO	0
EULSTRENGTH	0
MULTIEULHYDRO	0

where grouping level = 0: All elements of the same type in one subcycling group.

grouping level = 1: All elements of the same property in one subcycling group.

grouping level = 2: Elements within a property in different subcycling groups.

2. The elements are partitioned into groups with different time steps. The smallest group is updated with time step DTMAST. One such update with DTMAST is called a SUBCYCLE, while a CYCLE is completed when the group with the largest time step is updated.

3. Due to synchronization, the following output requests are completed at the end of a CYCLE even if the user asks for output at the end of an intermediate SUBCYCLE:

   CYCLE SUMMARY

   MATERIAL SUMMARY

   BOUNDARY SUMMARY

   TIMEHIS of MATERIAL VARIABLES

   TIMEHIS and ARCHIVE requesting the FORCES on the grid points

4. Due to synchronization, the wrap-up of the calculation is completed at the end of a CYCLE even if the user asks for the wrap-up at the end of an intermediate SUBCYCLE.

ELSUBDAC

Deactivation of Element Subcycling

The gain in CPU time by the element subcycling algorithm is checked with an interval as defined on PARAM,ELSUBCHK. This gain can decrease due to changes in the stable time steps. If it is found that the gain in CPU time falls below the value specified by this parameter, the subcycling is deactivated.

Format and Example	Default
PARAM,ELSUBDAC,value	0.0
PARAM,ELSUBDAC,10.0	

Option	Meaning	Type
value	Percentage for efficiency check of subcycling.	R

Remark:

1. By default, this check is made only once during a run. To get a regular check on the efficiency, the value of ELSUBCHK must be changed.

ELSUBMAX

Maximum Number of Groups in Element Subcycling

Defines the maximum number of groups in element subcycling.

Format and Example	Default
PARAM,ELSUBMAX,value	4
PARAM,ELSUBMAX,2	

Option	Meaning	Type
value	Defines the maximum number of groups.	1 < I < 21

Remark:

1. An element group is updated with: dtgroup=2∗∗ (ngroup) ∗ dtmin

   where ngroup =1,2,... elsubmax, and dtmin is the smallest stable time step.

   Increasing the value of this parameter allows a time-step ratio greater than 8. This may lead to inaccurate results and instabilities in the solution.

ELSUBRGP

Redefinition of Element Subcycling Groups

The efficiency of the element subcycle groups is checked with an interval as defined on PARAM, ELSUBCHK. A complete redefinition of the element groups is performed if the estimated savings increase more than the value as specified by this parameter. The change in savings is calculated in percentage.

Format and Example	Default
PARAM,ELSUBRGP,value	1.E20
PARAM,ELSUBRGP,10.0	(No redefinition)

Option	Meaning	Type
value	Percentage for efficiency check of subcycle groups.	R

Remarks:

1. By default, the element groups are defined only once during an analysis. To get a redefinition based on an efficiency check, the value of ELSUBCHK must also be set.

   For example, the following settings give a check on the efficiency every 100 steps and a redefinition is performed if the extra savings due to a regroup exceed 25%:

   PARAM,ELSUBCHK,100
   PARAM,ELSUBRGP,25.0

2. During a calculation, elements can reach a stable time step that necessitates a redefinition. This kind of redefinition is not influenced by this parameter.

3. Redefinition during the analysis is potentially dangerous since the time steps are not growing gradually but instead are increased to the new values once at the beginning of a new CYCLE.

4. Checking and redefinition results in overhead, so the checking and redefinition functions should only be activated if gain is expected.

5. Redefinition can be forced at the beginning of a restart run with the parameter ELSUBRRG.

ELSUBRRG

Redefinition of Element Subcycling Groups at Restart

The user can force a redefinition of the element subcycle groups at the beginning of a restart run.

Format and Example	**Default**
PARAM,ELSUBRRG,option	OFF
PARAM,ELSUBRRG,ON	

Option	**Meaning**	**Type**
ON	The element subcycle groups are redefined during the first CYCLE of a restart run.	C
OFF	No redefinition.	

Remarks:

1. By default, the element groups are defined only once during an analysis.

   At restart time, the user can force a redefinition of the groups with this parameter.

2. Automatic redefinition based on efficiency checks can be activated by the parameter ELSUBRGP. See the ELSUBRGP Remarks.

3. The value of the parameter is not carried along to the next restart. If a new redefinition is needed, the parameter must be specified again in the input of the next restart.

ERRUSR

Redefinition of severity and number of error message prints

Redefinition of severity and number of prints of error messages.

Format and Example

PARAM,ERRUSR,name,severity,prints

PARAM,ERRUSR,P2010053,I,2

Option	Meaning		Type	Default
name	Error name.		C	Required
severity	Error severity.		C	Required
	I	Informative.		
	W	Warning.		
	E	Error.		
	F	Fatal.		
	C	Catastrophic.		
	N	MSC.NASTRAN–ignore.		
prints	Number of times the message is printed.		I	5

Remarks:

1. An error name consists of a maximum of eight characters. The entry is also used as a wildcard by entering less than eight characters. The string then is matched with the actual names, and every match redefines the actual message.

2. See Chapter 5, Diagnostic Messages.

EULTRAN

Switch for the Multimaterial Euler Transport Scheme

Sets the definition of the face velocity used in the multimaterial Euler transport scheme.

Format and Example	**Default**
PARAM,EULTRAN,option	IMPULSE
PARAM,EULTRAN,AVERAGE	

Option	**Meaning**	**Type**
IMPULSE	The face velocity is impulse weighted.	C
AVERAGE	The face velocity is a simple average.	

Remarks:

1. The default value of IMPULSE is sufficient for most multimaterial Euler problems. Especially problems where the reference density of the different materials varies widely (e.g., orders of magnitude) are required to use the default option.
2. In case the IMPULSE option (default) is used, the multimaterial Euler transport scheme computes that the face velocity uses an impulse weighted average of the material velocity in the left and the right element adjacent to the face.
3. In case the AVERAGE option is used, the multimaterial Euler transport scheme computes the face velocity as one-half times the sum of the material velocity in the left and the right element adjacent to the face.

EXTRAS

Extra Input Constants

Input of extra constants that you can access from within other user-written subroutines.

Format and Example	Default
PARAM,EXTRAS,name,value,name,value,etc.	No extra constants.
PARAM,EXTRAS,MASSFLOW,1.E6,MASS,15.3	

Option	Meaning	Type
name	Constant name.	C
value	Constant value.	R

Remark:

1. Usage in a user subroutine as follows:

```
SUBROUTINE EXCOMP (...)
COMMON/EXTRAS/NMEXTR, IDEXTR
.
.
.
CHARACTER*16 CARGET, CVAR
.
.
.
IF (IDEXTR.GT.0) THEN
     DO NV = 1,NMEXTR
        CVAR = CARGET (IDEXTR, NV,’USER’)
        IF (CVAR(1:8).EQ.’MASSFLOW’) THEN
           VALMF = XARGET (IDEXTR, NV,’USER’)
        ELSEIF (CVAR(1:4).EQ.’MASS’) THEN
           VALMS = XARGET (IDEXTR, NV,’USER’)
        ENDIF
     ENDDO
ENDIF
RETURN
.
.
.
END
```

FAILOUT

Failed Element Output Parameter

Defines whether failed elements are written to the output file (ARCHIVES).

Format and Example	**Default**
PARAM,FAILOUT,option	YES
PARAM,FAILOUT,NO	

Option	**Meaning**	**Type**
NO	Failed elements are not written to the archive files.	C
YES	Failed elements are written to the archive files.	C

Remarks:

1. When the NO option is chosen, the archives are written out as one file per requested time step regardless of the number set in the SAVE command for the archive files that appear in the Case Control Section.

2. Failed elements are NOT filtered when written to a RESTART file or a TIMEHISTORY file.

FASTCOUP

Fast Coupling Algorithm

Defines the fast coupling algorithm.

Format and Example	Default
PARAM,FASTCOUP,option1,option2	See Remark 1.
PARAM,FASTCOUP,INPLANE,FAIL	

Option	Meaning		Type
option1	blank	No offset for inplane coupling surface segments.	See Remark 2.
	INPLANE	Small offset for inplane coupling surface segments.	
option2	NOFAIL	No failure of the coupling surface.	See Remark 3.
	FAIL	Failure of the coupling surface will be taken into account.	

Remarks:

1. Default value for option1 is blank and for option2 NOFAIL.
2. When option1 is set to INPLANE, a small offset is given to coupling surface segments that are on top of a face of an Eulerian element.
3. Option2 can only be used in combination with PARAM,LIMITER,ROE. The coupling surface must consist of CQUADs and/or CTRIAs and a failure model for the material of the surface must be defined.
4. This parameter can only be used when the Eulerian mesh is aligned with the basic coordinate system axes.

FBLEND

Blend Fraction

Eulerian elements with uncovered fractions smaller than FBLEND are blended with adjacent elements to form a clump so that they do not control the time step.

Format and Example	**Default**
PARAM,FBLEND,value	0.6667
PARAM,FBLEND,0.5	

Option	**Meaning**	**Type**
value	The uncovered fraction below which blending occurs.	$0.0 \le R < 1.0$

Remarks:

1. The default value is satisfactory for virtually all calculations.
2. Elements are blended only if they would have controlled the time step otherwise.
3. Elements with uncovered fractions greater than FBLEND are not blended and are allowed to control the time step.
4. Large values of FBLEND produce a larger time step but many blends. Small values produce a smaller time step and fewer blends.
5. In a calculation with a coupling surface, STEPFCT is smaller or equal FBLEND to avoid instabilities (see PARAM,STEPFCT).

FMULTI

Multimaterial Overflow Array Parameter

Defines the dimension of the multimaterial element array.

Format and Example	Default
PARAMS,FMULTI,value	.10
PARAMS,FMULTI,.25	

Option	Meaning	Type
value	The relative amount of multimaterial elements.	0 < R < 1.

Remark:

1. The multimaterial Eulerian elements use an overflow array in which to store material data. This array can hold FMULTI times the total number of Eulerian elements. In a problem where more than 10% of the elements have more than one material, the default value of FMULTI must be increased.

GEOCHECK

Define Geometry Consistency Check

This parameter forces a check of the geometry for consistent connectivity of the defined hexagonal elements and correction if needed.

Format and Example	Default
PARAM,GEOCHECK,option	OFF
PARAM,GEOCHECK,ON	

Option	Meaning	Type
ON	Geometry consistency check is performed.	C
OFF	No geometry consistency check is performed.	

Remarks:

1. The defined geometry is checked for consistent connectivity of the hexagonal elements. If an inconsistency is detected, the connectivity is corrected. CFACE entries with references to elements that have been corrected are corrected as well.

2. This option must be used when MSC/XL hexagonal meshes are generated with the EXTRUDE or MIRROR option.

3. If a hexagonal mesh is generated with other commercial preprocessors, this parameter can correct the connectivity of the hexagonal elements in case problems are encountered with face generation or volume computation.

HGCMEM

Shell Membrane Hourglass Damping Coefficient

Defines the default membrane damping coefficient for shell elements.

Format and Example	**Default**
PARAM,HGCMEM,value	See Remark 3.
PARAM,HGCMEM,0.07	

Option	**Meaning**	**Type**
Value	Hourglass damping coefficient.	$0.0 \le R \le 0.15$

Remarks:

1. The default value applies to all types of hourglass suppression methods and should be used unless there is a good reason to define the hourglass damping coefficient by another means.
2. The value specified on this entry is used whenever the coefficient is not explicitly defined on the HGSUPPR entry.
3. If this entry is omitted, the default value of the coefficient used in the hourglass suppression method for shell elements is either equal to the default value of 0.1 or is equal to the default value defined on a PARAM,HGCOEFF entry.
4. See Section 2.5.10.1 on page 2-79 for details on hourglass suppression methods.

HGCOEFF

Hourglass Damping Coefficient

Defines the global default hourglass damping coefficient.

Format and Example	Default
PARAM,HGCOEFF,value	See Remark 3
PARAM,HGCOEFF,0.14	

Option	Meaning	Type
Value	Hourglass damping coefficient.	$0.0 \leq R \leq 0.15$

Remarks:

1. The default applies to all types of hourglass suppression methods and should be used unless there is good reason to define the hourglass damping coefficient by another means.
2. The value specified on this entry is used whenever the coefficients are not explicitly defined on HGSUPPR entries or on a HGCMEM, HGCWRP, HGCTWS, or HGCSOL entry.
3. If this entry is omitted, the default value of the hourglass damping coefficients is either equal to the default value of 0.1 or is equal to the value specified on a HGCMEM, HGCTWS, HGCWRP, or HGCSOL PARAM entry.
4. The value of the coefficients can be explicitly defined for each property by using an HGSUPPR entry.
5. See Section 2.5.10.2 on page 2-80 for details on hourglass suppression methods.

HGCSOL

Solid Hourglass Damping Coefficient

Define the default damping coefficient for solid elements.

Format and Example	**Default**
PARAM,HGCSOL,value	See Remark 3.
PARAM,HGCSOL,0.11	

Option	**Meaning**	**Type**
Value	Hourglass damping coefficient.	$0.0 \le R \le 0.15$

Remarks:

1. The default value applies to all types of hourglass suppression methods and should be used unless there is a good reason to define the hourglass damping coefficient by another means.
2. The value specified on this entry is used whenever the coefficient is not explicitly defined on the HGSUPPR entry.
3. If this entry is omitted, the default value of the coefficient used in the hourglass suppression method for solid elements is either equal to the default value of 0.1 or is equal to the default value defined on a PARAM,HGCOEFF entry.
4. See Section 2.5.10.2 on page 2-80 for details on hourglass suppression methods.

HGCTWS

Shell Twisting Hourglass Damping Coefficient

Defines the default twisting damping coefficient for shell elements.

Format and Example	Default
PARAM,HGCTWS,value	See Remark 3.
PARAM,HGCTWS,0.02	

Option	Meaning	Type
Value	Hourglass damping coefficient.	0.0 ≤ R ≤ 0.15

Remarks:

1. The default value applies to all types of hourglass suppression methods and should be used unless there is a good reason to define the hourglass damping coefficient by another means.

2. The value specified on this entry is used whenever the coefficient is not explicitly defined on the HGSUPPR entry.

3. If this entry is omitted, the default value of the coefficient used in the hourglass suppression method for shell elements is either equal to the default value of 0.1 or is equal to the default value defined on a PARAM,HGCOEFF entry.

4. See Section 2.5.10.2 on page 2-80 for details on hourglass suppression methods.

HGCWRP

Shell Warping Hourglass Damping Coefficient

Defines the default warping damping coefficient for shell elements.

Format and Example	Default
PARAM,HGCWRP,value	See Remark 3.
PARAM,HGCWRP,0.0	

Option	Meaning	Type
Value	Hourglass damping coefficient.	$0.0 \le R \le 0.15$

Remarks:

1. The default value applies to all types of hourglass suppression methods and should be used unless there is a good reason to define the hourglass damping coefficient by another means.
2. The value specified on this entry is used whenever the coefficient is not explicitly defined on the HGSUPPR entry.
3. If this entry is omitted, the default value of the coefficient used in the hourglass suppression method for shell elements is either equal to the default value of 0.1 or is equal to the default value defined on a PARAM, HGCOEFF entry.
4. See Section 2.5.10.2 on page 2-80 for details on hourglass suppression methods.

HGSHELL

Shell Hourglass Suppression Method

Defines the default hourglass suppression method for shell elements.

Format and Example	Default
PARAM,HGSHELL,option	See Remark 2.
PARAM,HGSHELL,DYNA	

Option	Meaning	Type
FBV	Flanagan-Belytschko viscous hourglass damping.	C
DYNA	Viscous hourglass damping.	

Remarks:

1. The type of hourglass suppression method defined on this entry is used as the default whenever the type fields on the HGSUPPR for shell properties are left blank.
2. If this entry is omitted, the default suppression method used for shell elements is either FBV or the default method defined on the PARAM,HGTYPE entry.
3. See Section 2.5.10.2 on page 2-80 for details on hourglass suppression methods.

HGSOLID

Solid Hourglass Suppression Method

Defines the default hourglass suppression method for solid elements.

Format and Example	Default
PARAM,HGSOLID,option	See Remark 2.
PARAM,HGSOLID,FBS	

Option	Meaning	Type
FBS	Flanagan-Belytschko stiffness hourglass damping.	C
DYNA	Viscous hourglass damping.	

Remarks:

1. The type of hourglass suppression method defined on this entry is used as the default whenever the type fields on the HGSUPPR for solid properties are left blank.
2. If this entry is omitted, the default suppression method used for solid elements is either FBS or the default method defined on the PARAM,HGTYPE entry.
3. See Section 2.5.10.2 on page 2-80 for details on hourglass suppression methods.

HGTYPE

Hourglass Suppression Method

Defines the default type of hourglass suppression method.

Format and Example	Default
PARAM,HGTYPE,option	See Remark 2.
PARAM,HGTYPE,FBS	

Option	Meaning	Type
FBS	Flanagan-Belytschko stiffness hourglass damping.	C
FBV	Flanagan-Belytschko viscous hourglass damping.	
DYNA	Viscous hourglass damping.	

Remarks:

1. The type of the hourglass suppression method defined on this entry is used as the default whenever the type fields in the HGSUPPR entries are left blank.
2. If this entry is omitted, the type can be defined on a PARAM,HGSHELL entry for shell elements, a PARAM,HGSOLID entry for solid elements, or on the HGSUPPR entries; otherwise the defaults apply. For shell elements the default is FBV; for solid elements, the default is FBS.
3. See Section 2.5.10.2 on page 2-80 for details on hourglass suppression methods.

HVLFAIL

Failure at Hydrodynamic Volume Limit

Defines element failure on the hydrodynamic volume limit.

Format and Example	Default
PARAM,HVLFAIL,option	NO
PARAM,HVLFAIL,YES	

Option	Meaning	Type
YES	Element failure on hydrodynamic volume limit.	C
NO	No element failure on hydrodynamic volume limit.	

Remarks:

1. Lagrangian elements (CHEXA) that have a material model with a failure model fail when the hydrodynamic volume limit is reached and the parameter is set to YES.
2. The hydrodynamic volume limit by default allows for 10% expansion.

HYDROBOD

Hydro Body Force

Defines a body force for single hydro material in Euler.

Format and Example	Default
PARAM,HYDROBOD,XACC,YACC,ZACC	No body force applied.
PARAM,HYDROBOD,-300.,0.,150.	

Option	Meaning	Type
XACC	X-acceleration.	R
YACC	Y-acceleration.	R
ZACC	Z-acceleration.	R

Remark:

1. This parameter defines a constant body force load in Euler for single hydro material only.

IEEE

IEEE Binary Data Output Format

On CRAY computers, archive and time-history output is defined in IEEE format rather than in Block Data format.

Format and Example	**Default**
PARAM,IEEE,option	OFF
PARAM,IEEE,ON	

Option	**Meaning**	**Type**
ON	Activate IEEE output.	C
OFF	No IEEE output.	

Remarks:

1. On computers that by default write binary data in IEEE format, the parameter will have no effect.
2. Binary IEEE files can be transported to all IEEE computer systems. (Note that this in only valid for ARCHIVE and TIMEHISTORY files.)

IGNFRCER

Ignores Warnings

Ignores certain errors for FORCE and MOMENT entries.

Format and Example	Default
PARAM,IGNFRCER	See below.

Option	Meaning	Type
No option	Some warnings that are given when using the FORCE1, FORCE2, MOMENT1, or MOMENT2 entries are normally accompanied by an additional error message. By using this PARAM, the warnings are still issued, but the error message will not be issued.	

Additional information about the BJOIN and spotweld connectivity will be listed in the output file. The information listed is:

- Grid point pairs forming a BJOIN or a spotweld.
- BJOINS and spotwelds initially connected.

Format and Example	**Default**
PARAM,INFO-BJOIN,option	NO
PARAM,INFO-BJOIN,YES	

Option	**Meaning**	**Type**
YES	Information is issued.	C
NO	Information is not issued.	

INISTEP

Initial Time Step

Defines the time step used at the start of the analysis.

Format and Example	Default
PARAM,INISTEP,value	No default
PARAM,INISTEP,1.E–6	

Option	Meaning	Type
value	Time step (in analysis time units) used for the first iteration.	R > 0.0

Remarks:

1. This parameter is required to start an analysis.
2. See Section 3.11 on page 3-68 for details on time-step control.

INITFILE

Method of Initalization Definition

Defines the method of initializing a transient analysis from a MSC.Dytran solution file.

Format and Example	Default
PARAM,INITFILE,option	V3
PARAM,INITFILE,V1	

Option	Meaning	Type
V1	Version of initialization, where the prestress and the transient input files must obey the following rules: • The number of structural elements must be the same. • The number of structural grid points must be the same. • The boundary conditions on the grid points must be the same. • The material models must be the same. Eulerian grid points are allowed to be present in the prestress analysis, but are not written to or read from the Solution file. This version is available for the following element types: • One-dimensional elements. • Shell elements (including composites). • Membrane elements. • Lagrangian solid elements.	C
V2	Version of initialization, where the prestress and the transient input files must obey the following rules: • No restrictions on the number of elements and grid points. • No restrictions on the consistency of the boundary conditions. • Eulerian grid points are allowed to be present in the prestress analysis, but are not written to or read from the Solution file. This version is available for the following element types: • Shell elements (excluding composites) • Lagrangian solid elements	C

(Continued)

Option	Meaning	Type
V3	Version of initialization, where the prestress and the transient input files must obey the following rules: • The number of structural elements must be the same. • The number of structural grid points must be the same. • The boundary conditions on the grid points are allowed to change. • The material models must be the same. • When the Eulerian grid points are present in the prestress analysis they are written to the solution file during the prestress analyis and read from the solution file during the transient analysis. This version is available for the following element types: • One-dimensional elements. • Shell elements (including composites). • Membrane elements. • Lagrangian solid elements. • Eulerian elements.	C

Remarks:

1. The user is responsible for consistency upon choosing the V2 definition.
2. See Sections 3.10.3 on page 3-67 and 3.16 on page 3-142 for more detailed information about prestress analyses.

INITNAS

Defines the Type of Displacement Initialization File

Defines the type of file to be used for initialization from an MSC.NASTRAN prestress analysis.

Format and Example	Default
PARAM,INITNAS,option	XL
PARAM,INITNAS,XL	

Option	Meaning	Type
XL	File is an MSC/XL export file from an MSC.NASTRAN database (*.xdb).	C
PATRAN	File is an MSC.Patran displacement output file from NASPAT.	C
PUNCH	File is an MSC.NASTRAN punch file for the displacements.	

Remarks:

1. When MSC.Dytran uses the results of an MSC.NASTRAN analysis to start a transient analysis from a prestressed state, the grid-point displacement field, as computed by MSC.NASTRAN, is read from a formatted file written either by MSC/XL, MSC.Patran, or MSC.NASTRAN.

   The format of the formatted import files is as follows:

   MSC/XL Export File

Record 1:	Header 1	
Record 2:	Header 2	
Record 3:	Header 3	
Record 4:	Header 4	
Record 5:	Header 5	
Record 6:	Header 6	
Record 7:	Header 7	
Record 8:	Header 8	
Record 9 to n+8:	Grid point X-Dis Y-Dis Z-Dis	(A8, 3A15)

   MSC.Patran Nodal Results Data File

Record 1:	TITLE	(80A1)
Record 2:	NNODES, MAXNOD, DEFMAX, NDMAX, NWIDTH	(2I9,E15.6,2I9)
Record 3:	SUBTITLE 1	(80A1)
Record 4:	SUBTITLE 2	(80A1)
Record 5 to n+4:	NODID, (DATA (j), j=1, NWIDTH)	(I8, (5E13.7))

2. The punch file option will only be active for the Key-Hoff, Belytschko-Tsay-Lin (BLT), and Hughes-Liu shell element formulations.

LIMCUB

Contact Cube Sort Algorithm

Defines the maximum number of cubes used to sort the grid points in a contact definition.

Format and Example	Default
PARAM,LIMCUB,value	2000
PARAM,LIMCUB,2300	

Option	Meaning	Type
LIMCUB	Maximum number of cubes.	I > 0

Remark:

1. Each slave node has to search for master nodes that are close enough to have potential contact. It is too expensive to have each slave node check each master node. To limit the number of checks, the space in which the nodes reside is subdivided into cubes. This subdivision is done so that the slave nodes have to check only the master nodes in their own cube and those in the neighboring cubes. The maximum number of cubes used to subdivide the space is equal to the value of LIMCUB.

LIMITER

Euler Solver Scheme

Defines the type and the spatial accuracy of scheme used in the Euler solver.

Format and Example	Default
PARAM,LIMITER,type,option	See Remark 1.
PARAM,LIMITER,ROE	

Option	Meaning		Type
type	Type of scheme.		C
	ROE	Roe solver for single hydro materials.	
option	blank	Second order in space.	See Remark 2.
	NONE	First order in space.	

Remarks:

1. By default, the standard Euler solver is used.
2. By default, second order spatial accuracy is used. The temporal accuracy is defined using the PARAM,RKSCHEME entry.
3. When type ROE is defined, no void elements are allowed and it can not be used in combination with EOSJWL. Also ALE and options concerning air bag analyses are not supported.
4. For more details on the Euler solver see Section 2.21 on page 2-136.

MATRMERG

Merges MATRIG and RBE2-FULLRIG Assemblies

Merges MATRIG and/or RBE2-FULLRIG rigid bodies into a new FULLRIG assembly.

Format and Example	Default
PARAM,MATRMERG,FR<id1>,MR<id2>,MR<id3>,FR<id4>,...	None
PARAM,MATRMERG,FR1,MR2,MR6,MR7,FR4,MR8	
PARAM,MATRMERG,AUTO	

Option	Meaning	Type
FR<id1> or AUTO	Name of the new FULLRIG assembly, or the AUTO option (see Remark 2).	C
MR<idi> or FR<idi>	Names of MATRIG and/or RBE2-FULLRIG rigid bodies, which are merged into a new FULLRIG assembly with name FR<id1>. No names can be supplied for the AUTO option.	C

Remarks:

1. FR<id1> must be a nonexisting RBE2-FULLRIG. The properties of FR<id1> (as mass, center of gravity and moments of inertia) are computed by MSC.Dytran from the properties of each rigid body mentioned on the entry. Rigid body output can be asked for FR<id1>, and loads or rigid body constraints can be applied to FR<id1>. The other MATRIGs and RBE2-FULLRIGs mentioned on the MATRMERG entry disappear after they have been merged.

2. Instead of supplying rigid body names, the AUTO-option can be used. After all the normal PARAM,MATRMERG and PARAM,MATRMRG1 entries have been applied, a PARAM,MATRMERG,AUTO merges all the resulting MATRIGs and RBE2-FULLRIGs which have common grid points, into a new rigid assembly called FM<id>, where the id is a new FM number, starting from 1. As it is not known at the start of an MSC.Dytran analysis how many FM-assemblies will be created, no rigid body output can be asked for FM<id>, and no constraints or loads can be applied to FM<id>. The MATRIGs and RBE2-FULLRIGs which have been merged by the AUTO option into a new FM<id> assembly, disappear.

3. To supply predefined properties for the merged assembly, PARAM,MATRMRG1 can be used, where the first rigid body mentioned on the entry must be an existing RBE2-FULLRIG or MATRIG.

MATRMRG1

Merges MATRIG and RBE2-FULLRIG Assemblies

Merges MATRIG and/or RBE2-FULLRIG rigid bodies into one existing MATRIG or RBE2-FULLRIG assembly with predefined properties.

Format and Example	Default
PARAM,MATRMRG1,MR<id1>,MR<id2>,MR<id3>,FR<id4>,...	None
PARAM,MATRMRG1,MR1,MR2,MR6,MR7,FR4,MR8	

Option	Meaning	Type
MR<id1> or FR<id1>	Name of the new MATRIG or FULLRIG assembly (must be an existing one).	C
MR<idi> or FR<idi>	Names of MATRIG and/or RBE2-FULLRIG rigid bodies, which are merged with the existing MR<id1> or FR<id1> into a new MATRIG assembly, with name MR<id1> or FR<id1>.	C

Remarks:

1. MR<id1> or FR<id1> must be an existing MATRIG or RBE2-FULLRIG, respectively. For a FULLRIG, the properties of FR<id1> (as mass, center of gravity and moments of inertia) are computed by MSC.Dytran from the properties of each rigid body mentioned on the entry. For a MATRIG, the mass of MR<id1> is either the predefined mass on the MATRIG (id1) entry or the predefined density on the MATRIG (id1) entry times the total volume of all MATRIG members in the MATRMRG1 entry. The center of gravity and moments of inertia of MR<id1> are either predefined on the MATRIG (id1) entry, or are otherwise computed from the properties of each rigid body on the entry. The other MATRIGs and RBE2-FULLRIGs mentioned on the MATRMRG1 entry disappear after they have been merged.

MAXSTEP

Maximum Time Step

Defines the maximum allowable time step.

Format and Example	Default
PARAM,MAXSTEP,value	1.E20
PARAM,MAXSTEP,1.E–3	

Option	Meaning	Type
value	The maximum time step.	R > 0.0

Remark:

1. If the time step calculated by MSC.Dytran is greater than MAXSTEP, the time step is set to MAXSTEP.

MESHELL

Mesh Density for Covering Ellipsoids

Mesh density for covering rigid ellipsoids.

Format and Example	**Default**
PARAM,HESHELL,value	5
PARAM,MESHELL,8	

Option	**Meaning**	**Type**
value	Mesh density for covering ellipsoids.	I > 3

Remark:

1. The default is sufficient in most cases. When this value is increased, the representation of the (hyper)ellipoids is better, but the archives created will be larger.

MESHPLN

Mesh Density for Covering Rigid Planes

Mesh density for covering rigid planes.

Format and Example	Default
PARAM,HESHPLN,value	3
PARAM,MESHPLN,4	

Option	Meaning	Type
value	Rigid planes will be meshed with MESHPLN times MESHPLN dummy quad elements.	I > 1

Remark:

1. The default is sufficient in most cases.

Defines the accuracy of the initial conditions in Eulerian elements, when using the geometrical shape definition.

Format and Example	**Default**
PARAM,MICRO,value	10
PARAM,MICRO,15	

Option	**Meaning**	**Type**
value	Micro-zoning parameter.	I > 0

Remarks:

1. MICRO^3 is the number of micro zones into which an element is subdivided during initial condition generation.
2. The default MICRO = 10 results in material fractions as accurate as 0.001. If a higher accuracy is required, a greater value for MICRO can be used, but the CPU time for the generation increases rapidly.
3. Micro zoning is only used when the initial conditions of the Eulerian material are specified on a TICEUL entry.

MINSTEP

Minimum Time Step

Defines the minimum time step that causes the analysis to terminate.

Format and Example	Default
PARAM,MINSTEP,value PARAM,MINSTEP,1.E–6	10% of INISTEP

Option	Meaning	Type
value	When the time step is less than the MINSTEP value, the analysis terminates.	R > 0.0

Remarks:

1. When the elements become very distorted, in a poorly designed mesh for example, or when they have endured a very large distortion, the time step may drop dramatically. The analysis continues, however, and a lot of computer resources may be wasted. This option allows you to specify a minimum time step that causes the analysis to terminate.

2. See Section 3.12 on page 3-70 for details on analysis termination.

NASIGN

Echo Ignored Data Entries

Toggles the echo of valid MSC.NASTRAN and/or MSC/DYNA data entries that are ignored by MSC.Dytran.

Format and Example	Default
PARAM,NASIGN,value	YES
PARAM,NASIGN,NO	

Option	Meaning		Type
value	YES	Echo ignored entries.	C
	NO	Do not echo.	

Remarks:

1. The echo of the ignored data entries is output to a file with the extension IGN.
2. Large input that originates from MSC.NASTRAN or MSC.Dytran may produce a large amount of output and slow down the input processing.

PARALLEL

Parallel Excecution

Defines that parallel parts of the program will be executed on more than one CPU on Shared Memory Parallel SGI systems (R10000 processors).

Format and Example

PARAM,PARALLEL,option,value/option

PARAM,PARALLEL,INFPAR,[ON/OFF]
PARAM,PARALLEL,MAXCPUS,value

Option	Meaning	Type	Default
INFPAR	A report is written on the actual amount of work done at the reported parallel levels.	C	OFF
MAXCPUS	Defines the number of processors that are used in the parallel parts of MSC.Dytran	I	1

Remarks:

1. Before starting MSC.Dytran in parallel mode, type:

   ```
   %setenv NCPUS nn
   %setenv MP_SET_NUMTHREADS nn
   ```

   at the shell prompt. The number ‘nn’ must be equal to or greater than the value entered on the PARAM,PARALLEL,MAXCPUS entry.

   Note that MP_SET_NUMTHREADS is the environment variable (specific for SGI) to turn on the number of processors available for processing. If MP_SET_NUMTHREADS is not set, then all the available physical processors in the system will be used.

2. Available parallel parts of MSC.Dytran are the Belytschko-Tsay-Lin (BLT), Hughes-Liu, and Key-Hoff Quad shell element solvers.

3. A summary on the parallel operation when using the shared-memory mode can be requested by including a PARAM,PARALLEL,INFPAR,ON entry in the input file. This request is not available in a restart of an analysis.

PLCOVCUT

Pressure Cut Off Time

Defines time when PLCOVER will be cut off.

Format and Example	Default
PARAM,PLCOVCUT,value	0.0
PARAM,PLCOVCUT,3.E-3	

Option	Meaning	Type
value	If there are one or more COUPLE definitions with a PLCOVER specified on the COUOPT entry, a cut off will be applied to the PLCOVER until time = PLCOVCUT. From time = 0 to time = PLCOVCUT, the PLCOVER will be cut off to the pressure in the intersected Eulerian element. For times greater than PLCOVCUT, the full PLCOVER will be applied to the coupling (SUB)SURFACE. This parameter is useful in air-bag analyses, where PLCOVER is used to model the environment pressure. During the early stages of the deployment of the air bag the pressure inside the bag may drop. Applying the full PLCOVER may lead to an unstable deployment of the air bag.	R

Remarks:

1. See also the COUPLE and COUOPT Bulk Data entries .

PMINFAIL

Switches Failure at Spall Limit

Defines Lagrangian solid element failure on reaching the spall limit.

Format and Example	**Default**
PARAM,PMINFAIL,option	NO
PARAM,PMINFAIL,YES	

Option	**Meaning**	**Type**
YES	Element failure on spall limit.	C
NO	No element failure on spall limit.	

Remarks:

1. Lagrangian elements (CHEXA) that have a material definition with a failure model will fail when the parameter is set to YES and the spall limit (minimum pressure) is reached, even when the other failure criterion is not yet reached.

2. The spall limit is set on the PMINC entry. (See also the DMAT entry).

RBE2INFO

Lists MATRIG and RBE2 Grid Points

The grid points attached to MATRIG and RBE2 assemblies are listed to the output file.

Format and Example	**Default**
PARAM,RBE2INFO,option	GRIDOFF
PARAM,RBE2INFO,GRIDON	

Option	**Meaning**	**Type**
GRIDON	Information is issued.	C
GRIDOFF	No information is issued.	

RHOCUT

Global Density Cut-Off Value

Defines the minimum density for all Eulerian elements.

Format and Example	**Default**
PARAM,RHOCUT,value	See Remark 4.
PARAM,RHOCUT,1.E–10	

Option	**Meaning**	**Type**
value	Density cutoff.	R > 0.0

Remarks:

1. Any Eulerian element with a density less than RHOCUT is considered to be empty. All of its variables are set to zero, and the equation of state is bypassed.
2. In the Eulerian transport calculation, if the material is flowing from element A to element B, and
   a. If the density of element B after transport is less than RHOCUT, then no transport is done.
   b. If the density of element A after transport is less than RHOCUT, then all of the mass is transported to element B.
3. A reasonable value of RHOCUT is 1.E–5 times the initial density.
4. If only RHOCUT is defined, all Eulerian elements use the RHOCUT value as cut-off density. If RHOCUT is omitted, all Eulerian elements use a cut-off density automatically set to 1.E–5 times a characteristic density. For single-material Eulerian elements, this characteristic density is the reference density.

RKSCHEME

Runge-Kutta Time-Integration Scheme

Defines the typeof time-integration scheme used in the Roe solver.

Format and Example	Default
PARAM,RKSCHEME,number	See Remark 1.
PARAM,RKSCHEME,3	

Option	Meaning	Type
number	Number of Runge-Kutta stages.	I > 0

Remarks:

1. This parameter can only be used in combination with PARAM,LIMITER,ROE. The default number of stages depends on the spatial accuracy of the solution scheme. One stage is used for first order spatial accuracy, and three stages for second order spatial accuracy.

2. For more details on the Runge-Kutta time-integration scheme see Section 2.21 on page 2-136.

RJSTIFF

Rigid-Joint Stiffness

Defines the stiffness of a rigid joint.

Format and Example	**Default**
PARAM,RJSTIFF,value	1.0
PARAM,RJSTIFF,100.	

Option	**Meaning**	**Type**
value	Multiplication factor for the stiffness of all rigid joints.	R > 0.0

Remarks:

1. The absolute stiffness of rigid joints is calculated automatically by MSC.Dytran. The stiffness of joints is taken so that a stable solution is guaranteed. The stiffness calculation takes into account the fact that a rigid body can be constrained by more than one joint.
2. This parameter can be used to increase or decrease the stiffness of the joints. Care must be taken because too high a value may lead to an unstable calculation.

ROHYDRO

Density Cut-Off Value

Defines the minimum density for hydrodynamic, single-material Eulerian elements.

Format and Example	Default
PARAM,ROHYDRO,VALUE	See Remark 3.
PARAM,ROHYDRO,1.E–6	

Option	Meaning	Type
value	Density cutoff.	R > 0.0

Remarks:

1. Hydrodynamic, single-material Eulerian elements with a density less than ROHYDRO are considered to be empty. All of the variables are set to zero, and the equation of state is bypassed.
2. In the Eulerian transport calculation, if the material is flowing from element A to element B, and
   a. If the density of element B after transport is less than ROHYDRO, then no transport is done.
   b. If the density of element A after transport is less than ROHYDRO, then all of the mass is transported to element B.
3. By default, the cutoff density for hydrodynamic Eulerian elements is set to 1.E–5 times the material reference density.

ROMULTI

Density Cut-Off Value

Defines the minimum density for multimaterial Eulerian elements.

Format and Example	Default
PARAM,ROMULTI,VALUE	See Remark 3.
PARAM,ROMULTI,1.E–6	

Option	Meaning	Type
value	Density cutoff.	R > 0.0

Remarks:

1. Multimaterial Eulerian elements with a density less than ROMULTI are considered to be empty. All of the variables are set to zero, and the equation of state is bypassed.
2. In the Eulerian transport calculation, if the material is flowing from element A to element B, and
   a. If the density of a specific material in element B after transport is less than ROMULTI, no transport is done.
   b. If the density of a specific material in element A after transport is less than ROMULTI, all of the mass of that material is transported to element B.
3. By default, the cut-off density is set for each material separately as 1.E–5 times the material reference density.

ROSTR
Density Cut-Off Value

Defines the minimum density for single-material Eulerian elements with shear strength.

Format and Example	Default
PARAM,ROSTR,VALUE	See Remark 3.
PARAM,ROSTR,1.E–6	

Option	Meaning	Type
value	Density cutoff.	R > 0.0

Remarks:

1. Single-material Eulerian elements with shear strength with a density less than ROSTR are considered to be empty. All of the variables are set to zero, and the equation of state is bypassed.
2. In the Eulerian transport calculation, if the material is flowing from element A to element B, and
   a. If the density of element B after transport is less than ROSTR, then no transport is done.
   b. If the density of element A after transport is less than ROSTR, then all of the mass is transported to element B.
3. By default the cut-off density for Eulerian elements with shear strength is set to 1.E–5 times the material reference density.

RSTDROP

Type of Elements Dropped at Restart

Defines the type of elements to be removed from the calculation when restarting an analysis.

Format and Example	Default
PARAM,RSTDROP,option	No elements are dropped.
PARAM,RSTDROP,LAGRANGE	

Option	Meaning	Type
LAGRANGE	Lagrangian solid elements are dropped.	C
EULER	Eulerian elements are dropped.	
MEMBRANE	Membrane elements are dropped.	
SURFACE	Rigid bodies and coupling surfaces are dropped.	

Remarks:

1. All elements of the specified type are removed from the calculation. It is not possible to drop a part of a Eulerian or Lagrangian mesh.

2. If Lagrangian solids or membranes are dropped from a coupled calculation, the surfaces should also be dropped to prevent surfaces being present that are not attached to anything.

3. The EULER option only works for a Eulerian mesh containing a single hydrodynamic material.

SCALEMAS

Mass Scaling Definition

Defines the activation of mass scaling.

Format and Example	Default
PARAM,SCALEMAS,DTMIN,MXPERC,STEPS	See Remark 3.
PARAM,SCALEMAS,1E-6, 100.0,1	

Option	Meaning	Type
DTMIN	Minimum allowable time step.	R > 0.0 See Remark 1.
MXPERC	Maximum percentage of added numerical mass with respect to original mass.	R ≥ 0.0 See Remark 1.
STEPS	Number of steps.	I > 0 See Remark 2.

Remarks:

1. Numerical mass will be added to all Lagrangian solid, triangular, quadrilateral, rod, bar and beam elements such that its time step will never become less than:

   ```
   dt = STEPFCT*DTMIN.
   ```

   where dt = timestep of calculation

   STEPFCT = timestep safety factor (see PARAM,STEPFCT)

   DTMIN = value specified on the PARAM,SCALEMAS entry

   If the added mass of a certain element exceeds the maximum percentage (MXPERC) of its original mass, no more mass will be added, and subsequently, the time step may decrease again.

2. The value of STEPS determines the checking frequency against the mass-scaling criterion: the check will be done every defined number of STEPS. STEPS = 1 is recommended.

3. The values for DTMIN, MXPERC, and STEPS are required input.

4. By requesting MSMASS in an output request, the ratio of scaled mass to original mass of the elements can be retrieved. By making fringe plots of this parameter, a check can be made if mass has not been added in a critical area.

5. See Section 2.16 on page 2-125 for instructions on how to use this entry.

SHELLFORM

Sets the Default of the Shell Formulation

Sets the default for the shell formulation for quadrilateral elements.

Format and Example	**Default**
PARAM,SHELLFORM,option	KEYHOFF
PARAM,SHELLFORM,BLT	

Option	**Meaning**	**Type**
BELY	The shell-formulation default is BELY.	C
BLT	The shell-formulation default is BLT.	
KEYHOFF	The shell-formulation default is KEYHOFF.	

Remarks:

1. The PARAM,SHELLFORM changes the default formulation for quadrilateral shell elements. All shell properties entries that do not explicitly define the formulation, use the default as specified on the PARAM entry.
2. Triangular shell elements have only one formulation (C0-TRIA). Therefore, the PARAM will be ignored for triangular elements.
3. For more information, see also Section 2.15 on page 2-118 on application-sensitive defaults.

SHELMSYS

Shell Element System Definition

Defines the shell element system for the BLT shells.

Format and Example

PARAM,SHELMSYS,option

PARAM,SHELMSYS,SIDE21

Option	Meaning	Type	Default
SIDE21	x-axis along side21.	C	MIDSIDES
MIDSIDES	x-axis connecting midpoints.		

Remarks:

1. SIDE21 puts the x-axis along side21 of the element, whereas MIDSIDES puts the x-axis along the vector connecting the midpoints of the side14 and side32.
2. Using the SIDE21 option for the BLT shell will result in the same Belytschko-Lin-Tsay implementation as BELY.

SHPLAST

Type of Plane-Stress Plasticity for Shells

Specifies the type of calculation used to determine the plane-stress plasticity method for shells.

Format and Example	Default
PARAM,SHPLAST,option	ITER
PARAM,SHPLAST,VECT	

Option	Meaning	Type
RADIAL	Noniterative, approximate radial return.	C
VECT	Iterative, vectorized with three iterations.	
ITER	Nonvectorized iterations.	

Remarks:

1. The RADIAL approach does not require iterations and, therefore, is the most efficient. It is, however, an approximation.

2. The other two approaches iterate to find the solution. ITER is the best since it takes as many iterations as are necessary. On vector machines, such as CRAY or IBM-3090, this is inefficient since it cannot be vectorized. VECT always performs three vectorized iterations, which is more efficient. However, three iterations may not be enough, and inaccuracies could occur.

3. For more information, see Section 2.15 on page 2-118 on application-sensitive defaults.

SHSTRDEF

Composite Shell Stress and Strain Output Definition

Specifies the default coordinate system for the stress and strain output of composite shells.

Format and Example

PARAM,SHSTRDEF,option

PARAM,SHSTRDEF,ELEM

Option	Meaning	Type	Default
FIBER	Stresses and strains are output in the fiber and matrix directions.	C	FIBER
ELEM	Stresses and strains are output in the element coordinate system.		

Remarks:

1. The default setting can be overruled per property on a PCOMPA entry on the STRDEF field.

SHTHICK

Shell-Thickness Modification Option

Specifies whether or not the thickness of the shell changes with membrane straining.

Format and Example	Default
PARAM,SHTHICK,option	YES
PARAM,SHTHICK,YES	

Option	Meaning	Type
YES	Shell thickness is modified according to the membrane strain.	C
NO	Shell thickness is constant.	

Remarks:

1. The YES option gives a true large-strain shell but requires some extra computation.
2. The NO option should give adequate results as long as the membrane strains are not very large (i.e., not more than 5–10%).
3. This option applies to all the formulations of the shell elements, except for the PCOMP. The thickness of PCOMP shell elements will always remain constant.
4. For more information, see Section 2.15 on page 2-118 on application-sensitive defaults.

SLELM

Store Shell Sublayer Variables

Defines whether shell sublayer variables are to be stored in the element arrays.

Format and Example	Default
PARAM,SLELM,option	YES
PARAM,SLELM,NO	

Option	Meaning	Type
YES	Store as an element variable.	C
NO	Do not store as an element variable.	

Remarks:

1. This parameter applies only to shell elements.
2. The shell sublayer variables are primarily stored in sublayer arrays. They can be copied into the element arrays only for specific output purposes.
3. Specifying NO reduces the CPU overhead time.
4. Irrespective of the entry on this parameter, sublayer variables are accessible in the sublayer arrays.

   For example, requesting TXX1 retrieves the stress from the element array, whereas TXX01 retrieves it from the sublayer arrays.
5. NASTRAN initialization always causes SLELM = YES.

SNDLIM

Sound Speed Minimum Value

Defines the minimum value for the speed of sound.

Format and Example	**Default**
PARAM,SNDLIM,value	1.E–3
PARAM,SNDLIM,1.E–6	

Remarks:

1. This parameter is used to avoid the possibility of division by zero in the time-step calculation.
2. SNDLIM has the units of velocity.

STEPFCT

Time-Step Scale Factor

Defines a scale factor to be used on the internally calculated time step.

Format and Example	**Default**
PARAM,STEPFCT,value	0.666
PARAM,STEPFCT,0.5	

Option	**Meaning**	**Type**
value	Time-step safety factor.	$0.0 < R \leq 1.0$

Remarks:

1. The actual time step used in MSC.Dytran is the product of the internal time step and the time-step safety factor.
2. The default value works well in the majority of situations and gives an efficient solution while maintaining a stable solution.
3. In a calculation with a coupling surface, FBLEND must be greater than or equal to STEPFCT to avoid instabilities (see PARAM,FBLEND).
4. For many calculations, STEPFCT can be set to 0.9, unless you are running a problem that has coupling surfaces defined.

STRNOUT

Shell Sublayer Strain Output

Saves the total strains and equivalent effective stress (von Mises stress) at shell sublayers for output.

Format and Example	Default
PARAM,STRNOUT,option	YES
PARAM,STRNOUT,YES	

Option	Meaning	Type
YES	Save.	C
NO	Do not save.	

Remarks:

1. A limited output set saves memory.
2. Perfectly elastic materials only have the limited output set.
3. Total strain output for shell composite materials can be requested from the PCOMPA Bulk Data entry.

USA_CAV

USA Cavitation Model

Defines a cavitation model for the pressure given by USA.

Format and Example	Default
PARAM, USA_CAV,option,value PARAM, USA_CAV,YES,-300.	No cavitation model.

Option	Meaning		Type
option	NO	No cavitation model is defined.	C
	YES	Cavitation model is defined.	
value	Cavitation pressure.		R

Remark:

1. When the pressure given by USA falls below the cavitation pressure this pressure is set equal to the cavitation pressure.
2. Cavitation model can not be used when USA surface is defined by 1-D element type of grid points.

VARACTIV

Activation or Deactivation of Grid-Point, Element, or Face Variables

Grid point, element, or face variables are activated or deactivated by the Bulk Data entry, as well as the definition of new user variables. The name of the variable is redefined as well.

Format and Example

PARAM,VARACTIV,(elname),(GEF),(varname),(datatype),(ACTDEAC),(newname)

PARAM,VARACTIV,LAGSOLID,ELEM,MASS,FLT,ACTIVE,ELMASS renames the variable MASS to ELMASS.

PARAM,VARACTIV,LAGSOLID,ELEM,NEWVAR,FLT,ACTIVE,NEWVAR creates a new variable NEWVAR as an element float value.

Option	Meaning		Type	Default
elname	Name of the element.		C	Required
	ELEM1D	One-dimensional elements.		
	SHTRIA	Triangular shell.		
	SHQUAD	Quadrilateral shell.		
	MEMTRIA	Membrane.		
	DUMTRIA	Triangular dummy element.		
	DUMQUAD	Quadrilateral dummy element.		
	LAGSOLID	Lagrangian solid.		
	EULHYDRO	Eulerian hydrodynamic solid.		
	EULSTRENGTH	Eulerian solid with stress tensor.		
	MULTIEULHYDRO	Multimaterial Eulerian solid.		
	ALL	Activate all variables.		
	ALLPRINT	Activate all variables and print a summary.		
GEF	Entity name.		C	ELEM
	GRID	Grid point.		
	ELEM	Element.		
	FACE	Face.		
varname	Name of the variable.		C	Required

(Continued)

Option	Meaning		Type	Default
datatype	Data type of the variable.		C	FLT
	INT	Integer.		
	FLT	Float.		
	CHAR	Character.		
ACTDEAC	Activate or deactivate variable.		C	ACTIVE
	ACTIVE	Activate a variable.		
	DEACTIVE	Deactivate a variable. (See Remark 4.)		
newname	Redefined name of the variable. (See Remark 3.)		C	Required

Remarks:

1. The ALL entry will activate all variables for all elements, regardless of whether they are used or not. The ALLPRINT will, additionally, print a summary of all element variables, regardless whether there are any elements of a certain type or not. If ALL or ALLPRINT is entered, then no subsequent entries are required.

2. All entries are required if ALL or ALLPRINT are not specified, except for newname, which defaults to the original name.

3. When a variable is renamed, all subsequent references must be made to the new name, e.g., in output requests.

4. The deactivate option is potentially dangerous, since some options may require the variable in an indirect way. It is advised, therefore, not to deactivate standard MSC.Dytran variables.

5. This PARAM entry is a convenient way to introduce new variables to an entity, which can be used in user subroutines. The new variables will be written to restart files and can be requested for output.

6. See Subroutine EEXOUT as an example of how to address a newly created variable.

(Continued)

7. In the print file of an MSC.Dytran run, a summary will be given for all variables of the element types that are used in the calculation (except when ALLPRINT was specified). The format of the printout is:

   ###–(CHAR)–NAME

   where ### = the variable ident number when the variable is active. If the variable has been deactivated ### will be printed as <~>

   CHAR = N if the variable is standard MSC.Dytran

   = R if the variable has been renamed

   = C if the variable has been defined by the user

   = E if the variable is used for editing only

   NAME = the (new) name of the variable

VDAMP

Dynamic Relaxation Parameter

Controls the global damping in the dynamic relaxation.

Format and Example	Default
PARAM,VDAMP,value/option	0.0
PARAM,VDAMP,0.001	
PARAM,VDAMP,OFF	

Option	Meaning	Type
value	Dynamic relaxation parameter.	R ≥ 0.0
OFF	No dynamic relaxation.	C

Remarks:

1. The dynamic relaxation parameter is connected to the system natural frequency, ω, as $\beta = \varepsilon\omega\Delta t$, where ε denotes a percentage of critical damping. The damping occurs by factoring the velocities every time step as follows:

$$F_1 = (1 - \beta)/(1 + \beta)$$

$$F_2 = 1/(1 + \beta)$$

$$v^{n + 1/2} = F_1 v^{n - 1/2} + F_2 a^n \Delta t^n$$

   where v is the velocity, a is the acceleration, and β is the dynamic relaxation parameter.

2. At the restart of an analysis with dynamic relaxation, the dynamic relaxation can be switched off by PARAM,VDAMP or PARAM,VDAMP,OFF.
3. For a more comprehensive description of dynamic relaxation, see Section 2.12 on page 2-109.

VELCUT

Velocity Cutoff

Defines the minimum velocity.

Format and Example	Default
PARAM,VELCUT,value	1.E–6
PARAM,VELCUT,1.0E–6	

Option	Meaning	Type
value	Minimum velocity.	R > 0.0

Remark:

1. Any velocity less than VELCUT is set to zero. It is mainly used to eliminate harmless but annoying small values of velocity caused by round-off error and numerical dispersion.

VELMAX
Maximum Velocity

Defines the maximum velocity in Eulerian and Lagrangian meshes.

Format and Example	**Default**
PARAM,VELMAX,value,YES/NO	1.E10,YES
PARAM,VELMAX,1.E6	

Option	**Meaning**	**Type**
value	Maximum velocity.	R > 0.0
YES	Remove the mass in Eulerian elements in which the velocity exceeds the maximum specified velocity.	C
NO	Do not remove the mass in Eulerian elements in which the velocity exceeds the maximum specified velocity.	

Remarks:

1. Although it is not usually necessary to limit the velocity in Eulerian meshes, there are occasions, in regions of near-vacuous flow for example, where specifying a maximum velocity can be advantageous. The same applies to Lagrangian meshes, in contact regions for example. This parameter should be used with care.

2. Because very high velocities occur mostly in Eulerian elements with very small mass, the mass in these elements can be removed to keep the analysis stable. This option is not available for Lagrangian solid elements.

3. VELMAX must be greater than the minimum velocity specified by PARAM,VELCUT.

CHAPTER 5

Diagnostic Messages

Whenever MSC.Dytran encounters invalid or inconsistent data in the input file or a problem is encountered during the analysis, a diagnostic message is printed.

Diagnostic messages produced during the initialization and solution are written to the <jobname>_ERROR_SUMMARY-MSG file. These messages normally indicate incorrect or inconsistent data and problems encountered during the solution.

Each diagnostic message is produced a maximum of five times to prevent large quantities of output from being produced.

The diagnostic message is a set of short codes that indicate the severity of the message, its number, and the subroutine that generated it. One or more lines of text follows, indicating what the problem is.

The coded line has the basic form:

%x–<diagnostic number>–<subroutine name>

where x indicates the severity:

I	Information
W	Warning
E	Error
F	Fatal
C	Catastrophic

Information messages do not indicate a problem, and the analysis should continue successfully.

Warnings are not fatal, and the analysis will continue. However, warnings are an indication that something about analysis is not normal. You should review all warnings carefully and make sure you know what is causing the message.

```
%W–P3007905–P3XXTXX_CYCLE_ZERO
Authorization files will expire this month
```

Error messages indicate that there is almost certainly something wrong with your analysis. You should review your input and modify it appropriately. Errors in the solution cause termination at the end of the current time step. If you specified output at the end of the analysis, then the output is produced before the analysis terminates.

```
%E–CN000602–SYS_CHECK_RST_INPUT_FILE
Restart input file must have extension .RST
```

Fatal messages have the same effect as error messages but indicate a more serious problem.

```
%F–2039502–P2XXTXIX_PACKET_EDIT
Materials cannot be put on archive files, only on time history files.
```

Catastrophic errors are issued only when the program would otherwise crash. They may occur, for example, when the analysis would result in a division by zero.

```
%C–P2057301–C2_STORE_IZONER_FOR_CFACE
Face number –4 is illegal
```

The severity code letter is followed by the unique diagnostic number. There is a unique number for each diagnostic message and it can be used to reference the message. The subroutine name indicates which routine produced the diagnostic message. Note that the routine names in MSC.Dytran consist of up to 31 characters.

Internal program errors have the message (PROGRAM ERROR) at the end of the first line of the diagnostic message, e.g.,

```
%E–CN000502–SYS_CHECK_RST_INPUT_FILE (PROGRAM ERROR)
RESTART INPUT FILE HAS WRONG EXTENSION
```

You should never get program errors. If you do, check if there are other diagnostic messages indicating other problems. If not, please note the diagnostic number and contact your local MSC representative.

CHAPTER 6

References

1. *MSC/DYNA User's Manual*, The MacNeal-Schwendler Corporation.

2. *MSC/PISCES-2DELK User's Manual*, The MacNeal-Schwendler Corporation.

3. E. L. Lee et al., "Adiabatic Expansions of High Explosive Detonation Products," UCRL-50422, May 1968, Lawrence Livermore National Laboratory, Livermore, California.

4. *MADYMO User's Manual 3-D Version 4.3*, The Dutch Institute of Applied Scientific Research - Road/Vehicles Research Institute, Department of Injury Prevention, October 1988.

5. Louise A. Obergefell, Thomas R. Gardner, Ints Kaleps, and John T. Fleck, *Articulated Total Body Model Enhancements*, Volume 1: "Modifications," (NTIS No. ADA198726).

6. Louise A. Obergefell, Thomas R. Gardner, Ints Kaleps, and John T. Fleck, *Articulated Total Body Model Enhancements*, Volume 2: "User's Guide," (NTIS No. ADA203566).

7. Louise A. Obergefell, Thomas R. Gardner, Ints Kaleps, and John T. Fleck, *Articulated Total Body Model Enhancements*, Volume 3: "Programmer's Guide," (NTIS No. ADA197940).

8. J. T. Fleck, F. E. Butler, and N. J. Deleys, "Validation of the Crash Victim Simulator," Calspan Report Nos. ZS-5881-V-1 through 4, DOT-HS-806-279 through 282, 1982, Volumes 1 through 4, (NTIS No. PC E99, PB86-212420).

9. *USA User's Manual*, John A. DeRuntz.

10. J. A. DeRuntz, T. L. Geers, and C. A. Felippa, *The Underwater Shock Analysis Code (USA-Version 3)*, A Reference Manual, September 1980.

11. P. L. Roe, *Approximate Riemann Solvers, parameter vectors, and difference schemes,* Journal of Computational Physics, 43, 357-372, 1981.

12. P. L. Roe and J. Pike, *Efficient construction and utilisation of approximate Riemann solutions,* Computing Methods in Applied Sciences and Engineering (VI), R. Glowinski and J. L. Lions (Editors), Elsevier Publishers B.V. (North Holland), INRIA 1984.

13. Bram van Leer, Chang-Hsien Tai, and Kenneth G. Powell, *Design of Optimally Smoothing Multi-Stage Schemes for the Euler Equations,* AIAA-89-1933-CP, 1989.

14. Ch. Hirsch, *Numerical Computation of Internal and External Flows, Fundamentals of Numerical Discretization, 1,* Wiley, Chichester, 1988.

15. Ch. Hirsch, *Numerical Computation of Internal and External Flows, Computational Methods for Inviscid and Viscous Flows, 2,* Wiley, Chichester, 1990.

APPENDIX A

Using XDYTRAN

A.1 Introduction

XDYTRAN is an X11/Motif-based application for UNIX systems that can be used to “launch” MSC.Dytran runs and to create customized versions of the MSC.Dytran executable. It allows you to select input files or user-source routines interactively, or to create a customized version of the memory-allocation routine memory.f, by pressing a button and entering parameters in the user interface. XDYTRAN also enables you to preview output files while the program is running.

A.2 Features of XDYTRAN

XDYTRAN provides the following features:

- Submission of MSC.Dytran jobs.
- Automatic detection of RESTART statements in the input file.
- Selection of ATB-input files.
- Killing, suspending, and forced wrap-ups of running MSC.Dytran jobs.
- Customization of executables with user routines and/or modified memory allocations.
- Interactive creation of the memory-allocation routine memory.f.

A.3 Execution of XDYTRAN

To run XDYTRAN, type:

```
xdytran
```

A.3.1 Main Window

The XDYTRAN main window consists of a menu bar and three subwindows.

A.3.1.1 Main Window Menu Bar

Pressing the “Select” mouse button (usually the left button) while the pointer is on one of the buttons on the menu bar makes a pull-down menu appear. While pressing the “Select” button, scroll down the menu until the item you want to activate is highlighted; then release the button.

You can also click the “Select” mouse button while the pointer is on one of the buttons, and the menu will remain displayed until you click the “Select” mouse button again. If you click on a menu item, it will be activated. If you click outside the menu, the menu will disappear without taking any other action.

You can also display pull-down menus using the keyboard by holding down the <Alt> key and typing the underlined letter in the menu name (e.g., typing <Alt>-f displays the File menu).

Main Window Help Menu

Selecting the MSC logo on the right-hand side of the menu bar displays the Help menu. You can select “About xdytran” from this menu to see a short message containing a copyright notice and instructions on how to use XDYTRAN’s online help facility.

A.3.1.2 Main Window Subwindows

At the top of the main window is the “Process Info” subwindow, which contains information about the “Current Process.”

At the lower left you can find the “Process” window containing icons for each of the processes (the MSC.Dytran runs) and three buttons to create, delete, or modify a process. The control panel at the lower right allows the user to start, suspend, kill, and wrap up the “Current” process.

A.3.2 Creating an MSC.Dytran Job

Selecting the “Create Process” button at the lower left of the main window will create a new dialog window, consisting of four parts. At the left side there is the “Select Files” subwindow, at the top right the “Process Info” subwindow, and in the middle on the right the “Output Files” subwindow. At the bottom there are two buttons: the “OK” button to accept the current input and a “Cancel” button to reject everything. Both buttons will remove the “Create/Modify Process” subwindow and set control back to the main window.

A.3.2.1 Selecting Files

The “Select Files” subwindow allows the specification of the input file, the output directory, the executable, and an ATB-input file to be used with an MSC.Dytran run.

1. Bring the cursor to the “What to Select” button, keeping the left mouse button pressed. A list with the four types of files that can be selected will pop up.

2. Bring the cursor to the desired type and release the left mouse button. Now the “Select Files” subwindow is configured for the selection of the type of file displayed on the “What to Select” button.

   The “Select Files” subwindow contains four input/output fields. At the top is the “File Name Pattern” field, displaying the pattern used to filter the files that are listed in the “Files” field. This pattern can be compared with search patterns used with the UNIX “ls” command. For example, “*.dat” will display all files in the current directory with the extension “.dat”. In the “Files” field all filenames are listed that match the search pattern. The “Directories” field at the left shows all subdirectories of the current directory.

3. Use the scroll bars at the right and the bottom to display the covered parts of the “Files” and “Directories” fields. The search pattern is automatically adapted to the type of file to be selected, for example “*.dat” for input files, “*.exe” for an executable, and “*.ain” for the ATB-input file.

   For the actual selection, you can either type in the complete pathname in the “Selected...” field or use the “Directory” and “Files” fields by double clicking with the left mouse button on the item to be selected.

4. Click on the “..” subdirectory to go up one level. The “Show Files” button can be used to activate the filename filtering when the directory has been changed.

5. Finally, click on the file to be selected, it will be displayed in reverse video. Then click on the “OK” button. This will also update the output fields in the other subwindows.

A.3.2.2 The Process Info Subwindow

The process info subwindow displays information about the selected files:

Process Title	The title found on the input file.
Input File	The full pathname of the selected input file.
Output Directory	The complete pathname of the selected output directory (defaults to the current directory).
Executable Name	The full pathname of the selected executable (defaults to the standard MSC.Dytran executable).
ATB Filename	The full pathname of the selected ATB-input file (if any).

Note that the fields in this subwindow cannot be used to change the names of the selected items; they are only for displaying the items set by the “Select Files” subwindow.

A.3.2.3 The Output-files Subwindow

The output-files subwindow has four input/output fields. The “Output File Prefix” shows the prefix that will be used for all files generated by MSC.Dytran. For example if the prefix is ABC, archive files will be named ABC_name_cycle.ARC, where “name” is the name defined in the input file, and “cycle” is the number of the first cycle on the archive file. The prefix defaults to that of the input file, converted to uppercase letters, but it can be changed to anything you like. The “Print File Prefix” field can be used to specify an extra prefix for the print-output file. For example, if the output-file prefix is ABC and a print-file prefix of ZYX has been defined, the print-output file will be named ABC_ZYX.OUT. There is no default. The other two fields in the subwindow are used to display the names of the print file and the error file. These fields can not be used to change those names.

A.3.2.4 Accepting/Rejecting the Selected File Names

When the selection is complete, you can either accept or reject the input by clicking on the “OK” or “Cancel” buttons along the bottom of the “Create/Modify” pop-up window. When “OK” is clicked, XDYTRAN will create a “Process” with an icon that is displayed in the “Processes” subwindow of the main window. Both the “OK” and the “Cancel” button will bring you back to the main window.

A.3.3 Selecting a Process

Each process has an icon in the “Processes” subwindow. To select a process just click on the appropriate icon. The following information about the selected process will be displayed in the “Process Info” subwindow:

Process Title — The title found on the input file.

Input File — The full pathname of the selected input file.

Output Directory — The complete pathname of the selected output directory
(defaults to the current directory).

Executable Name — The full pathname of the selected executable
(defaults to the standard MSC.Dytran executable).

ATB Filename — The full pathname of the selected ATB-input file (if any).

Current Status — Status of the selected process; for example, “Not started,” “Running,” “Suspended,” “Killed,” or “Finished”.

A.3.4 Modifying a Process

To modify a process, first select the process by clicking on its icon. Then click on the “Modify Process” button at the bottom of the main window. This will bring up the same selection window as when “Create Process” is clicked, except that, instead of the defaults, all fields will be set to the values that were previously defined.

A.3.5 Deleting a Process

To delete a process, first select the process by clicking on its icon. Then, click on the “Delete Process” button. This removes the icon and all related definitions and settings for the process within XDYTRAN (the input and output files are not removed). Once a process is removed from XDYTRAN, it cannot be referred to later, unless it is redefined.

A.3.6 Previewing Output Files

When “View Normal Output” or “View Error Output” are clicked in the “Control Panel” subwindow, XDYTRAN will open a window to display the contents of the print-output file or the error-output file of the current process. Use the scroll bars along the right side and the bottom to browse through the files. With the “Close Windows” button the scroll bars can be removed from the screen. XDYTRAN will update the contents when more output is generated by the running process.

A.3.7 Executing a Process

Make the process to be executed the current process by clicking on its icon. Then click on the “Run” button in the “Control Panel” subwindow to activate it. The “Run” button will become the “Suspend” button, and also the “Kill” and “Wrapup and Exit” buttons will be made sensitive. These buttons have the following actions:

Kill	Can be used to stop the running process immediately. This may cause loss of output data.
Suspend	Will suspend the running process. It can be re-activated later on.
Wrapup and Exit	Sends a request to a running MSC.Dytran job to stop at the end of the current cycle. If there is any request for output at wrap-up (restart, archive files) in the input file, this output will be generated.

A.3.8 Quitting XDYTRAN

To quit XDYTRAN, bring the cursor to the “File” button at the top left of the main window and press the left mouse button. Keep the mouse button pressed and move the cursor downward until “Quit” is highlighted. Then, release the mouse button. Another option is to press the <Control> key on the keyboard together with the “q”.

A.3.9 Customizing MSC.Dytran

To customize MSC.Dytran, you must first activate the “Customize MSC.Dytran” dialog window by placing the cursor on the “File” button at the top left of the main window and pressing the left mouse button. Keep the mouse button pressed and move the cursor downward until “Customize” is hightlighted, and release the mouse button. The “Customize MSC.Dytran” dialog window is similar to the “Create/Modify Process” dialog window. At the left side, there is a subwindow to select an output directory to store the customized executable or the name of a source file with user routines to be linked in. At the right is a subwindow to set the sizes of the buffer arrays in the memory-allocation routine “memory.f,” and a subwindow displaying the names of the selected files. Along the bottom are the “OK” and “Cancel” buttons.

A.3.9.1 Selecting Files

The “Select File” subwindow can be used similarly to the “Select File” submenu in the “Create/Modify Process” dialog window. With “What to Select”, you can choose between selecting the output directory for the customized executable or the name of the source file containing user subroutines. Note that since the customize process is based on the standard MSC.Dytran link script, it is not possible to define both an output directory and a source file, because the script will always store the executable in the same directory as the source file.

A.3.9.2 Customizing Memory

The “Customize Memory” subwindow contains four fields where you can type new values for the sizes of the buffer arrays in the memory allocation routine, memory.f. Click on the “Reset to Defaults” button to reset all four to their standard default values. When one or more of the input fields have been changed, XDYTRAN will write out a new version of memory.f in the same directory where the customized executable will be generated. It will automatically be linked in by the MSC.Dytran link script.

A.3.9.3 Selected File

The “Selected Files” subwindow displays the names of the source file with the user routines and the directory where the executable will be stored. This subwindow is for output only and cannot be used to change the names.

A.3.9.4 Accepting/Rejecting the Selections for Customization

When the selection is completed, you can either accept or reject the input by clicking on the “OK” or “Cancel” buttons at the bottom of the “Customize MSC.Dytran” pop-up window. Both the “OK” and “Cancel” buttons will bring you back to the main window. When “OK” is clicked, XDYTRAN will create a “Process” with an icon that is displayed in the “Processes” subwindow. Submitting a customized process from the main window is similar to submitting a regular MSC.Dytran job. Also, the log file can be previewed in the same way as the print-output file generated by MSC.Dytran.

A.4 Limitations

When using XDYTRAN, be aware of the following limitations:

- With the current version of XDYTRAN, it is not possible to define RESTART files explicitly outside the input file. However, all RESTART files defined in the input file will be taken into account by MSC.Dytran.
- When there is an existing memory.f routine in the directory where a customized executable has to be generated, it cannot be overwritten by a new version with the default buffer sizes. XDYTRAN will only write out a new “memory.f” if one or more of the buffersizes are different from the defaults. You should either manually remove the existing version outside XDYTRAN, or specify one or more buffer lengths slightly different from the default values.

APPENDIX B

Using XDEXTR

B.1 Introduction

XDEXTR is an X-Window/Motif-based application that translates information stored in the archive and time-history files produced by MSC.Dytran to various third-party import file formats, including the MSC.Patran import file format. XDEXTR can also be used noninteractively through the use of a translation-description file.

XDEXTR can be run as often as necessary as long as the MSC.Dytran archive or time-history files still exist. Since the import files produced by XDEXTR are in text format, they can be used to transfer data from one computer to another even if the computers are of different types. XDEXTR can therefore be run on one machine, and the import files can be moved to another machine for postprocessing with MSC.Patran or another third-party application package.

Features of MSC/XDEXTR

XDEXTR provides the following features:

- Translation of MSC.Dytran archive files to MSC.Patran, MSC/XL, SDRC/I-DEAS, and ETA/FEMB formats.
- Viewing of archive-file contents.
- User selection of output directory.
- Minimum/maximum value plotting for variables in archive files.
- Entity-position filtering of translation.
- Cycle filtering of translation.
- Full geometry to translated-output option.
- Translation of MSC.Dytran archive files to time-history translation.
- Batch-mode functionality using translation-description files (TDF).

B.2 Executing XDEXTR

To run XDEXTR, type:

```
xdextr
```

B.2.1 Main Window

The main XDEXTR window selects a file for translation.

B.2.1.1 Main Window Menu Bar

Pressing the “Select” mouse button (usually the left button) while the pointer is on one of the buttons on the menu bar makes a pull-down menu appear. While pressing the “Select” button, move down the menu until the menu item you want to activate is highlighted; then release the button.

You can also click the “Select” mouse button while the pointer is on one of the buttons, and the menu will remain displayed until you click the “Select” mouse button again. If you click on a menu item, it will be activated. If you click outside the menu, the menu will disappear.

You can also display pull-down menus using the keyboard by holding down the <Alt> key and typing the underlined letter in the menu name (e.g., typing <Alt>-f displays the File menu).

File Option

Selecting “Quit” from the File menu exits XDEXTR.

Help Option

Selecting “About xdextr” from the Help menu displays a window showing the current version of XDEXTR, a copyright notice, and instructions on how to access the help feature in XDEXTR.

B.2.1.2 File Selection

Select the results file that you wish to translate by clicking the “Select” mouse button (usually the left button) on its name in the Files window, then clicking the “OK” button (or you can double-click on the filename).

Once you have selected a file, XDEXTR creates a new window that displays the contents of the results file, and lets you select the variables, entities, and cycles that you want to translate.

You can navigate around the directory structure to find the results file you want by double-clicking on entries in the “Directories” list. Double-clicking on a directory name displays the names of any results files contained in that directory in the “Files” list. Double-clicking on the “..” entry moves up one level in the directory hierarchy. You can also edit the contents of the “File Name Pattern” (but don’t change the ‘.[ATR][RHS][CST]’ part) and then press <Return>, or click the “Show files” button to go directly to a particular directory.

B.2.2 Results File Window

This displays archive information and allows filtering of variables, entities, and cycles.

B.2.2.1 Results File Menu Bar

Pressing the “Select” mouse button (usually the left button) while the pointer is on one of the buttons on the menu bar makes a pull-down menu appear. While pressing the “Select” button, scroll down until the menu item you want to activate is highlighted; then release the button.

You can also click the “Select” mouse button while the pointer is on one of the buttons, and the menu will remain displayed until you click the “Select” mouse button again. If you click on a menu item, it will be activated. If you click outside the menu, the menu will disappear.

You can also display pull-down menus using the keyboard by holding down the <Alt> key and typing the underlined letter in the menu name (e.g., typing <Alt>-t displays the Translate menu).

Window Option

Selecting “Close” from the Window menu closes the results file window and releases the memory that XDEXTR uses to hold information about this results file.

Do Option

This menu button has three option selections:

Selecting “Translate” from the “Do” menu displays a pop-up window that lets you set parameters for the translation of the results file with information that is displayed in this window. Selecting “Filter by variable values” displays a pop-up window that lets you set parameters for the filtering of the results file based on variable values. At present, this option is only available for archive files. Selecting “Filter by entity positions” displays a pop-up window that lets you set parameters for the filtering of the results based on initial entity positions.

B.2.2.2 Information Header

This displays information about the archive such as heading, file type, creation date, and creation machine.

B.2.2.3 Entity Types

This is the list of entity types that are present within the selected archives.

Each entity type is represented by two bodies of text separated by a hyphen. The body of text to the left of the hyphen is the basic entity type while the body of text to the right of the hyphen gives more specific information about the entity type. The entity types represented for archive files are:

CBEAM	Beam Element.
CBAR	Bar Element.
CROD	Rod Element.
CELAS	Spring Element (6 DOF).
CSPR	Spring Element (3 DOF).
CDAMP	Damper Element (6 DOF).
CVISC	Damper Element (3 DOF).
CTRIA3	Triangular shell element.
CQUAD4	Quadrilateral shell element.
CTRIA3-M	Triangular membrane element.
CTRIA3-D	Triangular dummy element.
CQUAD4-D	Quadrilateral dummy element.
CHEXA-L	Langrangian solid element.
CHEXA-EH	Eulerian solid element (Hydrodynamic).
CHEXA-ES	Eulerian solid element (Shear strength).
CHEXA-MM	Eulerian solid element (Multimaterial).

The entity types represented for time-history files are:

ELEMENT	Element results.
GRID POINT	Grid-point results.
MATERIAL	Material results.

RIGID	Rigid-body results.
GASBAG	Gas-bag results.
ELLIPSOID	Rigid-ellipsoid results.
CONTACT	Contact results.
CROSS SECTION	Cross-section results.

For archive files that were created with MSC.Dytran prior to Version 2.3, the entity-type list would consist of one entry with the name “UNKNOWN”. This indicates that the entity-type name was not stored for this archive files.

By default, all entity types are selected, and the Entities list displays the entire list of entities. Any number of entity types can be selected, and their entities can be viewed in the Entities list. Dragging the mouse while depressing the “Select” mouse button (usually the left button) or dragging the mouse while depressing the <Shift> key and “Select” mouse button together selects all entity types between the entity type under the pointer, and the entity type on which the “Select” mouse button was pressed. This does not deselect any other selected entity types outside the selected range.

Selection and deselection of discontiguous ranges of entity types (to view the entity types they represent) is also possible. Depressing the <Control> key and the “Select” mouse button on an entity type toggles its selection state but does not deselect any other selected entity type. Dragging the <Control> key and the “Select” mouse button together can toggle ranges of all entity types between the entity type under the pointer and the entity type on which the “Select” mouse button was pressed. This does not deselect any other selected entity types outside the selected range. The contents of the “Entities” list reflects the selection state of the “Entity types” list. That is, the only entities shown in the “Entities” list are those of the set of types selected in the “Entity types” list.

Select Types

Pressing the “Select types” button selects all entities of the set of selected types. This allows you to quickly make selection of entities based on entity type or a group of entity types, rather than selecting the entities using the slider.

B.2.2.4 Variables

This is the list of variables that are available for translation. Only variables that are selected will have values written to the output files. All variables are selected by default (selected variables are highlighted).

You can select variables in several different ways:

1. Single select: Clicking the “Select” mouse button on a variable deselects all selected variables, and selects only the one you clicked on.

2. Multiple select: Pressing the “Select” mouse button and moving through the list deselects all currently selected variables, and then selects each variable that you drag the cursor through.

3. Extended select: Holding down the <Control> key and clicking on a variable toggles the selection state of that variable, leaving currently selected variables selected.

Holding down the <Control> key and stepping through the list selects each variable you move through, leaving currently selected variables selected.

B.2.2.5 Entities

This is the list of entities that are available for translation. Only entities that are selected will have values written to the output files. You can always tell which entities are selected by looking at the contents of the “Selected entities” list.

If the entities are named (not grid points or elements), you can select them from this list by double-clicking on them with the Select mouse button (usually the left button). Otherwise, you can use the sliders to the right of this list to select the entities.

Please note that the sliders have a predefined resolution of what value each increment step of the slider can have. This resolution depends upon the range of values the sliders have to represent. If the range that the sliders have to represent is large, then the minimum step of the sliders is large. To overcome this problem of accessing intermediate values, the arrow keys can be used to increment/decrement the slider that has the focus (the slider that has been highlighted with the mouse or <Tab> key) by single unit steps. Incrementing/decrementing in steps of ten is also possible by using the <Control> and arrow keys together or by clicking with the “Select” mouse button while above/below the slider block on the slider.

B.2.2.6 Selected Entities

This is the list of entities that have been selected and will have values written to the output files. You can select entities in one of two ways depending on whether they are numbered or named entities (named entities are anything except grid points and elements).

If the entities are named, you can select them from the Entities list by double-clicking the “Select” mouse button on the entity you wish to select. The entity name will then be copied into the “Select entities” list.

If the entities are numbered, you can use the sliders to the right of this list to select the entities. For more information on the use of sliders, see Section B.2.2.5 on page B-6.

B.2.2.7 Cycle

This is the list of cycles that are available for translation. Only cycles that are selected will have values written to the output files. You can always tell which cycles are selected by looking at the contents of the “Selected cycles” list.

You can select cycles from this list by using the sliders on the right-hand side of the window. For more information on the use of sliders, see Section B.2.2.5 on page B-6.

B.2.2.8 Selected Cycle

This is the list of cycles that have been selected and will have values written to the output files. Cycle selection is achieved by the use of sliders. For more information on the use of sliders, see Section B.2.2.5 on page B-6.

B.2.2.9 Select

This selects the use of the sliders and filtering with respect to either entities or cycles. To filter either entities or cycles, press the appropriate button using the “Select” mouse button before selection.

B.2.2.10 Selection Sliders

These sliders are used to select numbered entities (grid points and elements) or cycles for translation. Simply set the “From” slider to the ID of the first entity or cycle you wish to select, the “To” slider to the ID of the last entity or cycle you wish to select, and the “By” slider to an increment. When you click the ”Select” mouse button (usually the left button) on the “Select” button, all entities/cycles in the range you specified will be added to the selection set and their IDs will be displayed in the “Selected entities” or “Selected cycles” list, respectively.

You can also use the sliders to deselect a range of entities or cycles by setting the sliders to a range, then clicking on the “Deselect” button. For more information on the use of sliders, see Section B.2.2.5 on page B-6.

Another alternative is to type values directly into the input window below each slider. This can be achieved by choosing the input window below the slider you wish to set a value for by using the “Select” mouse button (usually the left button). A value can then be typed for the slider and, if valid, the sliders adjust to the new value given in the value window. If the input value is invalid, then the value is highlighted. The value can then be deleted and retyped. Moving any of the sliders immediately clears the value window to which that slider belongs.

B.2.2.11 Select/Deselect Buttons

This allows the selection/deselection of a range of entities or cycles using the input sliders.

B.2.3 Variable Filtering Window

This window lets you specify parameters for the filtering of the results file by variable values.

B.2.4 Min/Max Value Sliders

Min/max value sliders allow the input of the minimum and maximum values to be filtered for the variable chosen in the variables window. When no variable is selected, the sliders and value windows are grayed out. Upon selection of a variable, they are activated. The slider on the left represents the minimum filter value, and the slider on the right represents the maximum filter value. The ranges of the maximum and minimum scales are set to the maximum and minimum of the variable range by default. The minimum scale is set the minimum value, and the maximum scale is positioned to its maximum value by default. Moving either slider causes the value window by the side of each window to update its value relative to the sliders position.

Values may also be typed directly into the value windows, and the sliders adjust to the new value set in the value window automatically. Selection is achieved by moving the slider by using the “Select” mouse button to the appropriate value and by pressing the “Enter filter” or “Apply all and close” buttons, depending on whether more filters are to be added or not. Selecting the “Coarse” and “Fine” buttons displays coarse and fine adjustments to the minimum and maximum values. Pressing the “Fine” button zooms the maximum and minimum of the scale to ten percent of the range either way of the slider’s current value. Pressing the “Coarse” button returns you to the maximum and minimum allowable range for the variable.

For more information on the use of sliders, see Section B.2.2.5 on page B-6.

B.2.4.1 Graph Window

The graph displays the maximum and minimum values of an archive variable across the cycle range. The graph displays the maximum and minimum for a variable when a variable is selected using the “Select” mouse button in the Variables window. If the maximum values equal the minimum values, then the graph is not displayed and the last valid graph displayed remains in the graph window. The graph allows a global view of the variable data in an archive and allows better evaluation of the variable filtering applied.

B.2.4.2 Set Range

The two buttons allow you to filter the value range selected with the sliders to either within the range or outside of the range specified. This option should be selected before applying the filter.

B.2.4.3 Filters

This displays the list of variable filters that have been selected for the translation of this archive. Upon pressing the “Enter filter” button, the defined filtering value settings appear in the window. This allows you to keep track of the filters defined for this variable.

B.2.4.4 Variables

This displays the list of variables within the selected archive. Selecting one of the variables using the “Select” mouse button sets the sliders to display the maximum and minimum values found for this variable and displays the graph of the selected variable across the cycle range. Filtering by variable values is then possible for this variable.

B.2.4.5 Variable Filter Buttons

Selecting “Apply all and close” registers all the filters defined in the Filter window and closes the window. The filters will be applied upon translation of the archive. Selecting “Enter filter” stores the variable filter setting to the Filter window and allows the setting of another variable filter definition. Selecting “Cancel all and close” closes the window and registers no filters defined in the Filter window. No variable filtering will be applied upon translation of the archive.

B.2.5 Entity Position Filter

The “Entity Position Filter” lets you choose which entities are to have their results written to the output files according to their initial grid-point positions.

B.2.5.1 Input Value Sliders

These allow the input of the minimum and maximum values for the filtering of initial entity coordinates. This allows selection of entities according to their initial coordinate position and translates only those entities that have been initially selected. If a coordinate axis is not used, the sliders and value windows are grayed out. The slider on the left represents the minimum filter value and the slider on the right represents the maximum filter value. The ranges of the maximum and minimum scales are set to the maximum and minimum of the entity range by default. The minimum scale is set the minimum value and the maximum scale is positioned to its maximum value by default. Moving either slider causes the value window by the side of each window to update its value relative to the slider's position.

Values may also be typed directly into the value windows and the sliders will adjust to the new value set in the value window automatically. Selection is achieved by moving the slider by using the "Select" mouse button to the appropriate value and by pressing the "OK" or the "Cancel" buttons depending on confirming or canceling the entity filtering settings respectively. The "Coarse" and "Fine" buttons allow coarse and fine adjustment to the minimum and maximum values displayed. Pressing the "Fine" button zooms the maximum and minimum of the scale to ten percent of the range either way of the slider's current value. Pressing the "Coarse" button returns you to the maximum and minimum allowable range for the variable.

For more information on the use of sliders, see Section B.2.2.5 on page B-6.

B.2.5.2 Entity Filter Buttons

Selecting "OK" registers the initial entity position settings with the application. Only values with the initial entity positions that are within the ranges specified will be translated. Selecting "Cancel" cancels the entity filtering option and closes the window.

B.2.6 Translate Window

This window lets you specify parameters for the translation of the results file.

B.2.6.1 Select Output Directory

You can use the "File Selection Box" to select the directory where the output files from the translation will be placed.

You can navigate around the directory structure to find the directory you want by double-clicking on entries in the "Directories" list. Double-clicking on a directory name makes that directory the output directory for translation files. Double-clicking on the ".." entry moves up one level in the directory hierarchy. You can also edit the contents of the "File-Name Pattern" and then press <Return> or click on the "Show files" button to go directly to a particular directory.

A pull-down option menu button exists in the "File Selection Box" to select directories for the translation output or the TDF (translation description file). The mode of file selection is changed by depressing the "Select" mouse button and moving the mouse to the required option. Selecting the "Translation Description" selects the name of the TDF file and selecting the "Output Directory" option selects the directory for the translated output. The default is set to "Output Directory."

B.2.6.2 Output Type

You can use these buttons to select whether you want to translate the data into time history, archive, or SAE output format. Depending on the type of results file, some of these options may not be available. For example, you cannot translate a time-history file into archive format, since time history files do not contain geometry data.

B.2.6.3 Format

This selects the format for translation. There are four formats to select from at present:

- Clicking the “Select” mouse button (usually the left button) on “XL” selects MSC/XL as the output format for this translation.
- Clicking the “Select” mouse button on “I-DEAS” selects I-DEAS as the output format for this translation.
- Clicking the “Select” mouse button on “FEMB” file selects FEMB as the output format for this translation.
- Clicking the “Select” mouse button on “PATRAN” selects MSC.Patran as the output format for this translation.

B.2.6.4 Options

There is only one option at present, and this is the full-geometry option.

When translating an archive file to archive format, the geometry is written to a separate file. Normally, when some subset of the available entities is selected, only the geometry for those entities will be written to the geometry file. For elements, only the geometry of those grid points that are connected to some selected element will be written. In XL format, grid-point displacements for each time step are only written for those grid points that are connected to some selected element. If you want the geometry and displacements for every grid point and element to be written to the geometry file, click the “Select mouse” button on the “Full geometry” button. When the rectangle to the left of the button text shows green, the full-geometry option is active. Note that upon selection of the full-geometry option, all initial grid-point filtering selections (made with the “Filter by entity positions” options) will be ignored.

B.2.6.5 Output Directory

This displays the name of the directory where results files will be placed.

B.2.6.6 Translation-Description File

Selecting “Write TDF” causes XDEXTR to write a description of the current translation to the named TDF file when you select “OK” in the “Translate” dialog box. The TDF file will be overwritten or appended to, according to the settings of the radio buttons.

B.3 Using XDEXTR Without the Graphical User Interface

For translations that are done repeatedly or on a remote machine where the graphical user interface imposes significant network overhead, XDEXTR can be driven by a translation-description file (TDF) and run as a batch process. This capability replaces DEXTR.

Using the graphical user interface, you can create a TDF that corresponds to a translation you have set up, including all selections and filters. You can recreate this translation later by running XDEXTR and specifying the TDF name on the command line.

Since a TDF is just a text file, you can also create one using your text editor. Section B.3.2 on page B-14 describes the format of a TDF.

B.3.1 The XDEXTR Command Line

XDEXTR accepts several command-line options:

-f <TDF name>

Specifying this option with the name of a file containing well-formed translation descriptions causes XDEXTR to read the descriptions in the file and perform them in sequence. No windows will be displayed and no prompts will be issued.

-confirm

Under some circumstances, XDEXTR will ask for confirmation for operations that may be harmful in some way. For example, if a translation will overwrite files in the output directory, XDEXTR would normally ask you to confirm the operation. Since running XDEXTR under the control of a TDF is intended to be a batch process that requires no interaction, it will stop with a System Warning Message if it encounters a condition that requires confirmation. Specifying “-confirm” on the command line causes XDEXTR to suppress any System Warning Messages and proceed with operations that would normally require confirmation.

-help

Specifying “-help” on the command line causes XDEXTR to print a brief summary of the available command-line options.

B.3.2 Format of a TDF

A TDF is a free-format text file containing translation descriptions. Each translation description specifies all the settings for a translation, such as the selected entities, output format, etc. In case a setting is not specified in the TDF, the following defaults apply:

```
Translation type: Archive
Output format: PATRAN
Output directory:.
Selected entities: all
Selected variables: all
Selected cycles: all
Variable value filters: no filters active
Entity position filter: none
Cutoff frequency: 60.0
Time factor: 0
```

Inside a TDF, spaces or TABs separate tokens, and all keywords and identifiers are case insensitive except for UNIX pathnames. Any part of a line beginning with an exclamation mark (!) or hash mark (#) is considered a comment and is not interpreted. Here is a BNF syntax for a TDF file:

```
TDF file:: <Translation Description List>

<Translation Description List>:: <Translation Description>
                              :: <Translation Description List> <Translation
                                  Description>

<Translation Description>     :: Translation {<File Spec> <Translation Type
                                 Spec> <Optional Spec List>}

<File Spec>                   :: File ”<full pathname>”;

<Translation Type Spec>       :: TranslationType <Translation Type>;

<Translation Type>            :: Archive
                                 TimeHistory
                                 SAEFilter

<Optional Spec List>          :: <Optional Spec>
                                 <Optional Spec List> <Optional Spec>

<Optional Spec>               :: <Output Format Spec>
                                 <Output Directory Spec>
                                 <Selected Entities Spec>
                                 <Selected Variables Spec>
                                 <Selected Cycles Spec>
                                 <Entity Position Filter Spec>
                                 <Variable Value Filter>
                                 <Cutoff Frequency>
                                 <Time Units>
                                 <Nothing>
```

```
<Output Format Spec>          :: OutputFormat <Output Format>;

<Output Format>               :: patran
                                 xl
                                 femb
                                 ideas

<Output Directory Spec>       :: OutputDirectory ”<pathname>”;

<Selected Entities Spec>      :: SelectedEntities {<Entities List>}

<Entities List>               :: <Name List>
                                 <Number List>

<Name List>                   :: Name
                                 <Name List>,Name

<Number List>                 :: Integer
                                 <Number List>,Integer

<Selected Variables Spec>     :: SelectedVariables {<Name List>}

<Selected Cycles Spec>        :: SelectedCycles {<Number List>}

<Entity Position Filter Spec> :: EntityPositionFilter {<xmin> <xmax> <ymin>
                                 <ymax> <zmin> <zmax>}

<xmin>                        :: xmin Float
<xmax>                        :: xmax Float
<ymin>                        :: ymin Float
<ymax>                        :: ymax Float
<zmin>                        :: zmin Float
<zmax>                        :: zmax Float

<Variable Value Filter>       :: VariableValueFilter {<Filter Spec>}

<Filter Spec>                 :: Name <InOut> <Range Spec>

<InOut>                       :: Inside
                                 Outside

<Range Spec>                  :: Float Float
<Cutoff Frequency>            :: Cutoff Integer

<Time Units>                  :: TimeFactor Integer
```

B.3.3 Sample TDF

The following TDF translates cycles 12 through 250 of the archive file BLANK_DISTR.ARC to MSC.Patran format. The output files are placed in directory /sample/tdf. The entity-position filter is commented out, so it is not used in the translation. Any entities for which the variable THICK has a value greater than 0.9 and less than 0.92 will not be written to the output file.

```
!
! this is a sample translation description file
!
Translation”BLANK_DISTR.ARC” {
outputformat xl;
TRANSLATIONTYPE archive;
OutputDirectory”/sample/tdf”;
#EntityPositionFilter {xmin 44 xmax 78.00
#                       ymin 0.0 ymax 10000.0
#                       zmax 7.6 zmin 1.266}
VariableValueFilter {thick outside 0.9 0.92}
SelectedCycles {12 through 250}
}
```

B.4 Files

The output files produced by XDEXTR are dependent on which XDEXTR function is used.

dextr.msg

In all cases, a user-message file is produced that contains all the necessary information concerning the last translation session in the output directory.

MSC/XL

The output produced is mostly dependent on whether the file to be translated is an archive or time-history file.

Files Produced During Archive Translation

<identifier>_<step number>.ext

A formatted ASCII file containing the translated contents of an MSC.Dytran archive file. This file is in the correct format to be used by MSC/XL (i.e., XL "import" file) to produce deformed/undeformed plots, contour plots, and vector/arrow plots.

<identifier>_post.dat

A formatted ASCII file containing the geometry of the problem as contained in the archive file. This is in the correct MSC.Dytran format to be read directly into MSC/XL if the database, which was created during preprocessing, is not available.

In the above filenames, <identifier> is the identifier of the archive file being translated.

Translations using variable filtering will result in a "_post.dat" file created for every cycle translated. This is to accommodate the entity set changing for each cycle as the entity values fall in and out of the selected range.

Files Produced During Time-History Translation

TH_<variable name>_<entity identifier>_<entity id>.ext

A formatted ASCII file containing the translated contents of an MSC.Dytran time-history file. This file is in the correct format to be used by MSC/XL (i.e., XL “import” file) to produce X-Y plots.

where <variable name> = the user-specified variable name to be plotted along the Y-axis (time is plotted along the X-axis).

<entity identifier> = “EL” for elements.
“GP” for grid points.
“GL” for global entity (e.g., rigid body).

<entity id> = ID of entity/name of entity.

SDRC/I-DEAS

The output produced for I-DEAS is always in universal File Format. This format contains the information required to produce deformed/undeformed plots, contour plots, and vector/arrow plots.

Translations are possible only for archive files, and one universal file is produced for each archive in the filename specifier <identifier>.unv.

ETA/FEMB

The output produced for FEMB is in Animation History File format. This format contains the information required to produce deformed/undeformed plots, contour plots, and animations. Translations are possible only for archive files.

File Produced During Archive Translation

<identifier>_<group number>_ani.his

A formatted ASCII file containing the translated contents of an MSC.Dytran archive file. Each group number represents a group of six or less variable results found in the MSC.Dytran archive file that has been translated.

<identifier>_post.nas

A formatted ASCII file containing the geometry of the problem as contained in the archive file. This is in the correct MSC.Dytran format to be read directly into ETA/FEMB if the database, which was created during preprocessing, is not available.

In the above filenames, <identifier> is the identifier of the archive file being translated.

Translations using variable filtering will result in a_post.nas file created for every cycle translated. This is to accommodate the entity set changing for each cycle as the entity values fall in and out of the selected range.

MSC.Patran

The output produced is mostly dependent on whether the file to be translated is an archive or time-history file.

File Produced During Archive Translation

<identifier>_<step number>.neu

A formatted ASCII file containing the translated geometry of an MSC.Dytran archive file.

<identifier> _ <step number>.els

A formatted ASCII file containing the translated element results of an MSC.Dytran archive file.

<identifier>_<step number>.xtr

A formatted ASCII file containing the extrapolated element results to grid-point results, which was extracted from an MSC.Dytran archive file.

Files Produced During Time History Translation

TH_<variable name>_<entity identifier>_<entity id>.pat

A formatted ASCII file containing the translated contents of an MSC.Dytran time-history file. This file is in the correct format to be used by MSC.Patran (i.e., Patran “import” file) to produce X-Y plots.

where <variable name> = the user-specified variable name to be plotted along the Y-axis (time is plotted along the X-axis).

<entity identifier> = “EL” for elements.

“GP” for grid points.

“GL” for global entity (e.g., rigid body).

<entity id> = ID of entity/name of entity.

B.5 Help Feature

A comprehensive help feature is included in XDEXTR. Access the help feature in XDEXTR by pressing the F1 key on your keyboard. The cursor will change from the normal cursor to a question mark cursor. Clicking the left-hand button on any area of the application will display a pop-up help window containing context-sensitive help information.

B.6 Known Problems

This section contains known errors that occur with XDEXTR. It lists the specified area in which the problem or enhancement occurred (Specific Area), the platform for which this problem occurs (Platform), and a description of the problem (Description).

Specific Area	Platform	Description
Motif/XDEXTR	CRAY	At runtime, warning messages are produced by Motif, but the application is performing correctly.

APPENDIX C

MSC.Dytran and Parallel Processing

C.1 Introduction

Since MSC.Dytran calculations can use a large amount of time and computing resources, optimization is an important part of the development of MSC.Dytran. One such optimization effort is in the area of parallel processing, which attempts to speed up a calculation by harnessing the combined power of a computer with several processors or several complete computers.

MSC.Dytran will run certain kinds of calculations in parallel if the right kind of hardware is available and if you request it to do so.

C.2 Shared Memory

Since MSC/DYTRAN calculations can use a large amount of time and computing resources, optimization is an important part of the development of MSC.Dytran. One such optimization effort is in the area of parallel processing, which attempts to speed up a calculation by harnessing the combined power of a computer with several CPUs.

MSC.Dytran will run certain kinds of calculations in parallel if the right kind of hardware is available and if you request it to do so. Shared memory parallelism has been built into MSC.Dytran for shared memory parallel configurations SGI with R10000. See the description of PARAM,PARALLEL on page 4-502 for detailed information about how to request MSC/Dytran to run in parallel.

If you are running on SGI machines that support shared-memory parallel operations, calculation on Belytschko-Lin_Tsay (BLT), Hughes-Liu, or Key-Hoff Quad shell elements will run in parallel on the number of CPUs requested on the PARAM,PARALLEL entry, see also the description on page 4-502.

The Parallel Execution Report

After completion of a job, a report may be written about the actual and potential distribution of work among the concurrent processes. This will show information such as the fraction of work executed on a particular number of processors. Note that for different runs this report may show different work distribution, caused by differences in overall load of the system.

To write out the parallel execution report, include a PARAM,PARALLEL,INFPAR,ON entry in your input data file. At this moment, the parallelization information option is available for BVS quad shells solvers only. The outline of the report can be seen as follows

```
*************************************************************************
*   *** INFORMATION ON PARALLELIZATION
*
*  CPU TIME IN PARALLEL SECTION :   0.23610E+03
*  AVERAGE PARALLEL CPUS        :   0.19929E+01
*  MAXIMUM PARALLEL CPUS        :   0.20000E+01
*  MINIMUM PARALLEL CPUS        :   0.10000E+01
*
*--FURTHER PARALLEL INFORMATION PER PROPERTY
*
*  PROPERTY NAME                : SHELL1
*  CPU TIME IN PARALLEL SECTION :   0.23610E+03
*  NUMBER OF CPUS ALLOCATED     :           2
*  CPU NUMBER                   :           1
*  CALLS INTO PARALLEL SECTION  :      143601
*  PERCENTAGE OF PARALLLEL WORK :   0.50307E+02
*  AVERAGE CALLS (MAXIMUM)      :   0.20000E+01
*  AVERAGE CALLS (MINIMUM)      :   0.10000E+01
*  NUMBER OF ELEMENTS PROCESSED :   0.91905E+07
*
*  CPU NUMBER                   :          2
*  CALLS INTO PARALLEL SECTION  :     141851
*  PERCENTAGE OF PARALLLEL WORK :   0.49693E+02
*  AVERAGE CALLS (MAXIMUM)      :   0.20000E+01
*  AVERAGE CALLS (MINIMUM)      :   0.10000E+01
*  NUMBER OF ELEMENTS PROCESSED :   0.90785E+07
*
*  PROPERTY AVERAGE OF WORK (%) :   0.10000E+03
*  PROPERTY AVERAGE OF TIME (%) :   0.10000E+03
*****************************************************************************
```

Note that this CPU time information is only measured in parallel region. In this example the average parallel CPUs is close to the maximum two CPUs available. It indicates that the workload of the two CPUs used are in balance. More detailed information per CPU such as % parallel work, number of calls or elements processed, etc. are also included.

C.3 Distributed Memory

Distributed memory parallel computing is not supported by MSC.Dytran.

APPENDIX D

Using ATB

D.1 Introduction

In order to perform analyses of occupant interaction with structures, the occupant-modeling program ATB (Version 4.2) is included in MSC.Dytran (References 5–8). This is not just a procedure for running one program, transferring results, and then running the other program. MSC.Dytran and ATB run concurrently, exchanging data as the analysis proceeds.

Contrary to running MSC.Dytran with MADYMO (See Appendix E), it is not necessary to create a special executable, because ATB is built as standard in MSC.Dytran. It is activated only when required.

D.2 Input Specification

ATB

The input specification for ATB (Reference 6) does not change. The ATB-input file name (with standard extension ".ain") must be specified by the user. This is done in the ATB field in XDYTRAN (See Appendix A), or on the command line of the MSC.Dytran script by using the keyword "atb =" followed by the ATB-input filename (without extension). Both methods will command MSC.Dytran to activate ATB.

The preprocessor GEBOD generates ATB input of various known dummy models like Hybrid II and Hybrid III. This interactive program also enables you to create dummy models with user-supplied dimensions.

Some useful options have been added to the ATB input file:

- If a nonzero value is set in the NDMI field (I4) right after the BDYTTL field on entry B.1, default moments of inertia are supplied to every segment for which PHI is set to zero on entry B.2.a. The defaults are the moments of inertia of a general ellipsoid and depend on the weight and the semi-axes of the ellipsoid, as specified on entry B.2.a.
- If a negative value for NFVNT is supplied on entry D.9, the force coordinate system is defined with respect to the basic coordinate system instead of the local coordinate system of the segment. This option can be used to define a time-dependent gravity field in a fixed-basic direction.
- If a nonzero value is set in the NSJF field (I4) right after the I3 field on entry G.1.a, initial joint forces are subtracted from internal forces. Thus, it is possible to create an initial equilibrium after positioning the dummy. Care should be taken when using this option.

Despite above-mentioned additions, any original ATB-input file is still valid.

Moreover, positioning of the dummy can be handled by MSC.Dytran. For a description, see the Bulk Data entries ATBJNT, ATBSEG, and the PARAM entry ATBSEGCREATE.

Additional copies of References 5-8, where Reference 6 serves as ATB User Manual, may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
UNITED STATES OF AMERICA
Telephone: (703) 487-4600

MSC.Dytran

Each segment contact ellipsoid used in the occupant model must be defined in the MSC.Dytran input file with the RELEX entry. The names of these ellipsoids must be the same as the segment names as defined in the ATB-input file on the B.2 entry. The use of the RELEX entry is described in Section 4.7.

Only the RELEX entry should be used. Internal ellipsoids defined by the RELLIPS entry cannot be used in combination with those defined by RELEX. A mixture of MADYMO and ATB ellipsoids is not allowed.

The interaction between the ATB ellipsoids and the structural parts of the MSC.Dytran model is achieved through the standard contact and connection entries CONTREL and RCONREL. This interaction applies to all structural elements: quadrilateral and triangular shells, membrane elements, Lagrangian solids, and rigid bodies.

D.3 Termination Conditions

All termination conditions selected in MSC.Dytran remain valid. If a termination condition occurs, ATB is forced into an end cycle, and MSC.Dytran terminates as usual.

ATB-termination conditions cannot be used to stop the analysis.

D.4 Postprocessing

ATB

ATB results can be postprocessed with the standard ATB postprocessors.

MSC.Dytran

The data relating to the rigid ellipsoids can also be output from MSC.Dytran using the standard Case Control command RELOUT.

If the ellipsoid geometry is requested in an archive file, the ellipsoids are covered with dummy shell elements at each requested output step. These dummy elements are written to the archive file. In this way the geometry, position, and orientation of the ellipsoids can be visualized within the deformed MSC.Dytran mesh.

APPENDIX E

Using MSC.Dytran With MADYMO

E.1 Introduction

In order to perform analyses of occupant interaction with structures, MSC has developed a system of coupling MSC.Dytran with the occupant modeling program MADYMO (Reference 4). This is not just a procedure for running one program, transferring results, and then running the other program. MSC.Dytran and MADYMO run concurrently, exchanging data as the analysis proceeds.

The coupling between MSC.Dytran and MADYMO in MSC.Dytran Version 4.7 uses a direct coupling. The MSC.Dytran libraries and MADYMO libraries are linked into one executable.

MSC/Dytran Version 4.7 supports MADYMO 5.3.

E.2 Input Specification

Interaction

The interaction between MSC.Dytran finite element entities and MADYMO ellipsoids is supported. The interaction is achieved through the standard contact and connection entries CONTREL, and RCONREL, in the MSC.Dytran input file. The interaction applies to all structural elements:

- spring and damper elements
- shell elements
- membrane elements
- Lagrangian solid elements
- rigid bodies (MATRIG, RBE2, and RIGID)

Special entries need be inserted in both the MADYMO and the MSC.Dytran input file to activate the coupling.

MADYMO Input

The following section needs to be added to the MADYMO input file:

```
FORCE MODELS
  COUPLING
    ELLIPSOIDS
    <label> <system> <ellipsoid>
    END ELLIPSOIDS
    PLANES
    <label> <system> <plane>
    END PLANES
  END COUPLING
END FORCE MODELS
```

where `label` = a user-defined label to identify the ellipsoid or plane. The label is transferred to MSC.Dytran to identify the ellipsoid or plane.

`system` = the system ID to which the ellipsoid or plane belongs.

`ellipsoid` = the ellipsoid ID.

`plane` = the plane ID

See the MADYMO 5.3 User’s Manual 3D for more details.

MSC.Dytran Input

The following entries need to be added to the MSC.Dytran input file:

1. Definition of MADYMO ellipsoids.

   `RELEX,<name>,MADYMO,<label>`

   where `name` = a user-defined name of an ellipsoid. The name does not have to correspond to a name of an ellipsoid in the MADYMO input file. The name is used within the MSC.Dytran input file to define interactions with finite element entities and for output requests.

   `label` = a user-defined label to identify the ellipsoid. The label is used by MSC.Dytran to identify the ellipsoid. The label value must be the same as the label value used in the MADYMO input file.

   Using the `RELEX,<name>,MADYMO` entry in combination with a `RELEX,<name>,ATB`, or `RELLIPS` entry is not allowed.

2. Definition of MADYMO planes.

   `RPLEX,<name>,MADYMO,<label>`

   where `name` = a user-defined name of a plane. The name does not have to correspond to a name of a plane in the MADYMO input file. The name is used within the MSC.Dytran input file to define interactions with finite element entities and for output requests.

   `label` = a user-defined label to identify the plane. The label is used by MSC.Dytran to identify the plane. The label value must be the same as the label value used in the MADYMO input file.

   Using the `RPLEX,<name>,MADYMO` entry in combination with a `RELEX,<name>,ATB`, or `RELLIPS` entry is not allowed.

E.3 Time-Step Control

The stable time step of the analysis is usually determined by MSC.Dytran. The time step is then enforced on MADYMO. It is possible, however, that MADYMO needs to run with a time step less than the stable MSC.Dytran time step. The situation is detected when it occurs, and MADYMO will then enforce its time step on MSC.Dytran. When this happens, it shows in the MSC.Dytran output file in the time-step summary that will contain the keyword MADYMO, for example:

```
NCYCLE NSUB   TIME     TOTAL-E     NZ     DLTH      SSPD    VELOCITY    LMIN

 1962  0  9.805E-02 3.7432E+00 MADYMO 5.000E-05  .000E+00  .000E+00  .000E+00

 1963  0  9.810E-02 3.7517E+00 MADYMO 5.000E-05  .000E+00  .000E+00  .000E+00

 1964  0  9.815E-02 3.7595E+00 MADYMO 5.000E-05  .000E+00  .000E+00  .000E+00

 1965  0  9.820E-02 3.7671E+00 MADYMO 5.000E-05  .000E+00  .000E+00  .000E+00
```

E.4 Termination Conditions

All termination conditions selected in MSC.Dytran remain valid. If a termination condition occurs, MADYMO is forced into an end cycle, and MSC.Dytran terminates as usual.

MADYMO termination conditions can also be used to stop the analysis. A termination signal from MADYMO terminates both programs.

E.5 Postprocessing

The regular result files of both MADYMO and MSC.Dytran are created. Thus, it is possible to postprocess the results from MADYMO and MSC.Dytran in the usual way.

Some time-history data related to the rigid ellipsoids can also be output from MSC.Dytran using the standard Case Control commands, for example:

```
RELS   (MOTION) = 600

RELOUT (MOTION) = XCG,YCG,ZCG,ZVEL

TIMES  (MOTION) = 0 THRU END BY 1.E-4

TYPE   (MOTION) = TIMEHIS

SAVE   (MOTION) = 1000000
```

It is also possible to visualize the combined motion of the MADYMO occupant and the MSC.Dytran finite element mesh. This is achieved by requesting the `GEOMETRY` of the MADYMO ellipsoids in the MSC.Dytran Case Control Section, for example:

```
TYPE   (SPHERE) = ARCHIVE

RELS   (SPHERE) = 600

RELOUT (SPHERE) = GEOMETRY

TIMES  (SPHERE) = 0 THRU END BY 1.E-3

SAVE   (SPHERE) = 1000000
```

When the ellipsoid geometry is requested on an archive file, the ellipsoids are covered by dummy shell elements at each requested output step. The dummy elements are written to the archive file. The archive file can then be postprocessed as usual.

MADYMO planes can also be visualized. This is achieved by requesting the `GEOMETRY` of the MADYMO planes in the MSC.Dytran Case Control Section. For example:

```
TYPE   (PLANES) = ARCHIVE

RELS   (PLANES) = 601

RELOUT (PLANES) = GEOMETRY

TIMES  (PLANES) = 0 THRU END BY 1.E-3

SAVE   (PLANES) = 1000000
```

When the geometry of the planes is requested on an archive file, the planes are covered by dummy shell elements at each requested output step. The dummy elements are written to the archive file. The archive file can then be postprocessed as usual.

E.6 Installation Instructions

The following steps need to be taken into account during the installation procedure.

1. Install MSC.Dytran and MADYMO on the same computer. Running MSC.Dytran and MADYMO on different computers is not supported.

   Please, see the MSC.Dytran *Installation and Operations Guide* on how to install MSC.Dytran.

2. Ensure the following environment variable is properly set. Adding this environment variable to the user's `.cshrc` file is recommended.

   `MDROOT` points to the MADYMO installation directory

   For example:

   ```
           setenv MDROOT /disk/part1/MADYMO_53
   ```

   Note: Example settings can be found in `$DYTRANDIR/com/login.csh`

E.7 Submission of a Coupled Analysis

A coupled analysis can be submitted using the command procedure `dytranmad`. The dytranmad procedure processes its command line arguments and creates a second command procedure, which is submitted to a batch queue or is executed interactively. Here is how to use the dytranmad procedure:

```
dytranmad      jid=JobId  mad=MadId  imm=IMMfile  notify=yes|no exe=executable
               user_rout=user_routines_file  ask_rem=yes|no
               print=print_prefix output=out_prefix
```

Examples:

```
% dytranmad jid=mydata mad=mdata
% dytranmad jid=mydata mad=mdata user_rout=memory.f
% dytranmad jid=mydata mad=mdata imm=immdata
```

Command Line Arguments	Purpose
jid=JobId	Name of the MSC.Dytran input file (file type is detected). Script will look for the file JobId.dat. No default is applicable here.
mad=MadId	Name of the MADYMO input file (file type is detected). Script will look for the file MadId.dat. No default is applicable here.
imm=IMMfile	Name of an Initial Metric Method file name (if any). Script will look for IMMfile.dat No default applicable here.
notify=yes/no	If notify=yes, a message will be sent to the terminal when the job finishes. If notify=no, no message is sent. Default is yes.
exe=my_exe.exe	This command line argument enables you to use your own modified MSC.Dytran executable with name my_exe.exe. Default executable is <installation_dir>/dytran47/dytranexe/dytran.exe.
user_rout=my_file.f	The Fortran file with name my_file.f in which user routines are stored, will be compiled and linked with the dytran library. The executable produced will be used to perform the calculation. When the input file is a JobId.dat file, the USERCODE statement is looked for in this file and user_rout is set accordingly. Specifying user_rout on the command line will overrule the USERCODE statement.

Command Line Arguments	Purpose
ask_rem=yes/no	Parameter with which you can specify that you want be asked to remove all files beginning with the generic name derived from the JobId. Ask_rem=yes means that you will be asked to remove these files or restart with a different JobId. Default is yes.
print=my_file	Command line argument that can be used in case of interactive runs, that forces the output to be written to a file instead of the terminal screen. Example: with print=my_file the output will be written to a file named JID_MY_FILE.OUT.
output=prefix	Specifies the prefix to be used for all output files. Default is the JobId.

.

Notes:

1. Jid and mad must be specified.
2. Keywords are separated by a blank.
3. Keywords may appear in any order.
4. Dytran output file will be named JobId.OUT.
5. Archive files will have the extension .ARC.
6. Time History files will have the extension .THS.
7. All output files will be prefixed with the JobId.
8. All output filenames will be in uppercase letters.
9. Only CTRIA3 and GRID are valid entries in an IMM file.

E.8 Coupled Analyses With MSC.Dytran User Subroutines and/or memory.f

It is also possible to include MSC.Dytran user-written subroutines in a coupled analysis. The file that contains the user-written subroutines must be stored in the current working directory. MSC.Dytran must then be submitted by the following command:

```
% dytranmad jid=<dytran_input_file> mad=<MADYMO_input_file>
  user_rout=<file>.f
```

At some occasions, you may need a bigger memory definition that is specified in the `memory.f` subroutine. This can be done as usual. Take the following steps:

1. Copy the memory allocation routine `memory.f` to the current working directory. This memory routine resides on the `<installation_directory>/dytran47/dytranexe` directory. You need to modify the memory definitions in the subroutine.
2. If the analysis does not include a user-written subroutine, MSC.Dytran must be submitted by the following command:

   ```
   % dytranmad jid=<dytran_input_file> mad=<MADYMO_input_file>
     user_rout=memory.f
   ```

3. If the analysis includes a user-written subroutine, MSC.Dytran must be submitted by the following command:

   ```
   % dytranmad jid=<dytran_input_file> mad=<MADYMO_input_file>
     user_rout=<file>.f
   ```

Note that the modified version of MSC.Dytran is compiled and linked again each time you run an analysis with user code. If you want to use the same modified executable many times without compiling and linking each time, it is more efficient to use the exe option to run with the modified executable:

```
%dytranmad jid=<dytran_input_file> mad=<MADYMO_input_file>
exe=my_executable.exe
```

For more information concerning memory definitions and user-written subroutines, see the *MSC.Dytran Installation and Operations Guide*.

APPENDIX F

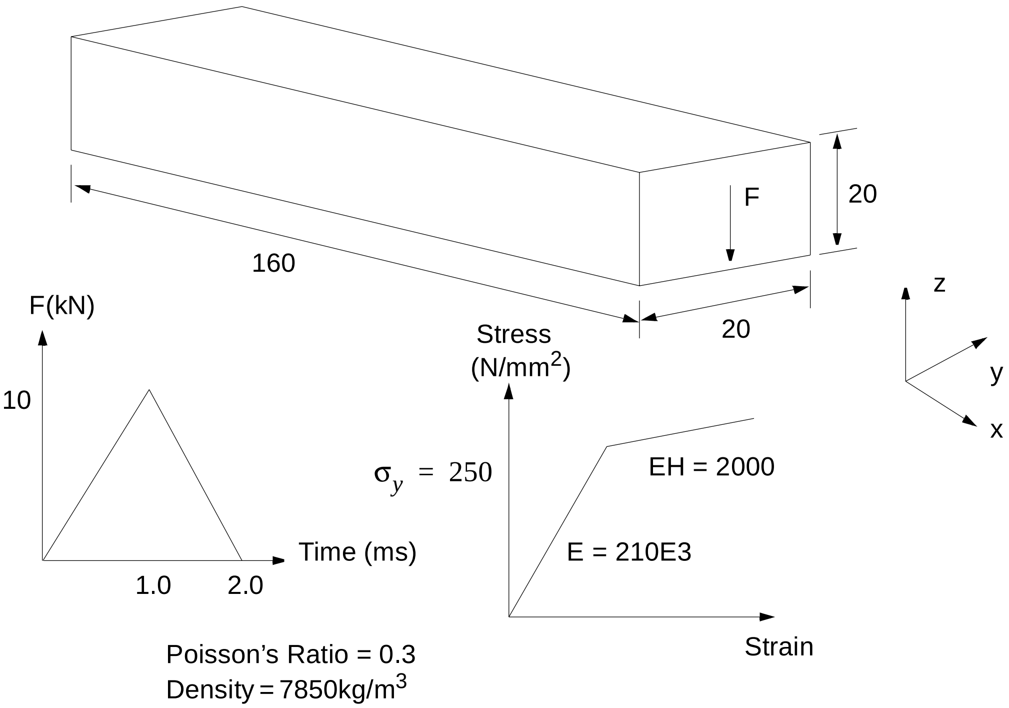
Example Input Data

F.1 Cantilever Beam

This example shows the input file for a simple cantilever beam modeled with shell elements and subjected to a dynamic tip load.

F.1.1 The Problem

The diagram below summarizes the problem. (All dimensions are in millimeters):



F.1.2 The Model

The following mesh is used to model the beam. This mesh is purely for demonstrating how to create an MSC.Dytran input file; it is not intended to be indicative of good modeling practice. The mesh was deliberately kept as simple as possible to limit the size of the input file. If this were a real problem that you were solving, then a considerably finer mesh would be required.

The grid-point and element numbers are shown on the plot shown in Section F.1.1 on page F-1.

F.1.3 Input File

The input file for this problem is listed below. The input is in free format, and the file has extensive comments to explain the operation of the various File Management Systems, Case Control commands, and the Bulk Data entries.

```
$ File Management Section
$ -----------------------
$
$ Define the type of output files to be used. We need one Archive file (logical
$ name ARC), one Time History file (logical name HIST) and one Restart file
$ (logical name RST).
$
TYPE(ARC) = ARCHIVE
TYPE(HIST) = TIMEHIS
TYPE(RST) = RESTART
$
$ CEND marks the end of the FMS and Executive Control Section
$
CEND
$
$ Case Control Section
$ --------------------
$
$ Set the termination time to 2.0ms
$
ENDTIME = 2.0E-3
$
$ Set CHECK to NO to switch of the data check and run the analysis
$
CHECK = NO
$
$ The Archive file will contain the effective plastic strain and the
$ effective stress for all elements at 0.2 and 2. msecs
$
TIMES(ARC) = 0.2E-3, END
SAVE(ARC) = 100000
ELEMENTS(ARC) = 10
SET 10 = 1,THROUGH,7
ELOUT(ARC) = EFFST01,EFFST03,EFFPL01,EFFPL03
$
$ The Time History file will contain the coordinates and velocities of grid
$ points 8 and 18. The results are stored at intervals of .1 msec.
$
TIMES(HIST) = 0 THROUGH END BY .1E-3
SAVE(HIST) = 100000

GRIDS(HIST) = 20
SET 20 = 8,18
GPOUT(HIST) = COORD,VEL

$
$ Write restart data at step 500
$
STEPS(RST) = 500
$
$ Select the constraints and loading to be used from the Bulk Data
$ section
$
SPC = 10
TLOAD = 11
$
$ Bulk Data Section
$ -----------------
$
$ The BEGIN BULK entry marks the end of Case Control and the start of
$ Bulk Data
$
BEGIN BULK
$
$ Define the grid points
$
GRID           1              0.      0.      0.
GRID           2            0.02      0.      0.
GRID           3            0.04      0.      0.
GRID           4            0.08      0.      0.
GRID           5            0.10      0.      0.
GRID           6            0.12      0.      0.
GRID           7            0.14      0.      0.
GRID           8            0.16      0.      0.
GRID          11              0.    0.02      0.
GRID          12            0.02    0.02      0.
GRID          13            0.04    0.02      0.
GRID          14            0.08    0.02      0.
GRID          15            0.10    0.02      0.
GRID          16            0.12    0.02      0.
GRID          17            0.14    0.02      0.
GRID          18            0.16    0.02      0.
$
$ Define the elements
$
CQUAD4         1     100       1       2      12      11
CQUAD4         2     100       2       3      13      12
CQUAD4         3     100       3       4      14      13
CQUAD4         4     100       4       5      15      14
CQUAD4         5     100       5       6      16      15
CQUAD4         6     100       6       7      17      16
CQUAD4         7     100       7       8      18      17
```

```
$
$ The properties of the elements are defined using a PSHELL entry.
$ This entry defines the thickness of the elements (20mm) and a material
$ identification number.
$
PSHELL       100     100    0.02
$
$ The bilinear elastic-plastic material properties are defined using a DMATEP
$ entry. The hourglass and bulk viscosity data is set to the default values
$ so the continuation line is omitted. The DMATEP defines the density and the
$ elastic properties of the material. The yield stress and hardening modulus
$ are defined on a YLDVM entry.
$

DMATEP, 100, 7850.0, 210.E9, 0.3, , , 100
YLDVM, 100, 250E6, 2000.E6
$
$ Use the SPC entry to constrain the grid points where the beam is built-in.
$ All the degrees of freedom are constrained for grid points 1 and 11.
$
SPC, 10, 1, 123456
SPC, 10, 11, 123456
$
$ The transient load is defined using three entries: TLOAD1, TABLED1 and FORCE.
$ The TLOAD1 entry gives the identification numbers of the other entries and
$ indicates that a force is being applied rather than enforced motion.
$
TLOAD1, 11, 11, , 0, 11
$
$ The TABLED1 entry gives the variation of load with time. Note that a
$ horizontal portion at zero load is added to the end of the curve. If the
$ analysis is continued past 2.0ms, this ensures that zero load will be applied,
$ since the table will be extrapolated from the last two points. The entry has
$ two continuation lines to define all the points and it is terminated with an
$ ENDT
$
TABLED1, 11, , , , , , , ,+
+, 0.0, 0.0, 1.0E-3, 10.0E3, 2.0E-3, 0.0, 3.0E-3, 0.0,+
+, ENDT
$
$ The position of the forces is specified using the FORCE entry. Since the load
$ is split between grid points 8 and 18, a scale factor of 0.5 is used. The load
$ is applied in the negative z direction so a vector of 0.0, 0.0, -1.0 is used.
$
FORCE, 11, 8, , 0.5, , , -1.0
FORCE, 11, 18, , 0.5, , , -1.0
$
$ The initial time step is set to 1 microsecond using the parameter   INISTEP.
$
PARAM, INISTEP, 1.E-6
$
$ The input must end with an ENDDATA entry.
$
ENDDATA
```

APPENDIX G

Using USA

G.1 Introduction

In order to enhance the capability to perform underwater shock analysis, the USA (Underwater Shock Analysis) code by DeRuntz et al. is included in MSC.Dytran (References 9-10). The interface is accomplished by a staggered solution method in which MSC.Dytran is used to calculate the structural response and USA calculates the fluid pressure response at the interaction surface. As a result, MSC.Dytran and USA run concurrently, exchanging data as the analysis proceeds.

To run MSC.Dytran with USA, it is not necessary to create a special executable, because USA is standard integrated in MSC.Dytran. The program is only activated when required and a license has been obtained.

The interface between MSC.Dytran and USA is not supported on CRAY an CONVEX systems.

G.2 Input Specification

USA

The input specification for USA (Reference 9-10) does not change. The USA-input files must be specified by the user.

Note: All input data can be typed in using free field format with a blank space or comma used to seperate the distinct entries. However, any individual pieces of data must not be longer than 10 characters with the exception of names. File name plus qualifier are restricted to 20 characters. Fixed or floating point entries should have a decimal point in the entry

MSC.Dytran

The interface between MSC.Dytran and USA must be defined in the MSC.Dytran input file with the USA entry. The use of the USA entry is described in Section 4.7.

Only one USA entry is allowed, and the interaction applies to all structural elements. Specific pressure gauges can be defined by using the SGAUGE entry.

G.3 Running MSC.Dytran with USA

Running MSC.Dytran in combination with USA is not possible using XDYTRAN or the script file dytran.csh. It must be run interactively using the executable. The combined analysis consists of four steps:

- Initialization of the problem

  ```
  dytran.exe jid=JobId,usaout=usa_out_prefix,pri=print_prefix&
  ```

  This run will stop after cycle zero and writes out an output file print_prefix.OUT and a restart file JobId_0.RST for MSC.Dytran and a preprocessing file for USA called usa_out_prefix.DAT. If requested, archive files for MSC.Dytran are written.

- FLUMAS preprocessor for USA to construct the fluid mass matrix

  ```
  dytran.exe flu=FluId,usafil=usa_out_prefix,pri=print_prefix2&
  ```

  The FluId must have extension ".ifl". This analysis will write an MSC.Dytran output file print_prefix2.OUT and an USA output file usa_out_prefix.OFL. The USA output file must be checked on negative eigenvalues before you continue. Negative eigenvalues in most cases mean that the USA input file is not correct. (See References 9-10). Other preprocessing files that can be obtained are usa_out_prefix.GEO, usa_out_prefix.DAA and usa_out_prefix.FLU.

- AUGMAT preprocessing for USA to set data from fluid mass matrix processor

  ```
  dytran.exe aug=AugId,usafil=usa_out_prefix,pri=print_prefix3&
  ```

  The AugId must have extension ".iau". This analysis will write an MSC.Dytran output file print_prefix3.OUT and an USA output file usa_out_prefix.OAU. Moreover, a preprocessing file called usa_out_prefix.PRE is written.

- Interaction of MSC.Dytran and USA by restarting MSC.Dytran

  ```
  dytran.exe jid=Job2Id,restart=JobId_0.RST,usain=UsaId,usafil=usa_out_prefix,pri=print_prefix4
  ```

  The UsaId must have extension ".ius". This analysis is the combined analysis of MSC.Dytran and USA.

Before running USA and its preprocessors, the environment must be set to define the memory. Depending on what type of problem is analyzed, the following variables must be set:

USA_CORE for total memory used in USA.

USA_MAXFLU is the maximum number of DAA fluid degrees of freedom.

USA_MAXSTR is the maximum number of structural nodal grid points.

USA_MFVNOD is the maximum number of fluid volume nodal grid points.

USA_MAXBEM is the maximum number of structural nodal grid points on beams.

USA_MFVELM is the maximum number of fluid volume elements grid points.

Example

```
setenv USA_CORE 15000000
setenv USA_MAXFLU 100000
setenv USA_MFVNOD 1
setenv USA_MAXBEM 200
USA_MFVELM 1
```

```
dytran.exe jid=sphere,usaout=quart_sphere,pri=step1
dytran.exe flu=sphflu,usafil=quart_sphere,pri=flu
dytran.exe aug=sphau4,usafil=quart_sphere,pri=aug
dytran.exe jid=restart,restart=QUART_SPHERE_0.RST,usain=sphere,usafil=quart_sphere,pri=out
```

G.4 Termination Conditions

All termination conditions selected in MSC.Dytran remain valid. If a termination condition occurs, USA is forced into an end cycle, and MSC.Dytran terminates as usual.

USA-termination conditions cannot be used to stop the analysis.

G.5 Postprocessing

USA

USA results can be postprocessed with the standard ATB postprocessor POSTPR.

MSC.Dytran

The pressure given by USA to the interface can be output from MSC.Dytran using the standard Case Control command USASOUT.

Special pressure gauge time-history files can be requested by defining SGAUGE entries and using the standard Case Control command SGOUT.