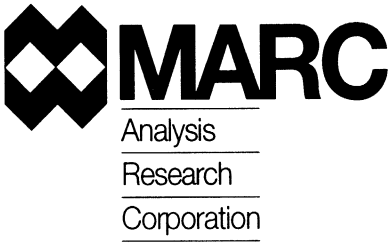


Volume A
User Information

PROPRIETARY NOTICE

MARC Analysis Research Corporation reserves the right to make changes in specifications and other information contained in this document without prior notice.

Although due care has been taken to present accurate information, MARC Analysis Research Corporation DISCLAIMS ALL WARRANTIES WITH RESPECT TO THE CONTENTS OF THIS DOCUMENT (INCLUDING, WITHOUT LIMITATION, WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE) EITHER EXPRESSED OR IMPLIED. MARC Analysis Research Corporation SHALL NOT BE LIABLE FOR DAMAGES RESULTING FROM ANY ERROR CONTAINED HEREIN, INCLUDING, BUT NOT LIMITED TO, FOR ANY SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF, OR IN CONNECTION WITH, THE USE OF THIS DOCUMENT.

RESTRICTED RIGHTS NOTICE

MARC Analysis Research Corporation:

North America

Corporate Headquarters
MARC Analysis Research Corporation
260 Sheridan Avenue
Palo Alto, CA 94306
Telephone: (415) 329-6800
FAX: (415) 323-5892

Europe

European Headquarters
MARC-Europe
Bredewater 26
2715 CA Zoetermeer, The Netherlands
Telephone: 31-79-510411
FAX: 31-79-517560

German Office
MARC Software Deutschland GmbH
Ismaninger Strasse 9
8011 Aschheim, Germany
Telephone: 49-89-904-50-33
FAX: 49-89-903-06-76

Italian Office
Espri-MARC
Piazza Rossetti 5/16A
16129 Genova, Italy
Telephone: 39-10-595949
FAX: 39-10-585949 (eve)

Pacific Rim

Far East Headquarters
Nippon MARC Co., Ltd.
P.O. Box 5056
Shinjuku Daiichi Seimei Bldg.
2-7-1 Nishi-Shinjuku
Shinjuku-ku, Tokyo 163, Japan
Telephone: 81-3-3345-0181
FAX: 81-3-3345-1529

Osaka Office
Nippon MARC Co., Ltd.
Dai 2 Kimi Bldg., 4F
2-11 Toyotsu-cho
Suita-city, Osaka 564, Japan
Telephone: 81-6-385-1101
FAX: 81-6-385-4343

Chapter Preface

A 1.1 ABOUT THIS MANUAL

This manual is Volume A, the first in a series of six volumes documenting the MARC Finite Element program. The documentation of the MARC program is summarized below. You will find references to these documents throughout this manual

MARC Program Documentation

TITLE	VOLUME
MARC User Information	Volume A
MARC Element Library	Volume B
MARC Program Input	Volume C
MARC User Subroutines	Volume D
MARC Demonstration Problems	Volume E
MARC Background Papers	Volume F

A 1.2 PURPOSE OF VOLUME A

The purpose of this volume is three-fold:

1. To help you define your finite element problem by describing MARC's capabilities to model physical problems.
2. To identify and describe complex engineering problems and introduce MARC's scope and capabilities for solving these problems; and
3. To assist you in accessing the MARC program features that are applicable to your particular problems and to provide you with references to the rest of the MARC literature.

A 1.3 CONTENTS OF VOLUME A

This volume describes how to use the MARC program. It explains the capabilities of the MARC program and gives pertinent background information. The principal categories of information are found under the following chapter titles:

1. Introduction
2. Program initiation
3. Data entry
4. Mesh definition
5. Structural procedure library
6. Material library
7. Element library
8. Boundary conditions
9. Output results
10. Plotting capabilities

The information in this manual is both descriptive and theoretical. You will find engineering mechanics discussed in some detail. You will also find specific instructions for operating the various options offered by MARC.

A 1.4 HOW TO USE THIS MANUAL

Volume A organizes the features and operations of the MARC program sequentially. This organization represents a logical approach to problem solving using Finite Element Analysis. First, the database is entered into the system, as described in Chapter 3. Next, a physical problem is defined in terms of a mesh overlay. Techniques for mesh definition are described in Chapter 4. Chapter 5 describes the various structural analyses that can be performed by MARC, while Chapter 6 describes the material models that are available in MARC. Chapter 7 explains the type of elements that may be used to represent the physical problem. Chapter 8 discusses constraints, in the form of boundary conditions. Finally, the results of the analysis, in the form of outputs and plots, are described in Chapters 9 and 10.

This volume is also designed as a reference source. This means that all users will not need to refer to each section of the manual with the same frequency or in the same sequence.

Contents

Page

Chapter 1 THE MARC SYSTEM

Page

Chapter 2 PROGRAM INITIATION

Chapter 3 DATA ENTRY

Chapter 4 INTRODUCTION TO MESH DEFINITION

Page

Chapter 5 STRUCTURAL PROCEDURE LIBRARY

Page

Page

Chapter 6 MATERIAL LIBRARY

Page

Chapter 7 ELEMENT LIBRARY

Page

Chapter 8 BOUNDARY CONDITIONS

Page

Chapter 9 OUTPUT RESULTS

Page

Chapter 10 PLOTTING CAPABILITIES

Appendix INTRODUCTION TO THE FINITE ELEMENT TECHNOLOGY IN MARC

Page

List of Figures

Figure **Page**

Figure **Page**

Figure **Page**

List of Tables

Table **Page**

Chapter 1 THE MARC SYSTEM

The MARC system contains a series of integrated programs that facilitate analysis of engineering problems in the fields of structural mechanics, heat transfer, and electromagnetics. The MARC system consists of the following programs:

- MARC
- MESH3D
- MARC-PLOT
- MARC-PIPE
- MENTAT

These programs work together to:

- Generate geometric information that defines your structure (MARC, MESH3D, PIPE, and Mentat)
- Analyze your structure (MARC)
- Graphically depict the results (MARC, MARC-PLOT, and Mentat)

Figure A 1.0-1 shows the interrelationships among these programs. Section A 1.1, MARC PROGRAMS, discusses the MARC component programs.

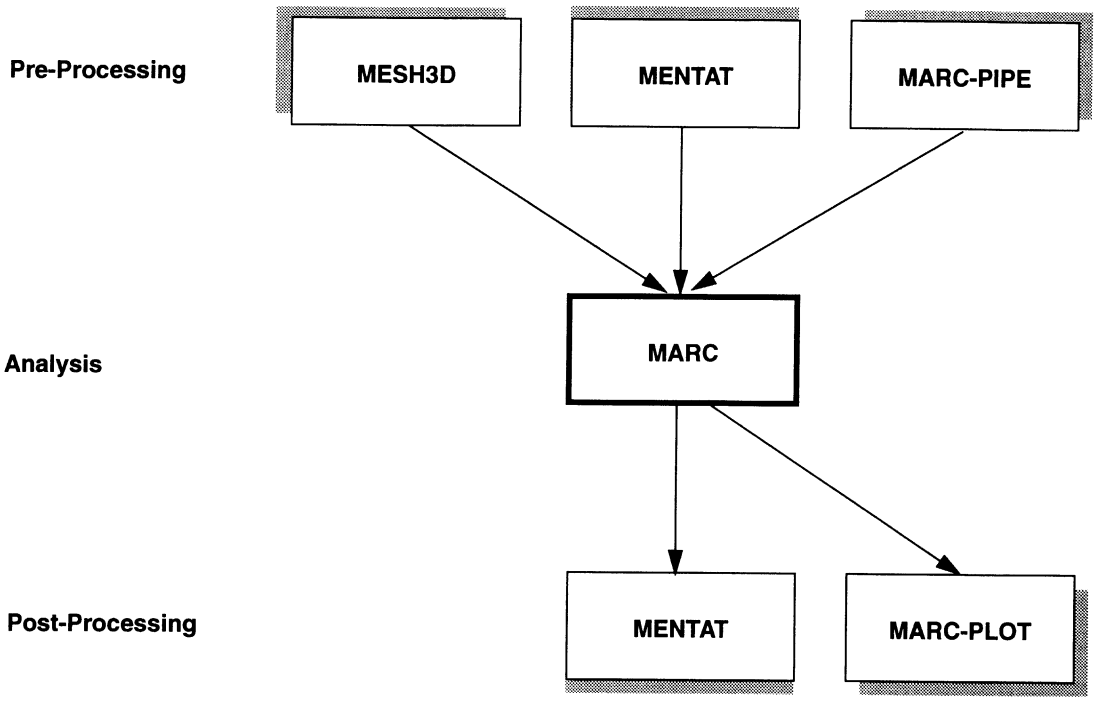


Figure A 1.0-1 The MARC System

A 1.1 MARC PROGRAMS

MARC

You can use MARC to perform linear or nonlinear stress analysis in the static and dynamic regimes and to perform heat transfer analysis. The nonlinearities may be due to either material behavior, large deformation, or boundary conditions. An accurate representation accounts for these nonlinearities.

Physical problems in one, two, or three dimensions may be modeled using a variety of elements. These elements include trusses, beams, shells, and solids. Mesh generators, graphics, and post-processing capabilities, which assist you in the preparation of input and the interpretation of results, are all available in the MARC system. The equations governing mechanics and implementation of these equations in the finite element method are discussed in the Appendix.

MESH3D

The mesh generating program, MESH3D, automatically generates three-dimensional mesh data, using either 8-node or 20-node brick elements. The output from the program consists of the three coordinates of each node and the connectivity of each element. This output can subsequently be used in MARC program analyses. MESH3D plotting capabilities allow you to view the generated mesh with hidden lines deleted.

MARC-PLOT

MARC-PLOT is a post-processor plotting program. It provides time history, as well as variable-versus-variable history plotting. As input, MARC-PLOT uses the post-tape written by MARC. You can plot any number of variables together as functions of time, increment number, or another variable. You can also generate several plots in a single job.

MARC-PIPE

MARC-PIPE is used to generate a mesh for analyzing a piping system consisting of straight and curved (elbow) sections. You can describe the system in simple terms in the input for MARC-PIPE. The output consists of the CONNECTIVITY and COORDINATE data, followed by the TYING data written on a tape. MARC can use this data to perform stress analyses.

Mentat

This program is an interactive computer program that prepares and processes data for use with the finite element method. Interactive computing can significantly reduce the human effort needed for analysis by the finite element method. Graphical presentation of data further reduces this effort by providing an effective way to review the large quantity of data typically associated with finite element analysis.

An important aspect of this program is that you can interact directly with the program. The program verifies keyboard input and returns recommendations or warnings when it detects questionable input. The program checks the contents of input files and generates warnings

about its interpretation of the data if the program suspects that it may not be processing the data in the manner in which you, the user, have assumed. This program allows the user to graphically verify any changes the input generates.

The Mentat program can process both two- and three-dimensional meshes to do the following:

- Generate and display a mesh
- Generate and display boundary conditions and loadings
- Perform post-processing to generate contour, deformed shape, and time history plots

The data that is processed includes:

- Nodal coordinates
- Element connectivity
- Nodal boundary conditions
- Nodal coordinate systems
- Element material properties
- Element geometric properties
- Element loads
- Nodal loads/nonzero boundary conditions
- Element and nodal sets

A 1.2 STRUCTURE OF MARC

The MARC program has four comprehensive libraries, making the program applicable to a wide range of uses. These libraries contain structural procedures, materials, elements, and program functions. The contents of each library are described below.

Procedure Library

The structural procedure library contains procedures such as static analysis, dynamic analysis, creep, buckling, and heat transfer.

The procedure library conveniently relates these various structural procedures to physical phenomena while guiding users through modules that allow, for example, nonlinear dynamic and heat-transfer analyses.

Material Library

The material library includes many material models that represent most engineering materials. Examples are the inelastic behavior of metals, soils, and rubber material. Many models exhibit nonlinear properties such as plasticity, viscoelasticity, and hypoelasticity. Linear elasticity is also included. All properties may depend on temperature.

Element Library

The element library contains approximately 140 elements. This library lets you describe any geometry under any linear or nonlinear loading conditions.

Program Function Library

The program functions such as selective assembly, user-supplied subroutines, and restart, are tailored for user-friendliness and are designed to speed up and simplify analysis work. MARC allows you to combine any number of components from each of the four libraries and, in doing so, puts at your disposal the tools to solve almost any structural mechanics problem.

A 1.3 FEATURES AND BENEFITS OF THE MARC PROGRAM

Since the mid-1970s, the MARC program has been recognized as the premier general purpose program for nonlinear finite element analysis. The program's modularity leads to its broad applicability. All components of the structural procedure, material, and element libraries are available for use, allowing virtually unlimited flexibility and adaptability.

The MARC system has helped analyze and influence final design decisions on

- Automotive parts
- Nuclear reactor housings
- Biomedical equipment
- Offshore platform components
- Coated fiberglass fabric roof structures
- Rocket motor casings
- Ship hulls
- Elastomeric motor mounts
- Space vehicles
- Electronic components
- Steam-piping systems
- Engine pistons
- Tires
- Jet engine rotors
- Welding, casting, and quenching processes
- Large strain metal extrusions

MARC's clients gained the following benefits not attainable through other numerical or experimental techniques. These benefits include:

- Accurate results for both linear and nonlinear analysis
- Better designs, which result in improved performance and reliability
- The ability to model complex structures and to incorporate geometric and material nonlinear behavior
- Documentation, technical support, consulting, and education provided by MARC
- Availability of MARC on most computers from workstations to supercomputers
- Efficient operation
- Availability of MARC programs to lease for in-house computers or access through a data center

A 1.4 MARC K6

Adaptive Meshing

An adaptive meshing capability has been developed for both linear and nonlinear analysis. The capability is available for 2D and 3D linear elements, triangles, quadrilaterals, tetrahedrals, hexahedrals, and shell elements. The capability may be used for structural analysis or heat transfer analysis. The support of other procedures will be added in subsequent releases. Over ten different error criteria are supported, and the user may select multiple criteria in an analysis. Special capabilities exist for contact which result in mesh redefinition whenever contact occurs.

Contact

The contact analysis capability has been significantly enhanced to allow the support of analytical contact surfaces. Circles, cylinders, and spheres are treated as exact quadratic surfaces. Other surfaces are modelled using NURBS. This capability improves both the accuracy and the convergence characteristics of the analysis. All Mentat generated surfaces may be supported by using NURBS.

Adaptive Time Stepping

The adaptive time stepping procedure AUTO INCREMENT may now be used with CONTACT analysis. This allows for better control of the time stepping when local buckling occurs. The AUTO TIME option has also been improved for CONTACT analysis.

Rigid Plastic Flow

The rigid plastic (R-P FLOW) capability has been rewritten with particular emphasis on sheet metal forming. An implicit-explicit approach is used to integrate the constitutive relation. This option has been expanded to support plane stress analyses. The thinning of the sheet is exactly treated to ensure no volumetric changes.

Membrane

The membrane element type 18 has been rewritten. It now gives better performance at a reduced cost.

New Shell

A new six-noded triangular shell element has been added. This element is based on a thin shell formulation that allows large rotations. The associated heat transfer element has also been added.

Follower Force

The follower force stiffness contribution may now be included in the calculation. This improves both the accuracy and stability of those analyses involving large deformation with follower force type loads. Follower force stiffness is available for the lower order continuum elements and shells (type 72 and type 75) and the membrane element type 18.

Convection

The heat transfer capability has been expanded to include convection. The user can either define a velocity history or the program will calculate it in a couple analysis.

Foam Model

A foam model has been added to the program. This model allows large elastic volumetric and shear strains. The curve fitting program has been updated to generate material properties for this model.

Explicit Dynamics

The explicit dynamics capability has been rewritten. The new procedure allows a variable time step. The time step is adjusted to ensure that the maximum stable time step is used. Analytical mass matrices are used for the lower order elements. The storage requirements have been considerably reduced.

Sparse Direct Solver

A new direct matrix solver using sparse matrix storage has been included. Both an in-core and an out-of-core memory capability is available. This procedure uses significantly less memory, and for many problems has improved computational performance.

Reduced Memory Requirements

The memory requirements for element storage has been reduced by 20% to 40%.

Coriolis Effects

Coriolis stiffness/damping effects may now be included. This is in addition to the centrifugal effects that were available in previous releases.

Wave Loading

A wave loading capability on beams/pipes has been added. Both the fluid drag and the buoyancy loads are included. This is available for both quasi-static and dynamic analyses.

Thermal Loads

Thermal loads may now be entered at the nodal points. This complements the previous capability where the thermal loads had to be entered at the integration points.

Base Motion

Base motion time histories of acceleration may now be prescribed.

Generalized Newmark Operator

The alpha and beta parameters associated with second order methods (Newmark-beta) may now be prescribed. This allows advanced users the ability to tune the dynamic operator for preferred damping/accuracy.

Chapter 2 PROGRAM INITIATION

Chapter 2 explains how to execute the MARC program on your computer. MARC runs on many types of machines. All MARC capabilities are available on each type of machine; however, program execution may vary among machine types. The allocation of computer memory depends on the hardware restrictions of the machine you are using. There is no limit to the size of the analysis that may be performed by MARC other than the limit imposed by your computing resources.

A 2.1 MARC HOST SYSTEMS

The MARC program will run on most computers. Table A 2.1-1 summarizes the types of machines and operating systems on which MARC will currently run.

Table A 2.1-1 MARC Computer Versions

Computer	Machine Type	Operating System
AMDAHL	All machines	MVS
APOLLO	DN 3000, 4000	UNIX
CONVEX	All machines	UNIX
CRAY	YMP, C90	UNICOS
DEC/VAX DEC/3100/5000 DEC/ALPHA	All machines All machines	VAX/VMS ULTRIX OSF
FUJITSU	M-Series	IV/F4
HEWLETT-PACKARD	HP9000	HP-UX
HITACHI	M-Series	VOS2, VOS3
IBM	All machines in 370 family All machines in 30XX family All machines in 43XX family	MVS/XA CMS/XA
IBM	RS6000	AIX
INTERGRAPH	C300, C400	UNIX
SILICON GRAPHICS	All machines	UNIX
SUN	All machines	UNIX

A 2.2 WORKSPACE REQUIREMENTS

Computing the amount of workspace required by the MARC program is a complex function of many variables. The most efficient method is to select a large enough workspace to handle a variety of runs, without sacrificing efficiency or wasting core space. The following sections discuss workspace requirements for the MARC program.

MARC Workspace Requirements

Both in-core and out-of-core data storage schemes are available in the MARC program. Elements may also be stored out-of-core if you use the ELSTO option. Therefore, this program presents some unique complications in the estimation of sizing. The program has both an in-core and out-of-core solver. The MARC program chooses the solution type automatically.

There are two out-of-core storage options in the MARC program. These options are:

- Out-of-Core Element Data storage
- Out-of-Core Solution

The Out-of-Core Element Data option stores element arrays (strains, stresses, temperatures, etc.) on an auxiliary device. Data connected with storage of all element quantities occupy a large amount of space for the more complex shell or three-dimensional elements. Putting these data in secondary storage does not cause long I/O times and allows a savings of core storage. To use this option, set the ELSTO parameter option. This information is written to Unit 3.

The Out-of-Core Solution is invoked automatically, as required. Data related to the master stiffness matrix occupy the most space and have significant effect on the I/O time. In many problems, there is insufficient memory to store the complete stiffness matrix in the core memory. In such cases, the program automatically selects the out-of-core solution procedure. When the direct solver is used, MARC uses a Gaussian elimination procedure with a blocked skyline storage technique for solving the stiffness matrix. The program tries to fit as much of the full system as possible into the core memory. The minimal necessary workspace equals

```
2*HBW*NDEG*IF
```

where HBW is the maximum nodal half-bandwidth and NDEG is the maximum number of degrees-of-freedom per node. The IF value varies depending on your system:

```
IF = 1 for CRAY
IF = 2 for all other machines
```

If the program uses the out-of-core solution, the output includes a graphical representation of the profile of the stiffness matrix.

You should optimize the nodal bandwidth for all nontrivial problems. The stiffness matrix requires approximately the following amount of memory:

```
NUMNP*NDEG*AHBW*NDEG*IF
```

where:

```
NUMNP = number of nodes in the structure
NDEG = number of degrees-of-freedom per node
AHBW = average nodal half-bandwidth
IF = 1 for CRAY
IF = 2 for all other machines
```

On machines with small core memory, you should use a workspace of 50,000 words for problems of less than 1,000 degrees-of-freedom. When the problem size is between 1,000 and 3,000 degrees-of-freedom, increase the workspace to 100,000 words and flag ELSTO in the MARC parameter options. When you analyze a problem of more than 3,000 degrees-of-freedom, flag ELSTO and allocate between 100,000 and 1,000,000 words. On many systems, especially those running under virtual operating systems, using more memory does not make running a job more expensive. Therefore, on these systems, you can be more liberal in allocating workspace than when you pay for memory resident time. On virtual operating systems, it is almost always more efficient to run in-core than out-of-core and to avoid the ELSTO option.

For large problems, you may want to see the exact workspace requirements for running a job, without actually executing the analysis. To do this, insert the STOP parameter option to exit the program normally after the workspace is allocated. The allocated workspace is based on the optimized bandwidth if you request the OPTIMIZE option in the model definition section.

MESH3D Workspace Requirements

The MESH3D program requires a workspace of approximately 30,000 words because of the large amount of temporary storage space needed to process a mesh. This 30,000-word workspace will handle almost all 8-node and 20-node brick runs in- core. For larger runs (that is, over 1,500 nodes), increase the workspace to approximately 40,000 words as a first try. Adjust this figure if necessary. The actual size needed is printed in the MESH3D output.

MARC-PIPE Workspace Requirements

The MARC-PIPE program is constructed so that the workspace covers all but the largest runs. The program executes in a small workspace; therefore, an over- estimated workspace will not increase costs noticeably.

MARC-PLOT Workspace Requirements

The MARC-PLOT program is designed to operate in a workspace sufficient for any possible run; therefore, allocation of workspace is not a concern.

A 2.3 FILE UNITS

The MARC program uses auxiliary files for data storage in various ways. Particular FORTRAN unit numbers are used for certain program functions (for example, ELSTO, RESTART). Table A 2.3-1 lists these file unit numbers.

NOTE

On most systems, these files are references by file names, as well as by the file unit numbers.

Several of these files are necessary for solving most problems. The program input file and program output file are always required.

On most computer systems (excluding IBM), you do not need to allocate file space, because the operating system does this automatically.

ELSTO and out-of-core solution use direct (random) access files. Therefore, you must take additional precautions when using IBM-compatible machines (AMDAHL, FUJITSU, HITACHI) under the MVS or CMS operating systems.

Table A 2.3-1 File Units

FORTRAN Unit Number	Purpose	Format
1	Mesh generation input/output	Card image
2	Out-of-core solution	Unformatted
3	ELSTO	Unformatted
4	Plot output file	Unformatted
5	Input file	Card image
6	Output file	Line printer
7	Dummy/logfile	Formatted
8	RESTART output	Unformatted
9	RESTART input	Unformatted
10	Dummy	
11	Out-of-core solution	Unformatted
12	Out-of-core solution	Unformatted
13	Out-of-core solution	Unformatted
14	Out-of-core solution	Unformatted
15	Out-of-core solution - Lanczos	Unformatted
16	POST output	Unformatted
17	POST input	Unformatted
18	Optimization	Formatted
19	POST output	Formatted
20	POST input	Formatted
21	Dummy	
22	Fluid/solid interaction - Lanczos	Unformatted
23	Lanczos	Unformatted
24	Temperature input	Formatted
25	Temperature input	Unformatted
31	Substructures Data Base	Unformatted

A 2.4 PROGRAM INITIATION

Volume C, Chapter 9 of the MARC User's Manual describes the procedure required for executing MARC programs on different computer systems. Chapter 9 also presents examples of executing MARC with and without user subroutines.

Procedures are set up that facilitate the execution of MARC on most computers. These procedures invoke machine-dependent control or command statements. These statements control the file or files associated with a job. If files other than default files are to be used, you must provide additional job control information.

Chapter 3 DATA ENTRY

The input data structure is made up of three logically distinct sections:

1. Parameter options describe the problem type and size.
2. Model definition options give a detailed problem description.
3. History definition options define the load history.

Input data is organized in (optional) blocks. Key words identify the data for each optional block. This form of input enables you to specify only the data for the optional blocks that you need to define your problem. The various blocks of input are "optional" in the sense that many have built-in default values which may be used by MARC in the absence of any explicit input from the user.

A 3.1 INPUT CONVENTIONS

The MARC program performs all data conversion internally so that the system will not abort because of data errors made by the user. The program reads all input data options alphanumerically (80A1) and converts them to integer, floating point, or keywords, as necessary. The program issues error messages and displays the illegal option image if it cannot interpret the option data field according to the specifications given in the manual. When such errors occur, the program attempts to scan the remainder of the data deck and ends the run with an exit error message at the END OPTION option or at the end of the input file.

Two input format conventions can be used: fixed and free format. You may mix fixed and free format options within a deck, but you may only enter one type of format on a single option.

The syntax rules for fixed fields are as follows:

- You must right-justify integers in their fields. (The right blanks are filled with zeroes).
- Give floating point numbers with or without an exponent. If you give an exponent, it must be preceded by the character E or D and must be right-justified.

The syntax rules for free fields are as follows:

- Check that each option contains the same number of data items that it would contain under standard fixed-format control. This syntax rule allows you to mix fixed-field and free-field options in the data deck because the number of options you need to input any data list will be the same in both cases.
- Separate data items on a option with a comma. The comma can be surrounded by any number of blanks. Within the data item itself, no embedded blanks may appear.
- Give floating point numbers with or without an exponent. If you given an exponent, it must be preceded by the character E or D and must immediately follow the mantissa (no embedded blanks).
- Give keywords exactly as they are written in the manual. Embedded blanks do not count as separators here (for example, BEAM SECT is one word only).
- If a option contains only one free-field data item, follow that item with a comma. For example, the number "1" must be entered as "1," if it is the only data item on a option. If the comma is omitted, the entry will be treated as fixed format and may not be properly right-justified.

Input of List of Items

The MARC program often requests that you enter a list of items in association with certain program functions. As an example, these items may be a set of elements as in the PROPERTY option, or a set of nodes as in the POINT LOAD option. Six types of items can be requested:

Element numbers
Node numbers
Degree-of-freedom numbers
Integration point numbers
Layer numbers
Increment numbers

This list can be entered using either the OLD format (compatible with the G, H, and J versions of MARC) or the NEW format (the K version).

Using the OLD format, you can specify the list of items in three different forms. You can specify:

1. A range of items as:

   m n p

   which implies items m through n by p. If p is not specified, the program assumes it is 1. Note that the range can either increase or decrease.

2. A list of items as:

   -n a_1 a_2 a_3 ... a_n

   which implies that you should give n items, and they are a ,...a .

3. A set name as:

   MYSET

   which implies that all items previously specified in the set MYSET are used. Specify the items in a set using the DEFINE model definition option.

   Using the NEW format, you can express the list of items as a combination of one or more sublists. These sublists can be specified in three different forms. The following operations can be performed between sublists:

   AND
   INTERSECT
   EXCEPT

When you form a list, subsets are combined in binary operations (from left to right). The following lists are examples.

1. SUBLIST1 AND SUBLIST2

   This list implies all items in subsets SUBLIST1 and SUBLIST2. Duplicate items are eliminated and the resulting list is sorted.

2. SUBLIST1 INTERSECT SUBLIST2

   This list implies only those items occurring both in subsets SUBLIST1 and SUBLIST2; the resulting list is sorted.

3. SUBLIST1 EXCEPT SUBLIST2

   This list implies all items in subset SUBLIST1 except those which occur in subset SUBLIST2; the resulting list is sorted.

4. SUBLIST1 AND SUBLIST2 EXCEPT SUBLIST3 INTERSECT SUBLIST4

   This list implies the items in subsets SUBLIST1 and SUBLIST2 minus those items that occur in subset SUBLIST3. Then, if the remaining items also occur in subset SUBLIST4, they are included in the list.

SUBLISTS can have several forms. You can specify:

1. A range of items as:

   m TO n BY p
   or
   m THROUGH n BY p

   which implies items m through n by p. If "BY p" is not included, the program assumes "BY 1". Note that the range can either increase or decrease.

2. A string of items as:

   $a_1\ a_2\ a_3\ \ldots\ a_n$

   which implies that n items are to be included. If continuation options are necessary, then either a C or CONTINUE should be the last item on the option.

3. A set name as:

   MYSET

   which implies that all items you previously specified to be in the set MYSET are used. You specify the items in a set using the DEFINE option.

NOTE

INTERSECT or EXCEPT may not be used when defining lists of degrees-of-freedom.

Examples of Lists

This section presents some examples of lists and entry formats.

Use the DEFINE model definition option to associate a list of items with a set name with items. Three sets are defined below: FLOOR, NWALL, and WWALL.

- DEFINE NODE SET FLOOR contains:

  1 TO 15 (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)

- DEFINE NODE SET NWALL contains:

  5 TO 15 BY 5 AND 20 TO 22 (5,10,15,20,21,22)

- DEFINE NODE SET WWALL contains:

  11 TO 20 (11,12,13,14,15,16,17,18,19,20)

Some possible lists are:

- NWALL AND WWALL, which would contain nodes:

  5 10 11 12 13 14 15 16 17 18 19 20 21 22

- NWALL INTERSECT WWALL, which would contain nodes:

  15 20

- NWALL AND WWALL EXCEPT FLOOR, which would contain nodes:

  16 17 19 20 21 22

A 3.2 PARAMETER OPTIONS

This group of options allocates the necessary working space for the problem and sets up initial switches to control the flow of the program through the desired analysis, This set of input options must be terminated with an END option, The input format for these options is described in Volume C, Section 2.

A 3.3 MODEL DEFINITION OPTIONS

This set of data options enters the initial loading, geometry, and material data of the model and provides nodal point data, such as boundary conditions. Model definition options are also used to govern the error control and restart capability. Model definition options can also specify print-out and post-processing options. The data you enter on model definition options provides the program with the necessary information for determining an initial elastic solution (zero increment solution). This group of options must be terminated with an END OPTION option. The input format for these options is described in Volume C, Section 3.

A 3.4 HISTORY DEFINITION OPTIONS

This group of options provides the load incrementation and controls the program after the initial elastic analysis. History definition options also include blocks which allow changes in the initial model specifications. Each set of load sets must be terminated with a CONTINUE option. This option requests that the program perform another increment or series of increments if you request the auto- incrementation features. The input format for these options is described in Volume C, Section 5.

A typical input deck setup for the MARC program is shown below.

- MARC Parameter Options
  Terminated by an END option
- MARC Model Definition Options
  (Zero Increment)
  Terminated by an END OPTION option
- MARC History Definition
  cards for the First Increment
  Terminated by a CONTINUE option
- (Additional History Definition
  Option for the second, third, ..., Increments)

Figure A 3.1-1 is a dimensional representation of the MARC input deck.

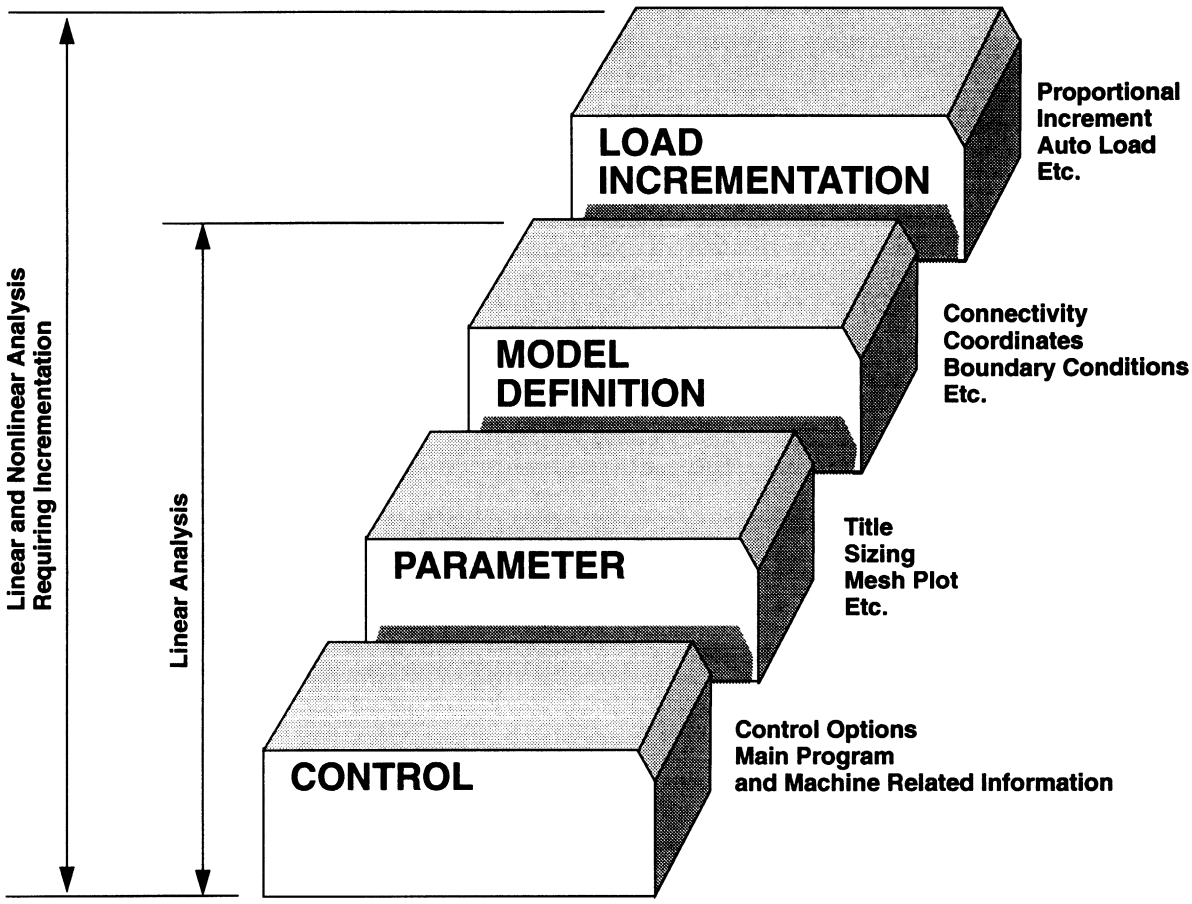


Figure A 3.1-1 The MARC Input Deck

A 3.5 MESH AND OUTPUT DISPLAY OPTION

When the MESH PLOT parameter option is inserted into the input deck, the program expects to read some selection of the mesh display options to control the plots. These options must immediately follow either the END OPTION option or a load incrementation CONTINUE option. You can generate any number of plots. A CONTINUE option ends the definition of an individual plot; an END PLOT option indicates the end of the current series of plots; a flag on the second option of the END PLOT set determines if the program will continue to perform an analysis or if it should stop after generating the plots. The input format for these options is described in Volume C, Chapter 4. It is recommended that graphical pre- and post-processing be performed interactively using Mentat.

A 3.6 REZONING OPTION

When the REZONE parameter option is inserted into the input deck, the program reads some selection of the rezoning option options to control the rezoning steps. These options must immediately follow the END OPTION option, a history definition CONTINUE option, or a mesh plot END PLOT option.

You may select as many rezoning steps in one increment as you need. Every rezoning step is defined by the data, starting with the REZONE option and ending with the CONTINUE option. The END REZONE option terminates the complete set of rezoning steps that form a complete rezoning increment. Follow the rezoning input with either a mesh plot option (if parameter option MESH PLOT is present), with normal history definition data, or again by rezoning data. The input format for these options is described in Volume C, Chapter 5A.

A 3.7 SUBSTRUCTURE

The MARC program is capable of multi-level substructuring that includes:

- Generation of superelements
- Use of superelements in subsequent MARC analyses
- Recovery of solutions (displacements, stresses, and strains) in the individual substructures

Each step can be done in an individual run, or an unlimited number of steps can be combined into a single run.

During superelement generation, you can create a complete new superelement or copy a previously defined superelement with identical or newly defined external load conditions.

During the use of superelements, you can rotate or mirror a single superelement. If the run is nonlinear, the superelements are treated as linear elastic parts. In this case, you can choose to perform a detailed analysis of certain superelements at any increment by descending down to the desired superelements. The load on the substructure can be controlled with the normal MARC control algorithms (AUTO INCREMENT, AUTO LOAD, and PROPORTIONAL INCREMENT).

Chapter 4 INTRODUCTION TO MESH DEFINITION

This chapter describes the techniques for mesh definition that are available internally in MARC. Mesh definition is the process of converting a physical problem into discrete geometric entities for the purpose of analysis. Before a body can undergo finite element analysis, it must be modeled into discrete physical elements. An example of mesh definition is shown in Figure A 4.0-1.

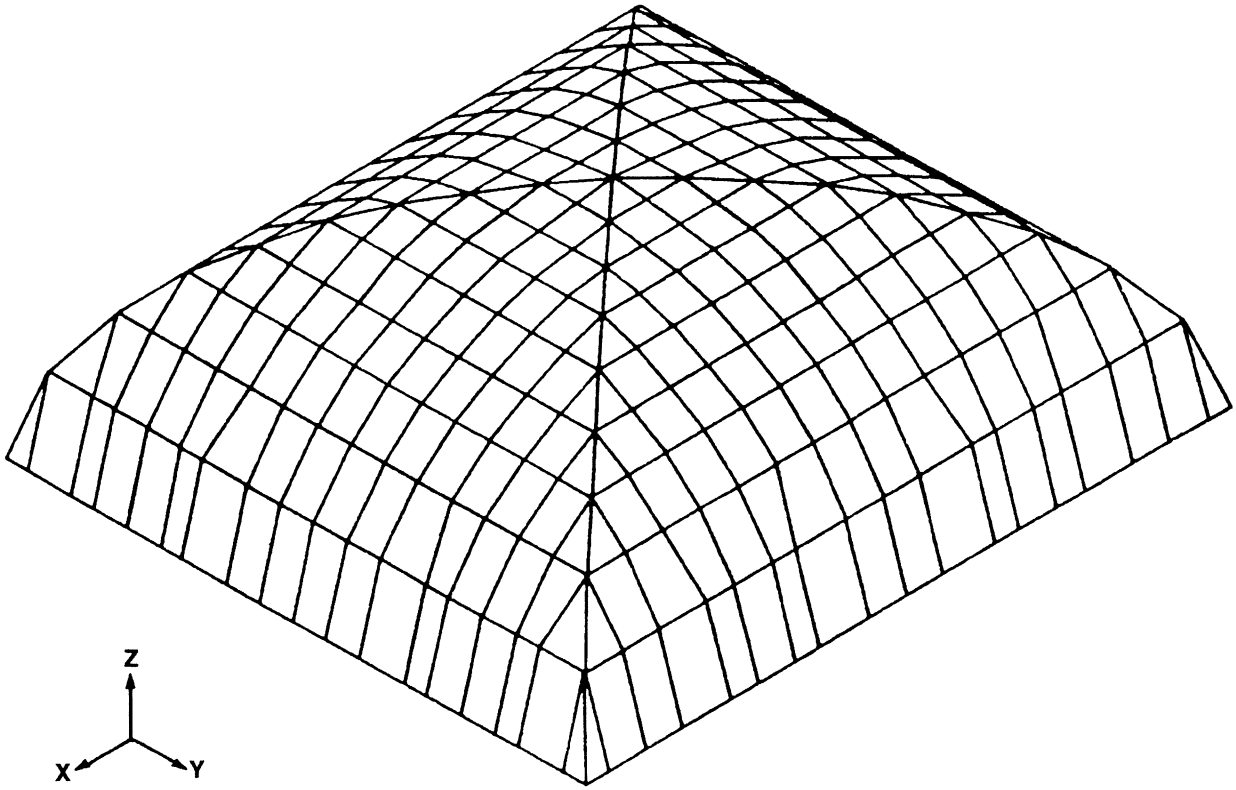


Figure A 4.0-1 Structure with Finite Element Mesh

Mesh definition encompasses the placement of geometric coordinates and the grouping of nodes into elements. For MARC to have a valid mesh definition, the nodes must have geometric coordinates and must be connected to an element.

First, describe the element by entering the element number, the element type, and the node numbers that make up the element.

Next, enter the physical coordinates of the nodal points.

NOTE

You do not need to enter element numbers and node numbers sequentially or consecutively.

A 4.1 DIRECT INPUT

You must enter two types of data into MARC for direct mesh definition: connectivity data, which describes the nodal points for each element, and coordinate data which gives the spatial coordinate of each nodal point. This section describes how to enter this data.

Element Connectivity Data

You can enter connectivity data from either the input option deck (FORTRAN unit 5) or from an auxiliary file. Several blocks of connectivity can be input. For example, the program can read one block from tape and subsequently read a block from the input option deck. Each block must begin with the word CONNECTIVITY. In the case of duplicate specification, the MARC program always uses the data that was input last for a particular element.

Enter the nodal points of two-dimensional elements in a counter-clockwise order. Figure A 4.1-1 illustrates correct and incorrect numbering of element connectivity data.

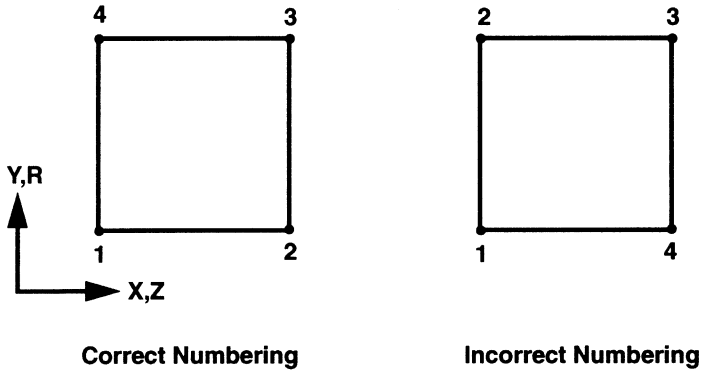


Figure A 4.1-1 Correct/Incorrect Numbering of Two-Dimensional Element Connectivity of 4-Node Elements

When there are eight nodal points on a two-dimensional element, number the corner nodes 1 through 4 in counter-clockwise order. The midside nodes 5 through 8 are subsequently numbered in counter-clockwise order. Figure A 4.1-2 illustrates the correct numbering of element connectivity of 8-node elements.

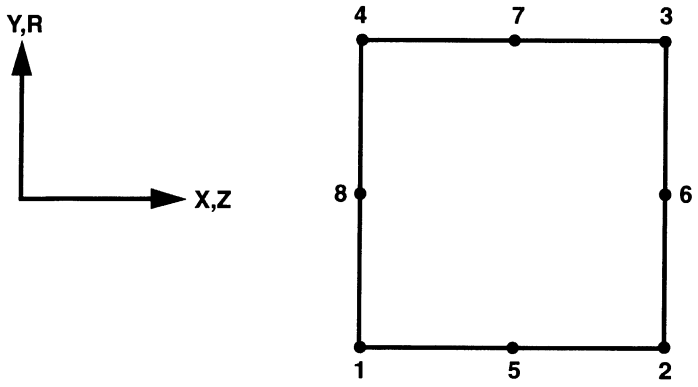


Figure A 4.1-2 Numbering of Two-Dimensional Element Connectivity for 8-Node Quadrilateral Elements

Lower-order triangular elements are numbered using the counter-clockwise rule.

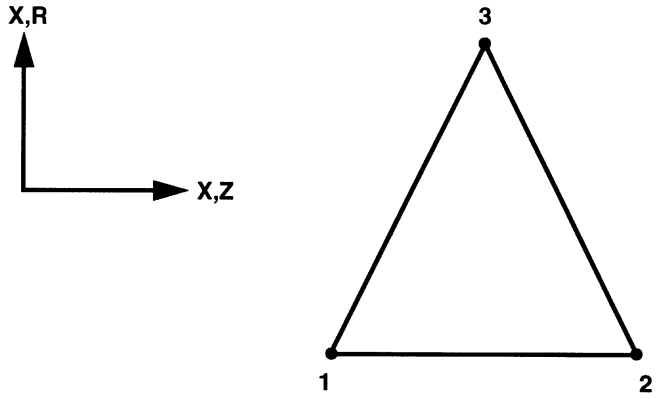


Figure A 4.1-3 Numbering of 3-Node Triangular Element

Note that quadrilateral elements may be collapsed into triangular elements by repeating the last node.

The higher order triangular elements have six nodes, the corner nodes are numbered first in a counter-clockwise direction. The midside nodes 4 through 6 are subsequently numbered as shown in Figure 4.1-4.

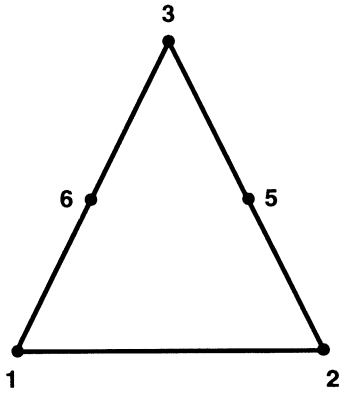


Figure A 4.1-4 Numbering of 6-Node Triangular Element

Number three-dimensional elements in the same order as two-dimensional elements for each plane. Enter nodes for an 8-node brick in counter-clockwise order as viewed from inside the element. First, enter nodes comprising the base, then enter ceiling nodes as shown in Figure A 4.1-5.

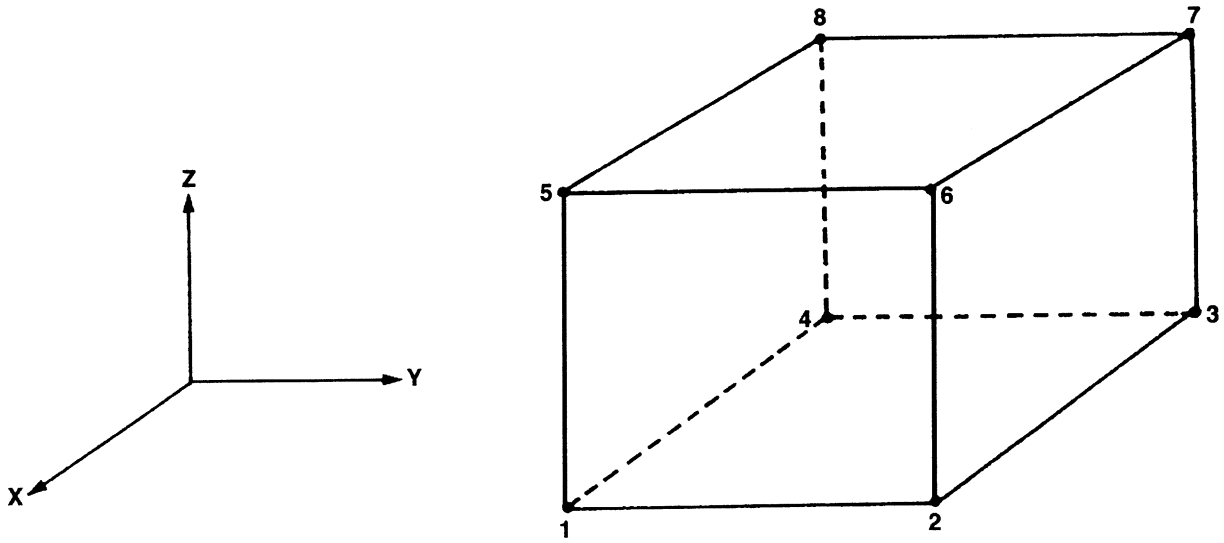


Figure A 4.1-5 Numbering of Element Connectivity for 8-Node Brick

A 20-node brick contains two 8-node planes and four nodes at the midpoints between the two planes. Nodes 1 through 4 are the corner nodes of one face, given in counter-clockwise order as viewed from within the element. Nodes 5 through 8 are on the opposing face; nodes 9 through 12 are midside nodes on the first face, while nodes 13 through 16 are their opposing midside nodes. Finally, nodes 17 through 20 lie between the faces with node 17 between 1 and 5. Figure A 4.1-6 illustrates the numbering of element connectivity for a 20-node brick.

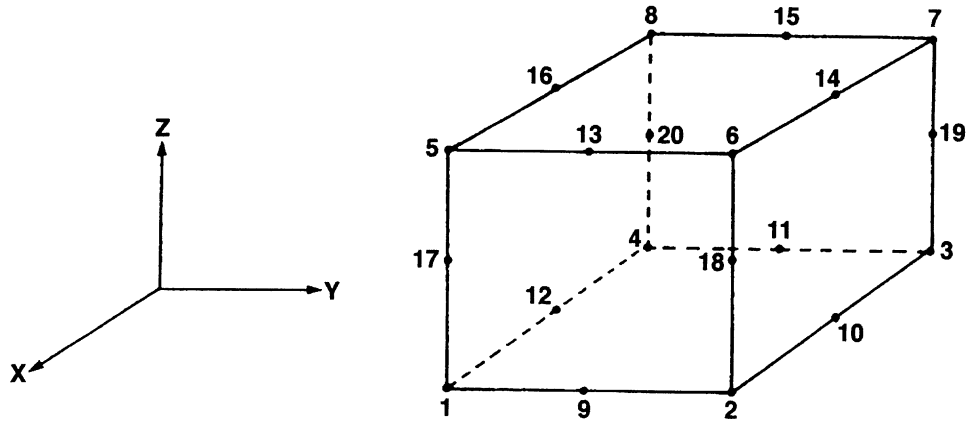


Figure A 4.1-6 Numbering of Three-Dimensional Element Connectivity for 20-Node Brick

The four node tetrahedral is shown in Figure A 4.1-7.

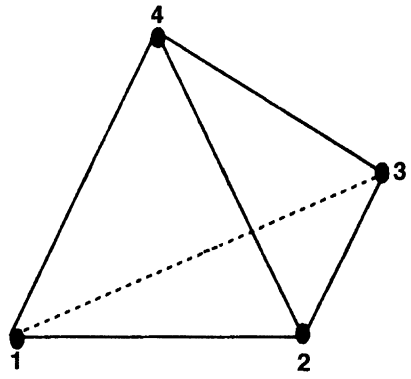


Figure A 4.1-7 Numbering of Four-Node Tetrahedral

The ten-node tetrahedral is shown in Figure A 4.1-8. The corner nodes 1-4 are numbered first. The first three mid-side nodes occur on the first face. Nodes 8, 9, and 10 are between nodes 1 and 4, 2 and 4, and 3 and 4, respectively.

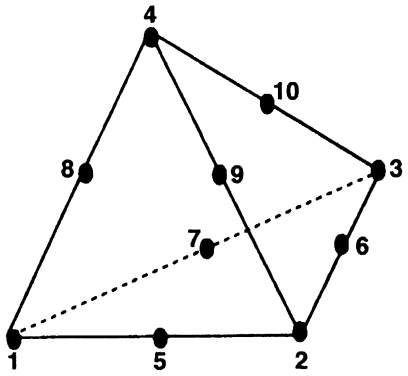


Figure A 4.1-8 Numbering of Ten-Node Tetrahedral

Nodal Coordinate Data

You can enter nodal coordinates directly from the input option deck (FORTRAN unit 5) or from an auxiliary file. You can enter several blocks of nodal coordinate data in a deck. In the case of duplicate specification, the program uses data that was entered last for a particular nodal point in the mesh definition.

Direct nodal input can be used to input local corrections to a previously generated set of coordinates. These options give the modified nodal coordinates.

The CYLINDRICAL option may be used to transform coordinates given in a cylindrical system to a cartesian system.

NOTE

MARC requires the final coordinate data in terms of a single cartesian system. Refer to Volume B to determine the required coordinate data for a particular element type.

Activate/Deactivate

The user has the ability to turn on and off elements using this option, which is useful when modeling ablation or excavation. When the user enters the mesh connectivity, the program assumes that all elements are to be included in the analysis unless they are deactivated. This effectively removes this material from the model. These elements can be reinstated later by using the ACTIVATE option. The previously calculated level of stress will also be reinstated. The use of these options results in nonlinear behavior and will have an effect upon convergence.

A 4.2 USER SUBROUTINE INPUT

User subroutines can be used to generate or modify the data for mesh definition. User subroutine UFCONN generates or modifies element connectivity data. The UFCONN model definition option activates this subroutine. The user subroutine is called once for each element requested. Refer to Volume D, Section 1 for a description of UFCONN and instructions for its use.

User subroutine UFXORD generates or modifies the nodal coordinates. The UFXORD model definition option activates this subroutine. The user subroutine is called once for each node requested. Refer to Volume D, Section 1 for a description of UFXORD and instructions for its use.

A 4.3 MESH2D

MESH2D generates a mesh of quadrilateral or triangular elements for a two-dimensional body of any shape. The generated mesh is written to a separate file, and must be read with the CONNECTIVITY, COORDINATE, and FIXED DISP, etc., options.

Block Definition

In MESH2D, a physical object, or domain, is divided into quadrilateral and/or triangular parts, called "blocks". Quadrilateral blocks are created by the program by mapping with polynomials of the third order from a unit square. These blocks may therefore be used to approximate curved boundaries. The geometry of a quadrilateral block is defined by the coordinates on 12 nodes shown in Figure A 4.3-1. If the interior nodes on an edge of the block are equal to zero or are not specified, the edge of the block is straight.

Triangular blocks have straight edges. The geometry of a triangular block is defined by the coordinates of the three vertices.

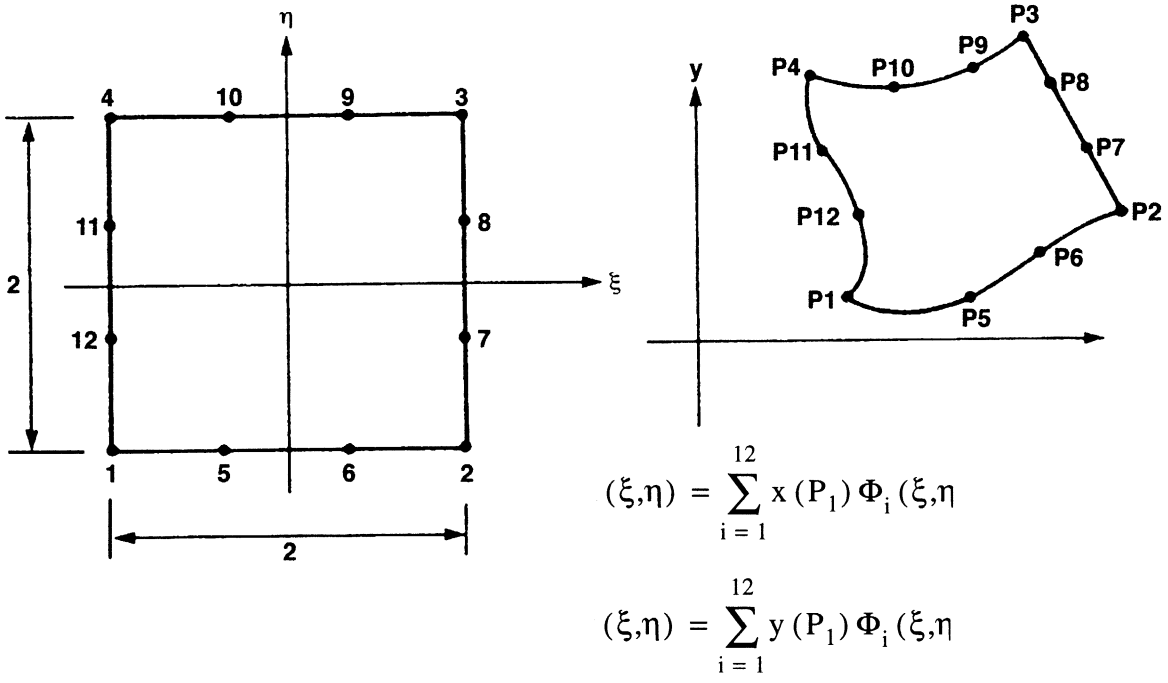


Figure A 4.3-1 Typical Quadrilateral Block

Merging of Nodes

The program creates each block with a unique numbering scheme. The MERGE option fuses all nodes that lie within a small circle, renumbers the nodes in sequence, and then fills up all gaps in the numbering system. You may select which blocks are to be merged together, or you can request that all blocks be merged. You must give the closeness distance for which nodes will be merged.

Block Types

MESH2D generates two types of quadrilateral blocks. Block Type 1 is a quadrilateral block that is covered by a regular grid. The program obtains this grid by dividing the block edge into M by N intervals. Figure A 4.3-2 illustrates the division of block edges into intervals with M=4, N=3.

The P_1 P_4 face of the block is the 1-4 face of triangular elements and the P_1 P_2 face of the block becomes the 1-2 face of quadrilateral elements.

Block Type 2 is a quadrilateral block that allows the transition of a coarse mesh to a finer one. In one direction, the block is divided into M, 2M, 4M ...; while in the other direction, the block is divided into N intervals. Figure A 4.3-3 illustrates the division of Block Type 2 edges into intervals.

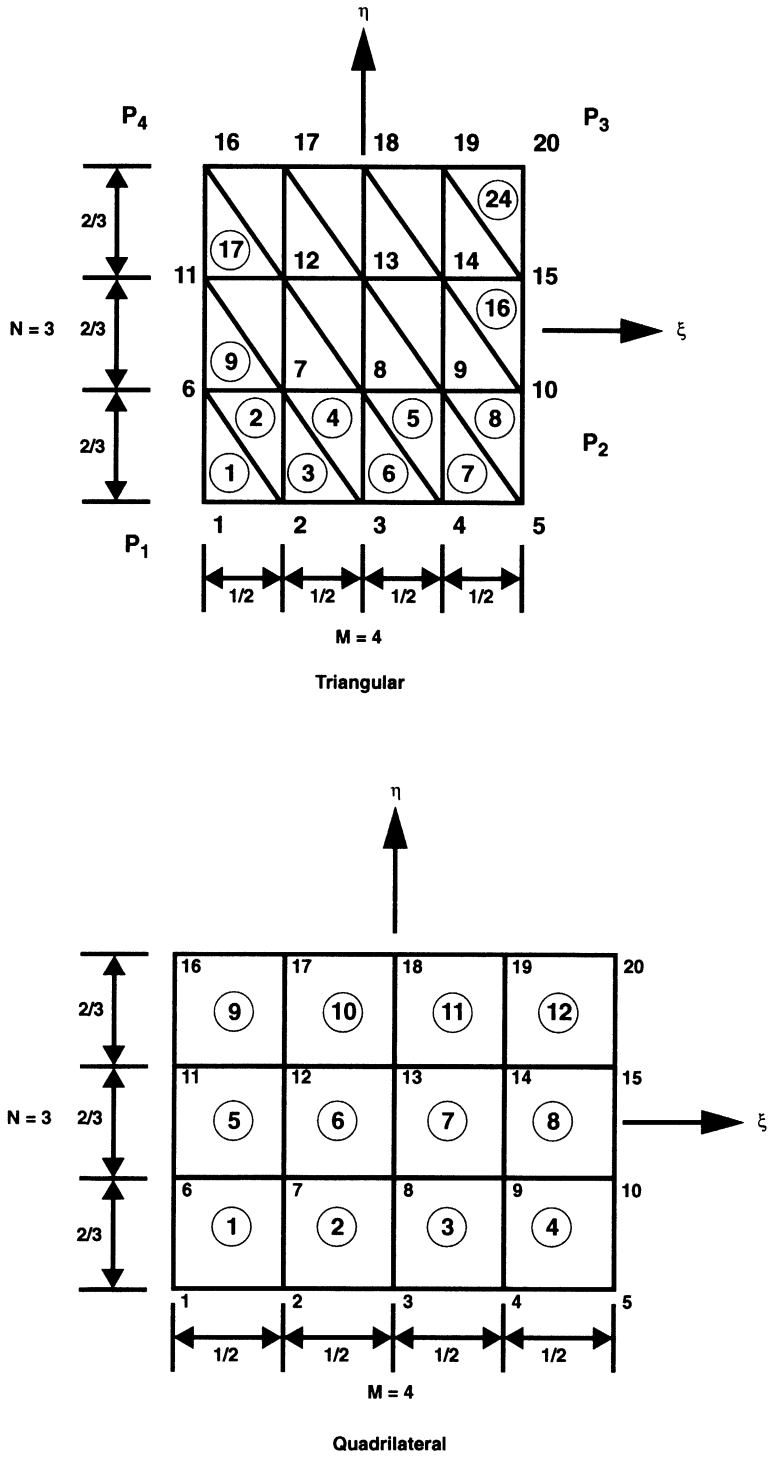


Figure A 4.3-2 Block Type 1

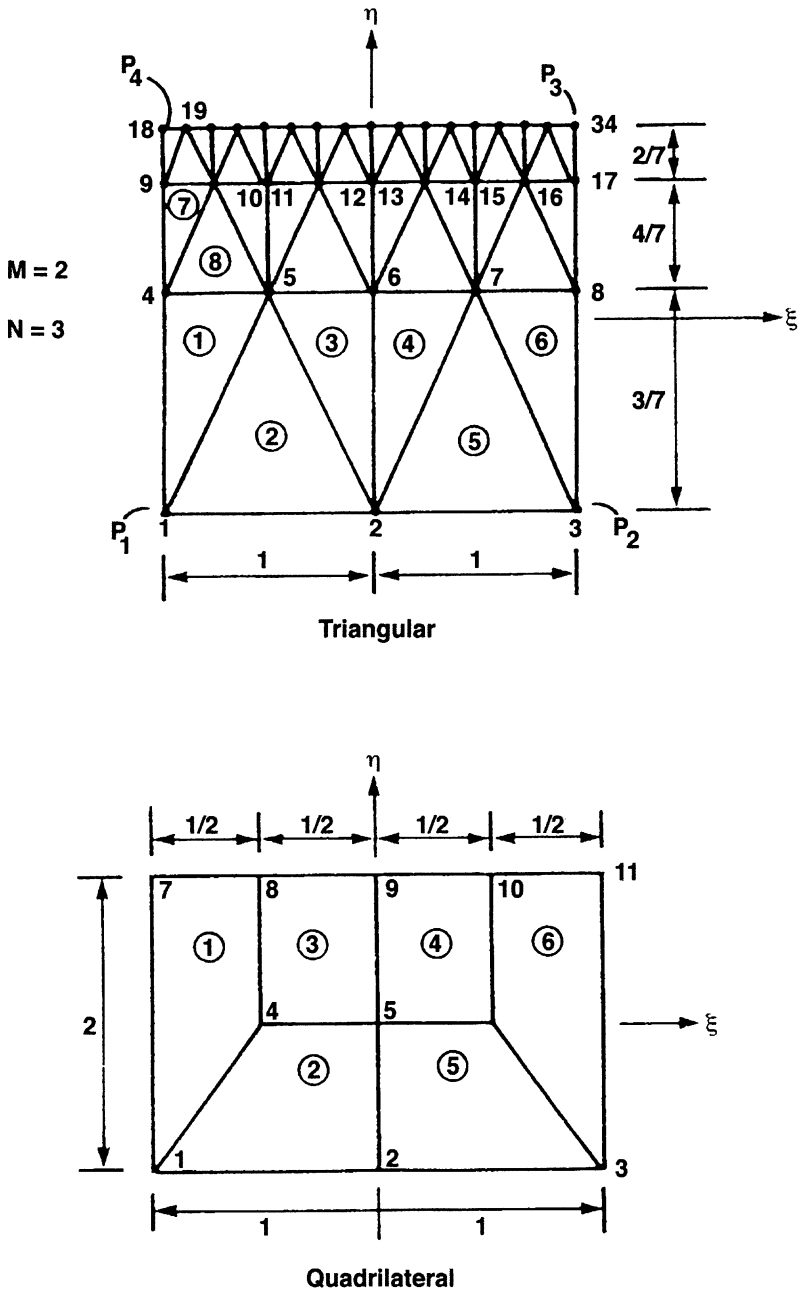


Figure A 4.3-3 Block Type 2

The P_1 P_2 face of the block becomes the 1-2 face of quadrilateral elements, and the P_2 P_3 face of the block is the 2-3 face of triangular elements.

Block Type 3 is a triangular block. The program obtains the mesh for this block by dividing each side into N equal intervals. Figure A 4.3-4 illustrates Block Type 3 for triangular and quadrilateral elements.

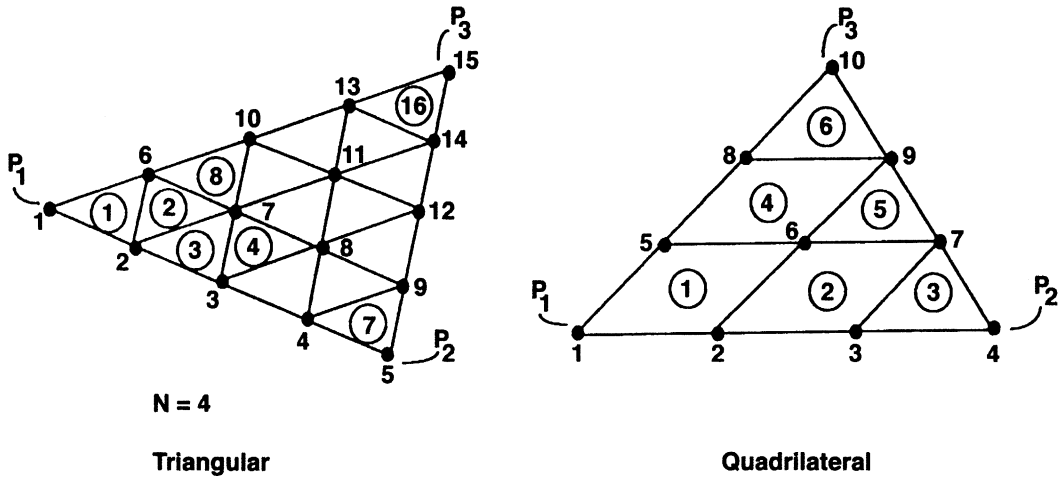


Figure A 4.3-4 Block Type 3

Block Type 4 is a refine operation about a single node of a block. The values of N and M are not used.

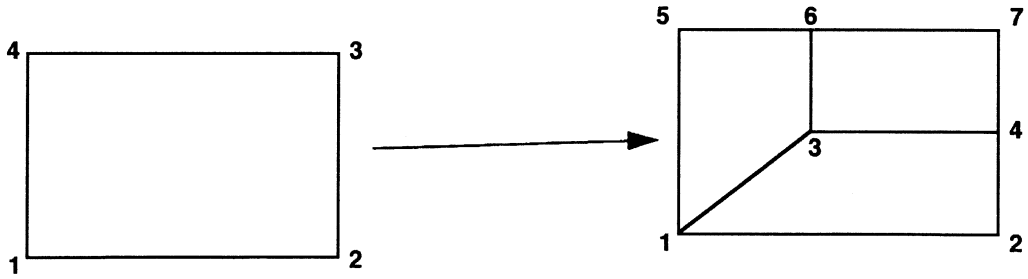


Figure A 4.3-5 Block Type 4

If quadrilateral elements are used in a triangular block, the element near the P_2 P_3 face of the block is collapsed by MESH2D in every row. The P_1 P_2 face of the block is the 1-2 face of the generated elements.

Symmetry, Weighting, and Constraints

MESH2D contains several features that facilitate the generation of a mesh: use of symmetries, generation of weighted meshes, and constraints. These features are discussed below.

MESH2D can use symmetries in physical bodies during block generation. An axis of symmetry is defined by the coordinates of one nodal point, and the component of a vector on the axis. One block may be reflected across many axes to form the domain. Figure A 4.3-6 illustrates the symmetry features of MESH2D.

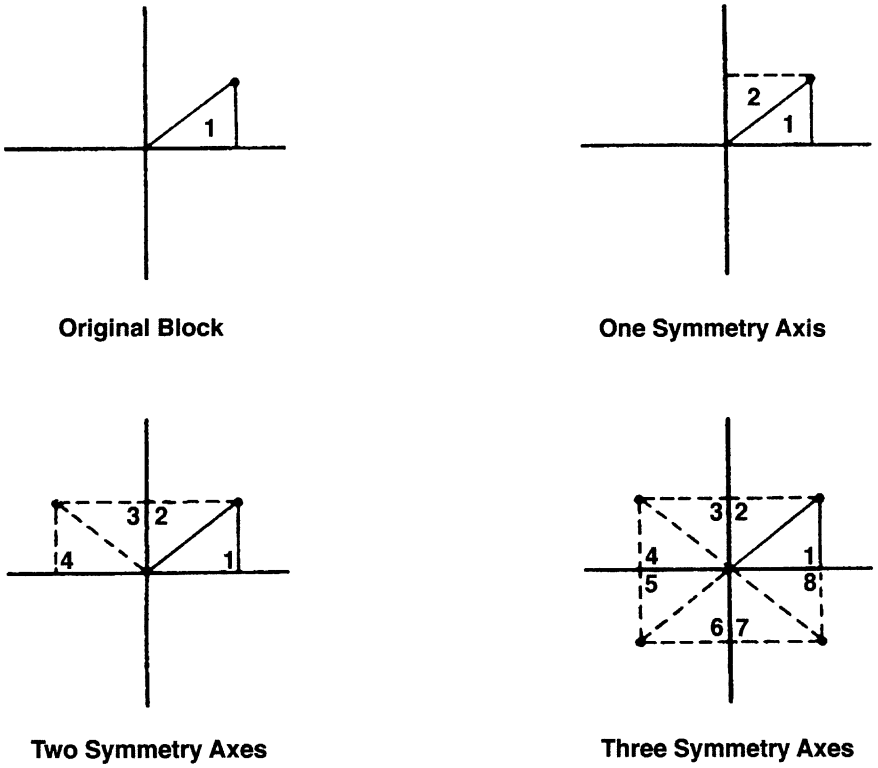


Figure A 4.3-6 Symmetry Option Example

A weighted mesh is generated by the program by spacing the two intermediate points along the length of a boundary. This technique biases the mesh in a way that is similar to the weighting of the boundary points. This is performed according to the third order isoparametric mapping function.

NOTE

If a weighted mesh is to be generated, be cautious not to move the interior boundary points excessively. If the points are moved more than 1/6 of the block length from the 1/3 positions, the generated elements may turn inside-out.

The CONSTRAINT feature generates boundary-condition restraints for a particular degree-of-freedom for all nodes on one side of a block. The feature then writes the constraints into the file after it writes the coordinate data. The FIXED DISP, etc., option must be used to read the boundary conditions generated from the tape.

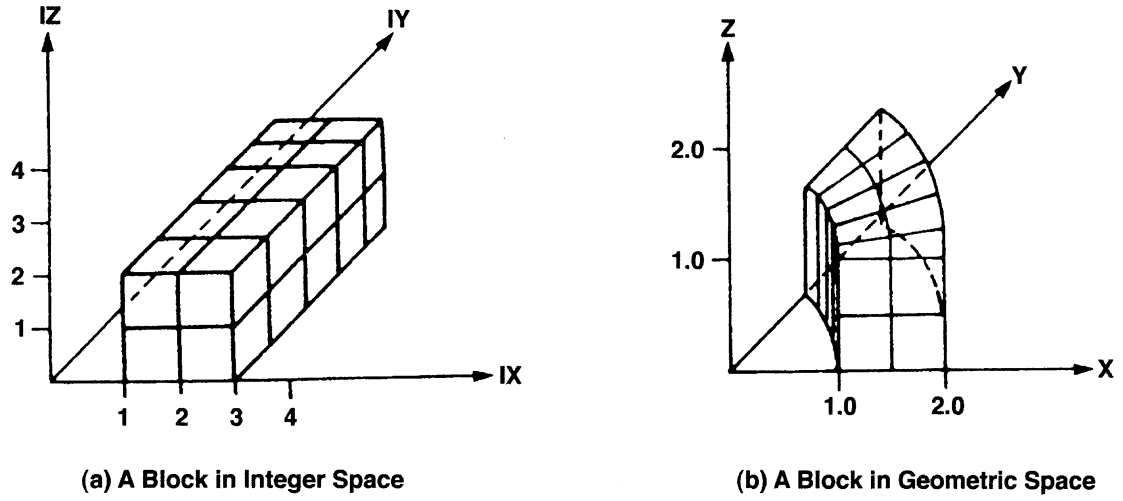
Additional Options

Occasionally, you may want to position nodes at specific locations. The coordinates of these nodes are entered explicitly and substituted for the coordinates calculated by the program. This is performed using the SPECIFIED NODE option. Some additional options in MESH2D are the following.

- MESH2D can be used several times within one input deck.
- The START NUMBER option gives starting node and element numbers.
- The CONNECT option allows forced connections and/or disconnections with other blocks. This option is useful when the final mesh has cracks, tying, or gaps between two parts.
- The MANY TYPES option specifies different element types.

A 4.4 MESH3D

The MESH3D program automatically generates three-dimensional structures using either node distorted cubes (Element Type 7) or 20-node distorted cubes (Element Type 21). The program outputs the three coordinates of each node and the eight or twenty nodes that define each element. If element types other than 7 or 21 are to be used, then use the ALIAS option in MARC.



(a) A Block in Integer Space

(b) A Block in Geometric Space

Figure A 4.4-1 MESH3D Coordinate Spaces

MESH3D builds structures by assembling blocks in both integer space and geometric space. The building blocks are first assembled in integer space, which is defined by a right-handed coordinate system (IX, IY, IZ). The blocks in integer space are rectangular and may have two types of nodal compositions. Blocks that are made up of 8-node elements are unit cubes (as shown in Figure A 4.4-1). Blocks that are made up of 20-node elements are cubes with sides that are two units long. A node in integer space has three integer coordinates.

The structure that is built by MESH3D is also generated in geometric space. Geometric space is defined by a right-handed orthogonal coordinate system (X, Y, Z).

A mapping process relates the block in integer space to the block in geometric space. This mapping is based on an isoparametric 20-node element. You must enter the X,Y,Z coordinates of key points on this 20-node cube. You can enter these coordinates block-by-block, or by cross-reference to a table for predefined stationary points. This mapping process specifies the location and shape of a block in geometric space.

To build the three-dimensional structure, several operations must be performed on blocks and between parts of blocks by the program. These operations are activated by a option which specifies which operation to perform, and provides input data for that operation. The operations are the following:

- POINT
- DEFINE
- MAP
- MERGE
- ERASE
- COPY
- JOIN
- DATA
- OUTPUT

The following sections explain these operations.

POINT Operation

The POINT operation inputs a list of points in space along with their geometric coordinates. This is an optional operation and it can be invoked before the DEFINE and MAP operations. Specify the position of the twenty principal nodes of the MAP block by cross-referencing the corresponding stationary point number. This saves respecifying the coordinates in each MAP operation when the same point in space is used as a principal node of several blocks.

DEFINE Operation

The DEFINE operation creates a new block in integer space. This operation defines element numbers, node numbers, element connectivity, and integer nodal coordinates. MESH3D divides the block into elements, as specified, and numbers the nodes and elements.

During the DEFINE operation, you input the following data:

- The block number
- The number of first element
- The number of first node
- The number of elements in integer directions
- The integer coordinates of the first node in the block

MAP Operation

The MAP operation maps the nodal integer coordinates into the actual (geometric) coordinates by specifying eight or more key points in geometric space. Mentally, superimpose this point numbering scheme on the block in integer space, and ignore the actual node numbers for this operation.

Use the following guidelines to number the blocks for the mapping process.

- Bottom Plane Corners: Number the corners 1, 2, 3, and 4.
- Top Plane Corners: Number these corners 5, 6, 7, and 8. In integer space:
  - Side 1-2 is the IX direction
  - Side 1-4 is the IY direction
  - Side 1-5 is the IZ direction
- Midsides of the Bottom Plane: Number these points 9, 10, 11, and 12.

- Midsides of the Top Plane: Number these points 13, 14, 15, and 16.
- Midsides between the Top and Bottom Planes: Number these points 17, 18, 19, and 20.
- Point 9 is between points 1 and 2.
- Point 13 is between points 5 and 6.
- Point 17 is between points 1 and 5.

You must input the geometric coordinates (X, Y, Z) of the eight corner points. You may also need to input the geometric coordinates of the twelve midside points. When you do not input a midside, the program assumes that edge to be a straight line with the midside point midway between the two corner points. When you input a midside point, that edge is a parabola in geometric space that passes through three key points on the edge.

NOTE

There are two ways to input geometric coordinates. You can specify the coordinates in a list associated with the MAP BLOCK, or you can cross-reference the local point number to a list or prespecified stationary points. Use the second option when the same point appears in several blocks.

MERGE Operation

The MERGE operation merges two blocks and then renumbers their elements and nodes. When two nodes coincide in integer space, this operation merges them into one node. The resultant node has the geometric coordinates and the node numbers of the nodes in the first block. The program renumbers the nodes and elements in the second block so there is no gap in the numbering of the new, merged block.

ERASE Operation

The ERASE operation deletes selected elements and nodes from a particular block. This operation renumbers elements and nodes in ascending order to close the gaps in the numbering schemes.

COPY Operation

The COPY operation creates a new block (N2) by copying an existing block (N1). The new block may be translated, rotated, and/or reflected about a plane in both integer space and geometric space.

To specify the translation, rotation, and reflection in geometric space, input the geometric coordinates of three non-colinear points in block N1 (P11, P12, P13), and three non-colinear points in block N2 (P21, P22, P23).

The local coordinate systems of block N1 and block N2 are defined by the following:

- Point P11 and P12 define axis direction X1.
- Point P13 defines a plane which determines axis direction Y1, perpendicular to X1.
- Z1 is the cross-product of the first two axis directions (right-hand rule).

The COPY operation rotates and translates block N1 so that X1 coincides with X2, Y1 coincides with Y2, and Z1 coincides with Z2. If you request reflection, Z1 coincides with Z2. This operation leaves block N1 intact and creates a new block N2.

To specify the translation, rotation, and reflection in integer space, input integer coordinates of six points. These six points in block N1 are K11, K12, K13; in block N2, they are K21, K22, and K23. In order to guarantee that all nodes in integer space are translated and rotated into points with integer coordinates, follow the guidelines listed below.

- Points K11 and K12 must be one unit apart.
- Points K21 and K22 must be one unit apart.
- The distance from K13 to the line K11-K12 must be one unit.
- The distance from K23 to the line K21-K22 must be one unit.

JOIN Operation

The JOIN operation joins selected pairs of nodes within one block. This operation is useful for forming circular and other multiple-connected structures. The JOIN operation retains the first node (N1) of the pair, along with its integer and geometric coordinates, and deletes the second node (N2). The nodes are then renumbered to close the gaps in the numbering scheme.

OUTPUT Operation

The OUTPUT operation prints out information on a selected block of elements. This information includes:

- The element numbers and their associated nodes
- Node numbers
- The three integer coordinates of the nodes
- The three geometric coordinates of the nodes

This operation is available so that you can check data.

DATA Operation

The DATA operation produces the printout of the OUTPUT operation and writes a file containing element connectivity and node point coordinates to FORTRAN unit 1. The MARC system of finite element programs uses the element connectivity and node point coordinates written to this file for analysis. This file contains the following information for each element:

- The element number
- The element type (7 or 21)
- Nodes of this element

The data file contains the following information for each node:

- The node number
- X, Y, and Z coordinates

A 4.5 MENTAT

Mentat is an interactive program which facilitates mesh definition by generating element connectivity and nodal coordinates. Some of the Mentat capabilities relevant to mesh generation are listed below.

- Prompts you for connectivity information and nodal coordinates. Accepts input from a keyboard or mouse.
- Accepts coordinates in several coordinate systems (cartesian, cylindrical, or spherical).
- Translates and rotates (partial) meshes.
- Combines several pre-formulated meshes.
- Duplicates a mesh to a different physical location.
- Generates a mirror image of a mesh.
- Subdivides a mesh into a finer mesh.
- Automatic mesh generation in two and three dimensions.
- Imports geometric and finite element data from CAD systems.
- Smooths nodal point coordinates to form a regular mesh
- Converts geometric surfaces to meshes.
- Refines a mesh about a point or line.
- Expands line mesh into a surface mesh, or a surface mesh into a solid mesh.
- Calculates the intersection of meshes.
- Maps nodal point coordinates onto prescribed surfaces.
- Writes input data file for connectivity and coordinates in MARC format for use in future analyses.
- Apply boundary conditions to nodes and elements.
- Define material properties.
- Submit MARC jobs.

Use the CONNECTIVITY and COORDINATE options to read the information generated by Mentat into the MARC program. These options are discussed in Section 4.1.

A 4.6 FXORD OPTION

The FXORD model definition option (Volume C, Section 2) generates doubly curved shell elements of Type 4, 8, or 24 for the geometries most frequently found in shell analysis. Since the mathematical form of the surface is well-defined, the program can generate the 11 or 14 nodal coordinates needed by Element Type 8, 24, or 4 to fit a doubly curved surface from a reduced set of coordinates. For example, you can generate an axisymmetric shell by entering only four coordinates per node. The FXORD option automatically generates the complete set of coordinates required by the elements in the program from the mathematical form of the surface.

A rotation and translation option is available for all components of the surface to give complete generality to the surface generation. The input to FXORD consists of the reduced set of coordinates given in a local coordinate system and a set of coordinates which orient the local system with respect to the global system used in the analysis. The program uses these two sets of coordinates to generate a structure made up of several shell components for analysis.

The FXORD option allows for the generation of several types of geometries. Volume C, Figure C4.6-1 illustrates these options. Because you may need to analyze shells with well-defined surfaces not available in this option, you can use the UFXORD user subroutine to perform your own coordinate generation (Volume D, Section 1). The FXORD option may also be used to convert cylindrical coordinates or spherical coordinates to Cartesian coordinates for continuum elements.

Major Classes of the FXORD Option

The following cases are considered:

- Shallow Shell
- Axisymmetric Shell
- Cylindrical Shell Panel
- Circular Cylinder
- Plate
- Curved Circular Cylinder
- Convert Cylindrical to Cartesian
- Convert Spherical to Cartesian

Shallow Shell (Type I)

Type 1 is a shallow shell with

$$\theta_1 = x_1, \quad \theta_2 = x_2 \tag{A 4.6-1}$$

The middle surface of Figure A 4.6-1 (Type 1) is defined by an equation of the form

$$x_3 = x_3(x_1, x_2) \tag{A 4.6-2}$$

and the surface is determined when the following information is given at each node.

$$x_1, x_2, x_3, \frac{\partial x_3}{\partial x_1}, \frac{\partial x_3}{\partial x_2}, \frac{\partial^2 x_3}{\partial x_1 \partial x_2} \qquad \text{(A 4.6-3)}$$

The last coordinate is only necessary for Element Type 4.

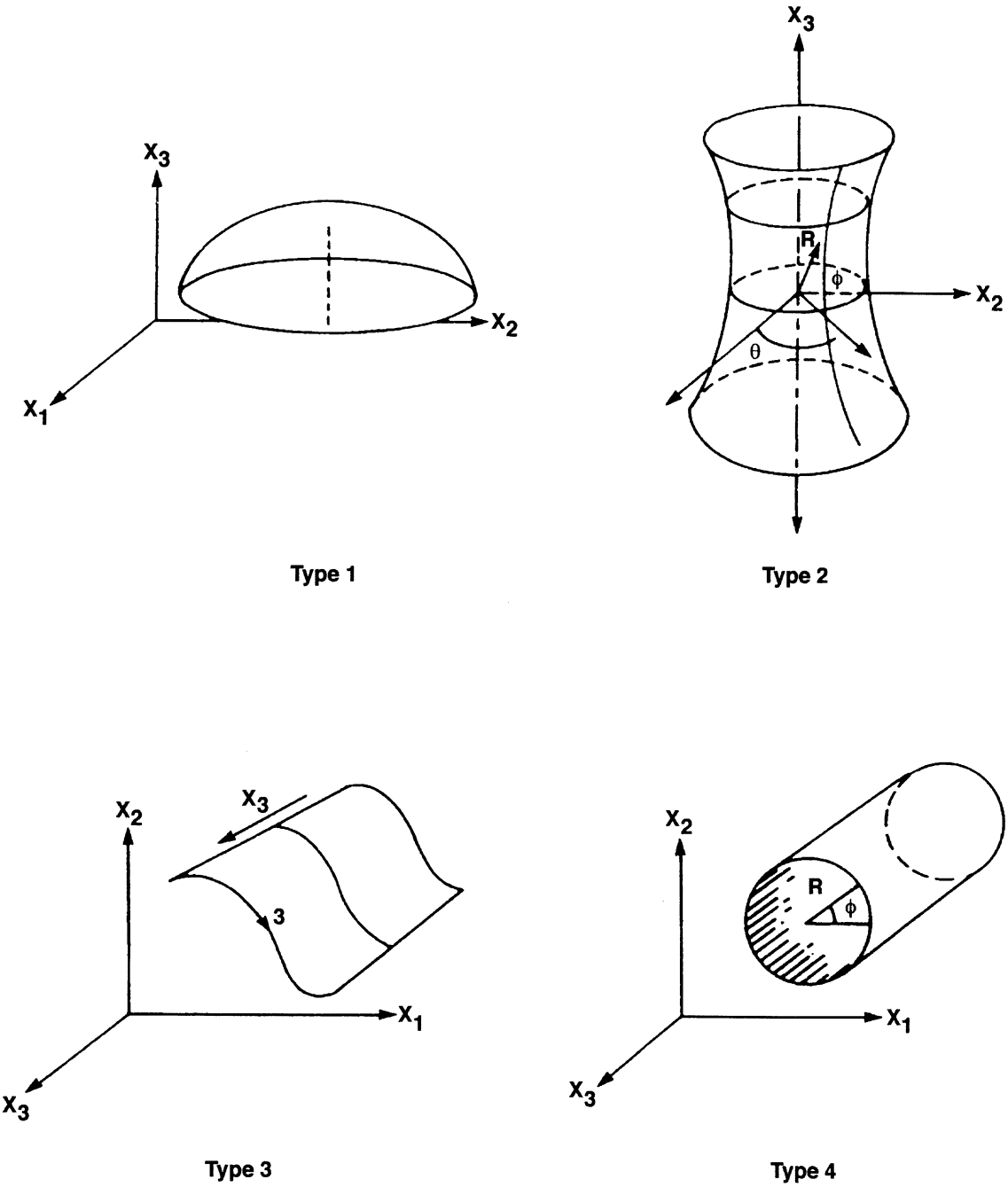


Figure A 4.6-1 Classification of Shells

Axisymmetric Shell (Type 2)

The middle surface symmetric to the x_3 -axis (Figure A 4.6-1, Type 2) is defined as:

$$\begin{aligned} x_1 &= R(\phi)\cos\phi\cos\theta \\ x_2 &= R(\phi)\cos\phi\sin\theta \\ x_3 &= R(\phi)\sin\phi \end{aligned} \qquad \text{(A 4.6-4)}$$

where ϕ and θ are the angles shown in Figure A 4.6-1. In this case, the surface is defined by

$$\theta, \phi, R, \frac{dR}{d\phi} \qquad \text{(A 4.6-5)}$$

The angles θ and ϕ are given in degrees.

Cylindrical Shell Panel (Type 3)

The middle surface is the cylinder defined by Figure A 4.6-1.

$$\begin{aligned} x_1 &= x_1(s) \\ x_2 &= x_2(s) \\ x_3 &= x_3 \end{aligned} \qquad \text{(A 4.6-6)}$$

The nodal geometric data required is

$$s, x_3, x_1, x_2, \frac{dx_1}{ds}, \frac{dx_2}{ds} \qquad \text{(A 4.6-7)}$$

Circular Cylinder (Type 4)

This is the particular case of Type 3 where the curve

$$x_1(s), x_2(s) \qquad \text{(A 4.6-8)}$$

is the circle given by Figure A 4.6-1 (Type 4).

$$\begin{aligned} x_1 &= R\cos\theta \\ x_2 &= R\sin\theta \end{aligned} \qquad \text{(A 4.6-9)}$$

The only nodal information is now

$$\theta, x_3, R \qquad \text{(A 4.6-10)}$$

Note that θ is given in degrees and, because R is constant, it needs to be given for the first nodal point only.

Plate (Type 5)

The shell is degenerated into the plate

$$x_3 = 0 \qquad \text{(A 4.6-11)}$$

The data is reduced to

$$x_1, x_2 \qquad \text{(A 4.6-12)}$$

Curved Circular Cylinder (Type 6)

Figure A 4.6-2 illustrates this type of geometry.

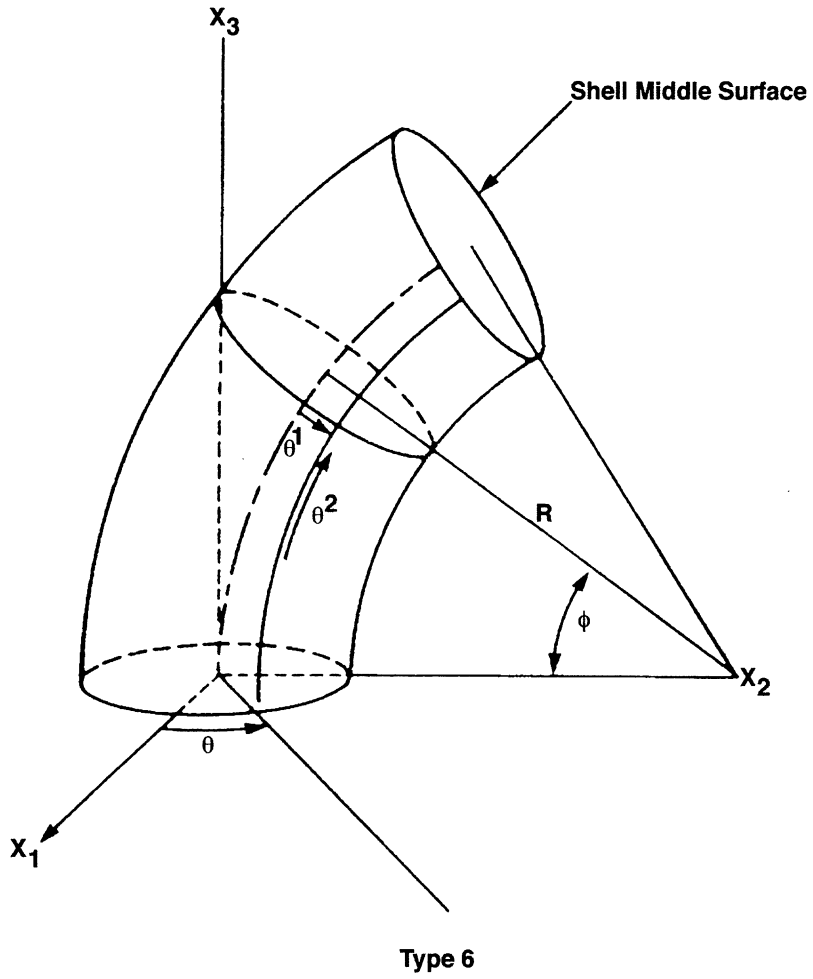


Figure A 4.6-2 Curved Circular Cylinder

The middle surface of the shell is defined by the equations

$$x_1 = r\cos\theta$$
$$x_2 = r\sin\theta\cos\phi + R(1 - \cos\phi) \qquad \text{(A 4.6-13)}$$
$$x_3 = (R + -r\sin\theta r\sin\phi)\sin\phi$$

The Gaussian coordinates on the surface are

$$\theta_1 = r\theta$$
$$\theta_2 = R\phi \qquad \text{(A 4.6-14)}$$

and form an orthonormal coordinate system. The nodal point information is

$$\theta, \phi, r, R \qquad \text{(A 4.6-15)}$$

θ and ϕ in degrees. You need to specify the radii r and R only for the first nodal point.

Convert Cylindrical to Cartesian (Type 7)

Type 7 allows you to enter the coordinates for continuum elements in cylindrical coordinates, which are converted by MARC to Cartesian coordinates. In this way, you can enter R, θ, Z and obtain x, y, z where θ is given in degrees and

$$x = R\cos\theta$$
$$y = R\sin\theta \qquad \text{(A 4.6-16)}$$
$$z = Z$$

Convert Spherical to Cartesian (Type 8)

Type 8 allows you to enter the coordinates for continuum elements in spherical coordinates, which are converted by MARC to Cartesian coordinates. In this way, you can enter R, θ, ϕ and obtain x, y, z where θ and ϕ are given in degrees and

$$x = R\cos\theta\sin\phi$$
$$y = R\sin\theta\cos\phi \qquad \text{(A 4.6-17)}$$
$$z = R\cos\phi$$

Recommendations on Use of the FXORD Option

When a continuous surface has a line of discontinuity, for example, a complete cylinder at $\theta = 0^o = 360^o$, you must place two nodes at each nodal location on the line to allow the distinct coordinate to be input. You must use tying Type 100 to join the degrees-of-freedom. Generally, when different surfaces come together, you must use the intersecting shell tyings.

The FXORD option cannot precede the COORDINATES option, because it uses input from that option.

A 4.7 MARC-PIPE

The MARC-PIPE program is used in conjunction with Element Types 14 and 17 to generate the meshes for piping systems.

The piping system consists of a set of joints that are identified and numbered. A joint is any change of section, branch point, or change of curvature of the pipe. The pipe is modeled with straight beams (Element Type 14) or elbow elements (Element Type 17) between any two joints.

For the MARC-PIPE program to generate a mesh for a piping system, you must enter the following information:

- Type of the section - straight (Type 0) or curved (Type 1 or 2)
- Number of the joints at the two ends of the section
- Number of divisions along the pipe
- Number of divisions around the pipe for the curved sections

For the MARC-PIPE program to generate a mesh for curved sections, you must enter the following information:

- The center of curvature for the bend or the intersection point of the two tangent pipes
- The pipe radius

NOTE

Only planar bends can be modeled with MARC-PIPE. When a bend curves in two different planes, you must insert a short straight section between them.

The output of the MARC-PIPE program consists of the CONNECTIVITY and COORDINATE sets for MARC input, and the TYING data. The data is written on unit 1. You can then input the tape directly to MARC for analysis.

A 4.8 INCREMENTAL MESH GENERATORS

Incremental mesh generators are a collection of options available in MARC to assist you in generating the mesh. Incremental mesh generators generate connectivity lists by repeating patterns and generate nodal coordinates by interpolation. Use these options directly during the model definition phase of the input.

During the model definition phase, you can often divide the structure into regions, or blocks, for which a particular mesh pattern can be easily generated. This mesh pattern is established for each region and is associated with a single element connectivity list. Use the CONNECTIVITY option to input this element connectivity list. The incremental mesh generators then generate the remainder of the connectivity lists.

Critical nodes define the outline of the regions to be analyzed. Use the COORDINATES option to enter the critical nodes. The incremental mesh generators complete the rest and join the regions by merging nodes.

A special connectivity interpolator option generates midside nodes for elements where these nodes have not been specified in the original connectivity. A separate mesh generation run is sometimes required to determine the position of these nodes. This run may be followed by mesh display plotting.

The incremental mesh generators are listed below:

Element Connectivity Generator – The CONN GENER option repeats the pattern of the connectivity data for previously defined master elements. One element can be removed for each series of elements, allowing the program to generate to tapered mesh. Two elements can be removed for each series with triangular elements.

Element Connectivity Interpolator – The CONN FILL option completes the connectivity list by generating midside nodes. You first generate the simpler quadrilateral or brick elements without the midside nodes. You can then fill in the midside nodes with this option.

Coordinate Generator – The NODE GENER option creates a new set of nodes by copying the spacing of another specified set of nodes.

Coordinate Interpolator – The NODE FILL generates intermediate nodes on a line defined by two end nodes. The spaces between the nodes can be varied according to a geometric progression.

Coordinate Generation for Circular Arcs – The NODE CIRCLE option generates the coordinates for a series of nodes which lie on a circular arc.

Nodal Merge – The NODE MERGE option merges all nodes which are closer than a specified distance from one another and it eliminates all gaps in the nodal numbers.

A 4.9 BANDWIDTH OPTIMIZATION

MARC can minimize the nodal bandwidth of a structure in several ways. The amount of storage is directly related to the size of the bandwidth, and the computation time increases in proportion to the square of the average bandwidth.

When the EBE iterative solver is used, the bandwidth optimizer has no influence. Using this method, it is beneficial if the elements are numbered in an orderly fashion.

The OPTIMIZE option allows you to choose from several bandwidth optimization algorithms. The minimum degree algorithm should only be used if the direct sparse solver is used.

The three available OPTIMIZE options are listed in Table A 4.9-1.

NOTE

This option creates an internal node numbering that is different from your node numbering. Use your node numbering for all inputs. All output appears with your node numbering. The occurrence of gap or Herrmann elements may change the internal node numbers. On occasion, this change may result in a non-optimal node numbering system, but this system is necessary for successful solutions.

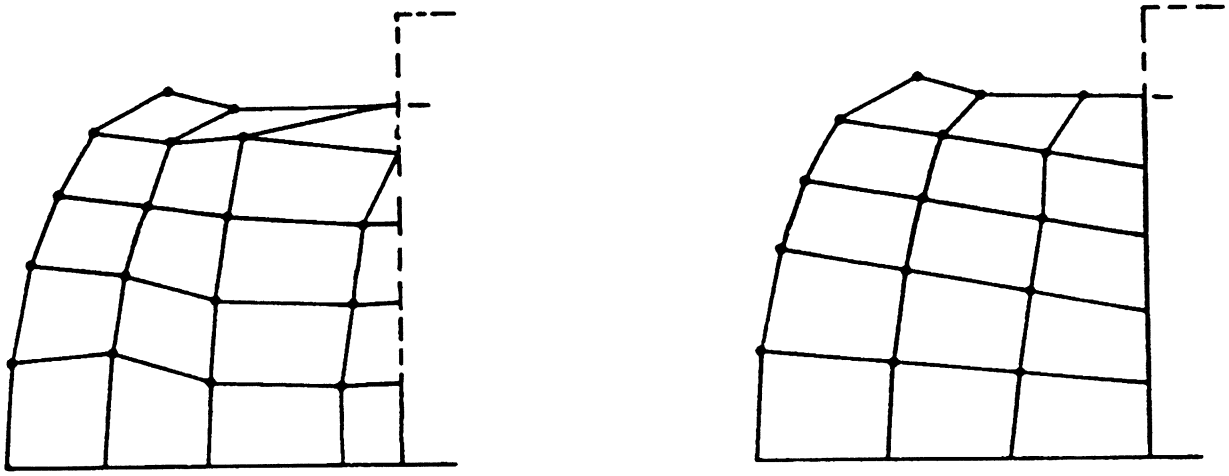
The nodal correspondence table obtained through this process can be saved and then used in subsequent analyses. This eliminates the need to go through the optimization step in later analyses. The correspondence table is used to relate the user-defined node (external) numbers to the program-optimized (internal) node numbers and vice versa.

Table A 4.9-1 Bandwidth Optimization Options

Option Number	Remarks
2	Cuthill-McKee algorithm
9	Sloan
10	Minimum Degree Algorithm

A 4.10 REZONING

The REZONING option defines a new mesh and transfers the state of the old mesh to the new mesh. Elements or nodes can be either added to or subtracted from the new mesh. This procedure requires a subincrement to perform the definition of the new mesh. The rezoning capability may be used for two- and three-dimensional continuum elements and for shell elements 22 and 75. See Figure A 4.10-1 for an example of REZONING.



Before Rezoning **After Rezoning**

Figure A 4.10-1 Mesh REZONING

A 4.11 SUBSTRUCTURE

The MARC multi-level procedure allows superelements to be used. One self-descriptive database stores all data needed during the complete analysis. The user has only to ensure that this database is saved after every step of the analysis.

The advantages of substructuring are the following:

- Separates linear and nonlinear parts of the model
- Allows repetition of symmetrical or identical parts of the model for linear elastic analysis
- Separates large models into multiple, moderate-size models
- Separates fixed model parts from parts of the model that may undergo design changes

A disadvantage of substructuring is the large amount of data that must be stored on the database.

Three steps are involved in a substructuring run.

- The superelement generation step is done for every superelement at a certain level.
- The use of superelements in subsequent MARC runs is done at the highest level, or is incorporated into Step 1 for the intermediate levels.
- Recovery of solutions within a certain superelement may or may not be done for every superelement.

Substructuring in MARC is only possible for static analysis. Nonlinearities are not allowed with a superelement. You, as a user, must ensure that nonlinearities are not present.

The maximum number of levels in a complete analysis is 26. The maximum number of substructures in the complete analysis is 676.

During superelement generation in MARC, you can generate a complete new superelement or you can copy a previously defined superelement with identical or newly defined external load conditions. Any number of superelements may be formed in a generation run.

MARC offers flexibility in the use of superelements by allowing rotation or mirroring of a superelement. If a run is nonlinear, superelements are treated as linear elastic parts. At every increment, you can perform a detailed analysis of certain substructures by descending down to the desired superelements. Use the normal MARC control algorithm (AUTO INCREMENT, AUTO LOAD, PROPORTIONAL INCREMENT) to control the load on the superelements.

Use parameter option SUBSTRUC to declare the formation of a substructure. Use the SUBSTRUCTURE model definition option to define the nodes and degrees-of-freedom that belong to the substructure. Use the SUPER parameter option combined with the SUPERINPUT model definition data to use the substructure in a later run. Use the user subroutine SSTRAN to rotate or mirror a substructure.

Technical Background

The system of equations for a linear static structure is

$$Ku = P \qquad \text{(A 4.11-1)}$$

When local degrees-of-freedom (subscripted l) and external degrees-of-freedom (subscripted e) are considered, this can be rewritten as

$$\begin{bmatrix} K_{ll} & K_{el} \\ K_{le} & K_{ee} \end{bmatrix} \begin{pmatrix} u_l \\ u_e \end{pmatrix} = \begin{pmatrix} P_l \\ P_e \end{pmatrix} \qquad \text{(A 4.11-2)}$$

To obtain both the stiffness matrix and the load vector of the substructure, it is necessary to eliminate u_l and rewrite the above system with u_e as the only unknown splitting the above equation.

$$K_{ll}u_l + K_{el}u_e = P_l$$

and (A 4.11-3)

$$K_{le}u_l + K_{ee}u_e = P_e$$

The first equation can be written as

$$u_l = -K_{ll}^{-1} \bullet K_{el}u_e + K_{ll}^{-1}P_l \qquad \text{(A 4.11-4)}$$

Substituting this equation into the second

$$-K_{le}K_{ll}^{-1}K_{el}u_e + K_{le}K_{ll}^{-1}P_l + K_{ee}u_e = P_e \qquad \text{(A 4.11-5)}$$

This can be rewritten as

$$K_{ee}^{*}u_e = P_e^{*} \qquad \text{(A 4.11-6)}$$

where

$$K_{ee}^{*} = K_{ee} - K_{le}K_{ll}^{-1}K_{el} \qquad \text{(A 4.11-7)}$$

and

$$P_e^{*} = P_e - K_{le}K_{ll}^{-1}P_l \qquad \text{(A 4.11-8)}$$

K_{ee}^{*} and P_e^{*} are solved by the triangularization of K_{ll}, the forward and backward substitution of K_{el} and P_l, respectively, and premultiplication with K_{le}, K_{ee}^{*} and P_e^{*} are used in the next part of the analysis with other substructures or with another element mesh. That analysis results in the calculation of u_e.

You can now calculate the local degrees-of-freedom and/or the stresses using the following procedure:

$$K_{el} \bullet u_l = P_l - K_{el}u_e \qquad \text{(A 4.11-9)}$$

which can be written as

$$u_l = K_{ll}^{-1} \bullet P_e^* \qquad \text{(A 4.11-10)}$$

where

$$P_e^* = P_e - K_{el} u_e \qquad \text{(A 4.11-11)}$$

The displacement of the substructure is therefore known, and stresses and strains can be calculated in the normal way.

Scaling Element Stiffness

Occasionally it is desirable to perform a scalar multiplication of the stiffness, mass and load matrix to represent a selective duplication of the finite element mesh.

The STIFFSCALE option may be used to enter the scaling factor for each element. In this case, the global stiffness, mass, and load matrices are formed as follows:

$$K^g = \Sigma s_i K_i^{el}, \quad M^g = \Sigma s_i M_i^{el}, \quad \text{and} \quad F^g = \Sigma s_i f_i^{el} \qquad \text{(A 4.11-12)}$$

Note that no transformation of the stiffness matrix occurs and that point loads are not scaled.

A 4.12 BEAM SECTION DEFINITION OPTION

The BEAM SECTION definition option inputs data to define the sectional properties for three-dimensional beam elements. Include this option if you are using Element Types 13, 77, or 79 or Element Types 14, 25, 76, or 78 with a noncircular section, or if you are using Element Types 52 or 98 and torsional and shear stiffness must be defined independently.

The convention adopted for the local (beam) coordinate system is: the first and second directions (local X and Y) at a point are normal to the beam axis; the third director (local Z) is tangent to the beam axis and is in the direction of increasing distances along the beam. The director set forms a right-handed system.

Orientation of the Section in Space

The beam axis in an element is interpolated from the two nodes of the element.

$$x, y, z, \frac{dx}{ds}, \frac{dy}{ds}, \frac{dz}{ds} \qquad \text{(A 4.12-1)}$$

where the last three coordinates are only used for element 13.

The beam section orientation in an element is defined by the direction of the first director (local X) at a point, and this direction is specified via the coordinates of an additional node or through the GEOMETRY option (see Volume B).

Definition of the Section

You can include any number of different beam sections in any problem. Data options following the BEAM SECTION definition parameter option of the MARC input (see Volume C) define each section. The program numbers the sections in the order they are entered. To use a particular section for a beam element, set EGEOM2 (GEOMETRY option, Option 2, Columns 11-20) to the floating-point value of the section number, for example, 1, 2, or 3. The program uses the default circular section for the closed section beam elements (14, 25, 76, 78) if EGEOM1 is nonzero. The program uses the default solid rectangular cross-section for elements 52 or 98 if EGEOM1 is nonzero. Figure 4.12-1 shows how the thin-walled section is defined using input data.

The rules and conventions for defining a section are listed below:

1. An x^1 - y^1 coordinate system defines the section, with x^1 the first director at a point of the beam. The origin of the x^1 - y^1 system represents the location of the node with respect to the section.
2. Enter the section as a series of branches. Branches can have different geometries, but they must form a complete traverse of the section in the input sequence so that the endpoint of one branch is the start of the next branch. It is often necessary for the traverse of the section to double back on itself. To cause the traverse to do this, specify a branch with zero thickness.

3. You must divide each branch into segments. The stress points of the section are the branch division points. The stress points are the points used for numerical integration of a section's stiffness and for output for stress results. Branch endpoints are always stress points. There must always be an even number of divisions (nonzero) in any branch. Not counting branches of zero thickness, you can use a maximum of 31 stress points (30 divisions) in a complete section.

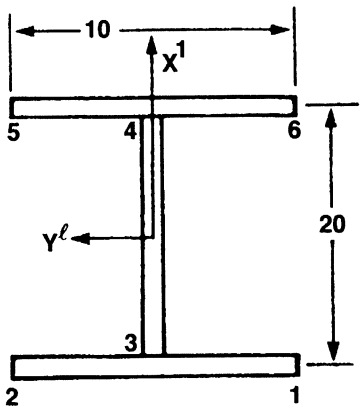
4. Branch thickness will vary linearly between the values given for branch endpoint thickness. The thickness can be discontinuous between branches. A branch is assumed to be of constant thickness equal to the thickness given at the beginning of the branch if the thickness at the end of the branch is given as an exact zero.

5. The shape of a branch will be interpolated as a cubic based on the values of x^1 and y^1 and their directions, in relation to distance along the branch. The data is input at the two ends of the branch. If both dx^1/ds and dy^1/ds are given as exact zeros at both ends of the branch, the branch is assumed to be straight. The section can have a discontinuous slope at the branch ends. The beginning point of one branch must coincide with the endpoint of the previous branch. As a result, x^1 and y^1 for the beginning of a branch need to be given only for the first branch of a section.

6. Stress points will be merged into one point if they are separated by a distance less than t/10, where t is a thickness at one of these points.

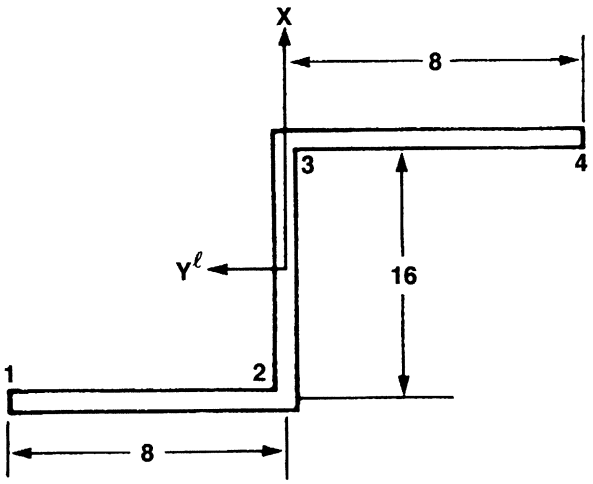
Figure A 4.12-1 shows three sections of a beam. Notice the use of zero thickness branches in the traverse of the I section. The program provides the following data: the location of each stress point in the section, the thickness at that point, the weight associated with each point (for numerical integration of the section stiffness), and the warping function at each section.

Section 1

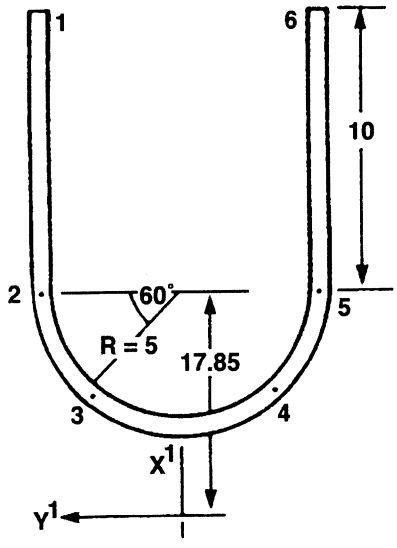


Branch	Divisions	Thickness
1-2	8	1.0
2-3	4	0.0
3-4	8	0.5
4-5	4	0.0
5-6	8	1.0

Section 2



Branch	Divisions	Thickness
1-2	10	1.0
2-3	10	1.0
3-4	10	1.0



Section 3

Branch	Divisions	Thickness
1-2	8	0.5
2-3	8	0.5
3-4	4	0.5
4-5	4	0.5
5-6	4	0.5

Figure A 4.12-1 BEAM SECTION Definition Examples

A 4.13 ERROR ANALYSIS

The user can determine the quality of the analysis by using the ERROR ESTIMATES option.

The ERROR ESTIMATES option can be used to determine the mesh quality (aspect ratio, and skewness), and how they change with deformation. While all of the MARC elements satisfy the patch test, the accuracy of the solution often depends on having regular elements. In analyses where the updated Lagrangian method is used, the mesh often becomes highly distorted during the deformation process. This option tells you when it would be beneficial to perform a rezoning step.

This option can also be used to examine the stress discontinuity in the analysis. This is a measure of the meshes ability to represent the stress gradients in the problem. Large stress discontinuities are an indication that the mesh is not of sufficient quality. This can be resolved by increasing the number of elements or choosing a higher order element.

A 4.14 ADAPTIVE MESH GENERATION

The adaptive mesh generation capability increases the number of elements and nodes to improve the accuracy of the solution. The capability is applicable for both linear elastic analysis and for nonlinear analysis. The capability may be used for lower order elements, 3-node triangles, 4-node quadrilaterals, 4-node tetrahedrals, and 8-node hexahedral elements.

When used in conjunction with the ELASTIC parameter for linear analysis, the mesh will be adapted and the analysis repeated until the error criteria is satisfied. When used in a nonlinear analysis, an increment will be performed. If necessary, this increment will be followed by a mesh adjustment which will be followed by the analysis of the next increment in time. While this may result in some error, as long as the mesh is not overly coarse, it should be adequate.

Number of Elements Created

The adaptive meshing procedure works by dividing an element and internally tying nodes to insure compatibility. Figure A 4.14-1 shows the process for a single quadrilateral element.

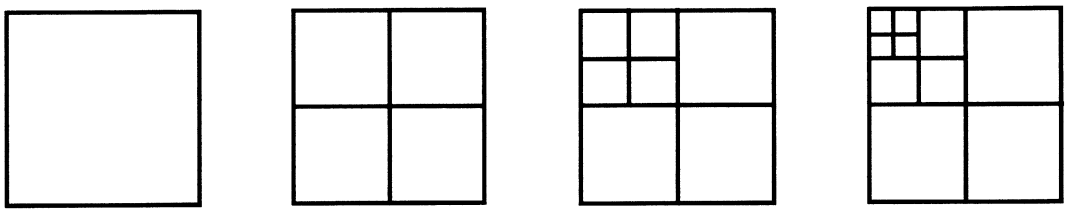


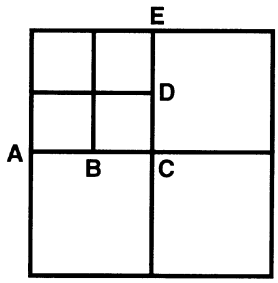
Figure A 4.14-1 Single Quadrilateral Element Process

A similar process occurs for the triangles, tetrahedral, and hexahedral elements. One can observe that for quadrilaterals the number of elements expands by four with each subdivision; similarly, the number of elements increases by eight for hexahedrals. If full refinement occurs, one observes that the number of elements is $2^{(\text{level x } 2)}$ for quadrilaterals and $2^{(\text{level x } 3)}$ for hexahedral elements.

Level	Number of Elements	
	Quadrilaterals	Hexahedrals
0	1	1
1	4	8
2	16	64
3	64	512
4	256	4096

For this reason, it is felt that the number of levels should, in general, be limited to three.

When adaptive meshing occurs, one can observe that discontinuities are created in the mesh as shown below:



To ensure compatibility node B is effectively tied to nodes A and C and node D is effectively tied to nodes C and E. All of this occurs internally and does not conflict with other user defined ties or contact.

Boundary Conditions

When mesh refinement occurs, boundary conditions are automatically adjusted to reflect the change in mesh. The rules listed below are followed:

1. FIXED DISPLACEMENT

   For both 2D and 3D, if both corner nodes on an edge have identical boundary conditions, the new node created on that edge will have the same boundary conditions. For 3D, if all four nodes on a face have identical boundary conditions, the new node created in the center of the face will have the same boundary conditions. Note that identical here means the same in the first degree of freedom, second degree of freedom, etc. independently of one another.

2. POINT LOADS

   The point loads remain unchanged on the original node number.

3. DISTRIBUTED LOADS

   Distributed loads are automatically placed on the new elements. Caution should be used when using user subroutine FORCEM as the element numbers may be changed due to the new mesh process.

4. CONTACT

   The new nodes generated on the exterior of a body are automatically treated as potential contact nodes. The elements in a deformable body are expanded to include the new elements created. After the new mesh is created, the new nodes are checked to determine if they are in contact.

Location of New Nodes

When an element is refined, the default is that the new node on an edge will be midside to the two corner nodes. As an alternative the SURFACE and ATTACH options may be used or user subroutine UCOORD may be used. The SURFACE option may be used to describe the mathematical form of the surface. If the corner nodes of an edge are attached to the surface, the new node is placed upon the actual surface.

This is illustrated in Figure A 4.14-2 and Figure A 4.14-3, where initially a single element is used to represent a circle. The circle is defined with the SURFACE option and the original four nodes are placed on it using the ATTACH option. Notice that the new nodes are placed on the circle.

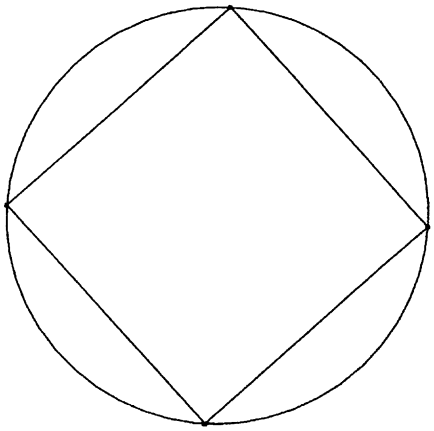
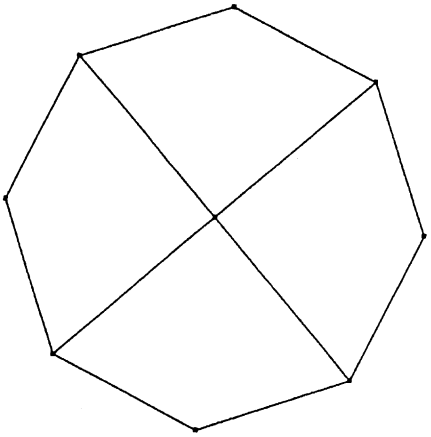
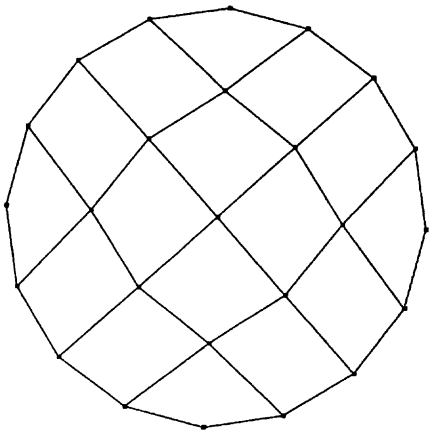


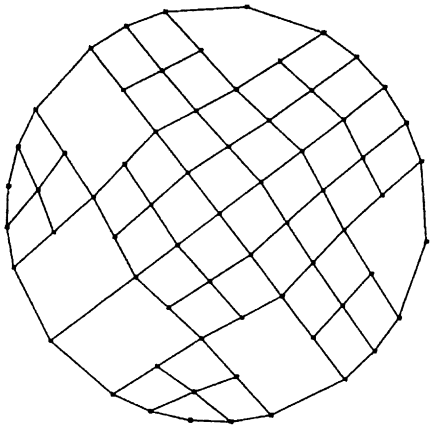
Figure A 4.14-2 Original Mesh and Surface



Level 1 Refinement



Level 2 Refinement



Level 2 and 3 Refinement

Figure A 4.14-3 Levels of Refinement

Error Criteria

The adaptive meshing subdivision occurs when a particular error criteria is violated. Multiple error criteria may be selected using the ADAPTIVE model definition option. These include:

Criterion Type 1 – Mean Strain energy method

The element is refined if the strain energy of the element is greater than the average strain energy in an element times a factor.

$$\text{element strain energy} > \frac{\text{total strain energy}}{\text{number of elements}} * f_1$$

Criterion Type 2 – Zienkiewicz – Zhu Error Criteria

The error norm is defined as either

$$\pi^2 = \frac{\int (\sigma^* - \sigma)^2 dV}{\int \sigma^2 dV + \int (\sigma^* - \sigma)^2 dV} \qquad \gamma^2 = \frac{\int (E^* - E)^2 dV}{\int E^2 dV + \int (E^* - E)^2 dV}$$

The stress error and strain energy errors are

$$X = \int (\sigma^* - \sigma)^2 dV \text{ and } Y = \int (E^* - E)^2 dV$$

where σ^* is the smoothed stress and σ is the calculated stress. Similarly, E is for energy.

An element is subdivided if

$\pi > f_1$ and

$X_{el} > f_2 * X/NUMEL + f_3 * X * f_1/\pi/NUMEL$

or

$\gamma > f_1$ and

$Y_{el} > f_4 * Y/NUMEL + f_5 * Y * f_1/\gamma/NUMEL$

where NUMEL is the number of elements in the mesh. If f_2, f_3, f_4, and f_5 are input as zero, $f_2 = 1.0$.

Criterion Type 4 – Location within a box

An element will be subdivided if it falls within the specified box.

Criterion Type 5 – Contact

An element will be subdivided if one of its nodes in associated with a new contact condition. In the case of a deformable-to-rigid contact, this implies that the node has touched a rigid surface. For deformable-to-deformable contact, the node may be either a tied or retained node. Note that if chattering occurs, there may be an excessive number of elements generated. Use the level option to reduce this problem.

Criterion Type 8 – Gradient

An element will be subdivided if the gradient in the element is greater than the average gradient in the solution. This is the recommended method for heat transfer.

Criterion Type 9 – von Mises Stress

If a stress intensity or singular point exists in the mesh, the mesh will subdivide indefinitely. Use the levels command to eliminate this problem.

This method is based upon either relative or absolute testing on either the von Mises stress, the equivalent strain or the equivalent plastic strain.

An element will be subdivided if either

von Mises > f_1 * maximum von Mises stress or

von Mises stress > f_2 or

equivalent strain > f_3 * maximum equivalent strain or

equivalent strain > f_4 or

equivalent plastic strain > f_5 * maximum equivalent plastic strain or

equivalent plastic strain > f_6

Criterion Type 10 – User defined

User subroutine UADAP may be used to prescribe a user defined error criteria.

Criterion Type 11 – Previously refined mesh

Use the refined mesh from a previous analysis as the starting point to this analysis. The information from the previous adapted analysis is read in.

Chapter 5 STRUCTURAL PROCEDURE LIBRARY

This chapter describes the analysis procedures available in MARC. These procedures range from simple linear elastic analysis to complex nonlinear analysis. A large number of options are available, but you need to consider only those capabilities that are applicable to your physical problem. This chapter provides technical background information as well as usage information about these capabilities.

A 5.1 LINEAR ANALYSIS

Linear analysis is the type of stress analysis performed on linear elastic structures. Because linear analysis is simple and inexpensive to perform and generally gives satisfactory results, it is the most commonly used structural analysis. Nonlinearities due to material, geometry, or boundary conditions are not included in this type of analysis. The behavior of an isotropic, linear, elastic material can be defined by two material constants: Young's modulus E, and Poisson's ratio ν.

The MARC program allows you to perform linear elastic analysis using any element type in the program. Various kinematic constraints and loadings can be prescribed to the structure being analyzed; the problem can include both isotropic and anisotropic elastic materials.

The principle of superposition holds under conditions of linearity. Therefore, several individual solutions can be superimposed (summed) to obtain a total solution to a problem.

Linear analysis does not require storing as many quantities as does nonlinear analysis; therefore, it uses the core memory more sparingly. The ELASTIC option uses the assembled and decomposed stiffness matrices to arrive at repeated solutions for different loads.

NOTE

Linear analysis is always the default analysis type in the MARC program.

Linear analysis in MARC requires only the basic input. Table A 5.1-1 shows a subset of the MARC program options which are often used for linear analysis.

Table A 5.1-1 Basic Input Options

Option Type	Option Name
Parameter	TITLE SIZING ELEMENTS ELASTIC ALL POINTS CENTROID ADAPTIVE FOURIER END
Model Definition	CONNECTIVITY COORDINATES GEOMETRY ISOTROPIC FIXED DISP DIST LOADS POINT LOAD CASE COMBINATION END OPTION

More complex linear analyses require additional data blocks.

1. Parameter option ELASTIC allows solutions for the same structural system with different loadings (multiple loading analysis). When using the ELASTIC option, you must apply total loads, rather than incremental quantities (e.g., total force, total moment, total temperature) in subsequent increments.
2. The RESTART option, used with the ELASTIC and/or CASE COMBINATION option, stores individual load cases on a restart tape. You can also store the decomposed stiffness matrix for later analyses.
3. Model definition option CASE COMBINATION combines the results obtained from different loading cases previously stored on a restart tape.
4. The ADAPTIVE option may be used to improve the accuracy of the analysis.
5. The J-INTEGRAL option allows the study of problems of linear fracture mechanics.
6. The FOURIER option allows the analysis of axisymmetric structures subjected to arbitrary loadings.
7. Model definition option ORTHOTROPIC or ANISOTROPIC activates the anisotropic behavior option. In addition, user subroutines ANELAS, HOOKLW, ANEXP, and ORIENT define the mechanical and thermal anisotropy and the preferred orientations.
8. You can use both the linear SPRING and ELASTIC FOUNDATION options in a linear stress analysis.

Accuracy

It is difficult to predict the accuracy of linear elastic analysis without employing special error estimation techniques. An inaccurate solution usually exhibits itself through one or more of the following phenomena:

- Strong discontinuities in stresses between elements
- Strong variation in stresses within an element
- Stresses that oscillate from element to element

Error Estimates

The ERROR ESTIMATES option may also be used to obtain an indication of the quality of the results. The user can have the program evaluate the geometric quality of the mesh by reporting the aspect ratios and skewness of the elements. In a large deformation updated Lagrange analysis, one can also observe how these change during the analysis, which indicates mesh distortion. When the mesh distortion is large, it is a good idea to do a rezoning step.

The ERROR ESTIMATES option can also be used to evaluate the stress discontinuity between elements. MARC first calculates a nodal stress based upon the extrapolated integration point values. These nodal values are compared between adjacent elements and reported. Large discrepancies indicate an inability of the mesh to capture high stress gradients, in which case you should refine the mesh and rerun the analysis.

The ERROR ESTIMATES option may be used for either linear or nonlinear analysis.

Fourier Analysis

Through Fourier expansion, the MARC program analyzes axisymmetric structures that are subjected to arbitrary loading. The FOURIER option is available only for linear analysis.

During Fourier analysis, a three-dimensional analysis decouples into a series of independent two-dimensional analyses, where the circumferential distribution of displacements and forces are expressed in terms of the Fourier series. Both mechanical and thermal loads may vary arbitrarily in the circumferential direction. You can determine the structure's total response from the sum of the Fourier components.

The Fourier formulation is restricted to axisymmetric structures with linear elastic material behavior and small strains and displacements. Therefore, conditions of linearity are essential and material properties must remain constant in the circumferential direction.

To use Fourier expansion analysis in MARC, the input must include the following information:

- The FOURIER parameter option allocates storage for the series expansion.
- FOURIER model definition blocks for as many series as are needed to describe tractions, thermal loading, and boundary conditions. Number the series sequentially in the order they occur during the FOURIER model definition input. Three ways to describe the series are listed below:
  - Specify coefficients a_0, a_1, b_1,... on the FOURIER model definition cards.
  - Describe $F(\theta)$ (where θ is the angle in degrees about the circumference) in point-wise fashion with an arbitrary number of pairs $[\theta,F(\theta)]$ given on the cards. The MARC program forms the corresponding series coefficients.
  - Generate an arbitrary number of $[\theta, F(\theta)]$ pairs using the user subroutine UFOUR and let the program calculate the series coefficients.

You may obtain the total solution at any position around the circumference by superposing the components already calculated after completion of all increments required by the analysis. The CASE COMBINATION option calculates this total solution by summing the individual harmonics which are stored on the restart tape.

The number of steps or increments needed for analysis depends on the number of harmonics that are chosen. For a full analysis with symmetric and antisymmetric load cases, the total number of increments equals twice the number of harmonics. Table A 5.1-2 shows which Fourier coefficients are used for a given increment.

Table A 5.1-2 Fourier Coefficients – Increment Number

LOAD TERMS			
INC.	**1st DOF,Z**	**2nd DOF,R**	**3rd DOF,θ**
0	a_0	a_0	0
1	0	0	a_0
2	a_1	a_1	b_1
3	b_1	b_1	a1
◊	.	.	.
◊	.	.	.
◊	.	.	.
2n	a_n	a_n	b_n
2n+1	b_n	b_n	a_n

The magnitude of concentrated forces should correspond to the value of the ring load integrated around the circumference. Therefore, if the Fourier coefficients for a varying ring load $p(\theta)$ are found from the $[\theta, p(\theta)]$ distribution, where $p(\theta)$ has the units of force per unit length, the force magnitude given in the POINT LOAD block should equal the circumference of the loaded ring. If $p(\theta)$ is in units of force per radian, the POINT LOAD magnitude should be 2π.

The Fourier series can be found for varying pressure loading from $[\theta, p(\theta)]$ input with p expressed in force per unit area. The MARC program calculates the equivalent nodal forces and integrates them around the circumference. The distributed load magnitude in the DIST LOADS block should be 1.0.

Table A 5.1-3 shows the elements in the program that can be used for Fourier analysis.

Table A 5.1-3 Elements Used for Fourier Analysis

Element Type	Description
62	8-node
73	8-node with reduced integration
63	8-node for incompressible behavior
74	8-node for incompressible behavior with reduced integration
90	3-node shell

Technical Background

The general form of the Fourier series expansion of the function F(θ) is shown in the equation below.

$$F(\theta) = a_o + \sum_{n=1}^{\infty} (a_n \cos n\theta + b_n \sin n\theta) \qquad \text{(A 5.1-1)}$$

This expression expands the displacement function in terms of sine and cosine terms. A symmetric and an antisymmetric problem are formulated for each value of n.

The displacements for the symmetric case, expressed in terms of their nodal values, are

$$\begin{aligned} u^n &= [N_1, N_2, \ldots] \cos n\theta \{u^n\}^e \\ v^n &= [N_1, N_2, \ldots] \cos n\theta \{v^n\}^e \\ w^n &= [N_1, N_2, \ldots] \sin n\theta \{w^n\}^e \end{aligned} \qquad \text{(A 5.1-2)}$$

Nodal forces are

$$\begin{aligned} Z &= Z_0 + \sum_{1}^{n} Z^n \cos n\theta \\ \underline{R} &= R_0 + \sum_{1}^{n} R^n \cos n\theta \\ T &= T_0 + \sum_{1}^{n} T^n \sin n\theta \end{aligned} \qquad \text{(A 5.1-3)}$$

The value n=0 is a special case in Fourier analysis. If only the symmetric expansion terms are used, the formulation defaults to the fully axisymmetric two-dimensional analysis. The antisymmetric case for n=0 yields a solution for the variable θ that corresponds to loading in the tangential direction. Analyze axisymmetric solids under pure torsion in this way.

Modal Shapes and Buckling Load Estimations During a Fourier Analysis

During a Fourier analysis, the MARC program can be asked to estimate both the modal shapes and buckling loads for each harmonic in the analysis. In either case, the program performs a Fourier analysis first and then estimates the modal shapes/buckling load at prescribed harmonic numbers. In addition to the input data required for a Fourier analysis (FOURIER parameter option and FOURIER model definition block), the following must also be added: DYNAMIC parameter option and MODAL INCREMENT model definition block; BUCKLE parameter option and BUCKLE INCREMENT model definition block, for Fourier modal shape and Fourier buckling load estimations, respectively.

In the Fourier modal analysis, the mass matrix in the eigenvalue equation is a constant matrix. The stiffness matrix in the eigenvalue equation is the one associated with a prescribed harmonic of the Fourier analysis. The expression of the eigenvalue equation is:

$$[K^m]\,\phi - \omega^2\,[M^o]\,\phi = 0 \qquad \text{(A 5.1-4)}$$

where $[K^m]$ is the stiffness matrix associated with the m^{th} harmonic of the Fourier analysis and $[M^0]$ is a constant matrix. Multiple modes for each harmonic may be extracted.

Similarly, in a Fourier buckling analysis, the stiffness matrices in the eigenvalue equation are (respectively); the linear elastic stiffness matrix and the geometric stiffness matrix associated with the prescribed harmonic of the Fourier analysis.

The eigenvalue equation is expressed

$$[K^m]\,\phi - \lambda\,[K_G^m]\,\phi = 0 \qquad \text{(A 5.1-5)}$$

where $[K^m]$ is the linear elastic stiffness matrix and $[K_G^m]$ is the geometric stiffness matrix, associated with the m^{th} harmonic of the Fourier analysis. The stresses used in the calculation of the geometric stiffness matrix are those associated with the symmetric load case, m = 0. Multiple buckling load estimations for each harmonic is also available.

A 5.2 NONLINEAR ANALYSIS

The finite element method can be used for nonlinear, as well as linear, problems. Early development of nonlinear finite element technology was mostly influenced by the nuclear and aerospace industries. In the nuclear industry, nonlinearities are mainly due to the nonlinear, high-temperature behavior of materials. Nonlinearities in the aerospace industry are mainly geometric in nature and range from simple linear buckling to complicated post-bifurcation behavior. In recent years, nonlinear finite element techniques have been applied to the metal forming industry and to manufacturing processes.

A problem is nonlinear if the force-displacement relationship depends on the current state (i.e., current displacement, force, and stress-strain relations). Let u be a generalized displacement vector, P a generalized force vector, and K the stiffness matrix. The expression of the force-displacement relation for a nonlinear problem is

$$P = K(P, u)\,u \qquad \text{(A 5.2-1)}$$

Linear problems form a subset of nonlinear problems. For example, in classical linear elastostatics, this relation may be written in the form

$$P = K \bullet u \qquad \text{(A 5.2-2)}$$

where the stiffness matrix K is independent of both u and P. If the matrix K depends on other state variables that do not depend on displacement or loads (such as temperature, radiation, moisture content, etc.), the problem is still linear.

Similarly, if the mass matrix is a constant matrix, the following undamped dynamic problem is also linear:

$$P = M \bullet \ddot{u} + K \bullet u \qquad \text{(A 5.2-3)}$$

There are three sources of nonlinearity: material, geometric, and nonlinear boundary conditions. Material nonlinearity results from the nonlinear relationship between stresses and strains. Models for these relationships cannot be derived along purely mathematical lines, but rather are based on experimental data. Considerable progress has been made in attempts to derive the macroscopic behavior of materials from microscopic backgrounds, but, up to now, commonly accepted constitutive laws are phenomenological. Nonlinearities due to plasticity, viscoplasticity, and creep (see Figure A 5.2-1) are of major practical importance. Nonlinear elastic behavior is also receiving increased attention.

Geometric nonlinearity results from the nonlinear relationship between strains and displacements on the one hand and the nonlinear relation between stresses and forces on the other hand. If the stress measure is conjugate to the strain measure, both sources of nonlinearity have the same form. This type of nonlinearity is mathematically well defined, but often difficult to treat numerically. Two important types of geometric nonlinearity occur: a) the analysis of buckling and snap-through problems (see Figure A 5.2-2 and Figure A 5.2-3) and b) large strain problems such as analysis of rubber components or metal forming processes. In large strain problems, the mathematical separation into geometric and material nonlinearity is nonunique.

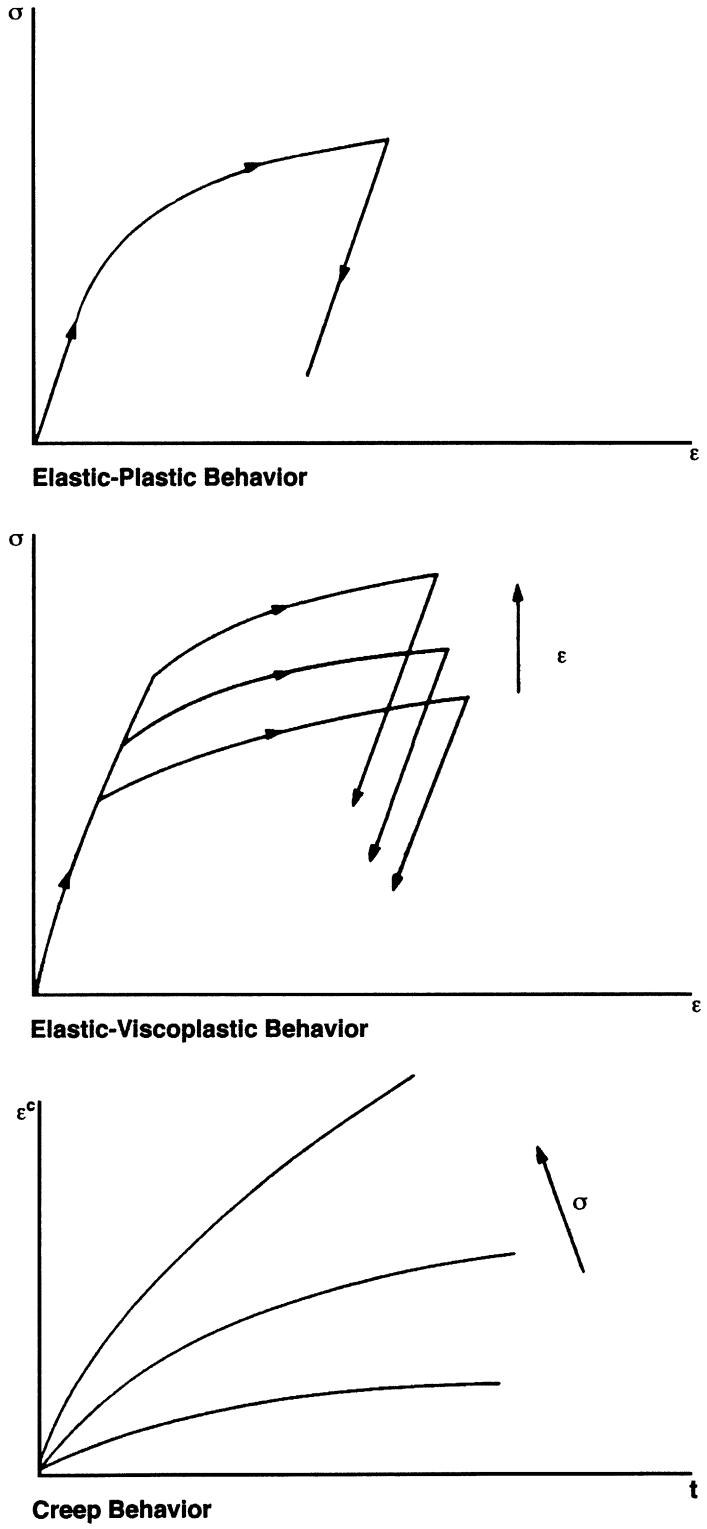


Figure A 5.2-1 Material Nonlinearity

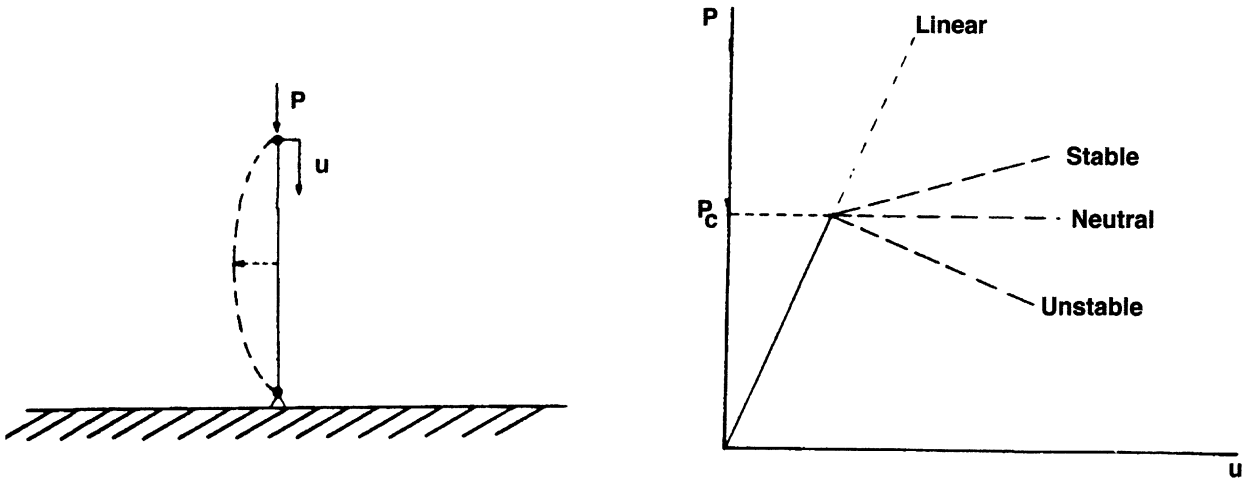


Figure A 5.2-2 Buckling

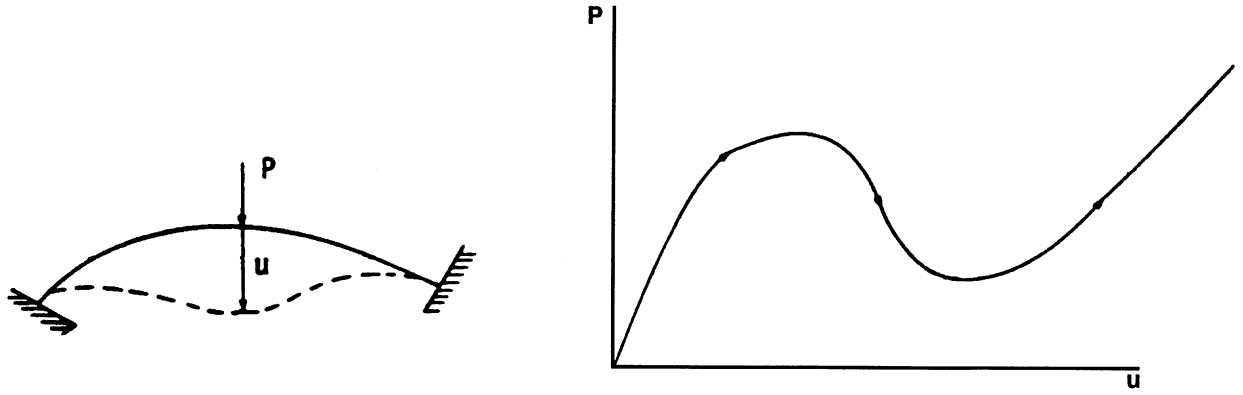


Figure A 5.2-3 Snap-Through

Boundary conditions and/or loads may also cause nonlinearity. Contact and friction problems lead to nonlinear boundary conditions. This type of nonlinearity plays an important role in metal forming problems (see Figure A 5.2-4). Loads on a structure cause nonlinearity if they vary with the displacements of the structure. These loads can be conservative, as in the case of a centrifugal force field (see Figure A 5.2-5); they can also be nonconservative, as in the case of a follower force on a cantilever beam (see Figure A 5.2-6). Also, such a following force can be locally nonconservative, but represent a conservative loading system when integrated over the structure. A pressurized cylinder (see Figure A 5.2-7) is an example of this.

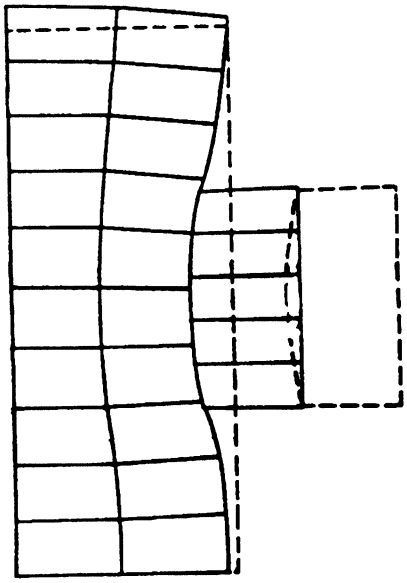


Figure A 5.2-4 Contact and Friction Problem

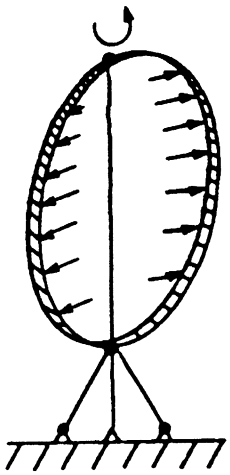


Figure A 5.2-5 Centrifugal Load Problem (Conservative)

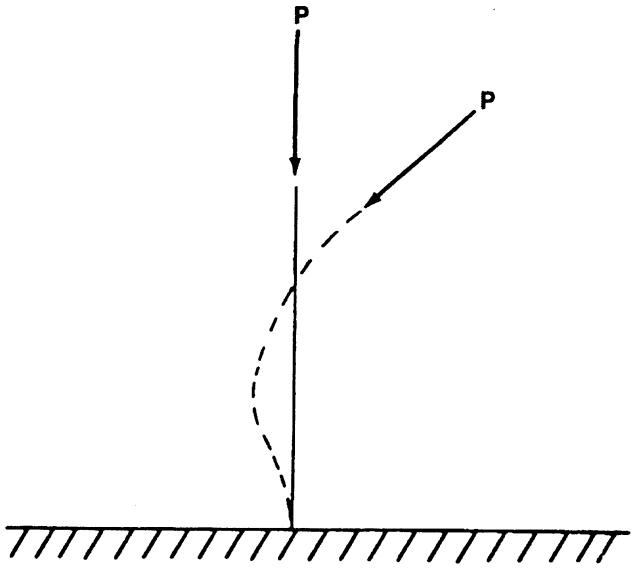


Figure A 5.2-6 Follower Force Problem (Non-Conservative)

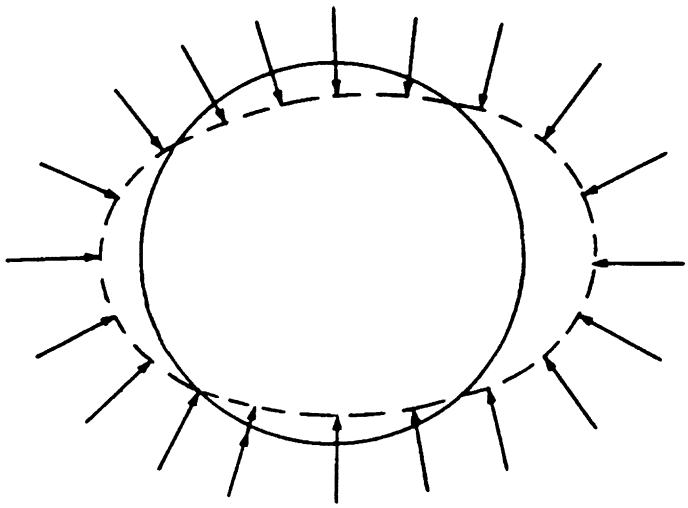


Figure A 5.2-7 Pressurized Cylinder (Globally Conservative)

Considerations for Nonlinear Analysis

Nonlinear analysis is usually more complex and expensive than linear analysis. Also, a nonlinear problem can never be formulated as a set of linear equations. In general, the solutions of nonlinear problems always require incremental solution schemes and sometimes require iterations (or recycles) within each load/time increment to ensure that equilibrium is satisfied at the end of each step. Superposition cannot be applied in nonlinear problems.

The four iterative procedures available in MARC are: Newton-Raphson, Modified Newton-Raphson, Newton-Raphson with strain correction modification, and a secant procedure. If the R-P flow contribution model is chosen, a direction substitution is used. See the appendix for a discussion of these iterative procedures.

A nonlinear problem does not always have a unique solution. Sometimes a nonlinear problem does not have any solution, although the problem may seem to be defined correctly.

Nonlinear analysis requires good judgment and uses considerable computing time. Nonlinear analyses often require several runs. The first run should extract the maximum information with the minimum amount of computing time. Some design considerations for a preliminary analysis are:

- Minimize degrees of freedom whenever possible.
- Halve the number of load increments by doubling the size of each load increment.
- Impose a coarse tolerance on convergence to reduce the number of iterations. A coarse run determines the area of most rapid change where additional load increments may be required. Plan the increment size in the final run by the following rule of thumb: there should be as many load increments as required to fit the nonlinear results by the same number of straight lines.

MARC solves nonlinear static problems according to one of the following two methods: tangent modulus or initial strain. Examples of the tangent modulus method are elastic-plastic analysis, nonlinear springs, nonlinear foundations, large displacement analysis and gaps. This method requires at least the following three controls:

- A tolerance on convergence
- A limit to the maximum allowable number of recycles
- Specification of a minimum number of recycles

An example of the initial strain method is creep or viscoelastic analysis. Creep analysis requires the following tolerance controls:

- Maximum relative creep strain increment control
- Maximum relative stress change control
- A limit to the maximum allowable number of recycles

To input control tolerances, use the model definition option CONTROL. These values may be reset upon restart or through the CONTROL history definition option. See the Appendix for further discussion on tolerance controls.

Behavior of Nonlinear Materials

Nonlinear behavior can be time- (rate-) independent, or time- (rate-) dependent. For example, plasticity is time-independent and creep is time-dependent. Both viscoelastic and viscoplastic materials are also time-dependent. Nonlinear constitutive relations must be modeled correctly to analyze nonlinear material problems. A comprehensive discussion of constitutive relations is given in Chapter 6.

Scaling the Elastic Solution

The parameter option SCALE causes scaling of the linear-elastic solution to reach the yield stress in the highest stressed element. Scaling takes place for small displacement elastic-plastic analysis, where element properties do not depend on temperature. The SCALE option causes all aspects of the initial solution to be scaled, including displacements, strains, stresses, temperature changes, and loads. Subsequent incrementation is then based on the scaled solution.

Selecting Load Incrementations

Several load incrementation options are available in the program to input mechanical and thermal load increments (see Table A 5.2-1, below).

Table A 5.2-1 Load Incrementation Options

Load Type	Options
Mechanical (Force/Displacement	PROPORTIONAL INCREMENT POINT LOAD DIST LOADS AUTO LOAD AUTO INCREMENT DISP CHANGE AUTO TIME
Thermal	THERMAL LOADS CHANGE STATE AUTO THERM

Restarting the Analysis

The model definition option RESTART creates a restart file for the current analysis which may be used in subsequent analyses. It also may be used to read in a previously generated file to continue the analysis. The RESTART option is very important for any multi-increment analysis because it allows you to continue the analysis at a later time. The default situation writes the restart information to unit 8 and reads a previously generated tape from unit 9. For post processing, option RESTART can be used to plot or combine load cases (see CASE COMBINATION). Upon restart, you may use the model definition REAUTO option to redefine parameters associated with an automatic load sequence.

To save storage space, it is not necessary to store each increment of analysis. The frequency can be set using the RESTART option, and subsequently modified using the RESTART INCREMENT option. It is also possible to store only the last converged solution, by using the RESTART LAST option. This should not be used with ELASTIC analysis because the stiffness matrix will not be stored.

Selecting Load Increment Size

MARC uses the incremental method in plasticity analysis. Load incrementation can be prescribed automatically or manually through a variety of options. The program controls iteration. The two extreme approaches to solving nonlinear equations are to solve a large number of small steps with few iterations or to solve one large step with many iterative cycles. Generally, it is best to employ a combination of both.

Residual Load Correction

The residual load is applied as a correcting force to ensure that equilibrium is maintained and, hence, that an accurate solution is obtained for nonlinear problems.

The residual load correction enforces global equilibrium at the start of each new increment. This prevents the accumulation of out-of-equilibrium forces from increment to increment and makes the solution less sensitive to the step size. Figure A 5.2-8 shows how stiffness is based on the state at the start of a step. The variables are defined below for increments i = 1,2,3:

- F_i applied forces for i = 1,2,3
- u_i calculated displacements for i = 1,2,3
- R_i residual loads for i = 1,2,3

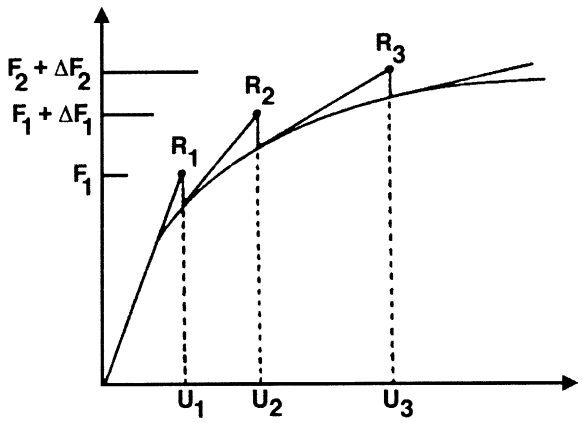


Figure A 5.2-8 Stiffness Based on State at Start of Step

The residual load correction is the difference between the internal forces and the externally applied loads. The residual load correction is expressed as

$$R = P - \int \beta^T \sigma dV \qquad \text{(A 5.2-4)}$$

where β is the differential operator which transforms displacements to strains, σ are the current generalized stresses, P is the total applied load vector, and R is the residual load correction.

In order to evaluate the residual load correction accurately, evaluate the integral by summing the contributions from all integration points. The residual load correction feature requires that stresses be stored at all the integration points. Data storage at all integration points is the default in MARC, but may be overridden in linear analysis by use of the CENTROID parameter option.

NOTE

The ALL POINTS parameter should be used for all nonlinear analysis. In the K3 version and all subsequent versions, this is the default.

Computational Procedures for Elastic-Plastic Analysis

MARC plasticity algorithms are unconditionally stable and accurate for moderate strain increments. However, they are somewhat less accurate for strong incremental stress changes and have poor convergence for strain increments greater than ten times the elastic strain. These considerations should guide the selection of appropriate load increment size.

The MARC program calculates the stress-strain relation at the mid-increment for each integration point based on an incremental strain prediction. For the first cycle of each step, this prediction is based on the strain increment of the preceding increment. The mean normal method or the radial return method establish the elastic-plastic response, calculating a secant stiffness matrix at each increment. If the residuals or displacements at the end of the increment satisfy the chosen tolerance, no recycling takes place. During recycles, the strains recovered from the previous iteration are used as estimated strains for the stiffness evaluation. The recycling procedure prevents poor strain estimates from adversely affecting the solution.

The program uses the mean normal method, by default, for all elements except those modeling plane stress. As an alternative, you can select the radial return method using the CONTROL option. The tangent modulus method with radial return is used for plane stress elements. In either case, the stress state remains on the yield surface.

MARC makes no distinction in elastic-plastic constitutive calculations between the assembly phase and recovery phase in calculating constitutive equations, except for updating the stress and strain quantities according to the process described below.

Given a strain increment vector,

1. Assume fully elastic response and find the corresponding stress increment using the equation

$$\Delta\sigma^{el} = C^{el}\Delta\varepsilon \qquad \text{(A 5.2-5)}$$

2. Check the stress state at the start of step 1 for possible violation of yield criterion. The yield criterion may not be exactly satisfied due to temperature effects, numerical integration of elastic-plastic relationships, or accumulated numerical inaccuracy. $F(\sigma) = 0$ indicates that the stress state is exactly on the yield surface.

   If $F(\sigma) > 0$, scale σ by a factor λ as follows:

$$F(\lambda\sigma) = 0 \qquad 0 < \lambda < 1 \qquad \text{(A 5.2-6)}$$

   For nonzero λ, the quantity

$$(1-\lambda)\,\sigma \tag{A 5.2-7}$$

is added to the elastic stress increment

$$\Delta\sigma^{el} = \Delta\sigma^{el} + (1-\lambda)\,\sigma \tag{A 5.2-8}$$

Now σ is guaranteed to be either inside or on the yield surface at the start of the increment.

3. Check the stress state at the end of the increment. If

$$F[\sigma + \Delta\sigma^{el}] < 0, \tag{A 5.2-9}$$

this is a purely elastic increment. If $F(\sigma) < 0$

$$F[\sigma + \Delta\sigma^{el}] > 0 \tag{A 5.2-10}$$

determine what fraction of the increment is elastic (designated m in Step 6) and what fraction is elastic-plastic (1-m).

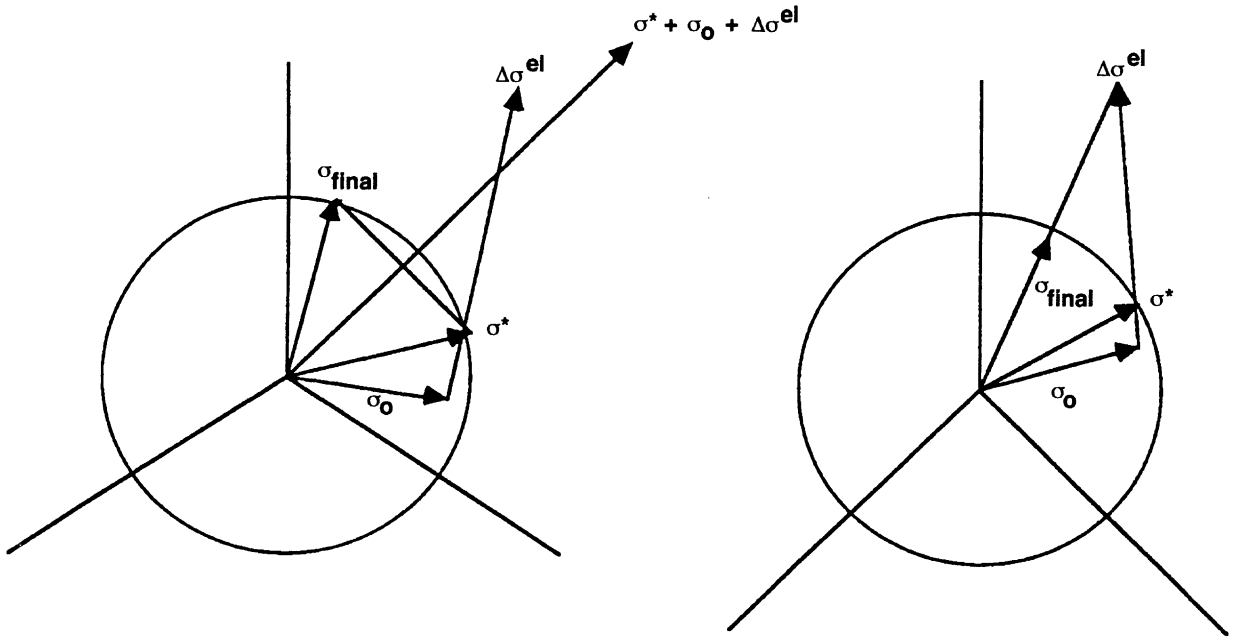
4. To find the increment of plastic strain and the constitutive matrix, determine the normal to the yield surface at the current stress state. For the von Mises yield criterion

$$F = \sqrt{3J_2} - \bar{\sigma} = 0 \tag{A 5.2-11}$$

the normal vector is

$$\left[\frac{\partial F}{\partial \sigma_{ij}}\right] = \frac{3}{2\bar{\sigma}}\left[\sigma_x', \sigma_y', \sigma_z', 2\tau_{xy}, 2\tau_{yz}, 2\tau_{zx}\right] \tag{A 5.2-12}$$

Use the mean stiffness, shown in Figure 5.2-9a, to determine the normal, or you can use the stiffness at the end of the increment, shown in Figure 5.2-9b, to determine the normal. This method guarantees that the final stress state satisfies the yield condition.



(a) Mean Normal Method

(b) Radial Return Method

Figure A 5.2-9 Mean Stiffness

5. Once $\partial F/\partial\sigma$ is found, you can determine the equivalent plastic strain increment from

$$\Delta\bar{\varepsilon}^{p} = \frac{\left[\frac{\partial F}{\partial\sigma}\right]^{T} C^{el}}{H + \left[\frac{\partial F}{\partial\sigma}\right]^{T} C^{el}\left[\frac{\partial F}{\partial\sigma}\right]}\Delta\varepsilon \qquad \text{(A 5.2-13)}$$

The work hardening slope H depends on the total plastic strain $\bar{\varepsilon}^{p}$, therefore, the equation for $\Delta\bar{\varepsilon}^{p}$ is solved iteratively in a maximum of five steps.

6. The elastic-plastic constitutive relationship can be formulated as

$$L^{e-p} = mC^{el} - (1-m)\frac{C^{el}\left[\frac{\partial F}{\partial\sigma}\right]^{T}\left[\frac{\partial F}{\partial\sigma}\right]C^{el}}{H + \left[\frac{\partial F}{\partial\sigma}\right]^{T} C^{el}\left[\frac{\partial F}{\partial\sigma}\right]} \qquad \text{(A 5.2-14)}$$

where m was calculated in Step 3.

7. The components of the increment of plastic strain are found by

$$\Delta\varepsilon^{p} = \Delta\bar{\varepsilon}^{p}\frac{\partial F}{\partial\sigma} \qquad \text{(A 5.2-15)}$$

8. The correct increment of stress is calculated as

$$\Delta\sigma = L^{e-p}\Delta\varepsilon \qquad \text{(A 5.2-16)}$$

9. During recovery update, stresses and strains are represented as

$$\sigma_0 + \Delta\sigma, \varepsilon_0 + \Delta\varepsilon \qquad \text{(A 5.2-17)}$$

10. During assembly, use $L^{e\text{-}p}$ for evaluation of the stiffness matrix

$$K = \int_v \beta^T L^{e-p} \beta dV \qquad \text{(A 5.2-18)}$$

11. For those integration points or layers that have gone plastic, an additional line will be printed giving the plastic strains, unless suppressed using the PRINT ELEM option.

CREEP

Creep is a time-dependent inelastic behavior that can occur at any stress level, either below or above the yield stress of a material. Creep is an important factor at elevated temperatures.

The MARC program offers the following options to be used in conjunction with creep analysis:

- Creep data can be entered directly through card input or user subroutine.
- Creep behavior can be either isotropic/or anisotropic.
- The Oak Ridge National Laboratory (ORNL) rules on creep can be activated.
- An automatic time stepping scheme maximizes the time step size in the analysis.
- Eigenvalues can be extracted for the estimation of creep buckling time.
- Use parameter option CREEP to activate the creep analysis option in MARC.

Input the creep time period and control tolerance information through the history definition option AUTO CREEP. This option can be used repeatedly to define a new creep time period and new tolerances.

Creep analysis is often carried out in several runs using the RESTART option. Save restart tapes for continued analysis. The REAUTO option allows you to reset the parameters defined in the AUTO CREEP option upon restart.

NOTE

It is always useful to check the accuracy of the creep law input before performing a creep analysis. You can easily carry out this check by running a simple one-element model creep analysis. Use the MARC truss element (Element Type 9) to simulate a uniaxial test. This simulated uniaxial test can either be a load control test or a displacement control test, depending on the prescribed boundary conditions. In a simulated creep test, the strain history predicted by MARC must be identical to the experimental data.

Adaptive Time Control

The AUTO CREEP option takes advantage of the diffusive characteristics of most creep solutions. Specifically, this option controls the transient creep analysis. The user specifies a period of creep time and a suggested time increment. The program automatically selects the largest possible time increment that is consistent with the tolerance set on stress and strain increments (see Section in 5.2 on creep control tolerances).

The algorithm is: for a given time step Δt, a solution is obtained. The program then finds the largest values of stress change per stress, $\Delta\sigma/\sigma$, and creep strain change per elastic strain, $\Delta\varepsilon^{cr}/\varepsilon^{el}$. It compares these values to the tolerance values, T_s (stress change tolerance) and T_e (strain change tolerance), for this period.

The value p is calculated as the larger of

$$\Delta\sigma/\sigma/T_\sigma$$

and (A 5.2-19)

$$\Delta\varepsilon^{cr}/\varepsilon^{el}/T_\varepsilon$$

If $p > 1$, the program resets the time step as

$$\Delta t_{new} = \Delta t_{old} \bullet 0.8/p \qquad \text{(A 5.2-20)}$$

The time increment is repeated until convergence is obtained or the maximum recycles control is exceeded. In the latter case, the run is ended.

Clearly, the first repeat should satisfy tolerances. If it does not, the possible causes are:

- excessive residual load correction
- strong additional nonlinearities such as creep buckling-creep collapse
- incorrect coding in user subroutine CRPLAW, VSWELL or UVSCPL.

Appropriate action should be taken before the solution is restarted.

If $p < 1$, the solution is stepped forward to $t + \Delta t$ and the next step is begun. The time step used in the next increment is chosen as

$$\Delta t_{new} = \Delta t_{old} \quad \text{if} \quad 0.8 \le p < 1 \qquad \text{(A 5.2-21)}$$

$$\Delta t_{new} = 1.25 \bullet \Delta t_{old} \quad \text{if} \quad 0.65 \le p < 0.8 \qquad \text{(A 5.2-22)}$$

$$\Delta t_{new} = 1.5 \bullet \Delta t_{old} \quad \text{if} \quad p < 0.65 \qquad \text{(A 5.2-23)}$$

Since the time increment is adjusted to satisfy the tolerances, it is impossible to predetermine the total number of time increments for a given total creep time.

Creep Control Tolerances

The MARC program performs a creep analysis under constant load or displacement conditions on the basis of a set of tolerances and controls you provide.These are as follows:

1. *Stress Change Tolerance* – This tolerance controls the allowable stress change per time step during the creep solution, as a fraction of the total stress at a point. Stress change tolerance governs the accuracy of the transient specify creep response. If you need accurate tracking of the transient, specify a tight tolerance of 1 percent or 2 percent stress change per time step. If you need only the steady-state solution, supply a relatively loose tolerance of 10-20 percent. It is also possible to check the absolute rather than the relative stress.

2. *Creep Strain Increment per Elastic Strain* – The MARC program uses either explicit or implicit integration of the creep rate equation. When the explicit procedure is used, the creep strain increment per elastic strain is used to control stability. In almost all cases the default of 50 percent represents the stability limit, so that you need not provide any entry for this value. It is also possible to check the absolute rather than the relative strain.

3. *Maximum Number of Recycles for Satisfaction of Tolerances* – During AUTO CREEP, the MARC program chooses its own time step. In some cases the program recycles to choose a time step that satisfies tolerances, but recycling rarely occurs more than once per step. Excessive recycling may be caused by physical problems such as creep buckling, poor coding of user subroutine CRPLAW, VSWELL or UVSCPL or excessive residual load correction that may occur when the creep solution begins from a state that is not in equilibrium.

   The maximum number of recycles allows you to avoid wasting machine time under such circumstances. If there is no satisfaction of tolerances after the attempts at stepping forward, the program will stop. The default of 5 recycles is conservative in most cases.

4. *Low Stress Cut-Off* – Low stress cut-off avoids excessive iteration and small time steps caused by tolerance checks that are based on small (round off) stress states. A simple example is a beam in pure bending. The stress on the neutral axis is a very small roundoff-number, so that automatic time stepping scheme should not base time step choices on tolerance satisfaction at such points. The default of 5 percent of the maximum stress in the structure is satisfactory for most cases.

5. *Choice of Element for Tolerance Checking* – The default option for creep tolerance checking checks all integration points in all elements. You may wish to check tolerances in only one element or in up to 14 elements of your choice. The most highly stressed element is usually chosen.

When you enter the tolerances and controls, the following conventions apply:

- All stress and strain measures in tolerance checks are second invariants of the deviatoric state (i.e., equivalent von Mises uniaxial values).
- You may reset all tolerances and controls upon the completion of one AUTO CREEP sequence.

Background Information

Creep behavior is based on a von Mises creep potential with isotropic behavior described by the equivalent creep law:

$$\dot{\bar{\varepsilon}}^{cr} = f(\bar{\sigma}, \bar{\varepsilon}^{cr}, T, t) \qquad \text{(A 5.2-24)}$$

The material behavior is therefore described by:

$$\Delta\varepsilon^{cr} = \dot{\bar{\varepsilon}}^{cr} \frac{\partial\bar{\sigma}}{\partial\sigma} \Delta t \qquad \text{(A 5.2-25)}$$

where $\partial\bar{\sigma}/\partial\sigma$ is the outward normal to the current von Mises stress surface and $\dot{\bar{\varepsilon}}^{cr}$ is the equivalent creep strain rate.

There are two numerical procedures used in implementing creep behavior. The default is an explicit procedure in which the above relationship is implemented in the program by an initial strain technique. In other words, a pseudo-load vector due to the creep strain increment is added to the right-hand side of the stiffness equation.

$$K\Delta u = \Delta P + \int_v \beta^T D \Delta\varepsilon^{cr} dV \qquad \text{(A 5.2-26)}$$

where K is the stiffness matrix, and Δu and ΔP are incremental displacement and incremental nodal force vectors, respectively. The integral

$$\int_v \beta^T D \Delta\varepsilon^{cr} dV \qquad \text{(A 5.2-27)}$$

is the pseudo-load vector due to the creep strain increment in which β is the strain displacement relation and D is the stress-strain relation.

As an alternative, an implicit creep procedure may be requested with the CREEP parameter option. In this case, the effect inelastic strain rate will have an influence on the stiffness matrix. Using this technique, significantly larger steps in strain space may be used.

Creep Buckling

The MARC program also predicts the creep time to elastic buckling due to stress redistribution under given load or repeated cyclic load.

The buckling option solves the following equation for the eigenvalue

$$(K + \lambda K_G)\,\Phi = 0 \qquad \text{(A 5.2-28)}$$

The geometric stiffness matrix K_G is a function of the increments of stress and displacement. These increments are calculated during the last creep time step increment. To determine the creep time to buckle, perform a BUCKLE step after a converged creep increment. Note that the incremental time must be scaled by the calculated eigenvalue.

AUTO-THERM-CREEP (Automatic Thermally Loaded Elastic-Creep/Elastic-Plastic-Creep Stress Analysis)

This option is intended to allow automatic, thermally loaded elastic-creep/elastic-plastic-creep stress analysis, based on a set of temperatures defined throughout the mesh as a function of time. The temperatures and transient times are presented to the program through the CHANGE STATE option, using input option 3 (POST FILE), and the program then creates its own set of temperature steps (increments) based on a temperature change tolerance provided on this option. The times at all temperature steps are calculated by the program for creep analyses.

At each temperature step (increment), an elastic/elastic-plastic analysis is carried out first to establish stress level in the structure. A creep analysis is performed next on the structure for the time period between current and previous temperature steps (increments). Both the elastic/elastic-plastic stress and the creep analyses are repeated until the total creep time provided on this option is reached. Convergence controls are provided on the CONTROL option for elastic-plastic analysis and on the AUTO THERM CREEP option for creep analysis. The analysis can be restarted at temperature steps (increments) or at creep steps (sub-increments). The results can be saved on a post file (POST option) for post-processing.

If no DIST LOADS, POINT LOAD or PROPORTIONAL INC option appears with the AUTO THERM CREEP set, all mechanical loads and kinematic boundary conditions are held constant during the AUTO THERM CREEP. However, DIST LOADS, POINT LOAD, PROPORTIONAL INC, or DISP CHANGE may be included in the set – the mechanical loads and kinematic boundary conditions which are then defined are assumed to change in proportion to the time scale of the temperature history defined by the CHANGE STATE option and will be applied accordingly. This is based on the fact that the increments of load and displacement correspond to the end of the transient time of the AUTO THERM CREEP input.

Viscoelasticity

In a certain class of problems, structural materials exhibit viscoelastic behavior. Two examples of these problems are quenching of glass structures and time-dependent deformation of polymeric materials. The viscoelastic material retains linearity between load and deformation; however, this linear relationship depends on time. Consequently, the current state of deformation must be determined from the entire history of loading. Different models consisting of elastic elements (spring) and viscous elements (dashpot) can be used to simulate the viscoelastic material behavior described in Section 6.6. A special class of temperature dependence known as the thermo-rheologically simple behavior (TRS) is also applicable to a variety of thermal viscoelastic problems. Both the equation of state and the hereditary integral approaches can be used for viscoelastic analysis.

In the MARC program, two options are available for viscoelastic analysis. The first option uses the equation of state approach and represents a Kelvin model. The second option is based on the hereditary integral approach and allows the selection of a generalized Maxwell model. The thermo-rheologically simple behavior is also available in the second option for thermal viscoelastic analysis. Section 6.6.4 discusses these models in detail. Automatic time stepping schemes AUTO CREEP and AUTO TIME can be used in a viscoelastic analysis for first and second options, respectively.

The first option for viscoelastic analysis uses the Kelvin model. To activate the generalized Kelvin model in MARC, use the VISCO ELAS or CREEP parameter option. To input the matrices [A] and [B] for the Kelvin strain rate computations, use the user subroutine CRPVIS. To input creep time period and the tolerance control for the maximum strain in an increment, use the history definition option AUTO CREEP.

Viscoplasticity

There are two procedures in MARC for viscoplastic analysis.

Explicit Method

The elasto-viscoplasticity model in MARC is a modified creep model to which a plastic element is added. The plastic element is inactive when the stress is less than the yield stress of the material. You can use the elasto-viscoplasticity model to solve time-dependent plasticity and creep as well as plasticity problems with a non-associated flow law.

The creep option in MARC has been modified to enable solving problems with viscoplasticity. The method is modified to allow solving elastic-plastic problems with non-associated flow rules which result in nonsymmetric stress-strain relations if the tangent modulus method is used.

The requirements for solving the viscoplastic problem are:

- CREEP parameter option and creep controls
- Load incrementation immediately followed by a series of creep increments specified by AUTO CREEP
- Use of user subroutine CRPLAW and/or user subroutine NASSOC

The following load incrementation procedure will help you solve a viscoplastic problem:

1. Apply an elastic load increment that will exceed the steady-state yield stress.
2. Relieve the high yield stresses by turning on the AUTO CREEP option.

You may repeat steps 1 and 2 as many times as necessary to achieve the required load history.

NOTE

The size of the load increments are not altered during the AUTO CREEP process so that further load increments can be effected by using the PROPORTIONAL INC feature.

The viscoplastic approach converts an iterative elastic-plastic method to one where a fraction of the initial force vector is applied at each increment with the time step controls. The success of the method depends on the proper use of the automatic creep time step controls. This means that it is necessary to select an initial time step that will satisfy the tolerances placed on the allowable stress change.

$$\text{The initial time step dt} = \frac{\text{allowable stress change X 0.7}}{\text{Maximum viscoplastic strain rate X Young's modulus}}$$

The allowable stress change is specified in the creep controls. The most highly stressed element usually yields the maximum strain rate. It is also important to select a total time that will give sufficient number of increments to work off the effects of the initial force vector. A total time of 30 times the estimated dt is usually sufficient.

The MARC program does not distinguish between viscoplastic and creep strains. A user subroutine NASSOC allows you to specify a non-associated flow rule for use with the equivalent creep strains (viscoplastic) that are calculated by subroutine CRPLAW. A flag is set in the CREEP parameter option in order to use the viscoplastic option with a non-associated flow rule.

The viscoplasticity feature can be used to implement very general constitutive relations with the aid of user subroutines ZERO and YIEL.

Implicit Method

A general viscoplastic material law can be implemented through user subroutine UVSCPL. When using this method, the user is responsible to define the inelastic strain increment at the current stress.

Since the viscoplasticity model in MARC is a modified creep model, you should familiarize yourself with the creep analysis procedure (see Section 5.2).

Large Deformation Effect

Geometric nonlinearity leads to two types of phenomena: change in structural behavior and loss of structural stability.

There are two natural classes of large displacement problems: the large displacement, small strain problem and the large displacement, large strain problem. For the large displacement, small strain problem, changes in the stress-strain law may be neglected, but the contributions from the nonlinear terms in the strain displacement relations must not be neglected. For the large displacement, large strain problem, the constitutive relation must be defined in the correct frame of reference and is transformed from this frame of reference to the one in which the equilibrium equations are written.

The collapse load of a structure may be predicted by performing an eigenvalue analysis. If performed after the linear solution (increment zero) the Euler buckling estimate is obtained. An eigenvalue problem may be formulated after each increment of load; this procedure can be considered a nonlinear buckling analysis even though a linearized eigenvalue analysis is used at each stage.

Coordinate Frame

There are two fundamentally different approaches for the description of mechanical problems: the Eulerian method and the Lagrangian method. In the steady state Eulerian method, the finite element mesh is fixed in space and the material flows through the mesh. This approach is particularly suitable for analysis of steady-state processes, such as the steady-state extrusion process (see Section 5.6 on rigid-plastic flow), shown in Figure A 5.2-10.

This approach is not advantageous when the constitutive equations depend on current strains or deformation histories or when treating free surfaces, particularly if there are distributed loads applied to these surfaces. These problems are more easily handled by the Lagrangian method.

In the Lagrangian method, the finite element mesh is attached to the material and moves through space along with the material. In this case, there is no difficulty in establishing stress or strain histories at a particular material point and the treatment of free surfaces is natural and straightforward.

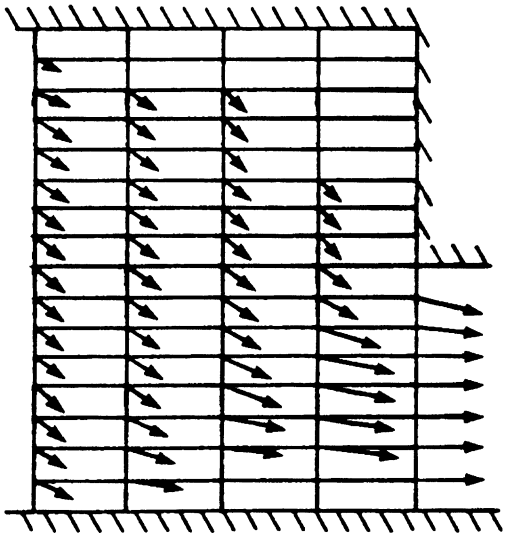


Figure A 5.2-10 Analysis of Steady-State Extrusion Process

The Lagrangian method is also the natural method to describe structural elements, i.e., shells and beams, and transient problems, such as the indentation problem shown in Figure A 5.2-11.

The Lagrangian method can also analyze steady-state processes such as extrusion. Drawbacks associated with the Lagrangian method are that a steady-state result can only be obtained as the limiting state of a transient analysis and that the distortion of the mesh is as severe as the deformation of the object. Severe mesh degeneration is shown in Figure A 5.2-12.

There are two varieties of Lagrangian methods: the total Lagrangian method and the updated Lagrangian method. In the total Lagrangian method, the mesh coordinates are not updated to the new positions. In the updated Lagrangian method, the coordinates of the mesh are updated after each increment.

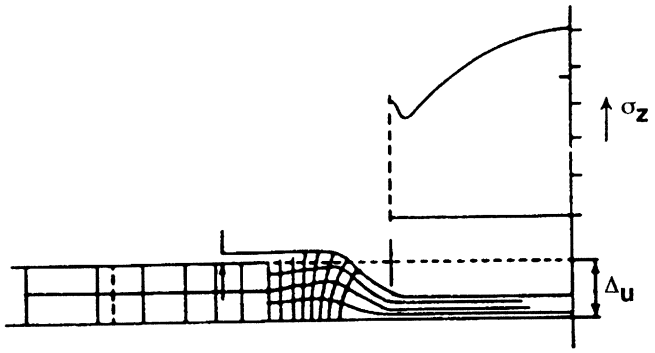


Figure A 5.2-11 Indentation Problem with Pressure Distribution on Tool

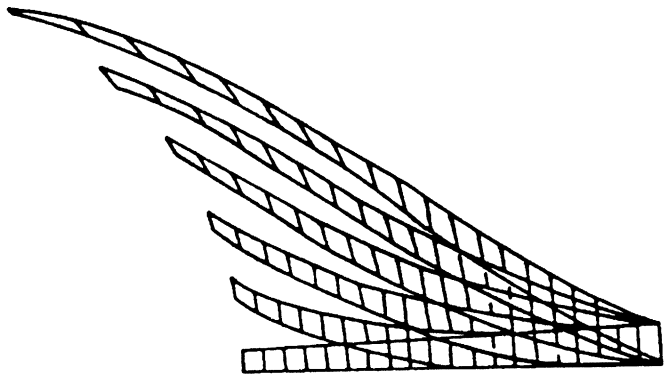


Figure A 5.2-12 In-Plane Torsion Test, Mesh Deformation

Depending on which option you use, the stress and strain results are given in different form as discussed below. If none of the following options, LARGE DISP, UPDATE or FINITE, is used, the program uses and prints "engineering" stress and strain measures. These measures are suitable only for analyses without large rotation or large strains.

Using only the LARGE DISP option, MARC uses the total Lagrangian method. The program uses and prints second Piola-Kirchhoff stress and Green-Lagrange strain. These measures are suitable for analysis with large rotations and small strains. Large strains may also be treated successfully with appropriate constitutive equations, such as Mooney-Rivlin, Ogden, or Foam model. The use of the Mooney-Rivlin, Ogden, or Foam material model will also cause the true (Cauchy) stress to be printed.

With the combination of LARGE DISP and UPDATE, MARC uses a variant on the engineering stress and strain measures. These measures are suitable for analyses with large rotations and small strains. Stress and strain components are printed with respect to the current state.

The combination of LARGE DISP, UPDATE and FINITE results in a complete large strain plasticity formulation. The program internally uses true (Cauchy) stress and the true deformation rate. This is also the stress printed, whereas the strain printed is an integrated form of the deformation rate. In a case of proportional straining, this method leads to logarithmic strains.

Other combinations of the options do not, in general, lead to good results.

Total Lagrangian Approach

The MARC program allows you to perform large displacement analysis based on the total Lagrangian approach. You can use this approach for linear or nonlinear materials, in conjunction with static or dynamic analysis. The total Lagrangian approach is particularly suitable for the analysis of nonlinear elastic problems (for instance, with the Mooney or Ogden material behavior or the user subroutine HYPELA). In these cases, the stress-free material will return to its original shape after load removal.

The total Lagrangian approach is also useful for problems in plasticity and creep, where moderately large rotations but small strains occur. This condition is typical in problems of beam or shell bending.

In the total Lagrangian approach, the large displacement formulation is based on the initial element geometry. Incremental stiffness matrices are formed to account for previously developed stress and changes in geometry.

To activate the large displacement (total Lagrangian approach) option in the MARC program, use parameter option LARGE DISP.

Include parameter option FOLLOW FOR for follower force (e.g., centrifugal or pressure load) problems. This option forms all distributed loads on the basis of the current geometry. Do not use the CENTROID parameter with this option. Always use residual load corrections with this option.

To input control tolerances for large displacement analysis, use model definition option CONTROL.

In the MARC program, the LARGE DISP option, when used without the UPDATE option, uses a Lagrangian initial coordinate frame of reference. The fundamental stress and strain measures used are second Piola-Kirchhoff stress and Green-Lagrange strain; therefore, you must ensure that any stress-strain data for large strain problems are based on these measures. These measures are not necessary for solving small strain problems.

Technical Background

The principle of virtual work defines equivalent forces P at the nodes for a virtual displacement δu

$$\delta u \cdot P = \int_V \delta\varepsilon\sigma dV = \int_V \delta u \beta^T \sigma dV \qquad \text{(A 5.2-29)}$$

Integration is performed over the initial volume V.

Canceling the non-zero virtual displacements from both sides and writing Equation A 5.2-29 in incremental form, we obtain

$$\Delta P = \left[\int_V \Delta\beta^T \Gamma dv + \int_V \Delta\beta^T L \beta dv\right] \Delta u \qquad \text{(A 5.2-30)}$$

The last term on the right of Equation A 5.2-30 can be divided into one matrix that depends on the current displacement and one that does not. With some rearrangement, we obtain the element stiffness matrices

$$K = K_1 + K_2 + K_0 \qquad \text{(A 5.2-31)}$$

where $K_1$is the initial stress matrix, obtained from the first term on the right of Equation A 5.2-30, K_2 is the initial displacement matrix, and K_0 is the small displacement stiffness matrix.

The element stiffness matrices and the nodal equivalent forces are then summed in the usual direct stiffness manner to obtain master-stiffness equations represented by equations in capitals

$$\Delta P = (K_1 + K_2 + K_0)\, \Delta u \qquad \text{(A 5.2-32)}$$

This Lagrangian formulation can be applied to problems if the undeformed configuration is known so that integrals can be evaluated, and if the second Piola-Kirchhoff stress is a known function of the strain. The first condition is not usually met for fluids, because the deformation history is usually unknown. For solids, however, each analysis usually starts in the stress-free undeformed state, and the integrations can be carried out without any difficulty. Also, stress can be written as a function of the deformation for many solids, particularly elastic and hyperelastic materials.

The second Piola-Kirchhoff stress for elastic and hyperelastic materials is a function of the Green-Lagrange strain defined below.

$$S_{ij} = S_{ij}(E_{kl}) \qquad \text{(A 5.2-33)}$$

If the stress is a linear function of the strain (linear elasticity)

$$S_{ij} = L_{ijkl}(E_{kl}) \qquad \text{(A 5.2-34)}$$

the resulting set of equations is still nonlinear because the strain is a nonlinear function of displacement.

For viscoelastic fluids and elastic-plastic and viscoplastic solids, the constitutive equations usually supply an expression for the rate of stress in terms of deformation rate, stress, deformation, and sometimes other (internal) material parameters. The relevant quantity for the constitutive equations is the rate of stress at a given material point.

It therefore seems most obvious to differentiate the Lagrangian virtual work equation with respect to time. The rate of virtual work is readily found as

$$\int_V \left[\dot{S}_{ij}\delta E_{ij} + S_{ij}\frac{\partial v_k}{\partial X_i}\frac{\partial \delta u_k}{\partial X_j}\right] dV = \int_V \dot{Q}_i \delta u_i dV + \int_S \dot{T}_i \delta u_i dS \qquad \text{(A 5.2-35)}$$

This formulation is adequate for most materials, because the rate of second stress may be written as

$$\dot{S}_{ij} = \dot{S}_{ij}(\dot{E}_{kl}, S_{mn}, E_{pq}) \qquad \text{(A 5.2-36)}$$

For many materials, the stress rate is even a linear function of the strain rate

$$\dot{S}_{ij} = \dot{L}_{ijkl}(S_{mn}, E_{pq})\,\dot{E}_{kl} \qquad \text{(A 5.2-37)}$$

Equation A 5.2-35 supplies a set of linear relations in terms of the velocity field. The velocity field may be solved noniteratively.

Updated Lagrangian Approach

In the updated Lagrangian approach, the element stiffness is assembled in the current configuration of the element, and the stress and strain output are given with respect to the coordinate system in the updated configuration of the element. The updated Lagrangian approach is useful in: a) analysis of shell and beam structures in which rotations are large so that the nonlinear terms in the curvature expressions may no longer be neglected, and b) large strain plasticity analysis, for calculations which the plastic deformations cannot be assumed to be infinitesimal.

The MARC program provides an updated Lagrangian option for large displacement analysis. To activate this option, use parameter options UPDATE and LARGE DISP. The UPDATE option in MARC (used with or without the large displacement option) defines a new (Lagrangian) frame of reference at the beginning of each increment. This option is suitable for analysis of problems of large rotation and small strain. If analysis of large plastic strain is required, use the FINITE option in addition to the UPDATE option. The program uses true stress (σ) and deformation rate (D). Integration of the deformation rate in a uniaxial tensile specimen furnishes logarithmic strains. Hence, you must input your stress-strain curves as true stress versus logarithmic strain.

In case the engineering stress curve S_E - ε_E is given, the following relations hold:

True Stress $$\sigma = \frac{P}{A} = \frac{P}{A_0} \bullet \frac{A_0}{A} = \frac{A_0}{A} S_E \qquad \text{(A 5.2-38)}$$

Logarithmic strain:

$$\varepsilon_T = \ln\left(1 + \frac{u}{L_0}\right) = \ln(1 + \varepsilon_E) \qquad \text{(A 5.2-39)}$$

Note that for (approximate) incompressible material behavior

$$A(1 + \varepsilon_E) = A_0 \qquad \text{(A 5.2-40)}$$

Hence, the expression for the true stress may then be approximated by

$$\sigma = (1 + \varepsilon_E) S_E \qquad \text{(A 5.2-41)}$$

The updated Lagrangian approach can be used to analyze structures where inelastic behavior (e.g., plasticity, viscoplasticity, or creep) causes the large deformations. The (initial) Lagrangian coordinate frame has little physical significance in these analyses since the inelastic deformations are, by definition, permanent. For these analyses, the Lagrangian frame of reference is redefined at the beginning of each increment.

One may use the updated procedure with or without the LARGE DISP option. When you use the LARGE DISP option, the program takes into account the effect of the internal stresses by forming the initial stress stiffness. Also, the program calculates the strain increment to second order accuracy to allow large rotation increments. It is recommended that the LARGE DISP option be used in conjunction with the UPDATE option.

Although you can theoretically derive a constitutive equation of the type shown in Equation A 5.2-36 and Equation A 5.2-37 for most materials, it is often more suitable to specify the constitutive equations with reference to the current state. This is the case if the moduli in Equation A 5.2-38 do not depend on the total strain E_{ij}. Obtain the desired formulation by taking Equation A 5.2-35 in the current state as the reference state. It follows that momentarily

$$F_{ij} = \delta_{ij}, \quad \delta E_{ij} = \delta D_{ij}, \quad \frac{\partial}{\partial X_i} = \frac{\partial}{\partial x_i}, \quad S_{ij} = \sigma_{ij} \tag{A 5.2-42}$$

where F is the deformation tensor, and D is the rate of deformation, and therefore, Equation A 5.2-35 transforms into

$$\int_V \left[\dot{\sigma}^T_{ij} \delta D_{ij} + \sigma_{ij} \frac{\partial v_k}{\partial x_i} \frac{\partial \delta u_k}{\partial x_j} \right] dv = \int_V \dot{q}_i \delta u_i dv + \int_S \dot{t}_i \delta u_i ds \tag{A 5.2-43}$$

In this equation, $\dot{\sigma}^T_{ij}$ is the Truesdell rate of Cauchy stress which can be obtained from the usual material rate of Cauchy stress by differentiation of

$$\dot{\sigma}_{ij} = J^{-1} F_{ik} \dot{S}_{kl} F_{jl} + J^{-1} \dot{F}_{ik} S_{kl} F_{jl} + J^{-1} F_{ik} S_{kl} \dot{F}_{jl} - \dot{J} J^{-2} F_{ik} S_{kl} F_{jl} \tag{A 5.2-44}$$

If you again take the current state as the reference state it follows that

$$\dot{\sigma}_{ij} = \dot{\sigma}^T_{ij} + \frac{\partial v_i}{\partial x_k} \sigma_{kj} + \sigma_{ik} \frac{\partial v_j}{\partial x_k} - \sigma_{ij} \frac{\partial v_k}{\partial x_k} \tag{A 5.2-45}$$

The Truesdell rate of Cauchy stress is materially objective. That is, if a rigid rotation is imposed on the material, the Truesdell rate vanishes, whereas the usual material rate does not vanish. The constitutive equations may well be formulated in terms of the Truesdell rate of Cauchy stress. Equation 5.2-37 may be written as

$$\dot{\sigma}^T_{ij} = L_{ijkl}(\sigma_{mn}) D_{kl} \tag{A 5.2-46}$$

Buckling Analysis

Buckling analysis allows you to determine at what load the structure will collapse. You can detect the buckling of a structure when the structure's stiffness matrix approaches a singular value. You can perform an eigenvalue analysis on the linear structure to obtain the linear buckling load. You can also use eigenvalue analysis for nonlinear buckling based on the incremental stiffness matrices.

The buckling option estimates the maximum load that can be applied to a geometrically nonlinear structure. To activate the buckling option in the program, use the parameter option BUCKLE. If a nonlinear buckling analysis is performed, also use the parameter option LARGE DISP.

Use the history definition option BUCKLE to input control tolerances for buckling load estimation (eigenvalue extraction by a power sweep method). You can estimate the buckling load after every load increment. The BUCKLE INCREMENT option may be used if a collapse load calculation is required at multiple increments.

For extremely nonlinear problems, the BUCKLE option may not produce accurate results. In that case, the history definition option AUTO INCREMENT allows automatic load stepping in a quasi-static fashion for both geometric large displacement and material (elastic-plastic) nonlinear problems. The option can handle elastic-plastic snap-through phenomena. Therefore, the post-buckling behavior of structures can be analyzed.

The linear buckling load analysis is correct when you take a very small load step in increment zero, or make sure the solution has converged before buckling load analysis (if multiple increments are taken). Linear buckling (after increment zero) can be done without using the LARGE DISP option, in which case the restriction on the load step size no longer applies.

In a buckling problem that involves material nonlinearity (e.g., plasticity), the nonlinear problem must be solved incrementally. In the MARC program, you know that plastic collapse has occurred when there is failure to converge in the iteration process or when the stiffness matrix turns nonpositive definite.

The buckling option solves the following eigenvalue problem by the inverse power sweep method

$$[K + \lambda \Delta K_G (\Delta u, u, \Delta \sigma)] \phi = 0 \qquad \text{(A 5.2-47)}$$

Where ΔK_G is assumed to be a linear function of the load increment ΔP to cause buckling. The geometric stiffness ΔK_G used for the buckling load calculation is based on the stress and displacement state change at the start of the last increment. However, the stress and strain states are not updated during the buckling analysis. The buckling load is therefore estimated by

$$P\,(\text{beginning}) + \lambda \Delta P \qquad \text{(A 5.2-48)}$$

where for increments greater than 1, P (beginning) is the load applied at the beginning of the increment prior to the buckling analyses, and λ is the value obtained by the power sweep.

The control tolerances for the inverse power sweep method are the maximum number of iterations in the power sweep and the convergence tolerance. The power sweep terminates when the difference between the eigenvalues in two consecutive sweeps divided by the eigenvalue is less than the tolerance.

Perturbation Analysis

The buckling mode may be used to perform a perturbation analysis of the structure. In the manual mode, a buckling increment is performed upon request and the coordinates are perturbed by a fraction of the buckle eigenvector. One enters the eigenvector number and the fraction. In the subsequent increments the coordinates will be:

$$X = X + f \bullet \frac{\phi}{|\phi|} \qquad \text{(A 5.2-49)}$$

The manual mode may be activated by using the BUCKLE INCREMENT model definition, or BUCKLE load incrementation option. In the automatic mode, the program will check for a non-positive definite system during the solution phase. When this occurs, it will automatically perform a buckle analysis during the next increment and then update the coordinates. The automatic mode may be activated by using the BUCKLE INCREMENT option. Also, be sure to force the solution of the nonpositive definite system through the CONTROL or PRINT option.

Large Strain Elasticity

Structures composed of elastomers, such as tires and bushings, are typically subjected to large deformation and large strain. An elastomer is a polymer, such as rubber, which shows an elastic nonlinear stress-strain behavior. The large strain elasticity capability in MARC deals primarily with elastomeric materials. These materials are characterized by the form of their elastic strain energy function. For a more detailed description of elastomeric material, see Section 6.4.

For the finite element analysis of elastomers, there are some special considerations which do not apply for linear elastic analysis. These considerations, discussed below, include:

- Large deformations
- Incompressible behavior
- Instabilities in the material description
- Existence of multiple solutions

Large Deformations

The formulation is complete for arbitrarily large displacements and strains; the solution, however, is obtained as a series of piecewise linear increments, so that suitable small load steps must be taken.

When extremely large deformations occur, the element mesh should be designed so that it can follow these deformations without complete degeneration of elements. You must first get some idea what the shape of the structure will be after the loads have been applied, and adapt the element mesh in order to obtain a reasonable solution. It is possible to use rezoning when using the lower order incompressible elements (80-84, 118-120).

Incompressible Behavior

One of the most frequent causes of problems analyzing elastomers is the incompressible material behavior. Lagrangian multipliers (pressure variables) are used to apply the incompressibility constraint. The result is that the volume is kept constant in a generalized sense, over an element.

In the finite element formulation, each Lagrangian multiplier (pressure variable) corresponds to a constraint on the volume. Be sure that the number of constraints is not too small or too large. Do this by comparing the actual number of kinematic degrees of freedom (hence not counting degrees of freedom which are prescribed with boundary conditions or tying) with the number of Lagrangian multiplier (constraints). Under all circumstances, the number of Lagrangian multiplier must be smaller than the actual number of kinematic degrees of freedom. Otherwise, no deformation can occur and the Lagrangian multiplier may take on arbitrary values. Although this condition is necessary, it is usually not sufficient to obtain an accurate solution. Therefore, the number of degrees of freedom should be considerably larger than the number of multipliers. As a guideline, the number of degrees of freedom should be at least twice the number of multipliers for 2D or axisymmetric analysis, and at least three times the number of multipliers for 3D analysis.

The large strain elasticity formulation may also be used with conventional plane stress, membrane and shell elements. Because of the plane stress conditions, the incompressibility constraint can be satisfied without the use of Lagrange multipliers.

Instabilities in the Material Description

Under some circumstances, materials can become unstable. This instability may be real or may be due to the mathematical formulation used in calculation.

Instability may also result from the approximate satisfaction of incompressibility constraints. If the number of Lagrangian multipliers is insufficient, local volume changes may occur. Under some circumstances, these volume changes may be associated with a decrease in total energy. This type of instability usually occurs only if there is a large tensile hydrostatic stress.

Existence of Multiple Solutions

It is possible that more than one stable solution exists (due to nonlinearity) for a given set of boundary conditions. An example of such multiple solutions is a hollow hemisphere with zero prescribed loads. Two equilibrium solutions exist: the undeformed stress-free state and the inverted self-equilibrating state. An example of these solutions is shown in Figure A 5.2-13 and Figure A 5.2-14. If the equilibrium solution remains stable, no problems should occur; however, if the equilibrium becomes unstable at some point in the analysis, problems may occur.

When incompressible material is being modeled, the basic linearized incremental procedure is used in conjunction with mixed variational principles similar in form to the Herrmann incompressible elastic formulation. These formulations are incorporated in plane strain, axisymmetric, generalized plane strain, and three- dimensional elements. These mixed elements may be used in combination with other elements in the library (suitable tying may be necessary) and with each other. Where different materials are joined, the pressure variable at the corner nodes must be uncoupled to allow for mean pressure discontinuity. Tying must be used to couple the displacements only.

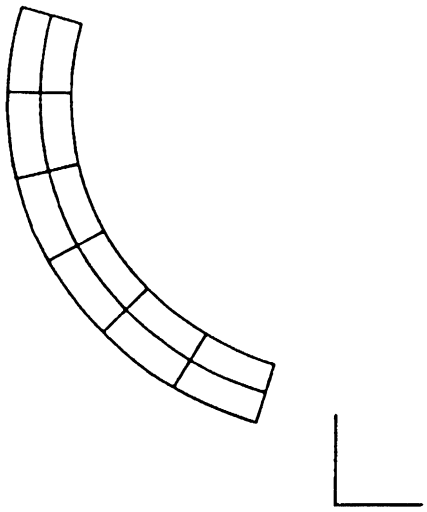


Figure A 5.2-13 Rubber Hemisphere

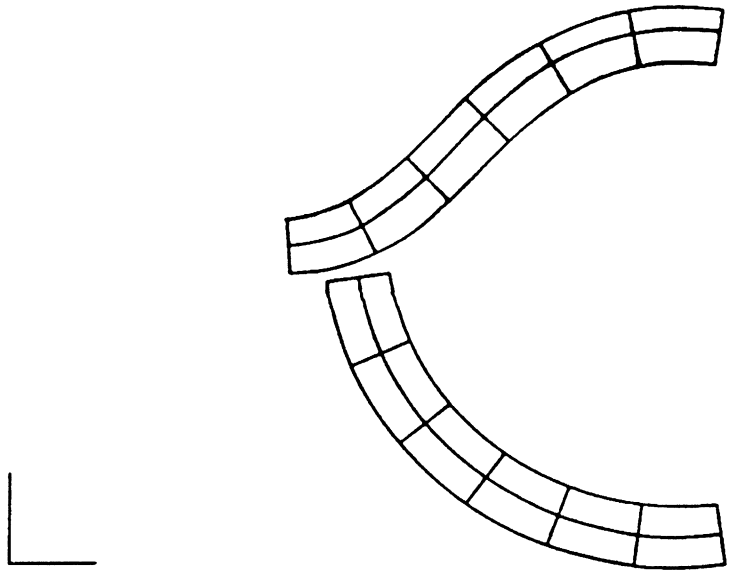


Figure A 5.2-14 Inverted Rubber Hemisphere

Large Strain Plasticity

In recent years there has been growing interest in the analysis of metal forming problems by the finite element method. Although an Eulerian flow-type approach and a total Lagrangian approach have been used for steady-state and transient problems, the updated Lagrangian procedure, pioneered by McMeeking and Rice, is most suitable for analysis of large strain plasticity problems. The main reasons for this are: a) its ability to trace free boundaries, and b) the flexibility of taking elasticity and history effects into account.

In contrast to the Eulerian method, the updated Lagrangian method allows you to take into account elasticity and work hardening effects. Residual stresses can be accurately calculated.

The large strain plasticity capability in MARC allows you to analyze problems of large-strain, elastic-plastic material behavior. These problems may include manufacturing processes such as forging, upsetting, extension or deep drawing, and/or large deformation of structures that occur during plastic collapse. The analysis involves both material, geometric and boundary nonlinearities.

In addition to the options required for plasticity analysis, the LARGE DISP, UPDATE, and FINITE options are needed for large strain plasticity analysis.

In performing finite deformation elastic-plastic analysis, there are some special considerations which do not apply for linear elastic analysis. These considerations include:

- Choice of finite element types
- Nearly incompressible behavior
- Treatment of boundary conditions
- Severe mesh distortion
- Instabilities in the material description

Choice of Finite Element Types

Accurate calculation of large strain plasticity problems depends on the selection of adequate finite element types. In addition to the usual criteria for selection, two aspects need to be given special consideration: the element types selected need to be insensitive to (strong) distortion; for plane strain, axisymmetric and three-dimensional problems, the element mesh must be able to represent non-dilatational (incompressible) deformation modes.

Nearly Incompressible Behavior

Most finite element types tend to lock during fully plastic (incompressible) material behavior. A remedy is to introduce a modified variational principle which effectively reduces the number of independent dilatational modes (constraints) in the mesh. This procedure is successful for plasticity problems in the conventional "small" strain formulation. Zienkiewicz pointed out the positive effect of reduced integration for this type of problem and demonstrates the similarity between modified variational procedures and reduced integration. MARC recommends the use of lower-order elements, invoking the constant dilatation option. The lower-order elements, which use reduced integration and hourglass control, also behave well for nearly incompressible materials.

Treatment of Boundary Conditions

In many large strain plasticity problems, specifically in the analysis of manufacturing processes, the material slides with or without friction over curved surfaces. This results in a severely nonlinear boundary condition. The MARC gap-friction element and CONTACT option can model such sliding boundary conditions.

Severe Mesh Distortion

Because the mesh is attached to the deforming material, severe distortion of the element mesh often occurs, which leads to a degeneration of the results in many problems. The ERROR ESTIMATES option may be used to monitor this distortion. To avoid this degeneration, generate a new finite element mesh for the problem and then transfer the current deformation state to the new finite element mesh. The REZONING procedure in the program is specifically designed for this purpose.

Instabilities in the Material Description

Elastic-plastic structures are often unstable due to necking phenomena. Consider a rod of a rigid-plastic incompressible work hardening material. With $\dot{\varepsilon}$ the current true uniaxial strain rate and h the current work hardening, the rate of true uniaxial stress σ is equal to

$$\dot{\sigma} = h\dot{\varepsilon} \qquad \text{(A 5.2-50)}$$

The applied force is equal to $F = \sigma A$, where A is the current area of the rod. The rate of the force is therefore equal to

$$\dot{F} = \dot{\sigma}A + \sigma\dot{A} \qquad \text{(A 5.2-51)}$$

On the other hand, conservation of volume requires that

$$A\dot{\varepsilon} + \dot{A} = 0 \qquad \text{(A 5.2-52)}$$

Hence, the force rate can be calculated as

$$\dot{F} = (h - \sigma)\, A\dot{\varepsilon} \qquad \text{(A 5.2-53)}$$

Instability clearly occurs if $\sigma > h$. For applied loads (as opposed to applied boundary conditions), the stiffness matrix will become singular (non-positive definite).

For the large strain plasticity option, the work hardening slope for plasticity is the rate of true stress versus the true plastic strain rate. The work hardening curve must therefore be entered as the true stress versus the logarithmic plastic strain in a uniaxial tension test.

With the updated Lagrangian option, you cannot use the anisotropic plasticity formulation, but you can use anisotropic elastic properties.

Technical Background

In the following equation, the rate of virtual work is derived in the updated Lagrangian method as

$$\int_v \left[\dot{\sigma}_{ij}^T \frac{\partial \delta u_i}{\partial x_j} + \sigma_{ij} \frac{\partial v_k}{\partial x_i} \frac{\partial \delta u_k}{\partial x_j} \right] dv = \int_v \dot{q}_i \ \delta u_i dv + \int_s \dot{t}_i \ \delta u_i ds \qquad \text{(A 5.2-54)}$$

In this equation, the Truesdell rate of Cauchy stress is related to the usual rate of Cauchy stress by

$$\dot{\sigma}_{ij}^T = \dot{\sigma}_{ij} - \frac{\partial v_i}{\partial x_k} \sigma_{kj} - \sigma_{ik} \frac{\partial v_j}{\partial x_k} + \sigma_{ij} \frac{\partial v_k}{\partial x_k} \qquad \text{(A 5.2-55)}$$

and the integrations are carried out over the current volume and surface. It is also postulated that the constitutive equations for elastic-plastic materials exist in the form

$$\dot{\sigma}_{ij}^T = L_{ijkl} D_{kl} \qquad \text{(A 5.2-56)}$$

Now the moduli L_{ijkl} are not equal to the classical elastic-plastic moduli

$$L_{ijkl}^{e-p} = 2G \left[\delta_{ik}\delta_{jl} + \frac{\lambda}{2G} \delta_{ij}\delta_{kl} - \frac{3}{2} \frac{\sigma'_{ij}\sigma'_{kl}}{\sigma_0^2} \right] \qquad \text{(A 5.2-57)}$$

Nagtegaal and deJong demonstrated that these moduli supply the relation between the Jaumann rate of Cauchy stress and the deformation rate

$$\dot{\sigma}_{ij}^J = L_{ijkl}^{e-p} D_{kl} \qquad \text{(A 5.2-58)}$$

The Jaumann rate of Cauchy stress is related to the material rate of Cauchy stress with the equation

$$\dot{\sigma}_{ij}^J = \dot{\sigma}_{ij} - \omega_{ik}\sigma_{kj} - \sigma_{ik}\omega_{jk} \qquad \text{(A 5.2-59)}$$

where the spin tensor ω_{ij} is defined by

$$\omega_{ij} = \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} - \frac{\partial v_j}{\partial x_i} \right) \qquad \text{(A 5.2-60)}$$

The Jaumann rate of Cauchy stress is the rate of change of the Cauchy stress in a corotational system. The material rate of Cauchy stress can be eliminated from Equation A 5.2-55 and Equation A 5.2-59 with use of Equation A 5.2-60 which yields the relation

$$\dot{\sigma}_{ij}^T = \sigma_{ij}^J - D_{ik}\sigma_{kj} - \sigma_{ik}D_{kj} + \sigma_{ij}D_{kk} \qquad \text{(A 5.2-61)}$$

Thus the relation between the large strain moduli and the classical elastic-plastic moduli is

$$L_{ijkl} = L^{e-p}_{ijkl} - \delta_{il}\sigma_{kj} - \sigma_{il}\delta_{kj} + \sigma_{ij}\delta_{kl} \qquad \text{(A 5.2-62)}$$

Note that the last term in Equation A 5.2-62 does not satisfy the usual symmetry relation.This is not relevant for metal plasticity problems since the deformations are approximately incompressible. Hence, Equation A 5.2-62 may be approximated by

$$L_{ijkl} \approx L^{e-p}_{ijkl} - \delta_{il}\sigma_{kj} - \sigma_{il}\delta_{kj} \qquad \text{(A 5.2-63)}$$

Nonlinear Boundary Conditions

There are three types of problems associated with nonlinear boundary conditions: contact, nonlinear support, and nonlinear loading. The contact problem may be solved through the use of special gap elements or the CONTACT option. Nonlinear support may involve nonlinear springs and/or foundations. Sometimes nonlinearities due to rigid links that become activated or deactivated during an analysis may be modeled through adaptive linear constraints. Nonlinear loading is present if the loading system is nonconservative, as is the case with follower forces or frictional slip effects.

Discontinuities are inherent in the nature of many of these nonlinearities, making the solution by means of incremental linear approximations difficult. Some of the most severe nonlinearities in mechanics are introduced by nonlinear boundary conditions. It is, therefore, very important to be aware of potential problem areas and to have a good understanding of the underlying principles. This awareness and understanding will enable you to validate numerical answers and to take alternative approaches if an initial attempt fails.

Contact Problems

Contact problems are common occurrences that are often ignored. They are, however, important problems, and demand serious attention in various engineering analyses. Some examples of contact problems are the interface between the metal workpiece and the die in metal forming processes, pipe whip in piping systems, and crash simulation in automobile designs.

Contact problems are characterized by two important phenomena: gap opening and closing, and friction. As shown in Figure A 5.2-4, the gap describes the contact (gap closed) and separation (gap open) conditions of two objects (structures). Friction influences the interface relations of the objects after they are in contact. The gap condition is dependent on the movement (displacements) of the objects, and friction is dependent on the contact force as well as the coefficient of (Coulomb) friction at contact surfaces. The analysis involving gap and friction must be carried out incrementally. Iterations may also be required in each (load/time) increment to stabilize the gap-friction behavior.

Two options are available in MARC for the simulation of a contact problem. A description of these options (gap-friction element and the CONTACT option) is given below.

Gap-Friction Element

In the MARC program, a gap-friction element is available for the analysis of contact problems. This element provides gap and frictional connections between any two nodes of a structure. This element also can model a gap in a fixed direction or sliding along a curved surface. In addition, the element can also simulate the condition where a point is not allowed to penetrate a given circular (2D) or spherical (3D) surface.

The gap-friction element (Element Type 12) is used to model contact problems. Refer to Volume B (element library) for detailed information about this element.

Enter the gap-friction data through the model definition option GAP DATA.

To update the contact conditions, use the user subroutine GAPU. Both the closure distance and the direction of closure may be modified.

Obtain additional output information regarding gap convergence by using the PRINT parameter option.

**** CAUTION ****

During increment 0, the MARC program does not iterate to deal with nonlinearities. Changes in gap status during the zeroth increment lead to inaccurate results. If gap-friction nonlinearities occur immediately, defer external loading until the first increment to avoid changes in gap status during increment 0. If you specify a negative closure distance for gap elements, MARC finds the linear solution to the interference fit problem that is created by the "overclosed" gaps in the zeroth increment. As long as the overclosed gaps do not open in increment 0, the solution is correct.

If higher-order isoparametric elements are used, the gap elements should be attached to only the corner nodes. If a complete element face is in contact with a rigid wall, the situation is comparable to a distributed pressure applied on the same element face. It is well known that in the case of the 20-node hexahedron, a uniform surface pressure results in consistent nodal forces of different sign so the consistent force at the corner nodes acts in the opposite direction to the applied pressure. This means that even for completely uniform contact, the gaps at the corner nodes would remain open. To avoid this problem, the midside nodes should be constrained to conform to a linear displacement variation from one corner node to another, and gap elements should be used only at the corners.

To keep the recycling (required due to changes in gap status) to a minimum during nonlinear boundary problem simulations, use as few gap elements as possible and avoid modes of loading that produce chatter among the different gaps.

Technical Background

In the MARC program, the modeling of gap-friction is based on the imposition of kinematic constraints. The minimization of the total potential energy is subject to these constraints. This necessitates the introduction of Lagrange multipliers and the solution of an expanded system of equations. Conditional logic determines when to enforce a constraint or when to ignore it.

During the evaluation of the stiffness matrix, the gap status is based on the estimated strain increment. The gap status is checked again after the solution is obtained, during the recovery of the strains and stresses. For cases where the gap status differs between assembly and recovery due to penetration or change from slipping to sticking friction, recycling of the current increment is enforced.

Once contact between two components has been established, the question of frictional forces and possible slip arises. If friction is assumed to be present, additional constraints are used. Two frictional constraints are activated upon contact. Two frictional directions are perpendicular to each other and form the plane normal to the direction in which contact is established. If the frictional force exceeds the maximum allowed, based on the normal force and the coefficient of friction, an additional slip constraint is necessary to limit the magnitude of the friction force.

A simple description of the method of Lagrange multiplier for analysis of constraint conditions follows:

Let the system equation

$$Ku = f \tag{A 5.2-63}$$

be subjected to constraint conditions

$$Cu = 0 \tag{A 5.2-64}$$

Through the minimization of the augmented functional

$$\Psi = \frac{1}{2}u^T Ku - u^T f + \lambda^T Cu \tag{A 5.2-65}$$

we obtain

$$\begin{bmatrix} K & C^T \\ C & O \end{bmatrix} \begin{pmatrix} u \\ \lambda \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \tag{A 5.2-66}$$

This equation can be solved simultaneously for both u (displacement) and λ (Lagrangian multiplier). In the gap-friction element, the values of the Lagrangian multipliers represent normal and frictional gap forces, as well as frictional slippage.

CONTACT Option

Whenever two or more bodies are in contact or come in contact with each other, it is necessary to impose non-penetration constraints. These constraints have been automated in MARC using the CONTACT model definition option.

It should be mentioned that, in many circumstances, the more traditional ways of imposing non-penetration (e.g., application of boundary conditions, FIXED DISPLACEMENTS, and the use of gap elements) may achieve better results.

The basic concept in the contact features of MARC is the definition of bodies. The boundary surfaces of these bodies contain all the geometrical information necessary to impose non-penetration. In most circumstances, the bodies correspond to the physical model being analyzed with finite elements. In some cases, however, a body may be present for the sole purpose of constraining other bodies, with no analysis required. This leads to the concept of defining both deformable bodies and rigid bodies. Deformable bodies are a simple collection of finite elements. Rigid bodies are pure geometrical entities. A requirement is that at least one deformable-body must be present for any finite element analysis to be performed.

The non-penetration constraint, as exemplified in Figure A 5.2-15 is expressed as

$$\underset{\sim}{U}_A \bullet \underset{\sim}{n} \leq D \qquad \text{(A 5.2-67)}$$

where $\underset{\sim}{U}_A$ is the displacement vector, $\underset{\sim}{n}$ is the normal vector, and D is the distance between the body and the rigid surface. In the finite element framework this constraint can be imposed by several means: Lagrange multipliers, penalty functions, or solver constraints. The CONTACT feature uses the latter approach.

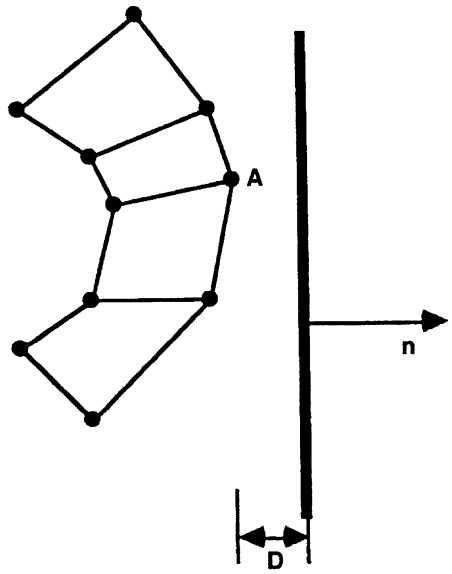


Figure A 5.2-15 The Non-penetration Constraint in CONTACT

Whenever contact between a deformable-body and a rigid-body is detected, imposed displacements are automatically created. Whenever contact between two deformable bodies is detected, multipoint constraints (called TIES) are automatically created. No contact between rigid bodies is ever considered.

As stated above, a deformable-body is defined by the user as a set of elements. MARC determines all of the nodes on the boundary, which become the set of candidate nodes for contact. That is, the contact algorithm will attempt to prevent penetration of any of these nodes into any defined body. Simultaneously, the body's boundary surface is stored as a set of geometrical entities. All boundary nodes will be prevented from crossing this surface. If it is known where contact will occur, the CONTACT NODE option may be used to define the candidate nodes for contact. Use of this option reduces computational costs.

A rigid-body is defined by means of a set of geometrical entities which define the boundary surface. It is not necessary to completely define a full body envelope. Only the regions of potential contact must be defined, provided they are all connected. Several geometrical primitives are accepted as building blocks to construct surfaces such as straight lines, circles, splines, and NURB curves in two dimensions and patches, ruled surfaces, surfaces of revolution, Bezier surfaces, and NURB surfaces in three dimensions. These primitives are described in this section. The rigid surfaces may be either represented in analytical form of in discrete form. If the discrete form is used, all geometrical primitives are subdivided into straight segments or flat patches. The user has control over the density of these subdivisions, in order to approximate a curved surface within a desired degree of accuracy (Figure A 5.2-16).

The way geometrical entities are entered determines which side of the described surface is the rigid-body. In two dimensions, the surface is described by a sequence of its geometrical components. It is assumed that this sequence follows the periphery of the rigid-body counter-clockwise; in other words, a normal defined by the right-hand rule points to the inside of the body.

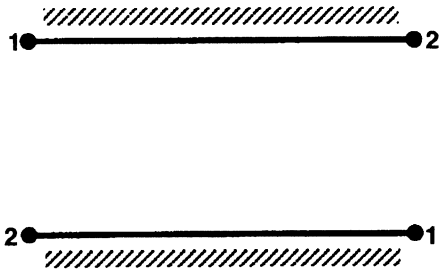


Figure A 5.2-16 Rigid-body Boundary

A rigid-body motion can be applied to any rigid-body. In order to allow for complex paths, motion is defined by means of the integration in time of instantaneous velocity fields. A rigid-body velocity field is uniquely defined by a translation plus a rotation around a point. Therefore, a point in space is associated with every rigid-body (regardless of whether it belongs to its surface) to which both translational and rotational velocities can be applied.

The motion path is obtained by an explicit integration of the velocity field. This motion definition and the need for synchronization have the consequence of all problems with CONTACT being time-controlled. A time increment needs to be defined, whether it is relevant for the mechanics problem being treated or not. The versatility of this approach can be demonstrated in Figure A 5.2-17. On the left, a rigid-body defined by three straight lines, and a translational motion completely defines a moving punch. On the right, two half circles and an angular velocity around the center completely define a rotating roll.

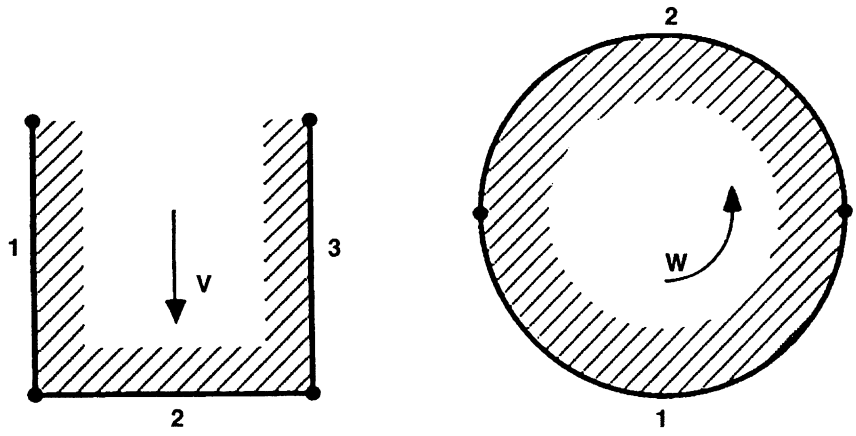


Figure A 5.2-17 Examples of Rigid Bodies

The MOTION CHANGE option may be used to define a new velocity. A user subroutine MOTION has been provided, which allows for continuous variations in the velocity field to be considered. In this way, any complicated path can be created, including stops and reversals, as well as synchronization between motions of different bodies.

Deformable-bodies cannot have rigid-body motions. The need to *move* a deformable-body other than by means of other contacting bodies, is done via traditional finite element procedures, such as fixed displacements, point loads or distributed loads.

NOTES

Although it is possible to freely move bodies around during an analysis, it remains the user's responsibility to make sure that deformable bodies do not have rigid-body modes. If a body has been deformed by several rigid bodies, and you want to calculate residual stresses by removing the contact with the rigid bodies, it becomes necessary to apply a minimum of boundary conditions to the body so that it will not *float* in the air. It is not necessary to constrain rigid-body motion in dynamic analysis.

The simultaneous occurrence of contact and a fixed boundary condition at a node is assumed to mean that the node is on a line of symmetry. Therefore, care should be taken not to apply fixed displacements to nodes that may come into contact. Also, once this simultaneous occurrence takes place, the boundary condition cannot be later removed.

In MARC, contact between a deformable-body and a rigid surface is insured such that the nodes do not penetrate the rigid surface. It is possible that an edge of an element penetrates a rigid surface, especially where high curvature is present because of the finite element discretization. The use of the ADAPTIVE mesh generation option may be used to reduce these problems. Contact between deformable bodies is such that each node on each body is mutually checked for contact with each edge of every other body. There is no master/slave relation; the default is that every body will be checked for contact versus every other body. Often we know, a priori, that two bodies may never contact one another. The CONTACT TABLE option may be used to so inform the program, saving computation time.

Whenever the CONTACT option is in use, increment 0 is a null increment. Certain initialization operations are done at this time. For instance, the determination of all the boundary nodes, the translation of the boundary surface into geometrical entities, or the subdivision of user-entered rigid surface definitions into internal primitives are among them. Rigid-body profiles are often complicated, making it difficult to determine exactly where the first point of contact is located. If a rigid-body has a non-zero motion associated with it, the initialization procedure will bring it to first contact with any deformable-body. If there is more than one rigid-body in the analysis, each rigid-body which has a non-zero initial velocity will be moved until it comes into contact.

The implementation of the contact algorithms requires that the following steps are performed every increment of an analysis.

1. Find all the nodes that are in contact. This is determined by the distance between the nodes and surfaces. Since the distance is a calculated number, there are always roundoff errors involved. Therefore, a contact tolerance is provided such that if the distance calculated is below this tolerance, a node is considered in contact. This tolerance is provided either by the user, or calculated by MARC as 1/15 of the smallest element size for solid elements or 1/2 of the thickness for shell elements. In general, the contact tolerance should be a small number compared to the geometrical features of the configuration being analyzed. Another important use of the contact tolerance is described in point 5 below.

2. For all the nodes that are in contact, determine either the tying constraints or the imposed displacement increments along the normal to the contacted surface, as well as a local transformation that defines such a normal. This step is repeated for each cycle of a nonlinear problem in order to permit a node to find its equilibrium position along the contact surface.

3. Once a convergent solution is found, contact forces are analyzed. If a contacting node has a tensile contact force greater than the separation tolerance, the node is released and another solution is found. The program will automatically calculate the separation tolerance force based upon the residual forces or you can enter it directly. A large separation tolerance ensures that there will not be any separation. If you want to move two contacting bodies apart, a solution requiring substantial artificial stretching might have to be found before the decision to separate is made. It is recommended that you use the RELEASE option to totally separate two bodies or that the separation be done in very small increments.

4. At this point, you must decide whether a node that was free at the beginning of the increment would go through the body by the end of the increment. If this is the case, the increment of time is reduced in size so that the first node barely reaches contact. In this way, the beginning of the following increment detects new contact and the analysis proceeds normally.

5. If the controls of the problem (HISTORY DEFINITION) require a set of equally spaced increments, the previous possible shrinking in size of an increment leads to an increment split. MARC repeats the same increment number in order to execute the remainder part. You can easily see that if several nodes reach contact almost simultaneously, there is a potential for a very large number of increment splits. However, in spite of the first split being made with respect to the first contacting node, Step 1 (above) of the remainder increment will detect and bring into contact all subsequent nodes that are within the tolerance band of the surfaces they will contact. There is an advantage in making the contact tolerance as large as possible, so that the largest possible value is still small compared with the geometrical features of the problem.

The CONTACT option is compatible with the REZONE option of MARC. If the number of elements that constitute a body has changed, it becomes necessary to redetermine all the boundary nodes as well as the geometrical description of the surface. For this reason alone, the CONTACT block can be repeated within the REZONE block of data. No other changes in geometrical descriptions are allowed other than the deformable-body element definition.

NOTE

It is often forgotten that if a rezoning increment increases the number of nodes, elements, or boundary conditions it is necessary to reserve space for these features from the very beginning of the analysis. This can easily be done on the SIZING parameter option.

In general, the data initially provided in the CONTACT option cannot be altered during an analysis other than by the user subroutines provided. The only exceptions are the REZONE feature described above, and the tolerances given in the third card of the option, which can be altered either upon RESTART or REZONE.

Friction at the interface between bodies is numerically modeled by one of two methods: 1) by means of any application of distributed shear forces, or 2) by direct application of nodal forces. Two classical models are incorporated, namely adhesive friction (or Coulomb friction), where friction forces are defined by the expression

$$f_t \leq -\mu f_n t \qquad \text{(A 5.2-68)}$$

where

f_t tangential force being applied
f_n normal pressure
μ friction coefficient
t tangent unit vector, in the direction of the relative velocity

$$t = \frac{V_r}{|V_r|} V_r \qquad \text{relative sliding velocity}$$

and cohesive friction (or constant shear friction), where friction forces are defined by the expression

$$f_t \leq -mk_y t = -m\frac{\sigma_y}{\sqrt{3}}t \qquad \text{(A 5.2-69)}$$

where

f_t is the tangential force being applied
σ_y is the flow stress of the material being deformed
m is the friction factor
t is the tangent unit vector, in the direction of the relative velocity

The distributed load method of implementation may be used with continuum elements using either friction model.

Whenever shell elements are used in conjunction with the adhesive model, it becomes obvious that contact pressures are not available in the analysis. In such cases, friction is then applied as a lumped nodal force, following the same as Equation A 5.2-59. This method may also be used optionally with the continuum elements.

Table A 5.2-2 Continuum Elements

Numerical Method		
Friction Method	**Distributed**	**Point**
Adhesive	possible	possible (optional)
Cohesive	possible	not available

Table A 5.2-3 Shell Elements

Numerical Method		
Friction Method	**Distributed**	**Point**
Adhesive	distributed	optional
Cohesive	not available	possible

NOTE

When friction forces are treated as distributed forces, the analyst should take into consideration that no friction will be applied at an interface that has a single node in contact.

Equation A 5.2-68 and Equation A 5.2-69 are inequalities whenever two contacting surfaces stick to each other, and equalities whenever surfaces slide. The character of the contact constraints changes depending on whether there is sticking or slipping. In order to avoid this distinction, an approximation was made within the CONTACT option in such a way that there is always slip. However, the friction forces decrease to zero when the amount of slip also decreases to zero. The slip or sliding velocity below which such decrease comes into effect is given by the user. MARC is able to calculate a value if there are moving rigid bodies. This modeling is quite robust and does not affect the quality of the solution provided the sliding velocity referred to is small (a couple of orders of magnitude) compared to typical sliding velocities in the analysis. In this way, only nodes that would otherwise stick are affected.

Sometimes the two standard friction models implemented are insufficient for an appropriate friction description. To address this problem, a user subroutine UFRIC has been created that permits the user to constantly monitor the interface conditions and modify friction. For instance, it is possible to switch models, and make friction dependent on location, pressure, temperature, amount of sliding, etc.

When a deformable-body contacts a rigid-body, the friction coefficient of the rigid surface is used. When deformable-deformable contact occurs, the friction coefficient is averaged. The CONTACT TABLE option may also be used if complex situations occur.

In metal forming analyses, the user often would like to remove the rigid surfaces and evaluate the mechanical springback. The RELEASE option may be used to release the nodes of a deformable-body which are in contact with a rigid-body. Either the MOTION CHANGE option must be used to move the rigid surface away, or the CONTACT TABLE option must be used to ensure that the nodes do not recontact the surface they were released from.

The CONTACT option has also been made compatible with the coupled thermal-mechanical analysis capabilities of MARC. This feature produces a simultaneous solution of both heat transfer and mechanics problems. Enhancements were made to the contact algorithms that lead to an easy modeling of realistic coupled problems. The concept of bodies also exists. In this case, if the bodies are deformable, a fully coupled analysis is done. A rigid-body, described so far by its geometry, is assigned a constant temperature. A new modeling possibility was introduced in which the body is rigid, but a heat transfer analysis is performed in it. In these cases, the body is defined as a set of HEAT TRANSFER elements. The same boundary node determination, as well as surface determination, is done as for deformable bodies. Mechanical contact between a deformable-body and these rigid bodies is also done by means of ties. However, a rigid-body motion is given to the full heat transfer mesh, controlled in the same way a rigid-body normally is.

Heat fluxes (FILMS) are automatically created on all the boundaries of the deformable-bodies. Two film coefficients are assigned to each body. One corresponds to free convection and is used whenever the boundary is free. The other is a contact film, and is used whenever the boundary is in contact with another body. In these cases the heat flux across the interface is given by

$$q = HD\,(TD - T) \qquad \text{(A 5.2-70)}$$

where

- T is the surface temperature
- HD is the film coefficient between the two surfaces
- TD is the temperature of the same contact location, as obtained from interpolation of nodal temperatures of the body being contacted.

When a deformable-body contacts a rigid-body, the coefficients associated with the rigid-body are used. When two deformable bodies are in contact, the average film coefficient is used. The CONTACT TABLE option may also be used to specify the film coefficient. As with all other coupled problems, heat generated by plastic deformation can be calculated and applied as a volumetric flux. The heat generated by friction is also calculated, and applied as a surface flux.

Two user subroutines are available to facilitate the creation of more sophisticated boundary flux definitions (such as radiation and convections with variations in space, temperature, pressure, etc.) UHTCON allows the user to specify a film coefficient while the surface is in contact, and UHTCOE allows the user to specify a film coefficient while the surface is free.

Nonlinear Support

The MARC program provides two options for the modeling of support conditions: springs and elastic foundations. In a nonlinear problem, the spring stiffness and the equivalent spring stiffness of the elastic foundation can be modified through a user subroutine. In the nonlinear spring option, the incremental force in the spring is

$$\Delta F = K(\Delta u_2 - \Delta u_1) \qquad \text{(A 5.2-71)}$$

where K is the spring stiffness, Δu_2 is the displacement increment of the degree of freedom at the second end of the spring, and Δu_1 is the displacement increment of the degree of freedom at the first end of the spring.

Use the SPRINGS model definition option for the input of spring data. User subroutine USPRNG may be used to specify the value of K based on the amount of previous deformation for nonlinear springs. In dynamic analysis, the SPRING option may also be used to define a dashpot.

In the elastic nonlinear foundation option, the elements in MARC may be specified as being supported on a frictionless (nonlinear) foundation. The foundation supports the structure with an increment force per unit area given by

$$\Delta P_n = K(u_n)\,\Delta u_n \qquad \text{(A 5.2-72)}$$

where K is the equivalent spring stiffness of the foundation (per unit surface area), and Δu_n is the incremental displacement of the surface at a point in the same direction as ΔP_n.

To input nonlinear foundation data, use the FOUNDATION model definition option.

To specify the value of K for the nonlinear equivalent spring stiffness based on the amount of previous deformation of the foundation, use the user subroutine USPRNG.

Nonlinear Loading

When the structure is deformed, the directions and the areas of the surface loads are changed. For most deformed structures, such changes are so small that the effect on the equilibrium equation can be ignored. But for some structures such as flexible shell structure with large pressure loads, the effects on the results can be quite significant so that the surface load effects have to be included in the finite element equations.

MARC forms both pressure stiffness and pressure terms based on current deformed configuration with the FOLLOW FOR option. The FOLLOW FOR option should be used with the LARGE DISP option and the parameter option CENTROID should not be included due to the use of the residual load correction.

A 5.3 FRACTURE MECHANICS

Linear Fracture Mechanics

Linear fracture mechanics is concerned with whether cracks will form or grow. In particular, it determines the length at which a crack propagates rapidly for specified load and boundary conditions. The concept of linear fracture mechanics stems from Griffith's work on purely brittle materials. Griffith stated that, for crack propagation, the rate of elastic energy release should at least equal the rate of energy needed for creation of a new crack surface. This concept was extended by Irwin to include limited amounts of ductility. In Irwin's considerations, the inelastic deformations are confined to a very small zone near the tip of a crack.

It is easy to demonstrate that in both Griffith's and Irwin's considerations the elastic energy release rate is determined by a single parameter: the strength of the singularity in the elastic stress field at the crack tip. This is the so-called *stress intensity* factor, and is usually denoted by capital K. The magnitude of K depends on the crack length, the distribution and intensity of applied loads, and the geometry of the structure. Crack propagation will occur when any combination of these factors causes a stress intensity factor K to be equal to or greater than the experimentally determined material property K_c. Hence, the objective of linear fracture mechanics calculations is to determine the value of K. A more detailed discussion on stress intensity factors is given in the next section.

The basic concept presented by Griffith and Irwin is an energy rate balance between elastic energy release rate, and the rate of energy dissipation for crack propagation. The rate of energy release G depends only on the stress intensity factors K_I, K_{II}, and K_{III}.

The following relations hold:

For plane strain or antiplane shear

$$G = \frac{1-\nu^2}{E}(K_I^2 + K_{II}^2) + \frac{1+\nu}{E}K_{III}^2 \qquad \text{(A 5.3-1)}$$

For plane stress

$$G = \frac{1}{E}(K_I^2 + K_{II}^2) \qquad \text{(A 5.3-2)}$$

Hence, if the opening mode is known, the elastic energy release rate is an alternative for direct calculation of the stress intensity factor.

The stiffness differential procedure is a method for deriving the energy release rate directly from the results of a single analysis. With this technique, the average release rate of elastic energy G can be expressed as

$$G = -\frac{1}{2}q^2\frac{\Delta S}{\Delta \ell} \qquad \text{(A 5.3-3)}$$

where ΔS is the change in stiffness and q is the generalized displacements.

In a finite element formulation based on the displacement method, Equation 5.3-3 may be rewritten as

$$G = \frac{1}{2} u^T \frac{\Delta K}{\Delta \ell} u \qquad \text{(A 5.3-4)}$$

where u is the nodal displacement vector and ΔK is the change in the stiffness matrix due to propagation of the crack, and Δl is the amount of crack movement. A small propagation of the crack only has influence on a limited number of elements in the vicinity of the crack tip. Therefore, it is only necessary to calculate the stiffness change in a small number of elements, and to premultiply and postmultiply those with the displacements to obtain the energy release rate. This method was originally developed by Parks and has been implemented in MARC as the J-INTEGRAL option. The method usually yields very accurate results at almost no increase in cost.

Stress Intensity Factors

There are three possible modes of crack extension in linear elastic fracture mechanics: the opening mode, sliding mode, and tearing mode (see Figure A 5.3-1)

The opening mode (see Figure A 5.3-1a), Mode I, is characterized by the symmetric separation of the crack surfaces with respect to the plane, prior to extension (symmetric with respect to the X-Y and X-Z planes).

The sliding mode, Mode II, is characterized by displacements in which the crack surfaces slide over one another perpendicular to the leading edge of the crack (symmetric with respect to the X-Y plane and skew-symmetric with respect to the X-Z plane).

The tearing mode, Mode III, finds the crack surfaces sliding with respect to one another parallel to the leading edge (skew-symmetric with respect to the X-Y and X-Z planes).

The superposition of these three modes is sufficient to describe the most general case of crack tip stress and deformation fields, whenever the previously described plane strain, or plane stress, conditions can be assumed to exist.

The solutions for the near crack tip stress field for the above modes of crack extension are shown in Table A 5.3-1. The coordinates used in Table A 5.3-1 for fracture mechanics are shown in Figure A 5.3-2.

The significance of the expressions provided in Table A 5.3-1 is that they hold for all stationary, sharp, and straight-through plane cracks, regardless of the configuration of the planar body or the location of the crack. It is only the expressions for the stress intensity factors, K, which change when going from configuration to configuration. Once the functional forms of K have been determined, the state of elastic stress and displacement in the near crack tip region of the planar body can be determined immediately.

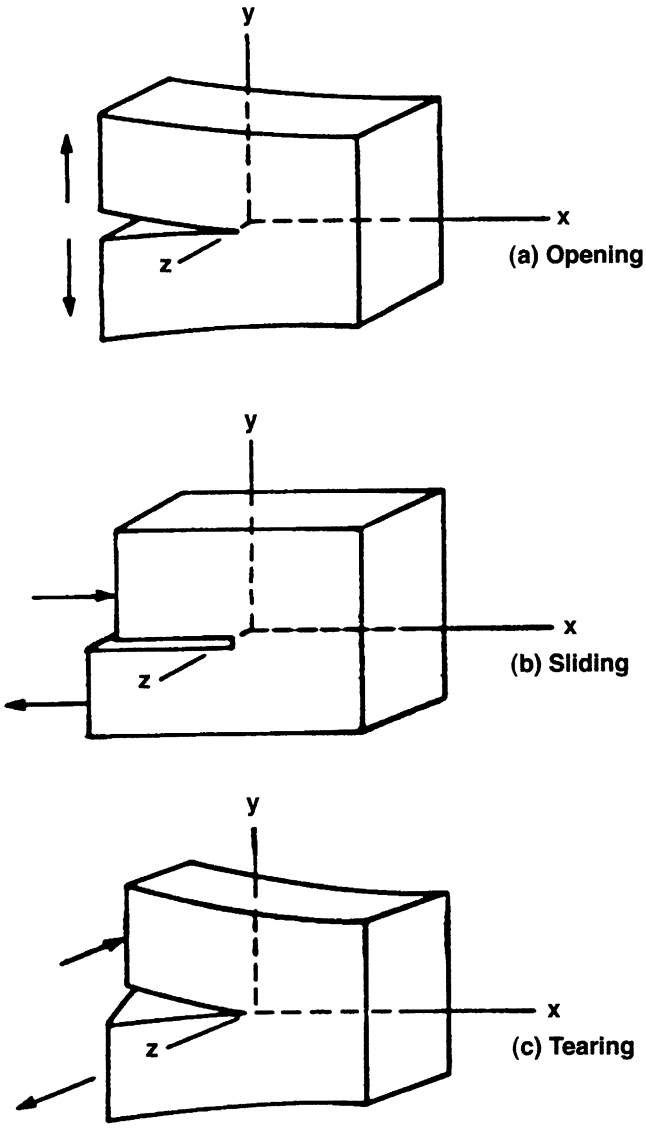


Figure A 5.3-1 Irwin's Three Basic Modes of Crack Extension

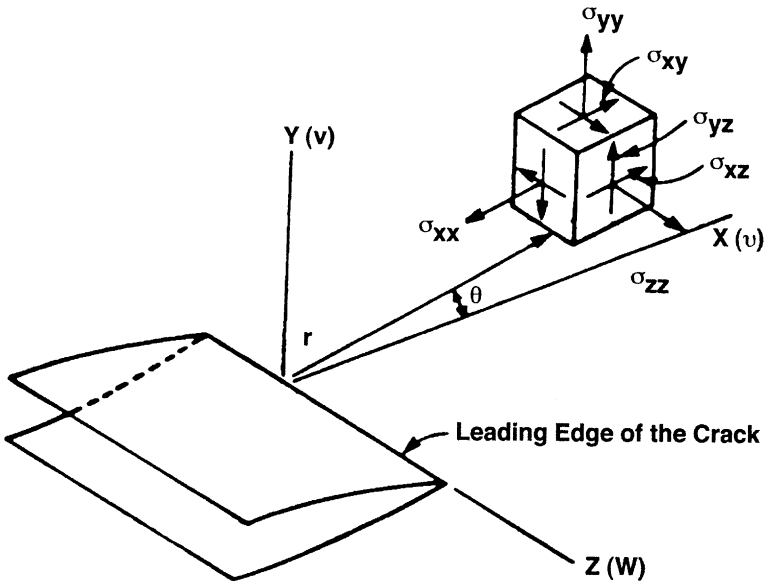


Figure A 5.3-2 Coordinates Measured from the Leading Edge of a Crack and the Stress Components in the Crack Tip Stress Field

Table A 5.3-1 Irwin's Near Crack Tip (Plane Strain) Solutions

	Equation
Mode I	$\sigma_{xx} = \frac{K_I}{\sqrt{2\pi r}} \cos\frac{\theta}{2}\left[1 - \sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right]$
	$\sigma_{yy} = \frac{K_I}{\sqrt{2\pi r}} \cos\frac{\theta}{2}\left[1 - \sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right]$
	$\sigma_{xy} = \frac{K_I}{\sqrt{2\pi r}} \sin\frac{\theta}{2}\cos\frac{\theta}{2}\cos\frac{3\theta}{2}$
	$\sigma_{zz} = \nu(\sigma_{xx} + \sigma_{yy})$
	$\sigma_{xz} = \sigma_{yz} = 0$
	$u = \frac{K_I}{G}\frac{\sqrt{r}}{\sqrt{2\pi}} \cos\frac{\theta}{2}\left[1 - 2\nu - \cos^2\frac{\theta}{2}\right]$
	$\nu = \frac{K_I}{G}\frac{\sqrt{r}}{\sqrt{2\pi}} \sin\frac{\theta}{2}\left[2 - 2\nu - \cos^2\frac{\theta}{2}\right]$
	$w = 0$
NOTE: In the above equations ν = Poisson's ratio, $G = E/2(1+\nu)$ = shear modulus, and E = Young's modulus	

Table A 5.3-1 Irwin's Near Crack Tip (Plane Strain) Solutions (Continued)

	Equation
Mode II	$\sigma_{xx} = \frac{K_{II}}{\sqrt{2\pi}} \sin\frac{\theta}{2}\left[2 + \cos\frac{\theta}{2}\cos\frac{3\theta}{2}\right]$ $\sigma_{yy} = \frac{K_{II}}{\sqrt{2\pi r}} \sin\frac{\theta}{2}\cos\frac{\theta}{2}\cos\frac{3\theta}{2}$ $\sigma_{xy} = \frac{K_{II}}{\sqrt{2\pi r}} \cos\frac{\theta}{2}\left[1 - \sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right]$ $\sigma_{zz} = \nu(\sigma_{xx} + \sigma_{yy})$ $\sigma_{xz} = \sigma_{yz} = 0$ $u = \frac{K_{II}}{G}\frac{\sqrt{r}}{\sqrt{2\pi}} \sin\frac{\theta}{2}\left[2 - 2\nu + \cos^2\frac{\theta}{2}\right]$ $\nu = \frac{K_{II}}{G}\frac{\sqrt{r}}{\sqrt{2\pi}} \cos\frac{\theta}{2}\left[-1 + 2\nu + \sin^2\frac{\theta}{2}\right]$ $w = 0$
Mode III	$\sigma_{zx} = \frac{K_{III}}{\sqrt{2\pi r}} \sin\frac{\theta}{2}$ $\sigma_{yz} = -\frac{K_{III}}{\sqrt{2\pi r}} \cos\frac{\theta}{2}$ $\sigma_{xx} = \sigma_{yy} = \sigma_{zz} = \sigma_{xy} = 0$ $w = \frac{K_{III}}{u}\frac{\sqrt{2r}}{\pi} \sin\frac{\theta}{2}$ $u = \nu = 0$
NOTE: In the above equations ν = Poisson's ratio, $G = E/2(1 + \nu)$ = shear modulus, and E = Young's modulus	

Nonlinear Fracture Mechanics

Nonlinear fracture mechanics is concerned with determining under which conditions crack propagation (growth) occurs. In this sense, nonlinear fracture mechanics is similar to linear fracture mechanics. However, there are additional questions addressed in nonlinear fracture mechanics. Will the crack propagation be stable or unstable? If it is stable, at which speed will it occur? After some propagation, will the crack be arrested?

Nonlinear fracture mechanics is still a research topic. Much work still needs to be done, especially on propagating cracks. The main problems with propagating cracks are the large number of parameters and the high cost of numerical analysis. Research on stationary cracks has been developed over the past years, and some important results have been obtained.

There also exists a singularity at the crack tip in fracture mechanics problems with nonlinear elastic-plastic material behavior, though the singularity is of a different nature. If one takes an exponential hardening law of the form

$$\frac{\sigma}{\sigma_0} = \left(\frac{\varepsilon}{\varepsilon_0}\right)^{1/n} \qquad \text{(A 5.3-5)}$$

it can be shown that the singularities in the strain and the stresses at the crack tip are of the form

$$\varepsilon = f(\theta)\, r^{-n/(n+1)}$$
$$\sigma = g(\theta)\, r^{-1/(n+1)} \qquad \text{(A 5.3-6)}$$

If n approaches infinity, the material behavior becomes perfectly plastic and the singularity in the stresses vanishes. The singularity in the strains, however, takes the form of

$$\varepsilon = f(\theta)\, r^{-1} \qquad \text{(A 5.3-7)}$$

It has not been possible to establish that the strength of the singularity is the only factor that influences initiation of crack propagation for nonlinear situations. In fact, it is doubted that initiation of crack propagation is dependent on only a single factor. The J-integral probably offers the best chance to have a single parameter to relate to the initiation of crack propagation.

Quarter Point Elements

The main difficulty in the finite element analysis of linear elastic fracture mechanics is representing the solution near the crack tip. The mesh at this point must be modeled so that the singularity is approximated with sufficient accuracy. Many methods have been devised to arrive at such an approximation; however, the most commonly used method uses a degenerate form of the standard 8-node quadrilateral element. This method is usually referred to as the "1/4 point node technique." This is also the preferred technique for analysis with the MARC program.

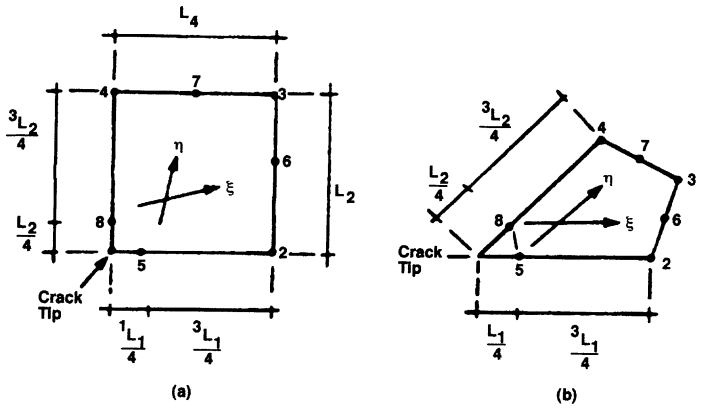
Best results have been obtained in planar or axisymmetric analysis by:

- using 8-node quadrilateral elements degenerated to triangles
- moving the nodes on the midsides adjacent to the cracktip to one quarter of the edge length (see Figure A 5.3-3)
- keeping the edge opposite the crack straight

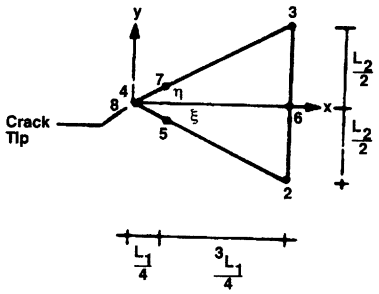
When these steps are followed, a singularity of the form $1/(\sqrt{r})$ in the stress field exists in all directions going radially outward from the crack tip. In three-dimensional analysis, best results are obtained if 20-node brick elements are degenerated to wedges in the same way as the 8-node quadrilaterals are degenerated to triangles. In this case, the edge nodes on the four edges adjacent to the crack front have to be moved to one-quarter of the edge length from the crack tip.

A more detailed description of these degenerated elements is given by Barsoum. Figure A 5.3-4 illustrates a typical arrangement of the quarter point elements at the tip of a crack.

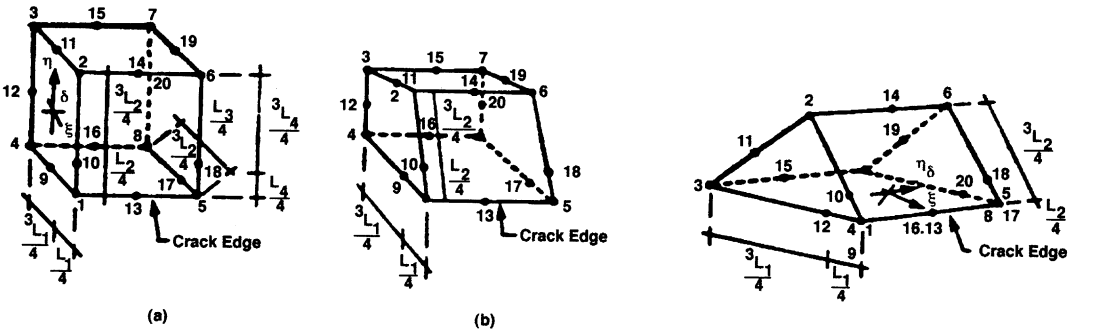
The 1/4 point node technique is also applicable to elastic-plastic analysis. In this type of analysis, strain singularities of $1/(\sqrt{r})$ and 1/r occur.



(a) 2D Rectangular Elements with Midside Nodes at the Quarter Points



(b) 2D Triangular Element with Midside Nodes at the Quarter Points



(c) 3D Brick Element with Midside Nodes at the Quarter Points

(d) 3D Prism with Midside Nodes at the Quarter Points

Figure A 5.3-3 Elements with Mid-Side Nodes at the Quarter Points

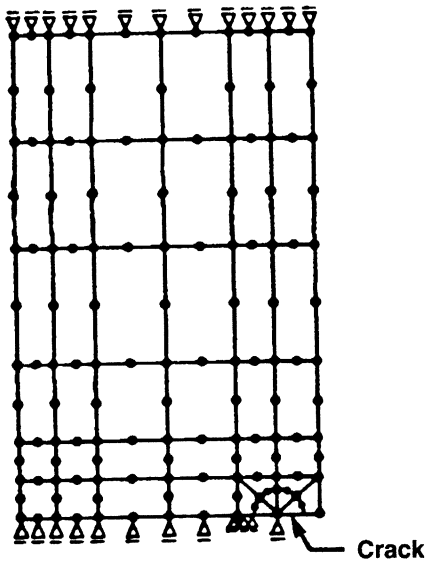


Figure A 5.3-4 Example of Finite Element Mesh for Crack Analysis

Dynamic Crack Propagation

The concepts of fracture mechanics discussed in previous sections have been applied to the prediction of crack initiation, as well as slow stable crack growth of statically loaded structures and for the prediction of fatigue crack growth in cyclically loaded structures. In problems where inertial effects cannot be ignored, application of quasi-static fracture mechanics techniques may lead to erroneous conclusions. The use of dynamic fracture mechanics concepts for these problems is clearly of necessity. The main emphasis on dynamic fracture mechanics is to predict the initiation of stationary cracks in structures, which are subjected to impact loading. It also focuses on the conditions for the continuous growth of fast propagating cracks, and on the conditions under which a crack is arrested.

The problem of predicting the growth rate and the possible crack arrest point is quite complicated. It is often treated by means of a so-called dynamic fracture methodology, which requires the combined use of experimental measurements and of detailed finite element analyses. An essential step in this approach is formed by the numerical simulation of propagating cracks by means of the finite element method.

An extended version of the J-integral originally proposed by Rice has been implemented into the MARC program. The extended J-integral takes into account the effect of inertial and body forces, thermal and mechanical loading and initial strains. The expression for the extended J-integral is then transformed into a surface integral for a straightforward evaluation of its value, by means of numerical integration techniques.

The use of parameter option DYNAMIC and the model definition option LORENZI allows for the calculation of dynamic energy release rates for cracked bodies which are subjected to arbitrary thermal and mechanical loadings including initial stresses.

Dynamic Fracture Methodology

In complete similarity with static fracture mechanics concepts, it is assumed that dynamic crack growth processes for linear materials are governed by the following condition:

$$K_I(t) = R_{ID}(\dot{a}, T, B) \qquad \dot{a} \geq 0 \qquad \text{(A 5.3-8)}$$

where $K_I(t)$ is the dynamic stress intensity factor for mode I, $\dot{a}$ is the crack velocity, and R_{ID} is the dynamic crack propagation toughness, which is assumed to be a material parameter that in general will depend on crack velocity $\dot{a}$, temperature T, and specimen thickness B.

The dynamic stress intensity factor will depend on crack length (a), applied loading (σ), time (t), specimen dimensions (D), temperature (T) and initial stress fields (σ_i) caused by residual stresses or by an initial strain field. The prediction of the crack propagation history and crack arrest event demands complete knowledge of the R_{ID} vs. $\dot{a}$ relation. The "Dynamic Fracture Methodology" procedure consists of the following two phases:

1. *Generation phase* – in this phase a crack arrest experiment is performed, yielding a crack propagation-versus-time curve. In addition, a numerical simulation of the experiment is carried out by using the measured crack propagation curve. This is used as input for the numerical model. This allows the calculation of dynamic stress intensity factors as a function of time. Combination of the latter relation with the measured crack propagation curve will result in a curve, which may be considered the dynamic crack propagation toughness-versus-crack velocity relation.
2. *Application phase* – in order to predict the crack growth and possible crack arrest point in a structural component, the inverse problem is solved. Now the actual stress intensity factors are calculated for the structural component, that is subjected to a particular loading history, by means of a dynamic finite element analysis. These calculated values are compared to the fracture toughness curve obtained during the generation phase, Equation A 5.3-8, and from this the crack growth is predicted.

J-integral Evaluation

The MARC program allows the evaluation of the J-integral through two different procedures. The first method described in this section evaluates the J-integral of Rice using the Parks method. The extended J-integral of Kishimoto and DeLorenzi is discussed in the following section.

The MARC program allows the evaluation of the J-integral by calculating the change in strain energy due to nodal movement during any analysis. Several values of the J-integral may be obtained by choosing several paths. The plastic strains are included in the definition of strain energy change for elastic-plastic analysis. In this way, the J-integral for the equivalent nonlinear elastic material is evaluated.

This J-integral evaluation feature is invoked through the use of the J-INTEGRAL model definition option. You must use the ALL POINTS option for this evaluation.

You can apply this method for the evaluation of the J-integral for structures with static loads. For dynamic or thermally driven problems, the extended J-integral discussed in the next section should be used. The method can be applied for elastic, elastic-plastic, and geometrically nonlinear responses. In the second case, the integral is evaluated for the equivalent nonlinear elastic material.

High accuracy can be achieved with a very coarse model through this method because it is based on the strain energy of the mesh, rather than on local values of stress or displacement.

The J-integral evaluation option can be used with either first- or second-order elements. The use of quarter-point elements at the crack tip is recommended.

To estimate the J-integral for each load case in elastic re-analysis, use the ELASTIC parameter option.

The program prints out the change in strain energy. This value must then be divided by the change in crack surface area to obtain the J-integral.

It can be proved that for a nonlinear elastic material, the J-integral is equal to the energy release rate G, and that it is path-independent.

For an elastic-plastic material, the J-integral is uniquely defined outside the plastic region. It approximately equals the energy release rate for crack propagation.

The evaluation of the J-integral in MARC is based on node movement and numerical differentiation to obtain the change in the potential energy as a function of nodal positions. It is assumed that the loads are not changed by node movement, so that the only concern is the change in strain energy

$$E = \sum_{\text{elements}} \int_{v} W^{e}(\varepsilon)\, dV \qquad \text{(A 5.3-9)}$$

where the Σ represents an appropriate sum over nodes associated with the element whose volume is represented in the integral and $W^e(\varepsilon)$ is the strain energy density.

$$\frac{dE}{dl} = \sum_{\text{elements}} \int_{v} \sigma_{ij} \frac{d\bar{\beta}_{ij}^{N} \bar{u}^{N}}{dl} dV + \sum_{\text{elements}} \int_{v^0} w \frac{d}{dl} \left| \frac{dV}{dV} \right| dV^0 \qquad \text{(A 5.3-10)}$$

This quantity is evaluated (by numerical differentiation) at each increment for each prescribed nodal movement to evaluate dE/dl at the end of each increment.

The J-integral procedure is not applicable for body forces, thermal stress analysis, creep analysis, or dynamics.

An Extended J-integral

Since the J-integral was proposed by Rice, various researchers have presented extensions of the initially proposed J-integral in order to account for various effects such as plastic deformation, body forces, thermal loading, inertial forces, large displacements, and large strains. In the following section a modified J-integral, which incorporates most of the aforementioned effects, is presented.

Conservation Laws

Consider a structure with a subregion Ω that has a volume V and a boundary Γ with surface S to which an arbitrary coordinate mapping $x' = x + \delta x$ is applied. When the coordinate change δx is infinitesimal, it can be shown that the following holds:

$$\int_{\Gamma} \delta x_k \left(W\delta_{jk} - \sigma_{ij}\frac{\partial u_i}{\partial x_k} \right) n_{ij} dS + \int_{\Omega} \{ \frac{\partial \delta x_k}{\partial x_j} \left(W\delta_{jk} - \sigma_{ij}\frac{\partial u_i}{\partial x_k} \right) + \delta x_k (f_i - \rho\ddot{u}_i) \frac{\partial u_i}{\partial x_k} - \sigma_{ij}\frac{\partial \varepsilon_{ij}^0}{\partial x_k} \} dV = 0 \qquad \text{(A 5.3-11)}$$

where W is the elastic strain energy density, and n is the outward normal to Γ.

For a unit virtual translation of Ω in the x_k direction and in the absence of body forces, inertial effects and initial stresses (Equation A 5.3-11) reduces to

$$\int_{\Gamma} \left(W n_k - t_i \frac{\partial u_i}{\partial x_k} \right) dS = 0 \qquad \text{(A 5.3-12)}$$

where the traction vector on Γ is defined by $t_1 = \sigma_{ij}\, n_j$.

Equation A 5.3-12 states that during such a translation of a subregion of the body the change of the strain energy is equal to the work done by the external tractions on the surface of the subregion.

Application to Sharply Notched Defects

The conservation law according to Equation A 5.3-11 will now be applied to a structure containing a sharply notched defect in order to derive a general path-independent integral, which characterizes the fracture process.

In the following, we will restrict ourselves to a plane structure only without the loss of generality. Consider a plate of thickness t containing a crack which coincides with the x_1 direction. The x_2 axis is perpendicular to the crack surface. As illustrated in Figure A 5.3-5, we consider a subregion Ω_1 with a closed surface $\Gamma_1 + \Gamma_{s1} + \Gamma_p$ where Γ_1 is any arbitrary chosen contour surrounding the crack tip, Γ_{s1} the fracture surface and Γ_p is a circular contour with radius $r \rightarrow 0$ surrounding the process zone that is of negligible size and in which continuum mechanics may not be applied. Note that the normal direction on Γ_p is chosen opposite to the normal on Γ_1.

If a crack extension Δa is considered, which may be treated as a constant virtual coordinate movement $\delta\underline{x} = (\Delta a, 0, 0)$ in Ω_1, application of the conservation law of Equation A 5.3-11 to the region Ω_1 with its closed surface consisting of Γ_1, Γ_{s1} and Γ_p will result in:

$$\int \left(\Gamma_1 W n_1 - t_i \frac{\partial u_i}{\partial x_1} \right) dS - \int \Gamma s_1 t_i \frac{\partial u_i}{\partial x_1} dS - \lim_{r \to 0} \int \left(W n - t_i \frac{\partial u_i}{\partial x_1} \right) dS - \int \Omega_1 \left((f_i - \rho\ddot{u}) \frac{\partial u_i}{\partial x_1} - \sigma_{ij} \frac{\partial \varepsilon_{ij}^0}{\partial x_1} \right) dV = 0 \quad \text{(A 5.3-13)}$$

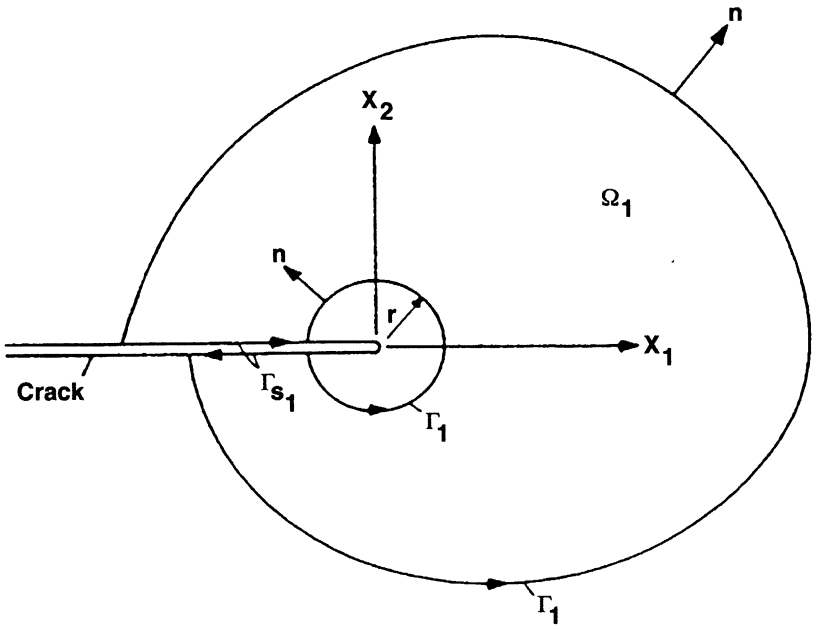


Figure A 5.3-5 Closed Contour Used in the Evaluation of the J-integral

The amount of energy that is dissipated in the process zone per unit crack extension; i.e., the rate of energy released at the crack tip, is expressed by the limiting value of the integral along Γp in the above relation. We will define this rate of energy as:

$$J = \lim_{r \to 0} \int_{\Gamma_1} \left(W n_1 - t_i \frac{\partial u_i}{\partial x_1} \right) dS \qquad \text{(A 5.3-14)}$$

Based on Equation A 5.3-13, J may be written as:

$$J = \int_{\Gamma_p} \left(W n_1 - t_i \frac{\partial u_i}{\partial x_1} \right) dS - \int_{\Gamma_{s_1}} t_i \frac{\partial u_i}{\partial x_1} dS - \int_{\Omega_1} \left((f_i - \rho \ddot{u}_i) \frac{\partial u_i}{\partial x_1} - \sigma_{ij} \frac{\partial \varepsilon_{ij}^0}{\partial x_1} \right) dV \qquad \text{(A 5.3-15)}$$

For the situation that body forces, inertia forces, initial strains and tractions on the crack surfaces are absent, J reduces to the path-independent integral proposed by Rice:

$$J = \int_{\Gamma_1} \left(W n_1 - t_i \frac{\partial u_i}{\partial x_1} \right) dS \qquad \text{(A 5.3-16)}$$

The stress/strain fields at the crack tip may be characterized by evaluating an integral away from the crack tip. In the case that the previously mentioned effects are present, the property that J can be determined based on the far field solution is no longer valid, because the value of the integral along Γ_1 will depend on the distance of Γ_1 to the crack tip. It is worth mentioning that no assumptions were made for the particular choice of Γ_1. This choice is, therefore,

arbitrary. Evaluation of J, however, requires full knowledge of the mechanical state within the Ω_1 region. The only restriction imposed so far is that a strain energy density function exists which relates stresses to strains.

Numerical evaluation of J

The solution of realistic fracture problems in most cases can only be carried out using numerical techniques (e.g. the finite element method). The numerical evaluation of J in 2D situations requires the calculation of both a line and a surface integral. In the following, an alternative formulation of J in which only surface integrals are involved has been derived. Consider a region Ω_2 surrounded by the contours Γ_1, Γ_2 and Γ_s (see Figure A 5.3-6) by using the conservation law of Equation A 5.3-11 and by choosing δx_k such that: $\delta x_2 = \delta x_3 = 0$ both in Ω_2 and on its boundary $\delta x_1 = \Delta a$ on Γ_1 and $\delta x_1 = 0$ on Γ_2, the following expression can be derived:

$$J = -\int_\Omega \frac{1}{\Delta a}\frac{\partial \delta x_1}{\partial x_j}\left(W\delta_{1j} - \sigma_{ij}\frac{\partial u_i}{\partial x_1}\right)dV - \int_\Omega \frac{\partial x_1}{\Delta a}\left((f_i - \rho\ddot{u}_1)\frac{\partial u_i}{\partial x_i} - \sigma_{ij}\frac{\partial \varepsilon_{ij}^0}{\partial x_1}\right)dV - \int_{\Gamma_s}\frac{\partial x_1}{\Delta a}t_i\frac{\partial u_i}{\partial x_1}dS \qquad \text{(A 5.3-17)}$$

where $\Omega = \Omega_1 + \Omega_2$ and $\Gamma_s = \Gamma_{s_1} + \Gamma_{s_2}$.

Within a finite element program, this expression for J can be readily evaluated by means of the commonly used numerical integration methods.

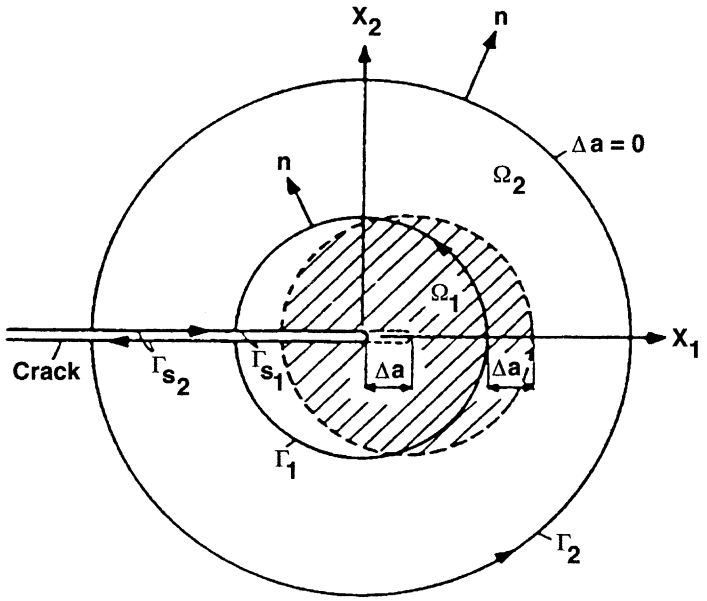


Figure A 5.3-6 Numerical Evaluation for J-integral

In case of the absence of body forces, free surface tractions, inertial effects, and initial strains the above expression reduces to:

$$J = \frac{-1}{\Delta a}\int_{\Omega} \frac{\partial \delta x_1}{\partial x_j}\left(W\delta_{1j} - \sigma_{ij}\frac{\partial u_i}{\partial x_1}\right)dV \qquad \text{(A 5.3-18)}$$

The extended J-integral is evaluated by using the LORENZI model definition option. In this option, you define the elements in region Ω_1 and Ω_2 and prescribe the displacements of the nodes in Ω_2.

Physical Interpretation of J

The J-integral which was proposed by Rice as defined in Equation 5.3-20 can be interpreted as the amount of energy which flows through the contour Γ_1 per unit crack advance. If no work is done by tractions within the contour Γ_1 or on the crack surface and no energy is dissipated other than the energy that is associated with crack growth in the process zone, J equals the rate of energy that is released at the crack tip. This energy rate is absorbed in the process-zone when advancing the crack forward. In case the previously mentioned conditions are not fulfilled, the energy rate J that flows through Γ_1 will depend on the choice of Γ_1 and thus does not equal the energy released at the crack tip.

In contrast, J according to Equation 5.3-19 defines the net difference between the rate of energy that flows through Γ_1 and the energy that is consumed within the region enclosed by Γ_1. J, therefore, expresses the energy flow to the crack tip per unit crack advance, and is independent of the choice of Γ_1.

A 5.4 DYNAMICS

MARC has a dynamic analysis capability which allows you to perform the following calculations:

- Eigenvalue extraction
- Transient analysis
- Harmonic response
- Spectrum response analysis

The program contains two methods for eigenvalue extraction and three time integration operators. Nonlinear effects, including material nonlinearity, geometric nonlinearity, and boundary nonlinearity, may be incorporated.

Linear problems can be analyzed using modal superposition or direct integration. All nonlinear problems should be analyzed using direct integration methods.

In addition to distributed mass, you can also attach concentrated masses associated with each degree of freedom of the system. You can include damping in either the modal superposition or the direct integration methods. You can also include (nonuniform) displacement and/or velocity as an initial condition, and apply time-dependent forces and/or displacements as boundary conditions.

Eigenvalue Analysis

The MARC program uses either the inverse power sweep method or the Lanczos method to extract eigenvalues and eigenvectors. The DYNAMIC parameter is used to determine which procedure to use, and how many modes are to be extracted. The inverse power sweep method is typically used for extracting a few modes while the Lanczos method is optimal for more modes. After the modes are extracted, they can be used in a transient analysis or spectrum response calculation.

In dynamic eigenvalue analysis, we find the solution to an undamped linear dynamics problem:

$$(K - \omega^2 M)\,\phi = 0$$

where K is the stiffness matrix, M is the mass matrix, ω are the eigenvalues (frequencies) and ϕ are the eigenvectors. In MARC, if the extraction is performed after increment zero, K is the tangent stiffness matrix, which may include material and geometrically nonlinear contributions. The mass matrix is formed from both distributed mass and point masses.

Inverse Power Sweep

The MARC program creates an initial trial vector. To obtain a new vector, the program multiplies the initial vector by the mass matrix and the inverse (factorized) stiffness matrix. This process is repeated until convergence is reached according to either of the following criteria: single eigenvalue convergence or double eigenvalue convergence. In single eigenvalue convergence, the program computes an eigenvalue at each iteration. Convergence is assumed when the values of two successive iterations are within a prescribed tolerance. In double eigenvalue convergence, the program assumes that the trial vector is a linear combination of two eigenvectors.

Using the three latest vectors, the program calculates two eigenvalues. It compares these two values with the two values calculated in the previous step; convergence is assumed if they are within the prescribed tolerance.

When an eigenvalue has been calculated, the program either exits from the extraction loop (if a sufficient number of vectors has been extracted) or it creates a new trial vector for the next calculation. If a single eigenvalue was obtained, the program uses the double eigenvalue routine to obtain the best trial vector for the next eigenvalue. If two eigenvalues were obtained, the program creates an arbitrary trial vector orthogonal to the previously obtained vectors.

After the program has calculated the first eigenvalue, it orthogonalizes the trial vector at each iteration to previously extracted vectors (using the Gram-Schmidt orthogonalization procedure). Note that the power shift procedure is available with the inverse power sweep method.

To select the power shift, set the following parameters:

- *Initial shift frequency.* – This is normally set to zero (unless the structure has rigid body modes, preventing a decomposition around the zeroth frequency).
- *Number of modes to be extracted between each shift.* – A value smaller than five is probably not economical because a shift requires a new decomposition of the stiffness matrix.
- *Auto shift parameter.* – When you decide to do a shift, the new shift point is set to

  $$\text{Highest frequency}^2 + \text{scalar x (highest frequency - next highest frequency)}^2$$

  You can define the value of the scalar through the MODAL SHAPE option.

The Lanczos Method

The Lanczos algorithm converts the original eigenvalue problem into the determination of the eigenvalues of a tri-diagonal matrix. The method can be used either for the determination of all modes or for the calculation of a small number of modes. For the latter case, the Lanczos method is the most efficient eigenvalue extraction algorithm. A simple description of the algorithm is as follows.

Consider the eigenvalue problem:

$$-\omega^2 M \bullet u + K \bullet u = 0 \qquad \text{(A 5.4-1)}$$

Equation (5.4-1) can be rewritten as:

$$\frac{1}{\omega^2} M \bullet u = M \bullet K^{-1} \bullet M \bullet u \qquad \text{(A 5.4-2)}$$

Consider the transformation:

$$u = Q \bullet \eta \qquad \text{(A 5.4-3)}$$

Substituting (Equation A 5.4-3) into (Equation A 5.4-2) and pre-multiplying by the matrix Q^T on both sides of the equation, we have

$$\frac{1}{\omega^2} Q^T \bullet M \bullet Q \bullet \eta = Q^T \bullet M \bullet K^{-1} \bullet M \bullet Q \bullet \eta \qquad \text{(A 5.4-4)}$$

The Lanczos algorithm will result in a transformation matrix Q such that:

$$Q^T \bullet M \bullet Q = I \qquad \text{(A 5.4-5)}$$

$$Q^T \cdot M \cdot K^{-1} \cdot MQ = T \qquad \text{(A 5.4-6)}$$

where the matrix T is a symmetrical tri-diagonal matrix of the form:

$$\underline{T} = \begin{bmatrix} \alpha_1 & \beta_2 & & 0 \\ \beta_2 & \alpha_2 & \beta_3 & \\ & & & \beta_m \\ 0 & & \beta_m & \alpha_m \end{bmatrix} \qquad \text{(A 5.4-7)}$$

Consequently, the original eigenvalue problem (Equation A 5.4-2) is reduced to the following new eigenvalue problem:

$$\frac{1}{\omega^2} \eta = T \bullet \eta \qquad \text{(A 5.4-8)}$$

The eigenvalues in equation (5.4-8) can be calculated by the standard QL-method.

Through the MODAL SHAPE option, you can either select the number of modes to be extracted, or a range of modes to be extracted. The Sturm sequence check can be used to verify that all of the required eigenvalues have been found. In addition, you can select the lowest frequency to be extracted to be greater than zero.

The Lanczos procedure also allows you to restart the analysis at a later time and extract additional roots. It is unnecessary to recalculate previously obtained roots using this option.

Convergence Controls

Eigenvalue extraction is controlled by:

A. The maximum number of iterations per mode in the power sweep method; or the maximum number of iterations for all modes in the Lanczos iteration method,

B. An eigenvalue has converged when the difference between the eigenvalues in two consecutive sweeps divided by the eigenvalue is less than the tolerance, and

C. the Lanczos iteration method has converged when the normalized difference between all eigenvalues satisfies the tolerance. The maximum number of iterations and the tolerance are specified through the MODAL SHAPE history definitions.

Modal Stresses and Reactions

After the modal shapes (and frequencies) are extracted, the RECOVER load incrementation option allows for the recovery of stresses and reactions at a specified mode. This option can be repeated for any of the extracted modes. The stresses are computed from the modal displacement vector ϕ; the nodal reactions are calculated from $F = K\,\phi - \omega^2 M\phi$. The RECOVER option is also used to place eigenvectors on the POST tape.

Transient Analysis

Transient dynamic analysis deals with an initial-boundary value problem. In order to solve the equations of motion of a structural system, it is important to specify proper initial and boundary conditions. You obtain the solution to the equations of motion by using either modal superposition (for linear systems) or direct integration (for linear or nonlinear systems). In direct integration, selecting a proper time step is very important. For both methods, you can include damping in the system.

The sections that follow discuss the six aspects of transient analysis listed below.

- Modal Superposition
- Direct Integration
- Time Step Definition
- Initial Conditions
- Boundary Conditions
- Damping

Modal Superposition

The modal superposition method predicts the dynamic response of a linear structural system. In using this method, we assume that the dynamic response of the system can be expressed as a linear combination of the mode shapes of the system. For the principle of superposition to be valid, the structural system must be linear. Damping can be applied to each mode used in the superposition procedure.

To select the eigenvalue extraction method and the number of modal shapes, use the parameter DYNAMIC.

To select a fraction of critical damping for each mode, use the DAMPING model definition option.

To select the time step, use the history definition option DYNAMIC CHANGE.

The program obtains the transient response on the basis of eigenmodes extracted. The number of modes extracted (rather than the choice of time step) governs the accuracy of the solution.

Consider the general linear undamped problem

$$M \bullet \ddot{u} + K \bullet u = f(t) \qquad \text{(A 5.4-9)}$$

and suppose that n eigenvectors $\phi_1,\ldots, \phi_n$ are known.

$$u = \Sigma \phi_i u_i(t) \qquad \text{(A 5.4-10)}$$

The eigenmodes are orthogonal with respect to the M and K matrices. After substitution in Equation A 5.4-9, we find a set of uncoupled dynamic equations

$$m_i \ddot{u}_i + k_i u_i = f_i(t) \qquad \text{(A 5.4-11)}$$

where:

$$m_i = \phi_i^T M \phi_i \qquad \text{(A 5.4-12)}$$

$$k_i = \phi_i^T K \phi_i \qquad \text{(A 5.4-13)}$$

$$f_i(t) = \phi_i^T \bullet f(t) \qquad \text{(A 5.4-14)}$$

We can introduce damping on each mode as a fraction of critical damping (μ_i) for that mode. We may rewrite Equation A 5.4-11 in the form

$$m_i \ddot{u}_i + 2 m_i \omega_i \mu_i \dot{u}_i + k_i u_i = f_i(t) \qquad \text{(A 5.4-15)}$$

Equation A 5.4-11 or Equation A 5.4-15 can be solved exactly for $u_i(t)$. The response of the structure is obtained by use of Equation A 5.4-10.

Direct Integration

Direct integration is a numerical method for solving the equations of motion of a dynamic system. It is used for both linear and nonlinear problems. In nonlinear problems, the nonlinear effects may include geometric, material, and boundary nonlinearities. For transient analysis, the MARC program offers three direct integration operators listed below.

- Newmark-beta implicit
- Houbolt "
- Central difference explicit

To select the direct integration operator, use the DYNAMIC option. Specify the time step size through the DYNAMIC CHANGE or AUTO TIME option. Direct integration techniques are imprecise; this is true regardless of which technique you use. Each technique exhibits at least one of the following problems: conditional stability, artificial damping, and phase errors.

The *Newmark-beta operator* is probably the most popular direct integration method used in finite element analysis. For linear problems, it is unconditionally stable and exhibits no numerical damping. The Newmark-beta operator can effectively obtain solutions for linear and nonlinear problems for a wide range of loadings. The procedure allows for change of time step, so it can be used in problems where sudden impact makes a reduction of time step desirable. This operator may be used with adaptive time step control. While this method is stable for linear problems, instability may develop if nonlinearities occur. By reducing the time step and/ or adding (stiffness) damping, you can overcome these problems.

The *Houbolt operator* has the same unconditional stability as the Newmark-beta operator. In addition, it has strong numerical damping characteristics, particularly for higher frequencies. This strong damping makes the method very stable for nonlinear problems as well. In fact, stability increases with the time step size. The drawback of this high damping is that the

solution may become inaccurate for large time steps. Hence, the results obtained with the Houbolt operator usually have a smooth appearance, but are not necessarily accurate. The Houbolt integration operator, implemented in MARC as a fixed time step procedure, is particularly useful in obtaining a rough *scoping* solution to the problem.

The explicit *central difference operators* for IDYN=4 and IDYN=5 are only conditionally stable. The program automatically calculates the maximum allowable time step. This method is not very useful for shell or beam structures because the high frequencies result in a very small stability limit. This method is particularly useful for analysis of shock-type phenomena. In this procedure, since the operator matrix is a **diagonal mass matrix no inverse of operator matrix is needed.** However, this fact also implies that you cannot use this method in problems having degrees of freedom with zero mass. This *restriction* precludes use of the Herrmann elements, gap-friction elements, the pipe bend element, and Element 72. Element 72 is precluded because it has a rotational degree of freedom, which does not have an associated mass. The mass is updated only if the UPDATE option or the CONTACT option is used.

Technical Background

Consider the equations of motion of a structural system:

$$Ma + Cv + Ku = F \qquad \text{(A 5.4-16)}$$

where M, C, and K are mass, damping, and stiffness matrices, respectively, and a, v, u, and F are acceleration, velocity, displacement, and force vectors. Various direct integration operators can be used to integrate the equations of motion to obtain the dynamic response of the structural system. The technical background of the three direct integration operators available in MARC is described below.

Newmark-Beta Operator

The generalized form of the Newmark-beta operator is

$$u^{n+1} = u^n + \Delta t v^n + (1/2 - \beta)\,\Delta t^2 a^n + \beta \Delta t^2 a^{n+1} \qquad \text{(A 5.4-17)}$$

$$v^{n+1} = v^n + (1 - \gamma)\,\Delta t a^n + \gamma \Delta t a^{n+1} \qquad \text{(A 5.4-18)}$$

where superscript n denotes a value at the nth time step and u, v, and a take on their usual meanings.

The particular form of the dynamic equations corresponding to the trapezoidal rule

$$\gamma = 1/2, \quad \beta = 1/4$$

results in

$$\left(\frac{4}{\Delta t^2}M + \frac{2}{\Delta t}C + K\right)\Delta u = F^{n+1} - I^n + M\left(a^n + \frac{4}{\Delta t}v^n\right) + Cv^n \qquad \text{(A 5.4-19)}$$

where the internal force I is

$$I = \int \beta^T \sigma dV \qquad \text{(A 5.4-20)}$$

Equation A 5.4-19 allows implicit solution of the system

$$u^{n+1} = u^n + \Delta u \qquad \text{(A 5.4-21)}$$

Notice that the operator matrix includes K, the tangent stiffness matrix. Hence, any nonlinearity results in a reformulation of the operator matrix. Additionally, if the time step changes, this matrix must be recalculated because the operator matrix also depends on the time step. It is possible to change the values of γ and β through the DYNAMIC CHANGE option.

Houbolt Operator

The Houbolt operator is based on the use of a cubic fitted through three previous points and the current (unknown) in time. This results in the equations

$$v^{n+1} = \left(\frac{11}{6}u^{n+1} - 3u^n + \frac{3}{2}u^{n-1} - \frac{1}{3}u^{n-2}\right)/\Delta t \qquad \text{(A 5.4-22)}$$

and

$$a^{n+1} = \left(2u^{n+1} - 5u^n + 4u^{n-1} - u^{n-2}\right)/\Delta t^2 \qquad \text{(A 5.4-23)}$$

Substituting this into the equation of motion results in

$$\left(\frac{2}{\Delta t^2}M + \frac{11}{6\Delta t}C + K\right)\Delta u = F^{n+1} - I^n + \frac{1}{\Delta t^2}\left(3u^n - 4u^{n-1} + u^{n-2}\right)M + \frac{1}{\Delta t}\left(\frac{7}{6}u^n - \frac{3}{2}u^{n-1} + \frac{1}{3}u^{n-2}\right)C \qquad \text{(A 5.4-24)}$$

This equation provides an *implicit* solution scheme. By solving Equation A 5.4-21 for Δu, you obtain Equation A 5.4-25, and so obtain v^{n+1} and a^{n+1}.

$$u^{n+1} = u^n + \Delta u \qquad \text{(A 5.4-25)}$$

Equation A 5.4-24 is based on uniform time steps – errors occur when the time step is changed. Also, a special starting procedure is necessary since u^{n-1} and u^{n-2} appear in Equation A 5.4-24.

Central Difference Operator

The central difference operator assumes a quadratic variation in the displacement with respect to time.

$$a^n = \left(v^{n+1/2} - v^{n-1/2}\right)/(\Delta t) \qquad \text{(A 5.4-26)}$$

$$v^n = \left(u^{n+1/2} - u^{n-1/2}\right)/(\Delta t) \qquad \text{(A 5.4-27)}$$

so that

$$a^n = (\Delta u^{n+1} - (\Delta u)^n) / (\Delta t^2) \qquad \text{(A 5.4-28)}$$

where

$$\Delta u^n = u^n - u^{n-1} \qquad \text{(A 5.4-29)}$$

for IDYN=4:

$$\frac{M}{\Delta t^2}\Delta u^{n+1} = F^n - I^n + \frac{M}{\Delta t^2}\Delta u^n \qquad \text{(A 5.4-30)}$$

for IDYN=5:

$$\frac{M}{\Delta t^2}\Delta u^{n+1} = F^n - I^n + \frac{M}{\Delta t^2}\Delta u^n - CV_{n-\frac{1}{2}} \qquad \text{(A 5.4-31)}$$

Since the mass matrix is diagonal, no inverse of the operator matrix is needed. Also, since the operator is only conditionally stable, the critical time step is evaluated at the beginning of the analysis. For IDYN=4, the critical time step is computed by a power sweep for the highest mode in the system only at the beginning of the analysis. For IDYN=5, an approximated method based on element geometry is used to compute the highest eigenvalue. The critical time step is calculated for each time step whenever the UPDATE option is used in the analysis. The variable time step can be used only for IDYN=5. Unless there is significant distortion in an element, the change of critical time step will not be significant.

Time Step Definition

In a transient dynamic analysis, time step parameters are required for integration in time. The DYNAMIC CHANGE option may be used for either the modal superposition or the direct integration procedure. The AUTO TIME option may be used for the Newmark-beta operator to invoke the adaptive time control. Enter parameters to specify the time step size and period of time for this set of boundary conditions.

When using the Newmark-beta operator, decide which frequencies are important to the response. The time step in this method should not exceed 10 percent of the period of the highest relevant frequency in the structure. Otherwise, large phase errors will occur. The phenomenon usually associated with too large a time step is strong oscillatory accelerations. With even larger time steps, the velocities start oscillating. With still larger steps, the displacement eventually oscillates. In nonlinear problems, instability usually follows oscillation. When using adaptive dynamics, you should prescribe a maximum time step.

As in the Newmark-beta operator, the time step in Houbolt integration should not exceed 10 percent of the period of the highest frequency of interest. However, the Houbolt method not only causes phase errors, it also causes strong artificial damping. Therefore, high frequencies are damped out quickly and no obvious oscillations occur. It is, therefore, completely up to the engineer to determine whether the time step was adequate.

In nonlinear problems, the mode shapes and frequencies are strong functions of time because of plasticity and large displacement effects, so that the above guidelines may be only a coarse approximation. To obtain a more accurate estimate, repeat the analysis with a significantly different time step (1/5 to 1/10 of the original) and compare responses.

The central difference integration method is only conditionally stable; the program automatically calculates the stable time step. This step size yields accurate results for all practical problems.

Initial Conditions

In a transient dynamic analysis, you can specify initial conditions such as nodal displacements and/or nodal velocities. To enter initial conditions, use the following model definition cards: INITIAL DISP for specified nodal displacements, and INITIAL VEL for specified nodal velocities. As an alternative, you may use the user subroutine USINC.

Time-Dependent Boundary Conditions

Simple time-dependent load or displacement histories may be entered on data cards. However, in general cases with complex load histories, it is often more convenient to enter the history through a user subroutine. MARC allows the use of subroutines FORCDT and FORCEM for boundary conditions. Subroutine FORCDT allows you to specify the time-dependent incremental point loads and incremental displacements. Subroutine FORCEM allows you to specify the time-dependent magnitude of the distributed load.

Damping

In a transient dynamic analysis, damping represents the dissipation of energy in the structural system. It also retards the response of the structural system.

The MARC program allows you to enter two types of damping in a transient dynamic analysis: modal damping and Rayleigh damping. Use modal damping for the modal superposition method and Rayleigh damping for the direct integration method. For modal superposition, you may include damping associated with each mode. To do this, use the DAMPING option block to enter the fraction of critical damping to be used with each mode.

During time integration, the program associates the corresponding damping fraction with each mode. The program bases integration on the usual assumption that the damping matrix of the system is a linear combination of the mass and stiffness matrices, so that damping does not change the modes of the system.

For direct integration damping, you can specify the damping matrix as a linear combination of the mass and stiffness matrices of the system. You can specify damping coefficients on an element basis.

Stiffness damping should not be applied to either Herrmann elements or gap elements because of the presence of Lagrangian multipliers.

Numerical damping is used to damp out unwanted high-frequency chatter in the structure. If the time step is decreased (stiffness damping may cause too much damping), use the numerical damping option to make the damping (stiffness) coefficient proportional to the time step. Thus, if the time step decreases, high-frequency response can still be accurately represented. This

type of damping is particularly useful in problems where the characteristics of the model and/or the response change strongly during analysis (e.g., problems involving opening or closing gaps).

Element damping uses coefficients on the element matrices and is represented by the equation:

$$C = \sum_{i=1}^{n} \alpha_i M_i + (\beta_i + \gamma_i \frac{\Delta t}{\pi}) K_i \qquad \text{(A 5.4-32)}$$

where

C is the global damping matrix

M_i is the mass matrix of i^{th} element

K_i is the stiffness matrix of the i^{th} element

α_i is the mass damping coefficient on the i^{th} element

β_i is the usual stiffness damping coefficient on the i^{th} element

γ_i is the numerical damping coefficient on the i^{th} element

Δt is the time increment

If the same damping coefficients are used throughout the structure, Equation A 5.4-32 is equivalent to Rayleigh damping.

The damping coefficients associated with springs (stiffness and numerical damping) and with mass points (mass damping) are zero. The damping on elastic foundations is the same as the damping on the element on which the foundation is applied. For springs, a dashpot can be added for nonlinear analysis.

Harmonic Response

Harmonic response analysis allows you to analyze structures vibrating around an equilibrium state. This equilibrium state may be unstressed or statically prestressed. Statically prestressed equilibrium states may include material and/or geometric nonlinearities. You can compute the damped response for prestressed structures at various states.

In many practical applications, components are dynamically excited. These dynamic excitations are often harmonic and usually cause only small amplitude vibrations. MARC linearizes the problem around the equilibrium state. If the equilibrium state is a nonlinear, statically prestressed situation, MARC considers all effects of the nonlinear deformation on the dynamic solution. These effects include the following:

- initial stress
- change of geometry
- influence on constitutive law

The vibration problem can be solved as a linear problem using complex arithmetic.

The analytical procedure consists of the following steps:

1. MARC calculates the response of the structure to a static preload (which may be nonlinear) based on the constitutive equation for the material response. In this portion of the analysis, the program ignores inertial effects.
2. MARC calculates the complex-valued amplitudes of the superimposed response for each given frequency, and amplitude of the boundary tractions and/or displacements. In this portion of the analysis, the program considers both material behavior and inertial effects.
3. You can apply different loads with different frequencies or change the static preload at your discretion. All data relevant to the static response is stored during calculation of the complex response.

To initiate a harmonic response analysis, use the HARMONIC parameter. To define the excitation frequency, use the HARMONIC history definition option. If you enter the HARMONIC history definition option with a set of incremental data, the program assumes those incremental data apply only for the harmonic excitation. This is true for applied boundary conditions as well as loads.

The small amplitude vibration problem can be written with complex arithmetic as follows

$$[K + i\omega D - \omega^2 M]\,\bar{u} = \bar{P} \qquad \text{(A 5.4-33)}$$

where

$\bar{u}$ $= u_{re} + iu_{im}$ complex response vector
$\bar{P}$ $= P_{re} + iP_{im}$ complex load vector

The notation is further defined below:

$$K = \Sigma K_{el} + \Sigma K_{sp} \qquad \text{(A 5.4-34)}$$

where

K_{el} are element stiffness matrices
K_{sp} are the spring stiffness matrices

$$M = \Sigma M_{el} + \Sigma M_{mp} \qquad \text{(A 5.4-35)}$$

where

M_{el} are element mass matrices
M_{mp} are masspoint contributions

$$D = \Sigma D_{el} + \Sigma D_{d} + \alpha K + \beta M + 2\omega\gamma K \qquad \text{(A 5.4-36)}$$

where

D_{el} are element damping matrices
D_d are damper contributions
α stiffness damping coefficient
β mass damping coefficient
γ numerical damping coefficient
i $= \sqrt{-1}$
ω = excitation frequencies
u $= u_{re} + iu_{im}$ complex response vector
p $= P_{re} + iP_{im}$ complex load vector

If all external loads and forced displacements are in phase and the system is undamped, this equation reduces to

$$(K - \omega^2 M)\, u_{re} = P_{re} \qquad \text{(A 5.4-37)}$$

which could be solved without activating the complex arithmetic on the parameter option HARMONIC.

The values of the damping coefficients (α, β, γ) are entered via the model definition option DAMPING. The spring and damper contributions are entered in the model definition option SPRINGS and mass points are specified in model definition option MASSES.

The element damping matrix (D_{el}) can be obtained for any material with the use of a material damping matrix which is specified in the user-entered subroutine UCOMPL. You specify the material response with the constitutive equation.

$$\sigma = B\varepsilon + C\dot{\varepsilon} \qquad \text{(A 5.4-38)}$$

where B and C may be functions of deformation and/or frequency.

The global damping matrix is formed by the integrated triple product. The following equation is used:

$$D = \Sigma_{el} \int_{vel} \beta^T C dV_{el} \qquad \text{(A 5.4-39)}$$

where β is the strain-displacement relation.

Similarly, the stiffness matrix K is based on the elastic material matrix B. The program calculates the response of the system by solving the complex equations:

$$[K + i\omega D - \omega^2 M]\, \bar{u} = \bar{P} \qquad \text{(A 5.4-40)}$$

where $\bar{u}$ now is the complex response vector

$$\bar{u} = u_{re} + iu_{im} \qquad \text{(A 5.4-41)}$$

A special application of the harmonic excitation capability involves the use of the elastomeric analysis capability in MARC. Here, the Mooney formulation (used in conjunction with the various Herrmann elements) is used to model the stress-strain behavior of the elastomeric compound. In MARC, the behavior is derived from the third order invariant form of the strain energy density function.

$$W(I_1, I_2) = C_{10}(I_1 - 3) + C_{01}(I_2 - 3) + C_{11}(I_1 - 3)(I_2 - 3) + C_{20}(I_1 - 3)^2 + C_{30}(I_{11} - 3)^3 \quad \text{(A 5.4-42)}$$

with the incompressibility constraint

$$I_3 = 1 \quad \text{(A 5.4-43)}$$

where I_1, I_2, and I_3 are the invariants of the deformation. For the harmonic excitation, the constitutive equation has the specific form

$$\Delta\bar{S}_{ij} = [D_{ijkl} + 2i\omega\bar{\phi}_{ijkl}]\,\Delta\bar{E}_{kl} \quad \text{(A 5.4-44)}$$

with D_{ijkl} as the quasi-static moduli following form the MOONEY strain energy density function and

$$\begin{aligned}\bar{\phi}_{ijkl} = {} & \bar{\phi}_0[C^{-1}_{ik}C^{-1}_{jl} + C^{-1}_{il}C^{-1}_{jk}] + \bar{\phi}_1[\delta_{ik}C^{-1}_{jl} + C^{-1}_{il}\delta_{jk}] \\ & + \bar{\phi}_2[C_{ik}C^{-1}_{jl} + C^{-1}_{il}C_{jk}] + \bar{\phi}_0\delta_{ij}C^{-1}_{kl} + \bar{\phi}_{11}\delta_{ij}\delta_{kl} \\ & + \bar{\phi}_{12}\delta_{ij}C_{kl} + \bar{\phi}_{20}C_{ij}C^{-1}_{kl} + \bar{\phi}_{21}C_{ij}\delta_{kl} + \bar{\phi}_{22}C_{ij}C_{kl}\end{aligned} \quad \text{(A 5.4-45)}$$

The output of MARC consists of stresses, strains, displacements and reaction forces, all of which may be complex quantities. The strains are given by

$$\bar{\varepsilon} = \beta \bullet \bar{u} \quad \text{(A 5.4-46)}$$

and the stresses by

$$\sigma = B\bar{\varepsilon} + C\dot{\bar{\varepsilon}} \quad \text{(A 5.4-47)}$$

The reaction forces are calculated with

$$\bar{R} = K\bar{u} - \omega^2 M\bar{u} + i\omega\Sigma D_{el}\bar{u} + i\omega\Sigma D_d\bar{u} \quad \text{(A 5.4-48)}$$

where $-\omega^2 M\bar{u}$ is only included if requested on the HARMONIC parameter option.

The printout of the nodal values consists of the real and imaginary parts of the complex values, but you can request that the amplitude and phase angle be printed. You do this with the PRINT CHOICE model definition data option.

Spectrum Response

The spectrum response capability allows the user to obtain maximum response of a structure subjected to known spectral base excitation response. This is of particular importance in earthquake analysis and random vibration studies. You can use the spectrum response option at any point in a nonlinear analysis and, therefore, ascertain the influence of material nonlinearity or initial stress.

The spectrum response capability technique operates on the eigenmodes previously extracted to obtain the maximum nodal displacements, velocities, accelerations, and reaction forces. You can choose a subset of the total modes extracted by either specifying the lowest n modes or by selecting a range of frequencies.

Enter the displacement response spectrum $S_D(\omega)$ for a particular digitized value of damping through user subroutine USSD. The program performs the spectrum analysis based on the latest set of modes extracted. The program lumps the mass matrix to produce $\overline{M}$. It then obtains the projection of the inertia forces onto the mode ϕ_j

$$P_j = \overline{M} \bullet \phi_j \qquad \text{(A 5.4-49)}$$

The spectral displacement response for the j^{th} mode is

$$\alpha_j = S_D(\omega_j) \bullet P_j \qquad \text{(A 5.4-50)}$$

MARC then calculates the root-mean-square values as

$$u = [\Sigma_j (\alpha_j \phi_j)^2]^{1/2} \quad \text{DISPLACEMENT} \qquad \text{(A 5.4-51)}$$

$$v = [\Sigma_j (\alpha_j \omega_j \phi_j)^2]^{1/2} \quad \text{VELOCITY} \qquad \text{(A 5.4-52)}$$

$$a = [\Sigma_j (\alpha_j \omega_j \phi_j)^2]^{1/2} \quad \text{ACCELERATION} \qquad \text{(A 5.4-53)}$$

$$f = \left[\Sigma_j (\alpha_j \omega_j^2 \overline{M} \phi_j)^2\right]^{1/2} \quad \text{FORCE} \qquad \text{(A 5.4-54)}$$

A 5.5 HEAT TRANSFER

MARC contains a solid body heat transfer capability for one-, two-, and three-dimensional, steady-state and transient analyses. This capability allows you to obtain temperature distributions in a structure for both linear or nonlinear heat transfer problems. The nonlinearities in the problem may include temperature-dependent properties, latent heat (phase change) effect, heat convection in the flow direction, and nonlinear boundary conditions (convection and radiation). The temperature distributions can, in turn, be used to generate thermal loads in a stress analysis.

MARC can be applied to solve the full range of two- and three-dimensional transient and steady-state heat conduction and heat convection problems. The MARC program provides heat transfer elements that are compatible with stress elements. Consequently, the same mesh can be used for both the heat transfer and stress analyses. Transient heat transfer is an initial-boundary value problem, so proper initial and boundary conditions must be prescribed to the problem in order to obtain a realistic solution. The MARC program accepts nonuniform nodal temperature distribution as the initial condition, and can handle temperature/time-dependent boundary conditions. Both the thermal conductivity and the specific heat in the problem can be dependent on temperature; however, the mass density remains constant at all times. The thermal conductivity can also be anisotropic. Latent heat effects (solid-to-solid, solid-to-liquid phase changes) can be included in the analysis. A time-stepping procedure is available for transient heat transfer analysis. Temperature histories can be stored on a tape (post tape) and used directly as thermal loads in subsequent stress analysis. User subroutines are available for complex boundary conditions such as nonlinear heat flux, convection, and radiation.

In addition, a number of thermal contact gap and fluid channel elements are available in the MARC program. These elements can be used for heat transfer problems involving thermal contact gap and fluid channel conditions.

The thermal contact elements provide perfect conduction or radiation/convection between surfaces of the thermal contact gap, depending on the surface temperatures. The perfect conduction capability in the elements allows for the enforcement of equal temperatures at nodal pairs and the radiation/convection capability allows for nonlinear heat conduction between surfaces, depending on film coefficient and emissivity. The perfect conduction is simulated by applying a tying constraint on temperatures of the corresponding nodal points and an automatic tying procedure has been developed for these elements. The radiation/convection capabilities in the elements are modeled by one-dimensional heat transfer in the thickness direction of the elements with variable thermal conductivity.

For the purpose of cooling, channels allowing coolant to flow through often appear in the solid. The fluid channel elements are designed for the simulation of one-dimensional fluid/solid convection conditions based on the following assumptions:

1. Heat conduction in the flow direction can be neglected compared to heat convection;
2. The heat flux associated with transient effects in the fluid (changes in fluid temperature at a fixed point in space) can be neglected.

For high velocity air flow, these assumptions are reasonable and the following approach is used:

Each cooling channel is modeled using channel elements. On the sides of the channel elements, convection is applied automatically. The film coefficient is equal to the film coefficient between fluid and solid, whereas the sink temperature represents the temperature of the fluid. A two-step staggered solution procedure is used to solve for the weakly coupled fluid and solid temperatures.

A summary of the MARC program capabilities for transient and steady-state analysis is given below.

- Selection of the following elements that are compatible with stress analysis:

  1-D: three dimensional link (2-node, 3-node)
  2-D: planar and axisymmetric element (3-, 4-, and 8-node)
  3-D: brick elements (8- and 20-node)
  reduced integration elements
  shell elements

- Specification of temperature-dependent materials (including latent heat effects) is performed with the model definition options ISOTROPIC, ORTHOTROPIC, TEMPERATURE EFFECTS, and ORTH TEMP.
- Selection of initial conditions is done using the INITIAL TEMPERATURE option.
- Selection of the temperature and time-dependent boundary conditions (prescribed temperature history, volumetric flux, surface flux, film coefficients, radiation, change of prescribed temperature boundary conditions during analyses) is done using the FIXED TEMP, TEMP CHANGE, DIST FLUXES, POINT FLUX, and FILM options.
- Selection of time steps using history definition option TRANSIENT.
- Application of a tying constraints on nodal temperatures using model definition option TYING.
- Generation of a thermal load (temperature) tape using the POST option, which can be directly interfaced with the stress analysis using model definition option CHANGE STATE.
- Use of user subroutines ANKOND for anisotropic thermal conductivity, FILM for convective and radiative boundary conditions or FLUX for heat flux boundary conditions.
- Selection of nodal velocity vectors for heat convection is done using VELOCITY and VELOCITY CHANGE options.
- Use of user subroutines UVELOC for heat convection.

Steady State Analysis

For steady state problems, use the STEADY STATE load incrementation option. If the problem is nonlinear, use the tolerance for property recycling on the CONTROL model definition set to obtain an accurate solution.

You must distinguish between two cases of nonlinearity in steady-state solutions: mild nonlinearities and severe nonlinearities. In the case of mild nonlinearities, variations are small in properties, film coefficients, etc., with respect to temperatures. The steady-state solution can be obtained by iteration. After a small number of iterations, the solution should converge.

The technique described above is not suitable for severe nonlinearities. Examples of severe nonlinearities are radiation boundary conditions and internal phase-change boundaries. In these cases, you must track the transient with a sufficiently small time step (Δt) to retain stability until the steady-state solution is reached. Clearly, the choice of Δt is dependent on the severity of the nonlinearity. The number of steps necessary to obtain the steady-state solution can often be reduced by judicious choice of initial conditions. The closer the initial temperatures are to the steady-state, the fewer the number of increments necessary to reach steady-state.

Transient Analysis

An automatic time stepping scheme can be used to adjust the time step for each increment. Use the TRANSIENT history definition option. The automatic time stepping scheme is based on a maximum allowable temperature change per step which is given on the CONTROL option. The scheme is as follows: after the program obtains a solution for a step, it calculates the maximum temperature change in the step. It checks this value against the specified control value. If the actual maximum change exceeds the specified value, the program repeats the step with a smaller time step -- it continues repeating this step until the maximum temperature change is smaller than the specified value or until the maximum number of recycles given on the control option is reached (in which case the program stops). If the actual maximum temperature change is between 80 percent and 100 percent of the specified value, the program goes on to the next step, using the same time step. If the actual maximum is between 65 percent and 80 percent of the specified value, the program tries the next step with a time step of 1.25 times the current step. If the actual maximum is below 65 percent of the specified value, the program tries the next step with a time step that is 1.5 times the current step. The purpose of the scheme is to increase the time step as the analysis proceeds.

You can switch off the automatic time stepping scheme, and force the program to step through the transient with a fixed time step you specify.

Convergence Control

Use model definition option CONTROL along with the automatic time stepping scheme to input convergence controls in transient heat transfer analysis. These options are:

- Maximum number of time steps in this run.
- Maximum nodal temperature change allowed. This is used only if the automatic time stepping scheme is used.
- Maximum nodal temperature change allowed before properties are re-evaluated and matrices reassembled.
- Maximum error in temperature estimate used for property evaluation. This control provides a recycling capability to improve accuracy in highly nonlinear heat transfer problems (for instance, for latent heat or radiation boundary conditions).

Temperature Effects

Both the thermal conductivity and specific heat in a heat transfer analysis can depend on temperatures; however, the mass density remains constant. Specify reference temperature values of thermal conductivity, specific heat and mass density with the ISOTROPIC option. Enter temperature variations of both the thermal conductivity and specific heat using the TEMPERATURE EFFECTS option. This option also allows input of latent heat information. The temperature- dependent data can be entered using either the slope-break point representation or the property vs. temperature representation. During analysis, an extrapolated/interpolated averaging procedure is used for the evaluation of temperature-dependent properties.

Latent heat can be induced because of a phase chance that can be characterized as solid-to-solid, solid-to-fluid, fluid-to-solid, or a combination of the above, depending on the nature of the process.

Phase change is a complex material behavior; a detailed modeling of this change of material characteristic is generally very difficult. The use of numerical models to simulate these important phenomena is possible; several major factors associated with phase change of certain materials have been studied numerically.

The basic assumption of the latent heat option in the MARC program is that the latent heat is uniformly released in a temperature range between solidus and liquidus temperatures of the materials (see Figure A 5.5-1). MARC uses a modified specific heat to model the latent heat effect. If the experimental data are sufficient and available, a direct input of the temperature-dependent specific heat data (see Figure A 5.5-2) may be used. Results of both approaches are comparable if the temperature increments are relatively small.

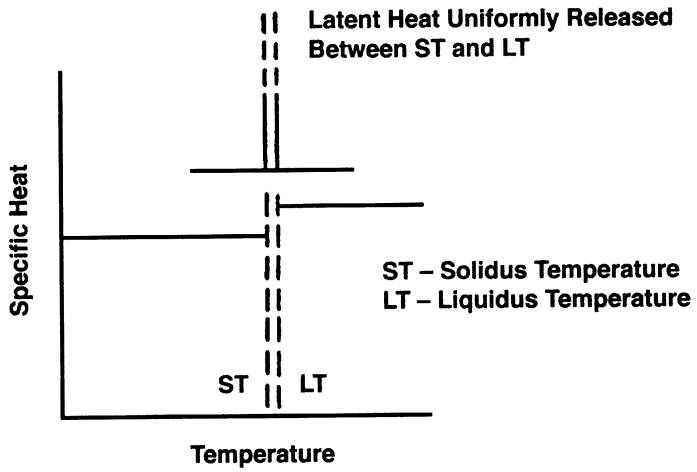


Figure A 5.5-1 Modeling Phase Changes with the Latent Heat Option

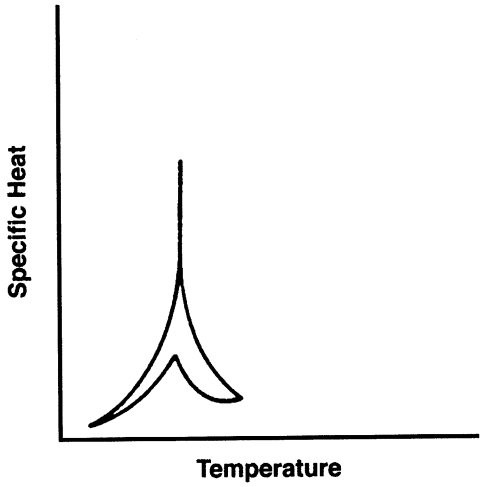


Figure A 5.5-2 Modeling Phase Changes with the Specific Heat Option

Initial Conditions

In a transient heat transfer analysis, the MARC program accepts nonuniform nodal temperature distribution as the initial condition. Enter the initial condition through the model definition option INITIAL TEMP or through user subroutine USINC. Initial conditions are not required in steady-state heat transfer analysis, even though they may improve convergence when temperature-dependent properties are included.

Boundary Conditions

There are two types of boundary conditions in transient/steady-state heat transfer analysis: prescribed nodal temperatures and nodal/element heat fluxes. These boundary conditions are entered directly through input cards or through user subroutines.

Prescribe nodal temperatures at boundary nodes using the model definition option FIXED TEMPERATURE. Prescribe time-dependent nodal temperatures through user subroutine FORCDT. You may change prescribed nodal temperatures with the TEMP CHANGE option.

The model definition option POINT FLUX allows you to enter constant concentrated (nodal) heat fluxes such as heat source and heat sink. Use the user subroutine FORCDT for time and/or temperature-dependent nodal fluxes.

Use the model definition option DIST FLUXES and the user subroutine FLUX for constant and time/temperature-dependent distributed (surface or volumetric) heat fluxes.

NOTE

In heat transfer analysis, you must always specify total values; for example, total temperature boundary conditions or total fluxes.

Use the model definition option FILMS to input the constant film coefficient and the ambient temperature associated with the convective boundary conditions. Use the user subroutine FILM for time/temperature-dependent convective boundary conditions. The expression of the convective boundary condition is

$$q = H(T_s - T_\infty) \qquad \text{(A 5.5-55)}$$

where q, H, T_s, and T_∞ are heat flux, film coefficient, unknown surface temperature, and ambient temperature, respectively.

The radiative boundary condition can be expressed as

$$q = \sigma\varepsilon(T_s^4 - T_\infty^4) \qquad \text{(A 5.5-56)}$$

where q is the heat flux, σ is the Stefan-Boltzmann coefficient, ε is emissivity, and T_s and T_∞ are unknown surface and ambient temperatures, respectively. The radiative boundary condition can be rewritten as

$$\begin{aligned} q &= \sigma\varepsilon\,(T_s^3 + T_s^2 T_\infty + T_s T_\infty^2 + T_\infty^3)\,(T_s - T_\infty) \\ &= H\,(\sigma, \varepsilon, T_s, T_\infty)\,(T_s - T_\infty) \end{aligned} \qquad \text{(A 5.5-57)}$$

This shows that the radiative boundary condition is equivalent to a nonlinear convective boundary condition, in which the equivalent film coefficient H (σ, ε, T_s, T_∞) depends on the unknown surface temperature T_s. This case requires the user subroutine FILM.

One important element in radiation is the view factor calculations which are generally tedious and non-trivial. In the MARC program, view factors will be automatically calculated for the user, in each cavity of an axisymmetric body involving radiative heat transfer. This capability is not yet available to two- dimensional plane and three-dimensional bodies. The parameter option RADIATION is used to activate this view factor calculation capability; the model definition option RADIATING CAVITY allows the user to input outlines of each cavity in terms of nodal numbers.

The user must subdivide the radiative boundary in his heat transfer problem into one or more cavities. For each cavity, the user defines the outline of the cavity in terms of an ordered sequence of nodes. Usually, the nodes coincide with the nodes of the finite element mesh. The user may add extra nodes, provided he also gives the appropriate boundary conditions.

The nodes must be given in counter-clockwise order with respect to an axis orthogonal to the plane of the figure and pointing to the viewer (see Figure A 5.5-1). If the cavity is not closed, the program adds the last side, by connecting the last node with the first one. This side is treated as a black body as far as radiation is concerned; its temperature is taken as the average between the temperatures of the adjacent nodes.

The program internally computes the view factor between every side of the cavity and all other sides. The matrix with the view factors can be stored onto a file, and read in again during a subsequent analysis, thus avoiding a new computation. During a transient heat transfer analysis, for every time step, the program estimates the temperature reached at the end of the step. From the estimated temperature, the properties of the materials (temperature dependent) are computed. In addition, the radiating heat fluxes are computed.

The temperatures at the end of the step are computed by solving the finite element equations.

At every node, the difference between estimated and computed temperature is derived. If the tolerance allowed by the model definition option CONTROL is exceeded, iterations within the time step will take place. Otherwise, the computation of the step is concluded.

The cavity is defined by its boundary, a list of nodes ordered counter-clockwise.

Insulated boundary condition (for example, symmetry boundaries) requires that the sum of the heat fluxes at a node be zero. This requirement is satisfied automatically. Therefore, no input is required for this type of boundary condition.

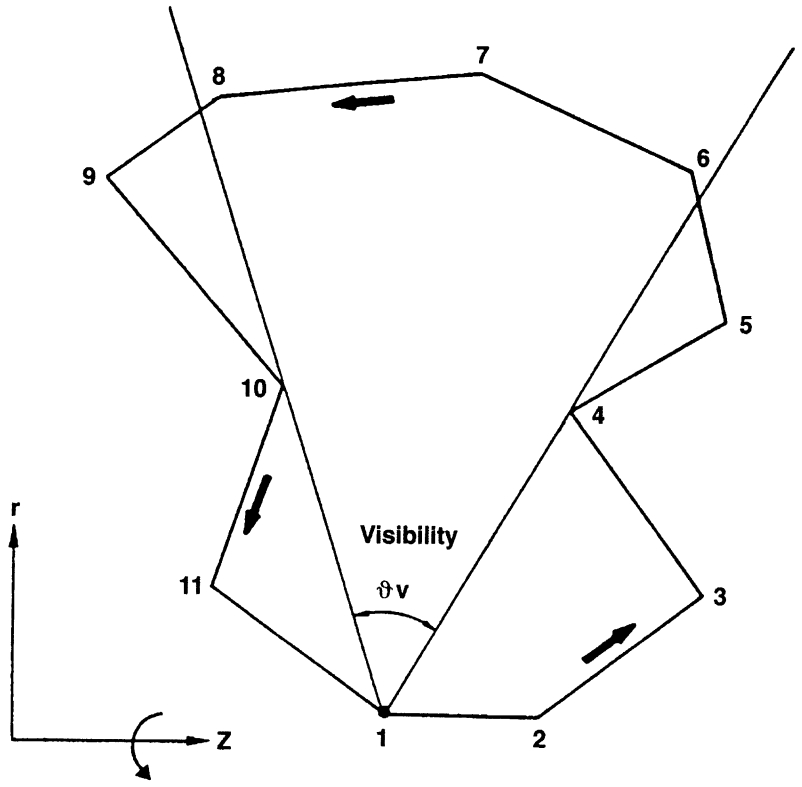


Figure A 5.5-3 Radiating Cavity (Axisymmetric)

Output

The program prints out both the nodal temperatures and the temperatures at the element centroid when the CENTROID option is used, or at the integration points if the ALL POINTS option is invoked. You can also indicate on the HEAT parameter option for the program to print out the temperature gradients ∇T and the resulting nodal fluxes.

To create a file of element and nodal point temperatures, use the POST model definition option. This file may be used as temperature input for performing a thermal stress analysis. This file would be processed using the CHANGE STATE option in the subsequent thermal stress analysis. This POST tape may also interface with the Mentat program to plot temperature as a function of time.

Technical Background

Let the temperature T(x) within an element be interpolated from the nodal values T of the element through the interpolation functions N(x),

$$T(x) = N(x)\,T \qquad \text{(A 5.5-58)}$$

The governing equation of the heat transfer problem is

$$C(T)\,\dot{T} + K(T)\,T = Q \qquad \text{(A 5.5-59)}$$

In Equation A 5.5-59, C(T) and K(T) are the temperature-dependent heat capacity and thermal conductivity matrices, respectively, T is the nodal temperature vector, $\dot{T}$ is the time derivative of the temperature vector, and Q is the heat flux vector.

The selection of the backward difference scheme for the discretization of the time variable in Equation A 5.5-59 yields the following expression:

$$\left[\frac{1}{\Delta t}C(T) + K(T)\right]T_n = Q_n + \frac{1}{\Delta t}C(T)\,T_{n-1} \qquad \text{(A 5.5-60)}$$

Equation A 5.5-60 computes nodal temperatures for each time increment Δt.

For the evaluation of temperature-dependent matrices, the temperatures at two previous steps provide a linear (extrapolated) temperature description over the desired interval

$$T(\tau) = T(t-\Delta t) + \frac{\tau}{\Delta t}T(t-\Delta t) - T(t-2\Delta t) \qquad \text{(A 5.5-61)}$$

This temperature is then used to obtain an average property of the material, f, over the interval to be used in Equation A 5.5-60, such that

$$f = \frac{1}{\Delta t}\int_{t-\Delta t}^{t} f(T(\tau))\,d\tau \qquad \text{(A 5.5-62)}$$

During iteration, the average property is obtained based on the results of the previous iteration.

Heat Transfer with Convection

The MARC program has the capability to perform heat transfer with convection if the velocity field is known. The numerical solutions of the convection-diffusion equation have been developed in recent years. The streamline-upwind Petro-Galerkin (SUPG) method has been implemented into the MARC heat transfer capability.

The elements which are available are described in Table A 5.5-1.

Table A 5.5-1 Heat Transfer Convection Elements

Element Type	Description
36, 65	2-, 3-node link
37, 39, 41, 50, 69, 85, 86	3-, 4-, 8-node planar
38, 40, 42, 70, 87, 88, 122	4-, 4-, 8-node axisymmetric
43, 44, 71, 123, 133	8-, 20-node hexahedron
135	4-node tetrahedral
133	10-node tetrahedron
131, 132	6-node triangular

To activate the convection contribution, use the HEAT parameter and set the fifth field to 2. Due to the nonsymmetric nature of the convection term, the nonsymmetric solver is used automatically. Specify the nodal velocity vectors using the VELOCITY option. To change velocity, use VELOCITY CHANGE. If nonuniform velocity vectors are required, user subroutine UVELOC is used. This capability may be used in conjunction with the with RIGID Perfectly-Plastic Flow in Section A 5.6 to perform a coupled analysis, in which the velocity fields are obtained.

Technical Background

The general convection-diffusion equation is:

$$\frac{\partial T}{\partial t} + v \bullet \nabla T = \nabla \bullet \kappa \nabla T + Q \qquad \text{(A 5.5-63)}$$

The perturbation weighting functions are introduced as:

$$W = N + \alpha \bullet \left(\frac{h}{2|v|} v \bullet \nabla N\right) + \beta \bullet \left(\frac{h}{4|v|} \Delta t v \bullet \nabla N\right) \qquad \text{(A 5.5-64)}$$

N is the standard interpolation function in Equation A 5.5-58. Alfa term (α), so-called upwinding parameter is the weighting used to eliminate artificial diffusion of the solution; while the beta term (β) is to avoid numerical dispersion. $|v|$ is the magnitude of local velocity vectors. T is the temperature, κ is the diffusion tensor. Q is the source term and Δt is the time increment.

The optimal choice for α and β are:

α = coth(Peclet/2) - (Peclet2)
β = C/3 - (2/Peclet) * (α/C)

where Peclet is the local Peclet number in the local element and C is the local Courant number:

Peclet = density * specific heat * characteristic length * magnitude of the fluid velocity/conductivity

Peclet = $\rho * c * h * |v| / k$

and

C = $|v| * (\Delta t)/h$ where (Δt) is the time increment.

The characteristic length h is defined in Reference 24 where $C \leq 1$ is required for numerical stability. When $C > 1$, the β is set to be zero and a large time step is recommended to avoid numerical dispersion.

NOTE

The interpolation function N is not the time-space functions defined in Reference 3, so that most MARC heat transfer elements can be used. The convection contribution of heat transfer shell elements is limited due to the definitions of the perturbation weighting function and the interpolation function.

Radiating Cavities

As previously mentioned, in a heat transfer analysis of axisymmetric body involving radiative boundary conditions, the program automatically calculates view factors for radiation. A description of the view factor calculation follows.

The amount of radiation exchanged between two surfaces will depend upon what fraction of the radiation from each surface impinges the other surfaces. Referring to Figure A 5.5-3, the radiation propagating from surface i to surface j will be:

$$q_{ij} = J_i A_i \left(\frac{\cos\phi_i \cos\phi_j A_j}{r^2} \right) = J_i A_i F_{ij} \qquad \text{(A 5.5-65)}$$

The term F_{ij} is solely geometrical in nature. For convenience, it is called the view factor F_{ij}. From the definition of F_{ij} we see that

$$A_j F_{ji} = A_i F_{ij} \qquad \text{(A 5.5-66)}$$

We are now ready to derive the heat transfer radiation equation. Steady-state (equilibrium) energy conservation requires:

$$q_i = A_i (J_i - G_i) \qquad \text{(A 5.5-67)}$$

Two independent expressions for G_i can be formed.

1. The incoming radiation on a surface must equal the radiation emitted by all other surfaces which strikes this surface,

$$A_i G_i = \Sigma_j J_j A_j F_{ji} = \Sigma_j J_j A_i F_{ij}$$

or (A 5.5-68)

$$G_i = \Sigma_j J_j F_{ij}$$

2. The other expression for G_i is

$$G_i = \frac{J_i - \varepsilon_i E_{n_i}}{\rho_i} = \frac{J_i - \varepsilon_i E_{n_i}}{1 - \varepsilon_i} \qquad \text{(A 5.5-69)}$$

where

E_n is the emissive power
ε is the emissivity
ρ is the reflectance.

Substituting Equation A 5.5-69 into Equation A 5.5-67 and rearranging it gives

$$J_i = E_{n_i} - \frac{1 - \varepsilon_i}{A_i \varepsilon_i} q_i \qquad \text{(A 5.5-70)}$$

This expression for J_i can be inserted into Equation A 5.5-68. After regrouping terms, one gets the governing equation for gray body diffuse radiation problems

$$\left[\frac{A_i\delta_i^j - (1-\varepsilon_j)\,A_iF_{ij}}{A_j\varepsilon_j}\right]\{q_j\} = [A_i\delta_i^j - A_iF_{ij}]\,\{E_{n_i}\} \qquad \text{(A 5.5-71)}$$

For the problem of black bodies, i.e., $\varepsilon = 1$, we have,

$$\{q_j\} = [A_i\delta_i^j - A_iF_{ij}]\,\{E_{n_i}\} \qquad \text{(A 5.5-72)}$$

This equation states the obvious; net radiation heat flow from a black surface is the difference between radiation given off and received; i.e., there is no reflection.

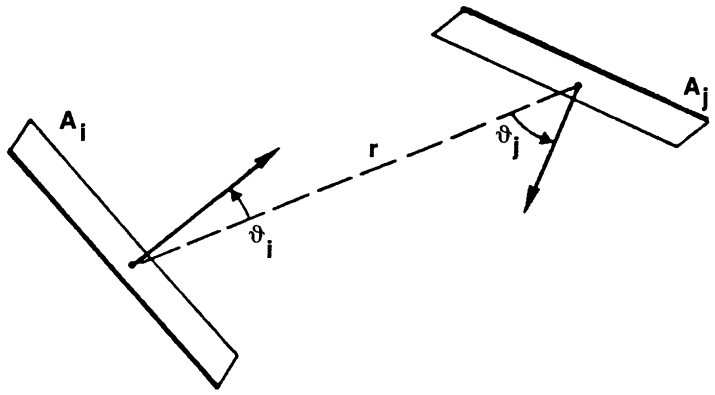


Figure A 5.5-4 View Factor Definition

The heat flux radiating from A_i to A_j is computed as

$$q_{ij} = J_i\,(A_i\cos\theta_i)\left(\frac{A_j\cos\theta_j}{\pi r^2}\right) \qquad \text{(A 5.5-73)}$$

where J_i is the power radiating from A_i, the first term within parentheses is the projection of A_i normal to the connecting line, and the second term is the solid angle under which A_j is seen from the center of A_i defining the viewfactor:

$$F_{ij} = \frac{\cos\theta_i\cos\theta_i}{\pi r^2}A_j \qquad \text{(A 5.5-74)}$$

$$q_{ij} = J_iA_iF_{ij}$$

Radiation-Gap

For the thermal contact gap element, in the *gap open* condition, two surface temperatures T_a and T_b at the centroid of the surfaces of a thermal contact element are obtained by interpolation from the nodal temperatures. These two surface temperatures are used for the computation of an equivalent conductivity for the radiation/convection link. The expression of the equivalent thermal conductivity k_1 is:

$$k_1 = \varepsilon \bullet \sigma \bullet L \bullet (T_{ka} + T_{kb})\,(T_{ka}^2 + T_{kb}^2) + H \bullet L \qquad \text{(A 5.5-75)}$$

where ε is the emissivity, σ is the Stefan-Boltzmann constant, L is the length of the element (distance between a and b), T_{ka}, T_{kb} are absolute temperatures at a and b converted from T_a and T_b; and H is the constant film coefficient.

The equivalent thermal conductivity k_1 for the thermal contact element is assumed to be in the gap direction. The thermal conductivities in other two local directions are all set to zero. A coordinate transformation from the local to the global coordinate system allows the generation of the thermal conductivity matrix of the thermal contact element in the global system for assembly.

Similarly, in the gap closed condition, tying constraints are automatically generated by the program for thermal contact elements. The constraint equation for each pair of nodes can be expressed as:

$$T_I + T_J \quad \text{if} \quad (T_{gap} > T_{close}) \qquad \text{(A 5.5-76)}$$

where

$$T_{gap} = \frac{1}{2}(T_I + T_J)$$

T_I, T_J = nodal temperatures at nodes I and J

T_{close} = gap closure temperature.

Channel

For the fluid channel element, the one-dimensional, steady-state, convective heat transfer in the fluid channel can be expressed as:

$$\dot{m}c\frac{\partial T_f}{\partial s} + \Gamma h\,(T_f - T_s) = 0$$
$$T_f(0) = T_{inlet} \qquad \text{(A 5.5-77)}$$

where $\dot{m}$ is mass flow rate, c is specific heat, T_f is fluid temperature, T_s is solid temperature, s is streamline coordinate, Γ is circumference of channel, h is film coefficient, and T_{inlet} is inlet temperature.

Similarly, the conductive heat transfer in the solid region is governed by the following equation:

$$C\dot{T}_s + KT_s = Q \qquad \text{(A 5.5-78)}$$

subjected to given initial condition and fixed temperature and/or flux boundary conditions. At the interface between the fluid and solid, the heat flux estimated from convective heat transfer is

$$q = h\,(T_s - T_f) \qquad \text{(A 5.5-79)}$$

In Equation A 5.5-78, C is the heat capacity matrix, K is the conductivity matrix and Q is the heat flux vector. Equation A 5.5-78 and Equation A 5.5-79 are coupled equations. The coupling is due to the unknown solid temperature T_s appearing in Equation A 5.5-77 and unknown fluid temperature T_f in Equation A 5.5-79 for the solution of Equation A 5.5-78.

The solutions for Equation A 5.5-77 and Equation A 5.5-78 are obtained from the introduction of a backward difference for the discretization of time variable in Equation A 5.5-78 and of streamline distance in Equation A 5.5-77. Let

$$\dot{T}_s = [T_s^i - T_s^{i-1}] / (\Delta t) \qquad \text{(A 5.5-80)}$$

we obtain

$$(\frac{1}{\Delta t}C + K)\,T_s^i = Q^i + \frac{1}{\Delta t}CT_s^{i-1} \qquad \text{(A 5.5-81)}$$

where Δt = time-step in transient analysis. Similarly, let

$$\frac{dT_f}{ds} = [T_f - T_f^{j-1}] / (\Delta s) \qquad \text{(A 5.5-82)}$$

we obtain

$$T_f^j = [\Delta s \bullet \beta + T_f^{j-1}] / (1 + \Delta s \bullet \alpha) \qquad \text{(A 5.5-83)}$$

where α is the streamline increment,

$$\alpha = \Gamma \bullet h / (\dot{m}c); \quad \text{and} \quad \beta = \Gamma \bullet h \bullet T_s^{j-1} / (\dot{m}c) \qquad \text{(A 5.5-84)}$$

A 5.6 RIGID-PLASTIC FLOW

The rigid-plastic flow analysis is an approach to large deformation analysis which can be used for metal forming problems. Two formulations are available: an Eulerian (steady state) and Lagrangian (transient) approach. The effects of elasticity are not included. If these effects are important, this option should not be used.

In the steady state approach, the velocity field (and stress field) is obtained as the solution of a steady-state flow analysis. The time period is considered as 1.0 and, hence, the velocity is equal to the deformation. In the transient formulation, the incremental displacement is calculated.

The R-P FLOW parameter invokes the rigid-plastic procedure. This procedure needs to enforce the incompressibility condition, which is inherent to the strictly plastic type of material response being considered.

Incompressibility may be imposed in three ways:

1. by means of Lagrange multipliers. Such procedure requires Herrmann elements which have a pressure variable as the Lagrange multiplier.
2. by means of penalty functions. This procedure uses regular solid elements, and adds penalty terms to any volumetric strain rate that develops. It is highly recommended that the constant dilatation formulation be used – by entering a nonzero value in the second field of the GEOMETRY section.
   a. A constant penalty is used when the bulk modulus is defined through the ISOTROPIC option.
   b. A variable penalty is used based upon the current value of the flow stress if the bulk modulus is not entered.
3. in plane stress analysis (shell and membrane elements), the incompressibility constraint is satisfied exactly by updating the thickness. This capability is not available for steady state analysis.

In R-P flow analysis, several iterations are required at any given increment, the greatest number occurring in the first increment. Subsequent increments require fewer iterations, since the initial iteration can make use of the solution from the previous increment.

Due to the simplicity of the rigid-plastic formulation, it is possible to bypass stress recovery for all iterations but the last in each increment, provided that displacement control is used. In such cases, considerable savings in execution time are achieved. If nodal based friction is used in a contact analysis, then a stress recovery is always performed after each iteration.

Steady State Analysis

The steady state R-P flow formulation is based on an Eulerian reference system. For problems in which a steady-state solution is not appropriate, an alternative method is available to update the coordinates. User subroutine UPNOD is used to update the nodal coordinates at the end of a step according to the relation

$$x_i^n = x_i^{n-1} + \dot{u}_i^n \Delta t \qquad \text{(A 5.6-1)}$$

where n refers to the step number, $\dot{u}^n$ is the nodal velocity components, and Δt is an arbitrary time step. Δt is selected in such a way as to allow only a reasonable change in mesh shape while ensuring stability with each step.

Updating the mesh requires judicious selection of a time step. This requires some knowledge of the magnitude of the nodal velocities that will be encountered. The time step should be selected such that the strain increment is never more than one percent for any given increment.

The quantities under the title of STRAIN in the printouts actually refer to the strain rate at an element integration point. The reaction forces output by the program give the limit loads on the structure.

Transient Analysis

In the transient procedure, there will be an automatic updating of the mesh at the end of each increment. During the analysis, the updated mesh may exhibit severe distortion and the solution may be unable to converge. Mesh rezoning may be used to overcome this difficulty.

Technical Background

The rigid-plastic flow capability is based on iteration for the velocity field in an incompressible, non-Newtonian fluid. The normal flow condition for a nonzero strain rate can be expressed as

$$\sigma'_{ij} = \left(\frac{2}{3}\frac{\bar{\sigma}}{\dot{\bar{\varepsilon}}}\right)\dot{\varepsilon}_{ij} = \mu\left(\dot{\bar{\varepsilon}}\right)\dot{\varepsilon}_{ij} \qquad \text{(A 5.6-2)}$$

where $\dot{\bar{\varepsilon}}$

$$\bar{\varepsilon} = \sqrt{2/3\dot{\varepsilon}_{ij}\dot{\varepsilon}_{ij}} \qquad \text{(A 5.6-3)}$$

is the equivalent strain rate, $\bar{\sigma}$ is the yield stress (which may be rate-dependent) and

$$\bar{\sigma}'_{ij} = \sigma_{ij} - \frac{1}{3}\delta_{ij}\sigma_{kk} \qquad \text{(A 5.6-4)}$$

gives the deviatoric stress.

The effective viscosity is evaluated as:

$$\frac{2}{3}\frac{\bar{\sigma}}{\dot{\bar{\varepsilon}}} \qquad \text{(A 5.6-5)}$$

Note that as $\dot{\bar{\varepsilon}} \rightarrow 0$, $\mu \rightarrow \infty$. A cut-off value of strain rate is used in the program to avoid this difficulty. An initial value for $\dot{\bar{\varepsilon}}$ is necessary to start the iterations. These values may be specified either in the CONTROL or the CONTACT option. The default cut-off value is 10^{-12}, and the default initial strain rate value is 10^{-4}.

The value of the flow stress is dependent upon both the equivalent strain, the equivalent strain rate, and the temperature. This dependence may be given through the WORK HARD, STRAIN RATE, and TEMPERATURE EFFECTS options, respectively. For steady state analysis, user subroutine UNEWTN may be used to define a viscosity. In this manner, a non-Newtonian flow analysis may be performed. For the steady state procedure, user subroutine URPFLO may be used to define the flow stress.

A 5.7 HYDRODYNAMIC BEARING

The MARC program has a hydrodynamic bearing analysis capability, which enables you to solve lubrication problems. This capability makes it possible to model a broad range of practical bearing geometries and to calculate various bearing characteristics such as load carrying capability, stiffness, and damping properties. It can also be used to analyze elasto-hydrodynamic problems.

The lubricant flow in hydrodynamic bearings is governed by the Reynolds equation. The bearing analysis capability has been implemented into the MARC program to determine the pressure distribution and mass flow in bearing systems.

MARC is capable of solving steady-state lubrication problems; the incremental procedure analyzes a sequence of different lubricant film profiles. The MARC program also can be used to solve coupled elasto-hydrodynamic problems. This analysis requires a step-by-step solution for both the lubrication and the stress problems using separate runs. Because the finite element meshes for each problem are different, the program does not contain an automated coupling feature.

Only one-dimensional or two-dimensional lubricant flow needs to be modeled, since no pressure gradient exists across the film height. This modeling is done with the available heat transfer elements. The library elements listed in Table A 5.7-1 may be used for this purpose:

Table A 5.7-1 Hydrodynamic Bearing Elements

Element	Description
36	2-node, three dimensional link
37	3-node, planar triangle
39	4-node, bilinear quadrilateral
41	8-node, planar biquadratic quadrilateral
65	3-node, three-dimensional link
69	8-node, biquadratic quadrilateral with reduced integration
121	4-node bilinear quadrilateral with reduced integration
131	6-node triangle

The MARC program computes and prints the following elemental quantities: lubricant thickness, pressure, pressure gradient components, and mass flux components. Each of these is printed at the element integration point.

The nodal point data consists of pressures, equivalent nodal mass flux at fixed boundary points, or residuals at points where no boundary conditions are applied. In addition, the program automatically integrates the calculated pressure distribution over the entire region to obtain consistent equivalent nodal forces. This integration is only performed in regions where the pressure exceeds the cavitation pressure.

The output includes the load carrying capacity (the total force on the bearing). This capacity is calculated by a vectorial summation of the nodal reaction forces. In addition, the bearing moment components with respect to the origin of the finite element mesh can be calculated and printed.

To activate the bearing analysis option, use the BEARING parameter. If the analysis requires modeling of flow restrictors, also include the RESTRICTOR parameter.

The values of the viscosity, mass density, and cavitation pressure must be defined in the ISOTROPIC block. Specify temperature-dependent viscosity values via TEMPERATURE EFFECTS. If thermal effects are included, the STATE VARS parameter is also required. In hydrodynamic bearing analyses, temperature is the second state variable. Pressure is the first state variable.

The fluid thickness field may be strongly position-dependent. A flexible specification of the film profile is allowed by using either the nodal thickness or elemental thickness option. Define nodal thickness values in the THICKNESS block. You may also redefine the specified values via the user subroutine UTHICK. Elemental values of lubricant thicknesses can be defined in the GEOMETRY block.

The program also enables the treatment of grooves. Constant groove depth magnitudes can be specified in the GEOMETRY block. If the groove depth is position-dependent, the contribution to the thickness field can be defined in user subroutine UGROOV.

The relative velocity of the moving surfaces is defined on a nodal basis in the VELOCITY block. In addition, you may redefine the specified nodal velocity components in user subroutine UVELOC.

Specify prescribed nodal pressure values in the FIXED PRESSURE block. Define restrictor type boundary conditions in the RESTRICTOR block. To specify nonuniform restrictor coefficients, use user subroutine URESTR.

Input nodal point mass fluxes using the POINT FLUX block. Specify distributed mass fluxes in the DIST FLUX block. If nonuniform fluxes are necessary, apply this via user subroutine FLUX.

Define variations of the previously specified lubrication film thickness field through the THICKNS CHANGE option. The program adds this variation to the current thickness values and solves the lubrication problem.

Activate the calculation of bearing characteristics (i.e., damping and stiffness properties) through the DAMPING COMPONENTS or STIFFNS COMPONENTS options. The program evaluates these properties based on the specified change in film thickness. This evaluation requires the formation of a new right-hand-side, together with a matrix back substitution. This is performed within so-called subincrements. The bearing force components calculated within these subincrements represent the bearing characteristics (i.e., the change in load carrying capacity for the specified thickness change or thickness rate). The previously specified total thickness change or thickness rate). The previously specified total thickness is not updated within subincrements. The calculated bearing characteristics are passed through to user subroutine UTHICK. This allows the user to define an incremental thickness change as function of the previously calculated damping and/or stiffness properties. This procedure may be applied when analyzing the dynamic behavior of a bearing structure. mechanical problems can often be represented by simple mass-damper-spring systems if the bearing structure is nondeformable. A thickness increment may be derived based on the current damping and stiffness properties by investigating the mechanical equilibrium at each point in time.

The bearing analysis capability deals with only steady-state solution and does not include the analysis of transient lubrication phenomena. Note that the incrementation procedure is only meant to analyze a sequence of film profiles. No nonlinearities are involved; each increment is solved in a single step without iteration.

To calculate the reaction forces that act on the bearing structure, the program requires information about the spatial orientation of the lubricant. This information is not contained in the finite element model because of the planar representation of the lubricant. Therefore, it is necessary to define the direction cosines of the unit normal vector that is perpendicular to the lubricant on a nodal basis in subroutine UBEAR. The resulting nodal reaction forces are printed.

The MARC program requires a step-by-step solution of both the lubrication problem and the stress problem in separate runs. The thickness changes need to be defined within the lubrication analysis based upon the displacements calculated in the stress analysis. The stress analysis POST file and user subroutine UTHICK may be used for this purpose. The tractions to be applied in the stress analysis may be read from the bearing analysis POST file in subroutines FORCDT.

Technical Background

The flow of a lubricant between two surfaces that move relative to each other is governed by the Reynolds equation

$$\nabla \bullet \left(\frac{\rho h^3}{12\eta} \nabla p \right) - \frac{\partial (\rho h)}{\partial t} - \frac{1}{2} \nabla \bullet (\rho h u) + M = 0 \qquad \text{(A 5.7-1)}$$

where:

p is lubricant pressure
ρ is mass density
h is film thickness
η is viscosity
t is time
u is the relative velocity vector between moving surfaces
M is the mass flux per unit area added to the lubricant

The following assumptions are involved in the derivation of this equation:

- The lubricant is a Newtonian fluid, i.e., the viscosity is constant.
- There is no pressure gradient across the film height.
- There is laminar flow.
- Inertial effects are negligible.
- The lubricant is incompressible, i.e., mass density is constant.
- Thermal effects are absent.

By introducing the film constant

$$\lambda = \frac{\rho h^3}{12\eta} \qquad \text{(A 5.7-2)}$$

Equation A 5.7-2 may be written as

$$\nabla \bullet (\lambda \nabla p) + M^r = 0 \qquad \text{(A 5.7-3)}$$

where M^r is the reduced mass flux given by

$$M^r = M - \frac{\partial(\rho h)}{\partial t} - \frac{1}{2}\nabla \bullet (\rho h u) \qquad \text{(A 5.7-4)}$$

In case of a stationary bearing, the transient term in Equation A 5.7-4 will vanish.

Three different kinds of boundary conditions may be distinguished for the lubrication problem: prescribed pressure on boundary, prescribed mass flux normal to the boundary, and mass flux proportional to pressure.

Prescribed pressure on boundary is specified as

$$p = \bar{p} \qquad \text{(A 5.7-5)}$$

where $\bar{p}$ is the value of the prescribed pressure.

Prescribed mass flux normal to the boundary has the form

$$-\lambda\frac{\partial p}{\partial n} = \bar{m}_n - \frac{1}{2}\rho h \bar{u}_n = \bar{m}_n^r \qquad \text{(A 5.7-6)}$$

where $\bar{m}_n^r$ is the reduced inward mass flux. Here, n refers to the inward normal on the boundary, and $\bar{m}_n$ and $\bar{u}_n$ are the inward components of total mass flux and relative velocity, respectively.

If a restrictor is used (as in hydrostatic bearings), the total mass flux is a linear function of the pressure on the boundary. This condition is specified as

$$m_n = -\lambda\frac{\partial p}{\partial n} + \frac{1}{2}\rho h u_n = c(\bar{p} - p) \qquad \text{(A 5.7-7)}$$

or, written in a slightly different form

$$-\lambda\frac{\partial p}{\partial n} = c(p^r - p) \qquad \text{(A 5.7-8)}$$

where c is the restriction coefficient and

$$\bar{p}^r = \bar{p} - \frac{\rho h \bar{u}_n}{2c} \qquad \text{(A 5.7-9)}$$

is the reduced pressure.

The set of differential Equation A 5.7-1, together with the boundary conditions (Equation A 5.7-6, Equation A 5.7-7, and Equation A 5.7-9) completely describe the lubrication problem. This is analogous to a heat conduction problem as shown in Table A 5.7-2.

Table A 5.7-2 Comparison of Lubrication and Heat Conduction

Lubrication		Heat Conduction	
Pressure	p	Temperature	T
Film constant	λ	Conductivity	k
Reduced body mass flux	M^r	Body heat flux	Q
Reduced boundary mass flux	m_n	Boundary heat flux	q_n
Restriction coefficient	c	Film coefficient	h
Reduced reference pressure	p^r	Reference temperature	T_r

A 5.8 COUPLED ANALYSES

The MARC program can solve five types of coupled problems: fluid/solid interaction, coupled thermo-electrical (Joule heating), coupled thermo-mechanical, fluid-soil (pore pressure), and electromagnetic.

The fluid/solid model allows for the effect of a fluid on the dynamic behavior of a structure. The fluid is assumed to be inviscid and incompressible. The effect of the fluid is to augment the mass matrix of a structure. Modal shapes can be obtained for a fluid/solid system; the modal superposition procedure predicts the dynamic behavior of the coupled fluid/solid system. This prediction is based on extracted modal shapes.

In the coupled thermo-electrical problem, the coupling takes place through the temperature-dependent electrical conductivity in the electrical problem and the internal heat generation caused by electrical flow in the thermal problem. The program solves for the voltage and temperature distribution.

Similarly, the coupling between the thermal and mechanical problems takes place through the temperature-dependent material properties in the mechanical (stress) problem and the internal heat generation in the mechanical problem caused by plastic work, which serves as input for the heat transfer problem. The temperature distribution and displacements are obtained.

In each of the coupled problems described above, two analyses are performed in each load/time increment. Iterations may also be carried out within each increment to improve the convergence of the coupled thermo-electrical and thermo-mechanical solutions.

In the coupled fluid-soil model, the fluid is assumed to be inviscid and incompressible. The effect of the fluid is to augment the stress in the soil material to satisfy equilibrium, and to influence the soil's material behavior.

In the electromagnetic analysis, the fully coupled Maxwell's equations are solved.

In the latter two analyses, the equations (fluid flow/structural or electrical/magnetic) are solved simultaneously.

Fluid/Solid Interaction

The fluid/solid interaction procedure investigates structures that are either immersed in, or contain a fluid. Examples of problems that make use of this feature are vibration of dams, ship hulls, and tanks containing liquids.

The MARC program is capable of predicting the dynamic behavior of a structural system that is subject to the pressure loading of fluid. The fluid is assumed to be inviscid and incompressible, for example, water. Use the FLU LOAD parameter to activate this option.

In MARC, the fluid is modeled with heat transfer elements (potential theory) and the structure is modeled with normal stress or displacement elements. The element choice must ensure that the interface between the structural and fluid models has compatible interpolation; i.e., both solid and fluid elements are either first order or second order. The tying option may be used to achieve compatibility if necessary. To identify the interfaces between the fluid and the structure, the FLUID SOLID model definition set is necessary.

During increment zero, the program calculates the stiffness matrix for the structure and the mass matrix for the structure augmented by the fluid effect. The program then extracts the eigenvalues of the coupled system using the MODAL SHAPE option. The modal superposition procedure can then be used to predict the time response of the coupled system. DYNAMIC CHANGE may be used to perform modal superposition (see Section 5.2).

To input properties of solids and the mass density of the fluid elements, use the model definition option ISOTROPIC.

The calculation of the structural mass augmentation requires triangularization of the fluid potential matrix: this matrix is singular, unless the fluid pressure is fixed at least at one point with the FIXED DISPLACEMENT option.

Technical Background

In a fluid/solid interaction problem, the equations of motion can be expressed as

$$M_s a + Ku = \frac{1}{\rho_f} S^T p \qquad \text{(A 5.8-1)}$$

The pressure vector p can be calculated from

$$-Sa = Hp \qquad \text{(A 5.8-2)}$$

The matrices in Equations 5.8-1 and 5.8-2 are defined as

$$M_s = \int_v N_i^\beta \; N_i^\alpha \; \rho_s dV \qquad \text{(A 5.8-3)}$$

$$K = \int_v \beta_{ij}^\beta \; D_{ijkl} \; \beta_{kl}^\alpha dV \qquad \text{(A 5.8-4)}$$

$$S = \int_{S^T} R^\nu \rho_f n_i N_i^\alpha dS \qquad \text{(A 5.8-5)}$$

$$H = \int_{v^f} \frac{\partial R^\varepsilon}{\partial x_i} \; \frac{\partial R^\delta}{\partial x_i} dV \qquad \text{(A 5.8-6)}$$

where

ρ_f and ρ_s are mass densities of the fluid and solid, respectively

a	is the acceleration vector
u	is the displacement vector
β_{ij}	is the strain displacement relation
D_{ijkl}	is the material constitutive relation
N_i	is the displacement interpolation function
R	is the pressure interpolation function
n_i	is the outward normal to the surface with fluid pressure p
S^T	is the surface on which the fluid pressure acts

In the present case, the fluid is assumed to be incompressible and inviscid. Only infinitesimal displacements are considered during the fluid vibration, so that the Eulerian and material coordinates coincide.

Substituting Equation A 5.8-2 into Equation A 5.8-1, we obtain

$$(M_s + S^T H^{-1} S/\rho_f)\, a + Ku = 0$$

or (A 5.5-7)

$$\overline{M}a + Ku = 0$$

This equation now allows the modes and frequencies of the solid structure immersed in the fluid to be obtained by conventional eigenvalue methods.

Coupled Thermal-Electrical Analysis (Joule Heating)

The coupled thermal-electrical analysis procedure can be used to analyze electric heating problems. The coupling between the electrical problem and the thermal problem in a Joule heating analysis is due to the fact that the resistance in the electric problem is dependent on temperatures, and the internal heat generation in the thermal problem is a function of the electrical flow.

The MARC program can analyze coupled thermo-electrical (Joule heating) problems. Use parameter JOULE to initiate the coupled thermo-electrical analysis. This capability includes the analysis of the electrical problem, the associated thermal problem, and the coupling between these two problems.

The electrical problem is a steady-state analysis and may involve current and/or voltage boundary conditions as well as temperature-dependent resistivity.

The thermal analysis is generally a transient analysis with temperature-dependent thermal properties and time/temperature-dependent boundary conditions.

Use model definition options ISOTROPIC, ORTHOTROPIC, TEMPERATURE EFFECTS and ORTHO TEMP to input reference values of thermal conductivity, specific heat, mass density, and electrical resistivity, as well as their variations with temperatures. The mass density must remain constant throughout the analysis.

Use model definition option VOLTAGE for nodal voltage boundary conditions, and model definition options POINT CURRENT and/or DIST CURRENT for current boundary conditions. No initial condition is required for the electrical problem, since a steady-state solution is obtained.

In the thermal problem, you can use the model definition options INITIAL TEMP, FILMS, POINT FLUX, DIST FLUXES, and FIXED TEMPERATURE to prescribe the initial conditions and boundary conditions. Use the user subroutines FILM and FLUX for complex convective and flux boundary conditions. To enter the unit conversion factor between the electrical and thermal problems, use the model definition option JOULE. The program uses this conversion factor to compute heat flux generated from the current flow in the structure.

Use history definition options STEADY STATE, TRANSIENT, POINT CURRENT, DIST CURRENT, VOLTAGE CHANGE, POINT FLUXES, DIST FLUXES, and TEMPERATURE CHANGE for the incrementation and change of boundary conditions.

A weak coupling between the electrical and thermal problems is assumed in the coupled thermo-electrical analysis, such that the distributions of the voltages and the temperatures of the structure can be solved separately within a time increment. A steady-state solution of the electrical problem (in terms of nodal voltages) is calculated first within each time step.

The heat generation due to electrical flow is included in the thermal analysis as an additional heat input.

The temperature distribution of the structure (obtained from the thermal analysis) is used to evaluate the temperature-dependent resistivity, which in turn is used for the electrical analysis in the next time increment.

Technical Background

In the coupled thermo-electrical analysis, the matrix equation of the electrical problem can be expressed as

$$\rho(T)\, V = I \qquad \text{(A 5.8-8)}$$

and the governing equation of the thermal problem is

$$C(T)\,\dot{T} + K(T)\,T = Q + Q^E \qquad \text{(A 5.8-9)}$$

In Equation A 5.8-8 and Equation A 5.8-9:

$\rho(T)$	is the temperature-dependent electrical conductivity matrix
I	is the nodal current vector
V	is the nodal voltage vector
$C(T)$ and $K(T)$	are the temperature-dependent heat capacity and thermal conductivity matrices, respective
T	is the nodal temperature vector
$\dot{T}$	is the time derivative of the temperature vector
Q	is the heat flux vector
Q^E	is the internal heat generation vector caused by the current flow.

The coupling between the electrical and thermal problems is through terms $\rho(T)$ and Q^E in Equation A 5.8-8 and Equation A 5.8-9.

The selection of the backward difference scheme for the discretization of the time variable in Equation 5.8-9 yields the following expression:

$$\frac{1}{\Delta t}[C(T)] + K(T)\,T_n = Q_n + Q_n^E + \frac{1}{\Delta t}C(T)\,T_{n-1} \qquad \text{(A 5.8-10)}$$

Equation A 5.8-10 is used for the computation of nodal temperatures in each time increment Δt.

The internal heat generation vector Q^E is computed from

$$Q^E = \int_v B^T q^E dv \qquad \text{(A 5.8-11)}$$

$$q^E = i^2 R \qquad \text{(A 5.8-12)}$$

where i and R are the current and electrical resistance, respectively.

The controls for the heat transfer option allow input of parameters that govern the convergence solution and accuracy of heat transfer analysis.

Thermal Mechanically Coupled Analysis

Many operations performed in the metal forming industry (such as casting, extrusion, sheet rolling, and stamping) may require a coupled thermo-mechanical analysis. The observed physical phenomena must be modeled by a coupled analysis if the following conditions pertain:

- The body undergoes large deformations such that there is a change in the boundary conditions associated with the heat transfer problem.
- Deformation converts mechanical work into heat through an irreversible process which is large relative to other heat sources.

In either case, a change in the temperature distribution contributes to the deformation of the body through thermal strains and influences the material properties.

MARC has a capability that allows you to perform mechanically coupled analysis. This capability is available for all stress elements and for small displacement, total Lagrangian, updated Lagrangian or rigid plastic analysis.

Use the COUPLE parameter to invoke this option. When defining the mesh, if you specify the element as a stress type through the CONNECTIVITY option, the program generates an associated heat transfer element, if possible. The region having an associated heat transfer element has coupled behavior. If you specify the element as a heat transfer type through the CONNECTIVITY option, that region is considered rigid. Only heat transfer is performed in that region.

The program uses a staggered solution procedure in coupled thermo-mechanical analysis. It first performs a heat transfer analysis, then a stress analysis. Use the CONTROL option to enter the control tolerances used in the analysis.

Load control and time step control are intimately coupled in these analyses. These controls may be specified in either of two ways.

- A fixed time step/load size may be specified by using either the DYNAMIC CHANGE option or the TRANSIENT NON AUTO option. In these cases, mechanical loads and deformations are incremental quantities that are applied to each step. Fluxes are total quantities.
- An adaptive time step/load size may be specified by using the AUTO TIME option. In this case, mechanical loads and deformations are entered as the total quantities that are applied over the load set. Fluxes are total quantities.

Exercise caution when applying boundary conditions. Use the FIXED DISP or FIXED TEMP options as well as DISP CHANGE or TEMP CHANGE for mechanical or thermal boundary conditions, respectively.

There are two primary causes of coupling. First, coupling occurs when deformations result in a change in the associated heat transfer problem. Such a change may be due to either large deformation or contact. Large deformation effects are coupled into the heat transfer problem only if the UPDATE option is invoked. The gap element in MARC (Type 12) has been modified so that if no contact occurs, the gap element acts as a perfect insulator. When contact does occur, the gap element acts as a perfect conductor.

The second cause of coupling is heat generated due to inelastic deformation. The irreversibility of plastic flow causes an increase in the amount of entropy in the body. This can be expressed as

$$T\dot{S} = f\dot{W}^{p} \qquad \text{(A 5.8-13)}$$

where $f\dot{W}^{p}$ is the fraction of the rate of plastic work dissipated into heat. Farren and Taylor measure f as approximately 0.9 for most metals. Using the mechanical equivalent of heat (M), the rate of specific volumetric flux is

$$R = Mf\dot{W}^{p}/\rho \qquad \text{(A 5.8-14)}$$

Use the CONVERT model definition option to define Mf.

Of course, all mechanical and thermal material properties may be temperature dependent. The governing matrix equations may be expressed as

$$M\ddot{u} + D(T)\dot{u} + K(T)u = f \qquad \text{(A 5.8-15)}$$

$$C(T)\dot{T} + K(T)T = Q + Q^{1} \qquad \text{(A 5.8-16)}$$

In Equation A 5.8-15, Q^{1} represents the amount of heat generated due to plastic work. The specific heat matrix C and conductivity matrix K may be evaluated in the current configuration if the updated Lagrange option is used.

NOTE

All terms except M may be temperature dependent.

A 5.9 ELECTROSTATIC ANALYSIS

The MARC program has the capability to perform electrostatic analysis. This allows the program to evaluate the electric fields in a body or media, where electrical charges are present. This can be solved for one-, two-, or three-dimensional fields. The semi-infinite elements may be used to represent an infinite domain. The electrostatic problem is governed by the Poisson equation for a scalar potential. This analysis is purely linear and has been implemented in MARC analogously to the steady state heat transfer problem. The available elements are described in Table A 5.9-3 below.

Table A 5.9-3 Element Types for Electrostatic Analysis

Element Type	Description
37, 39, 131, 41	3-, 4-, 6-, 8-node planar
69	8-node reduced integration planar
121	4-node reduced integration planar
101, 103	6-, 9-node semi-infinite planar
38, 40, 132, 42	3-, 4-, 6-, 8-node axisymmetric
70	8-node reduced integration axisymmetric
122	4-node reduced integration axisymmetric
43, 44	8-, 20-node 3-dimensional brick
71	20-node reduced integration brick
105, 106	12-, 27-node semi-infinite brick
123	8-node reduced integration brick
133	10-node tetrahedron
85, 86	4-, 8-node shell
87, 88	2-, 3-node axisymmetric shell

The MARC program computes and prints the following quantities: electric potential field vector (E) and electric displacement vector (D) at the element integration points. The nodal point data consists of the potential ϕ and the charge Q.

To activate the electrostatic option, use the ELECTRO parameter. The value of the isotropic permittivity constant is given in the ISOTROPIC option, orthotropic constants may be specified using the ORTHOTROPIC option. Optionally, user subroutine UEPS may be used.

Specify nodal constraints using the FIXED POTENTIAL option. Input nodal charges using the POINT CHARGE block. Specify distributed charges by using the DIST CHARGE block. If nonuniform charges are required, user subroutine FLUX can be used for distributed charges and user subroutine FORCDT for point charges.

The electrostatic capability deals with linear, steady-state problems only. The STEADY STATE option is used to begin the analysis. The resultant quantities may be stored on the post-tape for processing with Mentat.

Technical Background

The Maxwell equations to govern electrostatics are written in terms of the electric displacement vector D and the electric field vector E such that

$$\nabla \bullet D = \rho \qquad \text{(A 5.9-1)}$$

and

$$\nabla \times E = 0 \qquad \text{(A 5.9-2)}$$

where ρ is a given volume charge density.

The constitutive law is typically given in a form as:

$$D = \varepsilon E \qquad \text{(A 5.9-3)}$$

where ε is the dielectric constant.

Introducing the scalar potential ϕ such that

$$E = -\nabla\phi \qquad \text{(A 5.9-4)}$$

which satisfied the constraint Equation A 5.9-2 exactly.

Denoting the virtual scalar potential by ψ, the variational formulation is

$$\int_\Omega \nabla\psi \bullet \varepsilon\nabla\phi dV = \int_\Omega \psi\rho dV + \int_\Gamma \psi\,(\varepsilon\nabla\phi - n)\,dS \qquad \text{(A 5.9-5)}$$

The natural boundary condition is applied through the surface integral in terms of the normal electric displacement using

$$-\varepsilon\nabla\phi \bullet n = D \bullet n = D_n \qquad \text{(A 5.9-6)}$$

Consider a material interface Γ_{12}, separating two materials 1 and 2, and as ϕ is continuous over the material interface, the tangential electric field constraint is automatically satisfied.

$$n \times (E_1 - E_2) = 0 \qquad \text{on } \Gamma_{12} \qquad \text{(A 5.9-7)}$$

If charges are present on the interface, these are applied as distributed loads as follows:

$$n \bullet (D_1 - D_2) = \rho_s \qquad \text{on } \Gamma_{12} \qquad \text{(A 5.9-8)}$$

Using the usual finite element interpolation functions N and their derivatives β, we obtain

$\psi = N \qquad \phi = N\Phi$

$$K = \int_\Omega \beta^T \varepsilon \beta dV \tag{A 5.9-9}$$

$$F = \int_\Omega N^T \rho N dV + \int_\Gamma N^T N D_n dS + \int_{\Gamma_{12}} N^T N \rho_s dS \tag{A 5.9-10}$$

and finally

$$K\phi = F \tag{A 5.9-11}$$

A 5.10 MAGNETOSTATIC ANALYSIS

The MARC program has the capability to perform magnetostatic analysis. This allows the program to calculate the magnetic field in a media subjected to steady electrical currents. This can be solved for two- or three-dimensional fields. Semi-infinite elements may be used to represent an infinite domain.

The magnetostatic analysis for two-dimensional analysis is solved using a scalar potential, while for three-dimensional problems, a full vector potential is used. The magnetic permeability may be a function of the magnetic field, hence, creating a nonlinear problem. Only steady-state analyses are performed. The available elements which are described in Table A 5.10-1.

Table A 5.10-1 Elements Types for Magnetostatic Analysis

Element Type	Description
37, 39, 131, 41	3-, 4-, 6-, 8-node planar
69	8-node reduced integration planar
121	4-node reduced integration planar
101, 103	6-, 9-node semi-infinite planar
102, 104	6-, 9-node semi-infinite axisymmetric
38, 40, 132, 42	3-, 4-, 6-, 8-node axisymmetric
70	8-node reduced integration axisymmetric
122	4-node axisymmetric reduced integration
109	8-node brick
110	12-node semi-infinite brick

The MARC program computes and prints magnetic induction (B), the magnetic field vector (H), and the vector potential at the element integration points. The nodal point data consist of the potential (A) and the current (J).

To activate the magnetostatic option, use the MAGNETO parameter. The value of the isotropic permeability (μ) is given on the ISOTROPIC option; orthotropic constants may be specified using the ORTHOTROPIC option. Optionally, user subroutine UEPS may be used. Often, it is easier to specify ($1/\mu$), which is also available through these options. A nonlinear permeability may be defined using the B-H relation.

Specify nodal constraints using the FIXED POTENTIAL option. Input nodal currents using the POINT CURRENT block. Specify distributed currents by using the DIST CURRENT block. Permanent magnets may be introduced by using the PERMANENT option for isotropic materials, or by entering a remanence vector via the B-H RELATION option for orthotropic materials. In addition, user subroutine FLUX can be used for nonuniform distributed current and user subroutine FORCDT for fixed nodal potentials or point current.

The magnetostatic capability is linear unless a nonlinear B-H relation is defined. In such problems, convergence is reached when the residual current satisfies the tolerance defined in the CONTROL option. The STEADY STATE option is used to begin the analysis. The resultant quantities may be stored on the post tape for processing with Mentat.

Technical Background

The Maxwell equations for magnetostatics are written in terms of the magnetic flux density vector B such that

$$\nabla \times H = J \tag{A 5.10-1}$$

and

$$\nabla \bullet B = 0 \tag{A 5.10-2}$$

where J is the current density vectors.

For magnetic materials, the following relation between B, H, and M, the magnetization vector, holds:

$$B = \mu_0 (H + M) \tag{A 5.10-3}$$

with μ_0 being the permeability of vacuum. Denoting the magnetic susceptibility by X_m and the permanent magnet vector by M_0, we have

$$M = X_m H + M_0 \tag{A 5.10-4}$$

which cay be substituted into Equation A 5.10-3 to yield

$$B = \mu H + B_r \tag{A 5.10-5}$$

in which μ is the permeability, given by

$$\mu = \mu_0 (1 + X_m) \tag{A 5.10-6}$$

and B_r is the remanence, given by

$$B_r = \mu_0 M_0 \tag{A 5.10-7}$$

Notice that $(1 + X_m)$ is usually called the relative permeability μ_r.

For isotropic linear material, X_m and m_0 are scalar constants. If the material is orthotropic, X_m and m change into tensors. For real ferromagnetic material, X_m and μ will never be constant. Instead, they will depend on the strength of the magnetic field. Usually this type of material nonlinearity is characterized by a so-called magnetization curve or B-H relation specifying the magnitude of (a component of) B as a function of (a component of) H.

In MARC, the magnetization curve can be entered via the B-H RELATION option. For isotropic material, only one set of data points, representing the magnitude of the magnetic induction, $|B|$, as a function of $|H|$, the magnitude H, needs to be given. For orthotropic material, multiple curves are needed with each curve relating a component of B to the corresponding component of H.

From Equation A 5.10-5 and Equation A 5.10-7, it can be easily seen that for orthotropic material, a permanent magnetization or remanence can be entered through the B-H RELATION option, by putting in a nonzero B value for H = Q. For isotropic material, this does not work since the direction of the remanence vector would still be indeterminate. Therefore, in the isotropic case, the only possibility is to supply the magnetization vector through the PERMANENT option. Any offset of the $|B|$ ($|H|$)-curve, implying $|B| \neq 0$ at ($|H| = 0$) will be disregarded in this case. For orthotropic material, it is not allowed to use the PERMANENT option. In this case, the magnetization can exclusively be specified through the B-H RELATION option.

It is emphasized that the magnetization curve specified under B-H RELATION must be monotone and uniquely defined.

Introducing the vector potential A such that

$$\nabla \times A = B \tag{A 5.10-8}$$

which automatically satisfies the constraint, Equation (5.10-2), and we then have the final form

$$\nabla \times \mu^{-1} (\nabla \times A) = J + \mu^{-1} \beta_r \tag{A 5.10-9}$$

The vector potential (A) is not uniquely defined by Equation A 5.10-8. In 2D magnetostatic simulations, this indeterminacy is removed by the reduction of A to a scalar quantity. In 3D situations, MARC uses the Coulomb gauge for this purpose:

$$\nabla \bullet A = 0 \tag{A 5.10-10}$$

Denote the virtual potential by W; then, the variational formulation is

$$\int_V \mu^{-1} (\nabla \times W) \bullet (\nabla \times A)\, dV = \int_V W \bullet (J)\, dV + \int_V W \bullet (\nabla \times \mu^{-1} B_r)\, dV + \int_\Gamma W \bullet (H + n)\, dS \tag{A 5.10-11}$$

where n is the outward normal to V at the boundary S.

In the three-dimensional case, the Coulomb gauge, Equation A 5.10-10, is enforced with a penalty formulation. The resulting term added to the variational formulation, Equation A 5.10-11 reads:

$$\ldots + \int_V r \nabla \bullet W \nabla \bullet A\, dV \tag{A 5.10-12}$$

The default value used for r is:

$$r = 10^4 |\mu^{-1}| \tag{A 5.10-13}$$

Using the usual finite element interpolation functions N, the discrete curl operator G, and the weighting function W = N.

$$G = \begin{bmatrix} \partial N/\partial y \\ \partial N/\partial x \end{bmatrix} \quad \text{for 2-dimensional problems}$$

$$G = \begin{bmatrix} \partial N/\partial y - \partial N/\partial z \\ \partial N/\partial z - \partial N/\partial x \\ \partial N/\partial x - \partial N/\partial y \end{bmatrix} \quad \text{for 3-dimensional problems} \qquad \text{(A 5.10-14)}$$

Leads to the resulting system of algebraic equations

$$KA = F \qquad \text{(A 5.10-15)}$$

where

$$K = \int_V G^T \mu^{-1} G \; dV \qquad \text{(A 5.10-16)}$$

$$F = \int N^T N\,(J)\, dV + \int_V G^T M^{-1} B_r dV + \int_\Gamma N^T N\,(H \times n)\, dS \qquad \text{(A 5.10-17)}$$

The Coulomb gauge is based upon the principle of conservation of electric charge which in its steady state form reads:

$$\nabla \bullet J = 0 \qquad \text{(A 5.10-18)}$$

In MARC, it is up to the user to specify the current distribution. When doing so, it is recommended to ensure that this distribution satisfies Equation A 5.10-17. Otherwise, condition Equation A 5.10-10 could be violated. As a consequence, the results could become less reliable.

From the third term on the right-hand side of Equation A 5.10-17, it becomes clear that the natural boundary condition in this magnetostatic formulation is H x n, the tangential component of the magnetic field intensity. Consequently, if no other condition is specified by the user, by default a zero tangential magnetic field intensity at the boundaries will be assumed.

Besides, when there are no currents present on a Γ_{12} material interface separating two materials 1 and 2, the tangential magnetic field intensity is assumed to be continuous:

$$n \times (H_1 - H_2) = 0 \text{ on } \Gamma_{12} \qquad \text{(A 5.10-19)}$$

With H_1 and H_2, the magnetic field intensities in material 1 and 2, respectively. A discontinuous tangential magnetic field intensity can be modelled by assigning an appropriate distributed "shear" current density to the interface. This (surface) current density j is related to H_1 and H_2 by:

$$n \times (H_1 - H_2) = j \text{ on } \Gamma_{12} \qquad \text{(A 5.10-20)}$$

A 5.11 ACOUSTIC ANALYSIS

The MARC program has the capability to perform acoustic analysis in a cavity with rigid boundaries. This allows the program to calculate the fundamental frequencies of the cavity, as well as the pressure distribution in the cavity. This can be solved for two- or three-dimensional fields.

The acoustic problem with rigid reflecting boundaries is a purely linear problem analogous to dynamic analysis, but using the heat transfer elements. The elements which are available are described in Table A 5.11-1.

Table A 5.11-1 Element Types for Acoustic Analysis

Element Type	Description
37, 39, 131, 41	3-, 4-, 6-, 8-node planar
69	8-node reduced integration planar
121	4-node reduced integration planar
101, 103	6-, 9-node semi-infinite planar
38, 40, 132, 42	3-, 4-, 6-, 8-node axisymmetric
70	8-node reduced integration axisymmetric
122	4-node reduced integration axisymmetric
133	10-node tetrahedron

The MARC program computes and prints the following quantities: pressure, and pressure gradient at the integration points. The nodal point data consists of the pressure and the source.

To activate the acoustic option, use the ACOUSTIC parameter. The number of modes to be extracted should also be included on this card. The bulk modulus and the density of the fluid are given in the ISOTROPIC option.

Specify nodal constraints using the FIXED PRESSURE option. Input nodal sources using the POINT SOURCE block. Specify distributed sources using the DIST SOURCE block. If nonuniform sources are required, apply these via user subroutine FLUX.

To obtain the fundamental frequencies, use the MODAL SHAPE option after the END OPTION. The nodes can be used in a transient analysis by invoking the DYNAMIC CHANGE option. The point and distributed sources could be a function of time.

Technical Background

The wave equation of an inviscid fluid can be expressed in terms of the pressure p as

$$\frac{1}{c^2}\frac{\partial^2 p}{\partial^2 t} = \nabla^2 p \qquad \text{(A 5.11-1)}$$

where c is the sonic velocity

$$c = \sqrt{K/\rho} \qquad \text{(A 5.11-2)}$$

where K is the bulk modulus and ρ is the density.

Equation (5.11-1) can be rewritten as

$$K\nabla^2 p - \rho\frac{\partial^2 p}{\partial^2 t} = 0 \qquad \text{(A 5.11-3)}$$

Where the source terms are neglected, note that this is analogous to the dynamic equation of motion.

The modelling of rigid reflecting boundaries can be done as follows:

mathematically a reflecting boundary is described by

$$\frac{\partial p}{\partial n} = 0 \qquad \text{(A 5.11-4)}$$

where $\frac{\partial p}{\partial n}$ is the pressure gradient normal to the reflecting surface.

This is analogous to an insulated boundary in heat transfer. Hence, a reflecting boundary can be modelled by a set of nodes at the outer surface of the area which are not connected to another part of the medium. A reflecting plate in the middle of an acoustic medium can be modelled by double nodes at the same location

NOTE

Where there are no boundary conditions applied, there is a zero-valued eigenvalue, corresponding to a constant pressure mode. Hence, you will need to have a non-zero initial shift point.

A 5.12 ELECTROMAGNETIC ANALYSIS

The MARC program has the capability to perform both transient (dynamic) and harmonic fully coupled electromagnetic analysis. This allows the program to calculate the electrical and magnetic fields subjected to external excitation. This can be solved for both two- or three-dimensional fields. A vector potential for the magnetic field is augmented with a scalar potential for the electrical field. If a transient analysis is performed, the magnetic permeability may be a function of the magnetic field, hence a nonlinear problem. The elements available for electromagnetic analysis are described in Table A 5.12-1.

Table A 5.12-1 Element Types for Electromagnetic Analysis

Element Type	Description
111	4-node planar
112	4-node axisymmetric
113	8-node brick

The MARC program prints the magnetic flux density (B), the magnetic field vector (H), electric flux density (D), and the electrical field intensity at the integration points. In a harmonic analysis, these have real and imaginary components. The nodal point data consists of the vector potential A, the scalar potential V, the charge Q, and current I.

To activate the electromagnetic option, use the EL-MA parameter. The values of the isotropic permittivity (ε), permeability (μ), and conductivity (σ) are given in the ISOTROPIC option. Orthotropic constants may be specified using the ORTHOTROPIC option. User subroutines UEPS, UMU and USIGMA may be optionally used. A nonlinear permeability may be defined using the B-H relation.

Specify nodal constraints using the FIXED POTENTIAL option. Input nodal currents and charge using the POINT CURRENT block. Specify distributed currents by using the DIST CURRENT block and distributed charges by using the DIST CHARGE option. Nonuniform distributed currents and charges may also be specified by user subroutine FORCEM.

The electromagnetic capability is linear, unless a nonlinear B-H relation is defined. In such problems, convergence is reached when the residual satisfies the tolerance defined in the CONTROL option. The transient capability is only available with a fixed time step; use the DYNAMIC CHANGE option to activate this option. The resultant quantities may be stored on the post tape for processing with Mentat.

In electromagnetic analysis, you can enter the current and/or the charge. In a harmonic analysis, you can enter both the real and imaginary components.

Table A 5.12-2 Input Options for Electromagnetic Analysis

Load Description	Input Options		
	Model Definition	History Definition	User Subroutine
Nodal Current Nodal Charge	POINT CURRENT	POINT CURRENT	FORCDT
Distributed Current	DIST CURRENT	DIST CURRENT	FORCEM
Distributed Charge	DIST CHARGE	DIST CHARGE	FORCEM

Technical Background

Technical Formulation

Electromagnetic analysis is based upon the well-known Maxwell's equations. This has been implemented in MARC for both transient and harmonic analyses.

Transient Formulation

The Maxwell's equations are:

$$\nabla E + \dot{B} = 0 \qquad \text{(A 5.12-1)}$$

$$\nabla \times H - \varepsilon \dot{E} - \sigma E = 0 \qquad \text{(A 5.12-2)}$$

$$\nabla \cdot (\varepsilon \dot{E} + \sigma E) = 0 \qquad \text{(A 5.12-3)}$$

$$\nabla \cdot B = 0 \qquad \text{(A 5.12-4)}$$

where the constitutive relations are

$D = \varepsilon E$
$B = \mu_0(H+M)$
$J = \sigma E$

and

E is the electric field intensity
D is the electric flux density
H is the magnetic field intensity
B is the magnetic flux density
J is the current density
M is the magnetization

and

ε is the permittivity
μ is the permeability
σ is the conductivity
μ_0 is the permeability of free space.

Additionally, we have the conservation of charge:

$$\dot{\rho} + \nabla \cdot J = 0 \qquad \text{(A 5.12-5)}$$

where ρ is the charge density

We assume that the magnetization vector is given by

$$M = \chi H + M_o \qquad \text{(A 5.12-6)}$$

where M_o is the strength of the permanent magnet and χ is the susceptibility. Note that χ and μ are not independent, but satisfy the relation $\mu = \mu_0 (I + \chi)$.

A vector magnetic potential A and a scalar potential V are introduced, such that

$$B = \nabla \times A \qquad \text{(A 5.12-7)}$$

$$E = -(\nabla V + \dot{A}) \qquad \text{(A 5.12-8)}$$

Note that since only the curl of A is required, an arbitrary specification of the divergence may be made. The Coulomb gage is then introduced.

$$\nabla \cdot A = 0 \qquad \text{(A 5.12-9)}$$

This is implemented using a penalty condition. It is important to note that E depends not only on the scalar potential, but also upon the vector potential. Hence, interpretation of V as the usual voltage may lead to erroneous results.

Substituting into Maxwell's equations results in:

$$\nabla \times [\mu^{-1} (\nabla \times A - \mu_o M_o)] + \varepsilon (\nabla \dot{V} + \ddot{A}) + \sigma (\nabla V + \dot{A}) = 0 \qquad \text{(A 5.12-10)}$$

$$-\nabla \bullet [\varepsilon (\nabla \dot{V} + \ddot{A})] + \sigma (\nabla V + \dot{A}) = 0 \qquad \text{(A 5.12-11)}$$

It has been assumed that $\dot{\varepsilon} = 0$ i.e., the dielectric, is constant.

Due to the hyperbolic nature of the above system (similar to dynamics), a Newmark algorithm will be employed in the discretization.

The general form is

$$A^{n+1} = A^n + \Delta t \dot{A}^n + (0.5 - \beta) \Delta t^2 \ddot{A}^n + \beta \Delta t^2 A^{n+1} \qquad \text{(A 5.12-12)}$$

$$\dot{A}^{n+1} = \dot{A}^n + (1 - \gamma) \Delta t \ddot{A}^n + \gamma \Delta t \ddot{A}^{n+1} \qquad \text{(A 5.12-13)}$$

The particular form of the dynamic equations corresponding to the trapezoidal rule:

$$\gamma = \frac{1}{2} \quad \beta = \frac{1}{4} \qquad \text{(A 5.12-14)}$$

results in a symmetric formulation, which is unconditionally stable for linear systems.

In the current formulation, a fixed time step procedure must be used. The time step is defined through the DYNAMIC CHANGE option.

Harmonic Formulation

In harmonic analysis, it is assumed that the excitation is a sinusoidal function, and the resultant also has a sinusoidal behavior. This results in the solution of a complex system of equations. In this case, Maxwell's equations become

$$\nabla \times E + i\omega B = 0 \qquad \text{(A 5.12-15)}$$

$$\nabla \times H - i\varepsilon\omega E - \sigma E = 0 \qquad \text{(A 5.12-16)}$$

$$\nabla \bullet D - \rho = 0 \qquad \text{(A 5.12-17)}$$

$$\nabla \bullet B = 0 \qquad \text{(A 5.12-18)}$$

where ω is the excitation angular frequency.

Additionally, we have the conservation of charge

$$i\omega\rho + \nabla \bullet J = 0 \qquad \text{(A 5.12-19)}$$

where all symbols are the same as in the discussion above regarding transient behavior. Again, a vector potential A and a scalar potential V are introduced. In this case, these are complex potentials. Substituting into the Maxwell's equations results in

$$\nabla \times [\mu^{-1} (\nabla \times A - \mu_o M_o)] + \tilde{\sigma} (\nabla V + i\omega A) = 0 \qquad \text{(A 5.12-20)}$$

$$-\nabla \cdot [\tilde{\sigma} (\nabla V + i\omega A)] = 0 \qquad \text{(A 5.12-21)}$$

where $\tilde{\sigma} = \sigma + i\omega\varepsilon$

With a little manipulation, a symmetric complex formulation may be obtained. The excitation frequency is prescribed using the HARMONIC option.

Note that there is no capability in MARC to extract the natural frequencies of a complex system by modal analysis.

A 5.13 SOIL ANALYSIS

The soil analysis capability in MARC involves two parts. The first is the solution of the fluid-soil interaction. The second is the definition of the soil material behavior. In the current formulation, it is assumed that the fluid is of a single phase, and only slightly compressible. This formulation will not be adequate if steam-fluid-solid analysis is required. The dry soil can be modelled using one of the three models: linear elasticity, nonlinear elasticity and the modified Cam-Clay model.

There are three types of soil analysis available in MARC. In the first type, one is performing an analysis to calculate the fluid pressure in a porous medium. In such analyses, heat transfer elements 41, 42, or 44 are used. The soil model definition is used to define the permeability of the solid and the bulk modulus and dynamic viscosity of the fluid. The porosity is given either through the INITIAL POROSITY or the INITIAL VOID options and does not change with time. The prescribed pressures may be defined using the FIXED PRESSURE option, while input mass flow rates are given using either the POINT FLUX or DIST FLUXES option.

In the second type of soil analysis, the pore (fluid) pressure is directly defined, and the structural analysis is performed. Element types 27, 28, or 21 are available. In such analyses, the pore pressure is prescribed using the INITIAL PORE and CHANGE PORE options. The characteristics of the soil material are defined using the SOIL option. If an elastic model is used, the Young's moduli and Poison's ratio are important. If the Cam-Clay model is used, the compression ratios and the slope of the critical state line is important. For the Cam-Clay model, the preconsolidation pressure is defined using the INITIAL PC option. For this model, it is also important to define an initial (compressive stress) to ensure a stable model.

In the third type of soil analysis, a fully-coupled approach is used. Element types 32, 33, or 35 are available. These elements are "Herrmann" elements, which are conventionally used for incompressible analysis. In this case, the extra variable represents the fluid pressure. The SOIL option is now used to define both the soil and fluid properties. The porosity is given through the INITIAL POROSITY or the INITIAL VOID option. The prescribed nodal loads and mass flow rates are given through the POINT LOAD option, while distributed loads and distributed mass flow rates are given through the DIST LOADS option. The FIXED DISP option is used to prescribe either nodal displacements or pore pressures.

Technical Formulation

In soil mechanics, it is convenient to decompose the total stress ϕ_0 into a pore pressure component p and effective stress σ'.

$$\sigma = \sigma' - pI \qquad \text{(A 5.13-1)}$$

Note the sign convention used; a positive pore pressure results in a compressive stress. The momentum balance (equilibrium) equations are with respect to the total stresses in the system.

$$\nabla \bullet \sigma + f = \rho\ddot{u} \qquad \text{(A 5.13-2)}$$

where ρ is the density, and f, $\ddot{u}$ are the body force and the acceleration. The equilibrium equation can then be expressed as

$$\nabla\sigma' - \nabla p + f = \rho\ddot{u} \qquad \text{(A 5.13-3)}$$

The fluid flow behavior can be modeled using Darcy's law, which states that the fluid's velocity, relative to the soil's skeleton, is proportional to the total pressure gradient.

$$\dot{U}_f = \frac{-K}{\mu}(\nabla p + \rho_f g) \qquad \text{(A 5.13-4)}$$

where

$\dot{U}_f$ is the fluid relative velocity
K is the soil permeability
μ is the fluid viscosity
ρ_f is the fluid density
g is the gravity vector.

The fluid is assumed to be slightly compressible.

$$\dot{p} = K_f \frac{\dot{\rho}_f}{\rho_f} \qquad \text{(A 5.13-5)}$$

where K_f is the bulk modulus of the fluid.

However, the compressibility is assumed small enough such that the following holds:

$$\nabla \bullet \left[\rho_f \frac{K}{\mu}(\nabla p + \rho_f g) \right] \approx \rho_f \nabla \bullet \left[\frac{K}{\mu}(\nabla p + \rho_f g) \right] \qquad \text{(A 5.13-6)}$$

It is also assumed that the bulk modulus of the fluid is constant, introducing the fluid's compressibility β_f.

$$\beta_f = 1/K_f \qquad \text{(A 5.13-7)}$$

The solid grains are assumed to be incompressible. Under these assumptions, the governing equations for fluid flow is

$$\nabla \bullet \left[\frac{K}{\mu}(\nabla p + \rho_f g) \right] = \phi\beta_f \dot{p} + \nabla \bullet \dot{u} \qquad \text{(A 5.13-8)}$$

where σ is the medium's porosity.

It is important to note that the medium's porosity is only dependent upon the original porosity and the total strains. Letting V_f and V_s stand for the fluid and solid's volume

$$\phi = dV_f / (dV_f + dVs) = 1 - J^{-1}(1 - \phi_o) \qquad \text{(A 5.13-9)}$$

where J is the determinant of the deformation gradient and ϕ_o is the original porosity.

Introducing the weighting function $\bar{u}$ and $\bar{p}$, the weak form, which is the basis for the finite element system, then becomes

$$\int_v [\bar{u} \bullet f - \rho \bar{u} \bullet \ddot{u} - \nabla \bar{u} \bullet \sigma' + \nabla \bullet \bar{u} p] \, dV + \int_s \bar{u} \bullet t \, dS = 0 \qquad \text{(A 5.13-10)}$$

where V and S are the conventional volumes and surface area and t is the applied tractions. Note that the applied tractions is the combined tractions from both the effective stress σ' and the pore pressure p.

and

$$\int_v \left[\nabla \bar{p} \bullet \frac{K}{\mu} (\nabla p + \rho_f g) + \phi \beta_f \bar{p} \dot{p} + \bar{p} \nabla \bullet \dot{u} \right] dV - \int_s \bar{p} q_n dS = 0 \qquad \text{(A 5.13-11)}$$

where

$$q_n = \frac{k}{\mu} (\nabla p + \rho_f g) \bullet n \qquad \text{(A 5.13-12)}$$

is the normal volumetric inflow.

Three types of analyses may be performed. The simplest is a solution for only the fluid pressure based upon the porosity of the soil. In this case, a simple Poisson type analysis is performed and element types 41, 42, or 44 are used. In the second type of analysis, the pore pressures are explicitly defined and the structural analysis is performed. In this case, the element types 21, 27, and 28 should be used. Finally, a fully-coupled analysis is performed, in which case one should use element types 32, 33, or 35. Of course, the soil can be combined with any other element types, material properties to represent the structure, such as the pilings.

Soil Model

To be able to complete the soil mechanics formulation, it is also necessary to include the constitutive relation of the soil. Soil material modeling is considerably more difficult than conventional metals, because of its nonhomogeneous characteristics. Most soil models attempt to represent the fact that the material consists of a large amount of random particles. These models attempt to simulate behavior that is observed in the experimental laboratory. Among them is that: the bulk modulus increases as the particles are pressed together; when the preconsolidation stress is exceeded, the stiffness is dramatically reduced; and, when unloaded, the stiffness increases. At failure, there is no resistance to shear, and stiff clays or dense sands are dilatant. Over the years, many formulations have been used, including: linear elastic, nonlinear elastic, Drucker-Prager or Mohr Coulomb, and Cam-Clay and variations thereof. In MARC, we have implemented linear elasticity, nonlinear elasticity and the Cam-Clay model.

Elastic Models

Linear elasticity is defined in the conventional manner, defining the Young's moduli and the Poisson ratio in the SOIL option. The nonlinear elasticity model is implemented through user subroutine HYPELA. Some of the simplest models include the bilinear elasticity, where a different moduli is used during the loading and unloading path, or to represent total failure when a critical stress is obtained.

A more sophisticated elastic law is the hyperbolic model, where six constants are used. In this model the tangent moduli are

$$E = \left[1 - \frac{R_f(1-\sin\phi)\,(\sigma_1\sigma_3)}{2c\cos\phi + 2\sigma_3\sin\phi}\,\text{Mpa}\right]\text{MPa}\,(\frac{\sigma_3}{pa}) \qquad \text{(A 5.13-13)}$$

Two of the elastic models are the E-ν and K-G variable elastic models. In the E-ν model, Poisson's ratio is considered constant and

$$E = E_0 + \alpha_E p + \beta_E \tau \qquad \text{(A 5.13-14)}$$

while, in the K-G model

$$K = K_0 + \alpha_K p \qquad \text{(A 5.13-15)}$$

$$G = G_0 + \alpha_G p + \beta_G \tau \qquad \text{(A 5.13-16)}$$

The key difficulty with the elastic models is that dilatancy cannot be represented.

Cam-Clay Model

The Cam-Clay model was originally developed by Roscoe, and then evolved into the modified Cam-Clay model of Roscoe and Burland. This model, which is also called the critical state model, is implemented in MARC.

The yield surface is an ellipse in the p,τ plane as shown in Figure A 5.13-1, and is defined by

$$F = \frac{\tau^2}{M_{cs}^2} - 2pp_c + p^2 = 0 \qquad \text{(A 5.13-17)}$$

where p_c is the preconsolidation pressure, and M_{cs} is the slope of the critical state line.

The Cam-Clay model has the following properties of some note. At the intersection of the critical state line and the ellipse, the normal to the ellipse is vertical. Because an associated flow rule is used, this implies that at failure all plastic strain is distortional; the soil deforms at constant volume. Also, if the preconsolidation pressure is large, the soil will remain elastic for large stresses.

The evolution of the preconsolidation pressure is

$$\dot{p}_c = -\theta p_c \text{tr}\,(\dot{\varepsilon}^{pl}) \qquad \text{(A 5.13-18)}$$

where

$$\theta = \frac{1+e}{\lambda - \kappa} \qquad \text{(A 5.13-19)}$$

where

e is the void ratio
λ is the virgin compression index
κ is the compression index

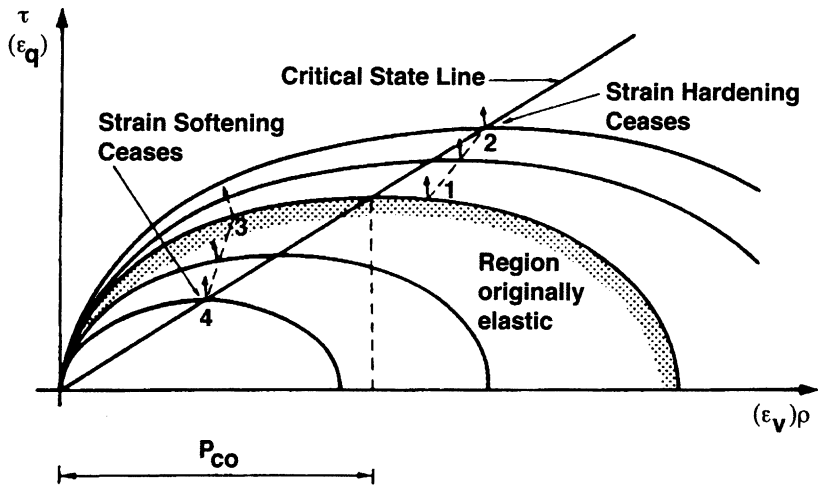


Figure A 5.13-1 Modified Cam-Clay Yield Surface

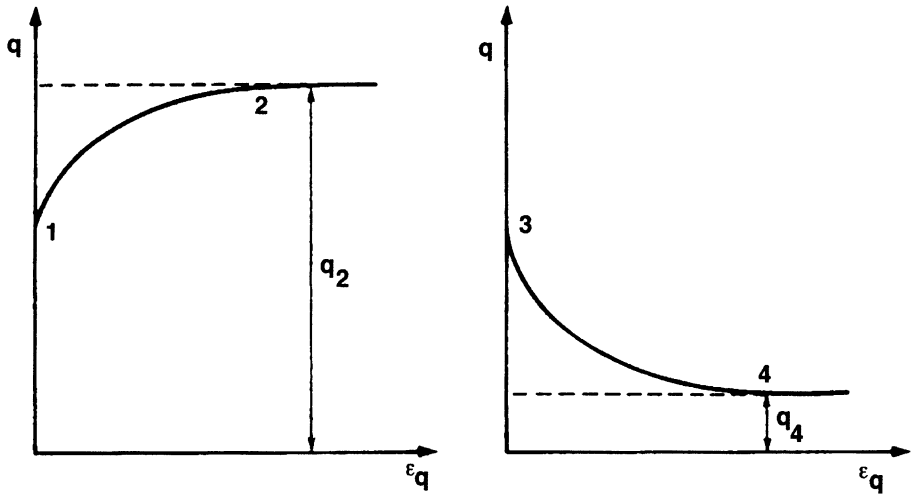


Figure A 5.13-2 Strain Hardening and Softening Behavior

The void ratio and the porosity are related by the expression

$$\phi = \frac{e}{1-e} \qquad \text{(A 5.13-20)}$$

In the modified Cam-Clay model, it is also assumed that the behavior is nonlinear elastic with a constant Poisson's ratio, the shear modulus behave as:

$$K = -\frac{1+e}{\kappa} p \qquad \text{(A 5.13-21)}$$

Note that this implies that at no hydrostatic stress, the bulk modulus is also zero. To avoid computational difficulties, a cutoff pressure of one percent of the preconsolidation pressure is used.

This constitutive law is implemented in MARC using a radial return procedure. It is available for either small or large strain analysis.When large displacements are anticipated, one should use the UPDATE option

References

1. Barsoum, R. S. "On the Use of Isoparametric Finite Elements in Linear Fracture Mechanics." *Int. J. Num. Methods in Engr.* 10, 1976.

2. Bathe, K. J., and Wilson, E. L. *Numerical Methods in Finite Element Analysis* Prentice-Hall, 1976.

3. Codina, R. "Finite element formulation for the numerical solution of the convection-diffusion equation" 1993.

4. Cornfield, G. C., and Johnson, R. H. "Theoretical Predictions of Plastic Flow in Hot Rolling Including the Effect of Various Temperature Distributions." *Journal of Iron and Steel Institute* 211, pp. 567-573, 1973.

5. Dunham, R. S. and Nickell, R. E. "Finite Element Analysis of Axisymmetric Solids with Arbitrary Loadings." No. 67-6. Structural Engineering Laboratory, University of Berkeley, June, 1967.

6. Hibbitt, H. D., Marcal, P. V. and Rice, J. R. "A Finite Element Formulation for Problems of Large Strain and Large Displacement." *Int. J. Solids Structures* 6 1069-1086, 1970.

7. Houbolt, J. C. "A Recurrence Matrix Solution for the Dynamic Response of Elastic Aircraft." *J. Aero. Sci.* 17, 540-550, 1950.

8. Hsu, M. B. "Modeling of Coupled Thermo-Electrical Problems by the Finite Element Method." Third International Symposium on Numerical Methods for Engineering, Paris, March, 1983.

9. McMeeking, R. M., and Rice, J. R. "Finite-Element Formulations for Problems of Large Elastic-Plastic Deformation." *Int. J. Solids Structures* 11, 601-616, 1975.

10. MARC Update. "The Inverse Power Sweep Method in MARC." U.S. Ed., vol. 3, no. 1, February, 1984.

11. MARC Reference manual Volume F.

12. Marcal, P. V. "Finite Element Analysis of Combined Problems of Nonlinear Material and Geometric Behavior." in *Proceedings of the ASME Computer Conference, Computational Approaches in Applied Mechanics*, Chicago, 1969.

13. Melosh, R. J., and Marcal, P. V. "An Energy Basis for Mesh Refinement of Structural Continua." *Int. J. Num. Meth. Eng.* 11, 1083-1091, 1971.

14. Morman, K. N., Jr., Kao, B. G., and Nagtegaal, J. C. "Finite Element Analysis of Viscoelastic Elastomeric Structures Vibrating about Nonlinear Statically Stressed Configurations." SAE Technical Papers Series 811309, presented at 4th Int. Conference on Vehicle Structural Mechanics, Detroit, November 18-20, 1981.

15. Morman, K. N., Jr., and Nagtegaal, J. C. "Finite Element Analysis of Small Amplitude Vibrations in Pre-Stressed Nonlinear Viscoelastic Solids." *Int. J. Num. Meth. Engng,* 1983.

16. Nagtegaal, J. C. "Introduction in Geometrically Nonlinear Analysis." Int. Seminar on New Developments in the Finite Element Method, Santa Marherita Ligure, Italy, 1980.

17. Nagtegaal, J. C., and de Jong, J. E. "Some Computational Aspects of Elastic-Plastic Large Strain Analysis." in *Computational Methods in Nonlinear Mechanics*, edited by J. T. Oden. North-Holland Publishing Company, 1980.

18. Newmark, N. M. "A Method of Computation for Structural Dynamics." *ASCE of Eng. Mech.* 85, 67-94, 1959.

19. Parks, D. M. "A Stiffness Derivative Finite Element Technique for Determination of Elastic Crack Tip Stress Intensity Factors." *International Journal of Fractures* 10, (4), 487-502, December 1974.

20. Peeters, F. J. H. "Finite Element Analysis of Elasto-Hydrodynamic Lubrication Problems." in *Proceedings of the XIth Int. Finite Element Kongress*, edited by IKOSS GmbH. Baden- Baden, Germany, Nov. 15-16, 1982.

21. Timoshenko, S., Young, D. H., and Weaver, Jr., W. *Vibration Problems in Engineering*. John Wiley, New York: 1979.

22. Wilkinson, J. H. *The Algebraic Eigenvalue Problem*. Oxford: Clarendon Press, 1965.

23. Yu, C C. and Heinrich, J. C. "Petro-Galerkin methods for the time-dependent convective transport equation." Int. J. Numer. Meth. Engrg., Vol. 23 (1986), 883-901.

24. Yu, C C. and Heinrich, J. C. "Petro-Galerkin methods for multidimensional time-dependent convective transport equation." Int. J. Numer. Meth. Engrg., Vol. 24 (1987), 2201-2215.

25. Zienkiewicz, O. C. *The Finite Element Method in Engineering Science*. Third Ed. London: McGraw-Hill, 1978.

26. Zienkiewicz, O. C., and Godbole, P. N. "A Penalty Function Approach to Problems of Plastic Flow of Metals with Large Surface Deformations." *Journal of Strain Analysis* 10, 180-183, 1975.

27. Zienkiewicz, O. C., and Godbole, P. N. "Flow of Plastic and viscoPlastic Solids with Special Reference to Extrusion and Forming Processes." *Int. Num. Methods in Eng.* 8, 1974.

28. Zienkiewicz, O. C., Loehner, R., Morgan, K., and Nakazawa, S. *Finite Elements in Fluid Mechanics – A Decade of Progress*, John Wiley & Sons Limited, 1984.

Chapter 6 MATERIAL LIBRARY

This chapter describes the material models available in the MARC program. The models range from simple linear elastic materials to complex time- and temperature-dependent materials. This chapter provides basic information on the behavior of various types of engineering materials and specifies the data required by the program for each material. For example, to characterize the behavior of an isotropic linear elastic material at constant temperatures, you need only specify Young's modulus and Poisson's ratio. However, much more data is required to simulate the behavior of material that has either temperature or rate effects. References to more detailed information are cited in this chapter.

The material models included in this chapter are listed below:

- Linear elastic
- Composite
- Hypoelastic
- Elastomeric
- Elastic-plastic
- Rate dependent
- Temperature dependent
- Time-Temperature transformation
- Low tension
- Non-structural

Data for these materials is entered to the MARC program either directly through the card deck or by user subroutines. Each section of this chapter discusses various options for organizing material data for input. Each section also discusses the constitutive (stress-strain) relation and graphic representation of the models and includes recommendations and cautions concerning the use of the models.

A 6.1 LINEAR ELASTIC MATERIAL

The MARC program is capable of handling problems with either isotropic linear elastic material behavior or anisotropic linear elastic material behavior.

The linear elastic model is the model most commonly used to represent engineering materials. This model, which has a linear relationship between stresses and strains, is represented by Hooke's Law. Figure A 6.1-1 shows that stress is proportional to strain in a uniaxial tension test. The ratio of stress to strain is the familiar definition of modulus of elasticity (Young's modulus) of the material.

E (modulus of elasticity) = (axial stress)/(axial strain) (A 6.1-1)

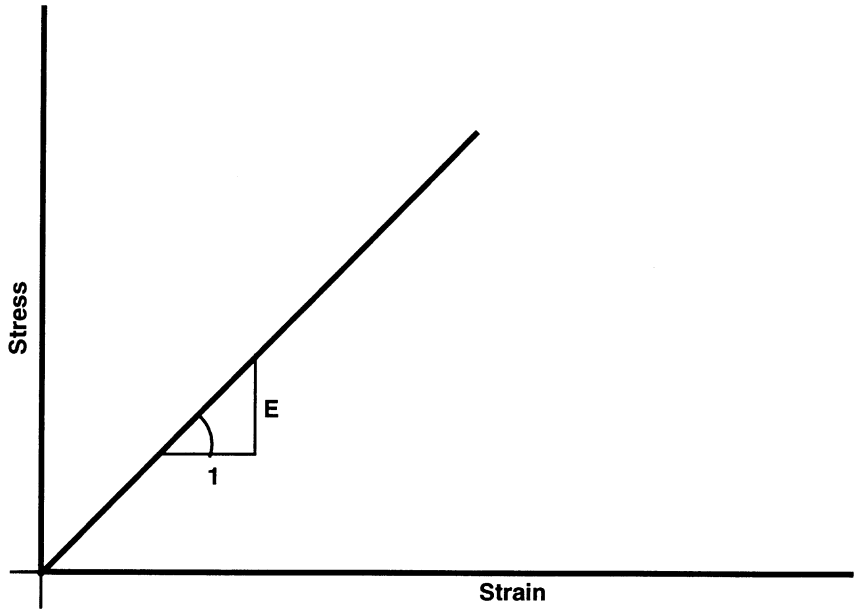


Figure A 6.1-1 Uniaxial Stress-Strain Relation of Linear Elastic Material

Experiments show that axial elongation is always accompanied by lateral contraction of the bar. The ratio for a linear elastic material is:

ν = (lateral contraction)/(axial elongation) (A 6.1-2)

This is known as Poisson's ratio. Similarly, the shear modulus (modulus of rigidity) is defined as:

G (shear modulus) = (shear stress)/(shear strain) (A 6.1-3)

It can be shown that for an isotropic material

$G = E / 2 (1+\nu)$ (A 6.1-4)

The shear modulus G, can be easily calculated if the modulus of elasticity E and Poisson's ratio, ν, are known.

Most linear elastic materials are assumed to be isotropic (their elastic properties are the same in all directions). Anisotropic material exhibits different elastic properties in different directions. The significant directions of the material are labeled as preferred directions, and it is easiest to express the material behavior with respect to these directions.

The stress-strain relationship for an isotropic linear elastic method is expressed as

$$\sigma_{ij} = \lambda\delta_{ij}\varepsilon_{kk} + 2G\varepsilon_{ij} \qquad \text{(A 6.1-5)}$$

where λ is the Lame constant and G (the shear modulus) is expressed as

$$\begin{aligned} \lambda &= \nu E/((1+\nu)(1-2\nu)) \text{ and} \\ G &= E/(2(1+\nu)) \end{aligned} \qquad \text{(A 6.1-6)}$$

The material behavior can be completely defined by the two material constants E and ν.

Use the model definition option ISOTROPIC for the input of isotropic linear elastic material constants E (Young's modulus) and ν (Poisson's ratio).

The stress-strain relationship for an anisotropic linear elastic material can be expressed as

$$\sigma_{ij} = C_{ijkl}\varepsilon_{kl} \qquad \text{(A 6.1-7)}$$

The values of C_{ijkl} (the stress-strain relation) and the preferred directions (if necessary) must be defined for an anisotropic material. For example, the orthotropic stress-strain relationship for a plane stress element is

$$C = \frac{1}{(1-\nu_{12}\nu_{21})} \begin{bmatrix} E_1 & \nu_{21}E_1 & 0 \\ \nu_{12}E_2 & E_2 & 0 \\ 0 & 0 & (1-\nu_{12}\nu_{21})G \end{bmatrix} \qquad \text{(A 6.1-8)}$$

To input anisotropic stress-strain relations, use the model definition options ORTHOTROPIC or ANISOTROPIC and user subroutine ANELAS, or user subroutine HOOKLW. The ORTHOTROPIC option allows as many as nine elastic constants to be defined. The ANISOTROPIC option allows as many as 21 elastic constants to be defined. If the anisotropic material has a preferred direction, use the model definition option ORIENTATION or the user subroutine ORIENT to input a transformation matrix. Refer to Volume D for information on user subroutines.

A Poisson's ratio of 0.5, which would be appropriate for an incompressible material, may be used for the following elements: Herrmann, plane stress, shell, truss, or beam. A Poisson's ratio which is close (but not equal) to 0.5 may be used for constant dilation elements and reduced integration elements in situations which do not include other severe kinematic constraints. Using a Poisson's ratio close to 0.5 for all other elements usually leads to behavior that is too stiff.

A 6.2 COMPOSITE MATERIAL

Composite materials are composed of layers of different materials (or layers of the same anisotropic material) with various layer thicknesses and different orientations. The material in each layer may be either linear or nonlinear. Tightly bonded layers (laminated composites) are often stacked in the thickness direction of beam, plate, or shell structures.

Figure A 6.2-1 identifies the locations of integration points through the thickness of beam and shell elements with/without the COMPOSITE option.

Note that when the COMPOSITE option is used, as shown on the left, the layer points are positioned midway through each layer. When the COMPOSITE option is not used, the layer points are equidistantly spaced between the top and bottom surfaces. MARC forms a stress-strain law by performing numerical integration through the thickness. If the COMPOSITE option is used, the trapezoidal method is employed; otherwise, Simpson's rule is used.

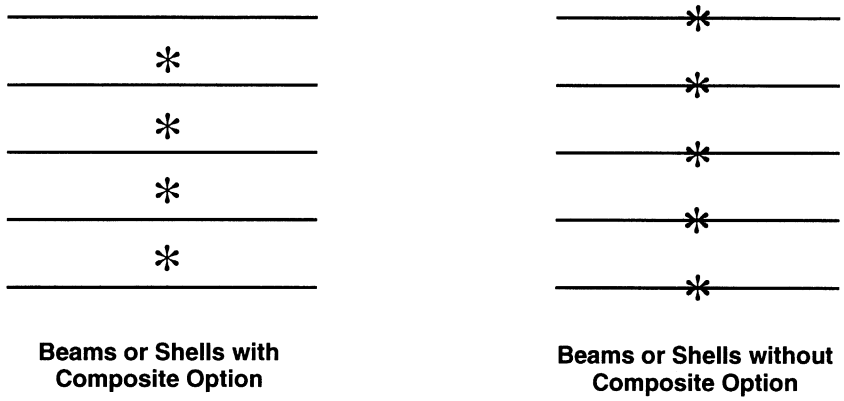


Figure A 6.2-1 Integration Points Through the Thickness of Beam and Shell Elements

Laminated Composite

To model laminated composite plates, shells, or beams with MARC, use the COMPOSITE option. In this option, three quantities are specified on a layer-by-layer basis: material identification number, layer thickness, and ply angle. The entire set of data (a "composite group") is then associated with a list of elements. For each individual layer, various constitutive laws may be used. The layer thickness may be constant or variable (in the case of variable total thickness elements), and the ply angle may change from one layer to the next. The orientation of the 0^{o} ply angle within each element is defined in the ORIENTATION option.

The material identification number specified in the COMPOSITE option, is cross-referenced with the material identification number supplied in the ISOTROPIC, ORTHOTROPIC, ANISOTROPIC, TEMPERATURE EFFECTS, ORTHO TEMP, WORK HARD, and RATE EFFECTS options. The ISOTROPIC, ORTHOTROPIC and ANISOTROPIC model definition options allow the user to input material constants such as Young's modulus, Poisson's ratio, shear modulus, etc. The TEMPERATURE EFFECTS and ORTHO TEMP options allow for input of temperature dependency of these material constants. Material constants for a typical layer are as follows:

t_i	thickness of the i^{th} layer
Young's moduli,	E_{xx}, E_{yy}, E_{zz}, ν_{xy}, ν_{yz}, ν_{zx},
Poisson's ratios, and shear moduli	G_{xy}, G_{yz}, G_{zx}
ρ	density
a_{xx}, a_{yy}	coefficients of thermal expansion
σ_y	yield stress
Mat	material identifier associated with temperature-dependent properties and work hardening data

User subroutines ANELAS, HOOKLW, ANEXP, and ANPLAS can be used for the anisotropic behavior of elastic constants, coefficient of thermal expansion, and yield condition.

There are seven given classes of strain-stress relations. The class of a particular element depends on the number of direct (NDI) and shear (NSHEAR) components of stress. Table A 6.2-1 lists the seven classes of elements.

Material Preferred Direction

Every element type in MARC has a default orientation (that is, a default coordinate system) within which element stress-strain calculations take place. This system is also assumed to be the coordinate system of material symmetry. This is especially important for non-isotropic materials (orthotropic, anisotropic, or composite materials). With the ORIENTATION option, the user specifies the orientation of the material axes of symmetry (relationship between the element coordinate system and the global coordinate system, or the 0^o ply angle line, if COMPOSITE) in one of four different ways:

1. as a specific angle offset from an element edge,
2. as a specific angle offset from the line created by two intersecting planes,
3. as a particular coordinate system specified by user-supplied unit vectors, or
4. as specified by user subroutine ORIENT. This is accomplished by the specification of an orientation type, an orientation angle, or one or two user defined vectors.

For the first option (EDGE I-J orientation type), the intersecting plane is defined by the surface normal vector and a vector parallel to the vector pointing from element node I to element node J. The intersection of this plane with the surface tangent plane defines the 0^o orientation axis. (See Figure A 6.2-2.) The orientation angle is measured in the tangent plane positive about the surface normal.

Table A 6.2-1 Classes of Stress-Strain Relations

Class 1	NDI = 1,NSHEAR = 0 Beam Elements 5, 8, 13, 16, 23, 46, 47, 48, 52, 64, 77, 79 $\{\varepsilon\} = [1/E_{xx}]\{\sigma\}$
Class 2	NDI = 2,NSHEAR = 0 Axisymmetric Shells 15 and 17 $\begin{Bmatrix}\varepsilon_{xx}\\ \varepsilon_{yy}\end{Bmatrix} = \begin{bmatrix}1/E_{xx} & -\nu_{yx}/E_{yy}\\ -\nu_{xy}/E_{xx} & 1./E_{yy}\end{bmatrix}\begin{Bmatrix}\sigma_{xx}\\ \sigma_{yy}\end{Bmatrix}$ $\nu_{yx} = \nu_{xy}E_{yy}/E_{xx}$
Class 3	NDI = 1,NSHEAR = 1 Beam Elements 14, 45, 76, 78 $\begin{Bmatrix}\varepsilon\\ \gamma\end{Bmatrix} = \begin{bmatrix}1./E_{xx} & 0.\\ 0. & 1./G_{xy}\end{bmatrix}\begin{Bmatrix}\sigma\\ \tau\end{Bmatrix}$
Class 4	NDI = 1,NSHEAR = 1 Plane Stress, Plates and Thin Shells 49 and 72 $\begin{Bmatrix}\varepsilon_{xx}\\ \varepsilon_{yy}\\ \varepsilon_{xy}\end{Bmatrix} = \begin{bmatrix}1./E_{xx} & -\gamma_{yx}/E_{yy} & 0.\\ -\gamma_{xy}/E_{xx} & 1./E_{yy} & 0.\\ 0. & 0. & 1./G_{xy}\end{bmatrix}\begin{Bmatrix}\sigma_{xx}\\ \sigma_{yy}\\ \tau_{xy}\end{Bmatrix}$ $\nu = \nu_{xy}(E_{yy}/E_{xx})$
Class 5	NDI = 2,NSHEAR = 1 Thick Axisymmetric Shells 1 and 89 $\begin{Bmatrix}\varepsilon_{mm}\\ \varepsilon_{\theta\theta}\\ \gamma_T\end{Bmatrix} = \begin{bmatrix}1./E_{mm} & -\nu_{\theta m}/E_{\theta\theta} & 0.\\ -\nu_{m\theta}/E_{mm} & 1./E_{\theta\theta} & 0.\\ 0. & 0. & 1/G_{m\theta}\end{bmatrix}\begin{Bmatrix}\sigma_{mm}\\ \sigma_{\theta\theta}\\ \tau_T\end{Bmatrix}$

Table A 6.2-1 Classes of Stress-Strain Relations (Continued)

Class 6	NDI = 3,NSHEAR = 1 Plain Strain, Axisymmetric with No Twist $\begin{Bmatrix}\varepsilon_{xx}\\ \varepsilon_{yy}\\ \varepsilon_{zz}\\ \gamma_{xy}\end{Bmatrix} = \begin{bmatrix} 1./E_{xx} & -\gamma_{yx}/E_{yy} & -\nu_{zx}/E_{zz} & 0. \\ -\nu_{zy}/E_{xx} & 1./E_{yy} & -\nu_{zy}/E_{zz} & 0. \\ -\nu_{xz}/E_{xx} & -\nu_{yz}/E_{yy} & 1./E_{zz} & 0. \\ 0. & 0. & 0. & 1./G_{xy} \end{bmatrix} \begin{Bmatrix}\sigma_{xx}\\ \sigma_{yy}\\ \sigma_{zz}\\ \tau_{zx}\end{Bmatrix}$ $\nu_{yx} = \nu_{xy}E_{yy}/E_{xx} \quad \nu_{zy} = \nu_{yz}E_{zz}/E_{yy} \quad \nu_{xz} = \nu_{zx}E_{xx}/E_{zz}$
Class 7	NDI = 2,NSHEAR = 3 Thick Shell, Element 22, 75 $\begin{Bmatrix}\varepsilon_{xx}\\ \varepsilon_{yy}\\ \gamma_{xy}\\ \gamma_{yz}\\ \gamma_{zx}\end{Bmatrix} = \begin{bmatrix} 1./E_{xx} & -\nu_{yx}/E_{xx} & 0. & 0. & 0. \\ -\nu_{xy}/E_{xx} & 1./E_{yy} & 0. & 0. & 0. \\ 0. & 0. & 1./G_{xy} & 0. & 0 \\ 0. & 0. & 0. & 1./G_{yz} & 0. \\ 0. & 0. & 0. & 0. & 1./G_{zx} \end{bmatrix} \begin{Bmatrix}\sigma_{xx}\\ \sigma_{yy}\\ \tau_{xy}\\ \tau_{yz}\\ \tau_{zx}\end{Bmatrix}$ $\nu_{xy} = \nu_{xy}E_{yy}/E_{xx}$
Class 8	NDI = 3,NSHEAR = 3 Three-Dimensional Brick Elements, Axisymmetric with Twist $\begin{Bmatrix}\varepsilon_{xx}\\ \varepsilon_{yy}\\ \varepsilon_{zz}\\ \gamma_{xy}\\ \gamma_{yz}\\ \gamma_{zx}\end{Bmatrix} = \begin{bmatrix} 1./E_{xx} & -\nu_{yx}/E_{zz} & -\nu_{zx}/E_{zz} & 0. & 0. & 0. \\ -\nu_{xy}/E_{xx} & 1./E_{yy} & -\nu_{zy}/E_{zz} & 0. & 0. & 0. \\ -\nu_{xz}/E_{xx} & -\nu_{yz}/E_{yy} & 1./E_{zz} & 0. & 0. & 0. \\ 0. & 0. & 0. & 1./G_{xy} & 0. & 0. \\ 0. & 0. & 0. & 0. & 1./G_{yz} & 0. \\ 0. & 0. & 0. & 0. & 0. & 1./G_{zx} \end{bmatrix} \begin{Bmatrix}\sigma_{xx}\\ \sigma_{yy}\\ \sigma_{zz}\\ \tau_{xy}\\ \tau_{yz}\\ \tau_{zx}\end{Bmatrix}$

For the second option (global plane orientation type), the intersecting plane is the chosen global coordinate plane. The intersection of this plane with the surface tangent plane defines the 0^o orientation axis. (See Figure A 6.2-3.)

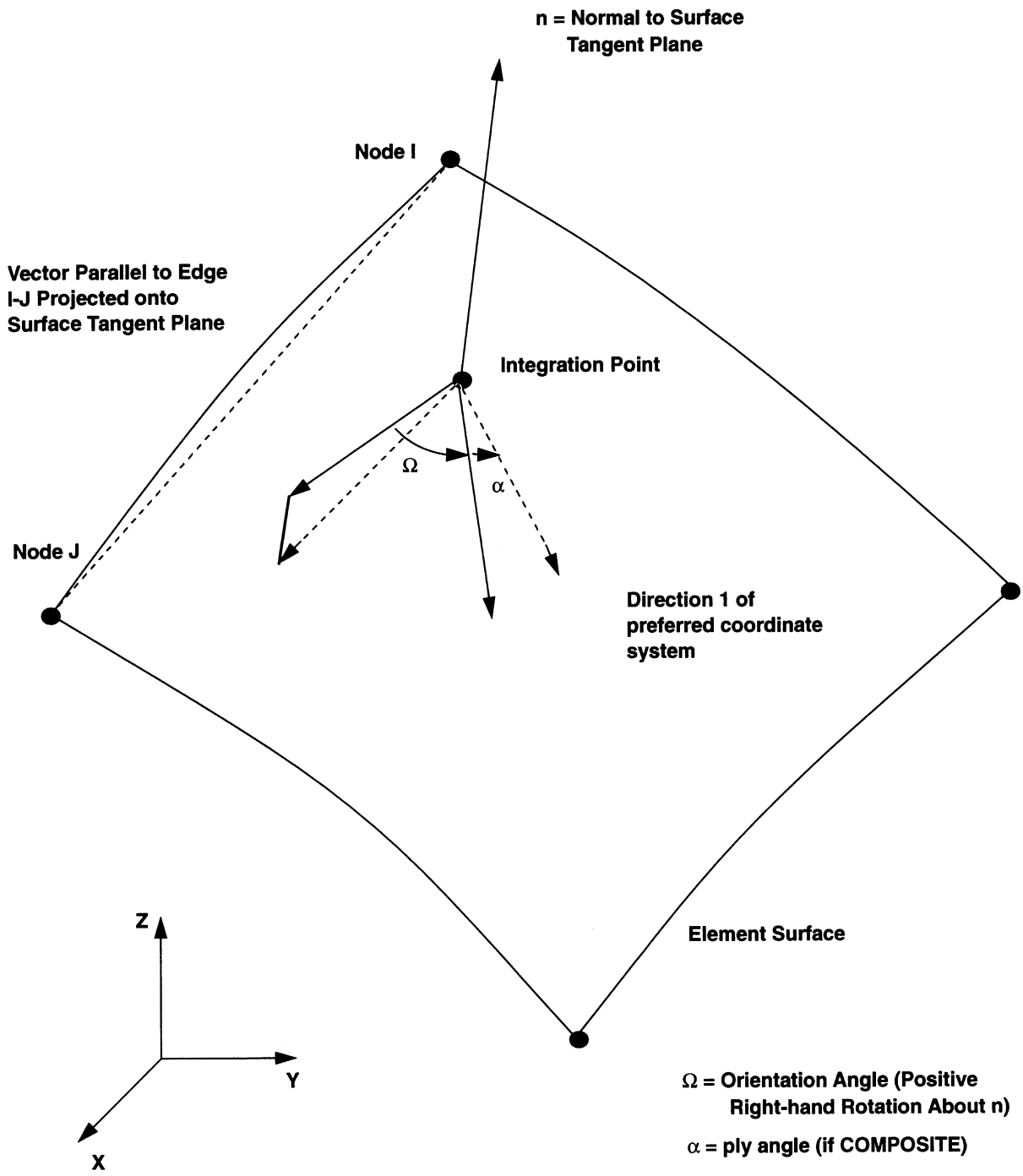


Figure A 6.2-2 Edge I-J Orientation Type

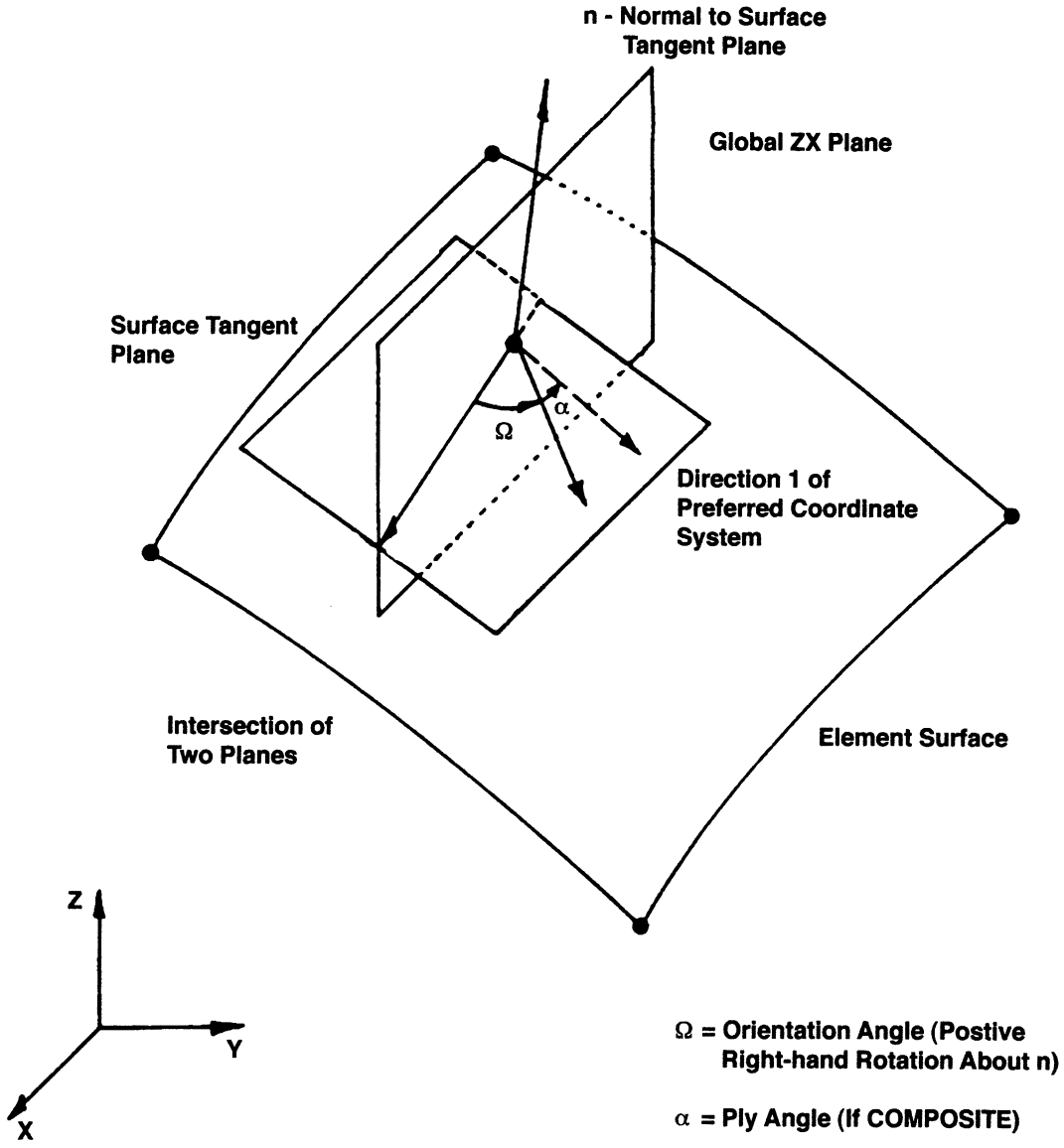


Figure A 6.2-3 Global ZX Plane Orientation Type

The third option (user defined plane orientation type) makes use of one or two user defined vectors to define the intersecting plane. Using a single vector, the intersecting plane is that plane which contains the user vector and the chosen coordinate axis. Using two user vectors, the intersecting plane is that plane which contains both of them. (See Figure A 6.2-4.)

Orientation type 3D ANISO also makes use of two user defined vectors, but in this case, the first vector defines the first (1) principal direction and the second vector defines the second (2) principal direction. (See Figure A 6.2-5.)

Finally, in the fourth option, the user subroutine ORIENT must be used for the definition of the orientation of the material axis of symmetry.

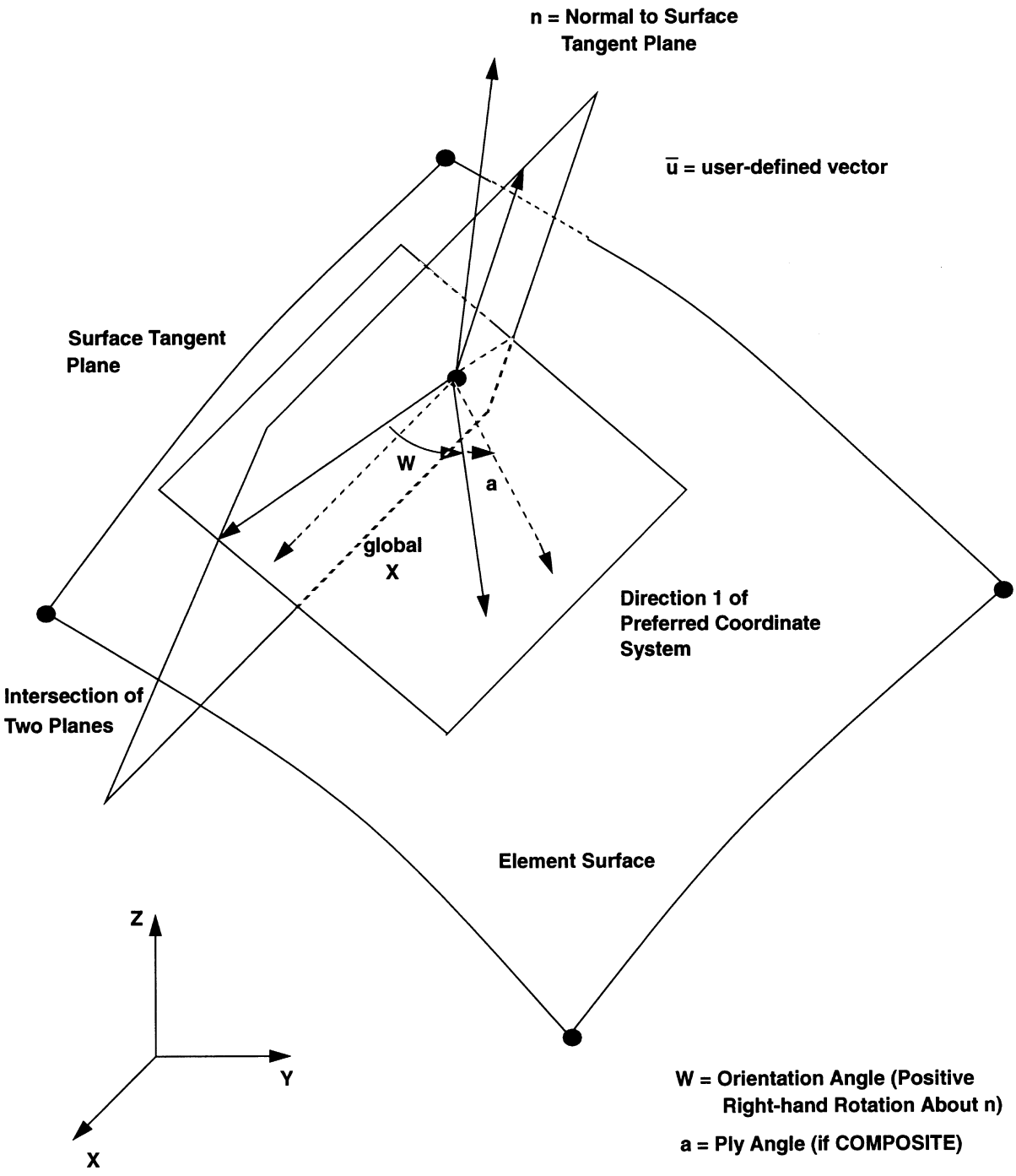


Figure A 6.2-4 User Defined XU Plane Orientation Type

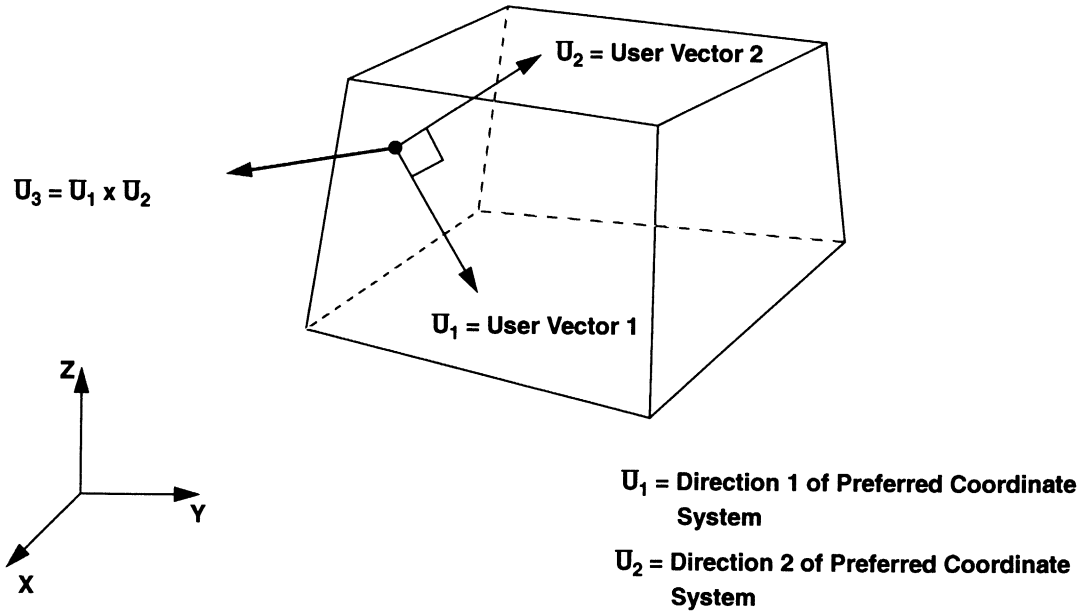


Figure A 6.2-5 3D ANISO Orientation Type

Material Dependent Failure Criteria

Calculations of user specified failure criteria on a layer by layer basis are available in the MARC program. They are maximum stress (MX STRESS), maximum strain (MX STRAIN), TSAI-WU, HOFFMAN, HILL or user-subroutine UFAIL. During each analysis, up to three failure criteria may be selected; failure indices are calculated and printed for every integration point. The model definition block FAIL DATA is used for the input of failure criteria data.

A simple description of these failure criteria is given below:

1. Maximum Stress Criterion

   At each integration point, MARC calculates six quantities:

$$1. \begin{cases} \left(\frac{\sigma_1}{X_t}\right)/F & \text{if} \quad \sigma_1 > 0 \\ \left(-\frac{\sigma_1}{X_c}\right)/F & \text{if} \quad \sigma_1 < 0 \end{cases} \tag{A 6.2-1}$$

$$2. \begin{cases} \left(\frac{\sigma_2}{Y_t}\right)/F & \text{if} \quad \sigma_2 > 0 \\ \left(-\frac{\sigma_2}{Y_c}\right)/F & \text{if} \quad \sigma_2 < 0 \end{cases} \tag{A 6.2-2}$$

3. $$\begin{cases} \left(\frac{\sigma_3}{Z_t}\right)/F & \text{if} \quad \sigma_3 > 0 \\ \left(-\frac{\sigma_3}{Z_c}\right)/F & \text{if} \quad \sigma_3 < 0 \end{cases} \qquad \text{(A 6.2-3)}$$

4. $$\left(\left|\frac{\sigma_{12}}{S_{xy}}\right|\right)/F \qquad \text{(A 6.2-4)}$$

5. $$\left(\left|\frac{\sigma_{23}}{S_{yz}}\right|\right)/F \qquad \text{(A 6.2-5)}$$

6. $$\left(\left|\frac{\sigma_{31}}{S_{zx}}\right|\right)/F \qquad \text{(A 6.2-6)}$$

where

F	is the failure index (normally, F =1.0).
X_t, X_c	are the maximum allowable stresses in the 1-direction in tension and compression.
Y_t, Y_c	are maximum allowable stresses in the 2-direction in tension and compression.
Z_1, Z_c	are maximum allowed stresses in the 3-direction in tension and compression.
S_{xy}	maximum allowable in-plane shear stress.
S_{yz}	maximum allowable yz shear stress.
S_{zx}	maximum allowable zx shear stress.

2. Maximum Strain Failure Criterion

At each integration point, MARC calculates six quantities:

1. $$\begin{cases} \left(\frac{\varepsilon_1}{e_{1t}}\right)/F & \text{if} \quad \varepsilon_1 > 0 \\ \left(-\frac{\varepsilon_1}{e_{1c}}\right)/F & \text{if} \quad \varepsilon_1 < 0 \end{cases} \qquad \text{(A 6.2-7)}$$

2. $$\begin{cases} \left(\frac{\varepsilon_2}{e_{2t}}\right)/F & \text{if} \quad \varepsilon_2 > 0 \\ \left(-\frac{\varepsilon_2}{e_{2c}}\right)/F & \text{if} \quad \varepsilon_2 < 0 \end{cases} \qquad \text{(A 6.2-8)}$$

3. $$\begin{cases} \left(\frac{\varepsilon_3}{e_{3t}}\right)/F & \text{if} \quad \varepsilon_3 > 0 \\ \left(-\frac{\varepsilon_3}{e_{3c}}\right)/F & \text{if} \quad \varepsilon_3 < 0 \end{cases} \qquad \text{(A 6.2-9)}$$

4. $$\left(\left|\frac{\gamma_{12}}{g_{xy}}\right|\right)/F \qquad \text{(A 6.2-10)}$$

5. $$\left(\left|\frac{\gamma_{23}}{g_{yz}}\right|\right)/F \qquad \text{(A 6.2-11)}$$

6. $$\left(\left|\frac{\gamma_{31}}{g_{zx}}\right|\right)/F \qquad \text{(A 6.2-12)}$$

where

F	is the failure index (normally, F=1.0).
e_{1t}, e_{1c}	are the maximum allowable strains in the 1 direction in tension and compression.
e_{2t}, e_{2c}	are maximum allowable strains in the 2 direction in tension and compression.
e_{3t}, e_{3c}	are maximum allowable strains in the 3 direction in tension and compression.
g_{xy}	is the maximum allowable shear strain in the xy plane.
g_{yz}	is the maximum allowable shear strain in yz plane.
g_{zx}	is the maximum allowable shear strain in zx plane.

3. Hill Failure Criterion

Assumptions:

a. Orthotropic materials only
b. Incompressibility during plastic deformation
c. Tensile and compressive behaviors are identical

At each integration point, MARC calculates:

$$\left(\frac{\sigma_1^2}{X^2} - \frac{\sigma_1\sigma_2}{X^2} + \frac{\sigma_2^2}{Y^2} + \frac{\sigma_{12}^2}{S^2}\right)/F \qquad \text{(A 6.2-13)}$$

where

F	is the failure index
X	is the maximum allowable stress in the 1 direction
Y	is the maximum allowable stress in the 2 direction
S	is the maximum allowable shear stress

4. Hoffman Failure Criterion

NOTE

Hoffman criterion is essentially Hill criterion modified to allow unequal maximum allowable stresses in tension and compression.

At each integration point, MARC calculates:

$$\{ (\frac{1}{X_t} - \frac{1}{X_c}) \sigma_1 + (\frac{1}{Y_t} - \frac{1}{Y_c}) \sigma_2 + \frac{\sigma_1^2}{X_t X_c} + \frac{\sigma_2^2}{Y_t Y_c} + \frac{\sigma_{12}^2}{S^2} - \frac{\sigma_1 \sigma_2}{X_t X_c} \} / F \qquad \text{(A 6.2-14)}$$

where: X_t, X_c, Y_t, Y_c, S, F are as before.

NOTE

For small ratios of, for example, $\frac{\sigma_1}{X_t}$, the Hoffman criteria may become *negative* due to the presence of the linear terms.

5. Tsai-Wu Failure Criterion

Tsai-Wu is a tensor polynomial failure criterion. At each integration point, MARC calculates:

$$\{ (\frac{1}{X_t} - \frac{1}{X_c}) \sigma_1 + (\frac{1}{Y_2} - \frac{1}{Y_c}) \sigma_2 + \frac{\sigma_1^2}{X_t X_c} + \frac{\sigma_2^2}{Y_t Y_c} + \frac{\sigma_{12}^2}{S_2} + 2F_2 \sigma_1 \sigma_2 \} / F \qquad \text{(A 6.2-15)}$$

where: F_{12} is the "normal stress interaction" parameter.

NOTE

In order for the Tsai-Wu failure surface to be closed,

$$F_{12}^2 < \frac{1}{X_t X_c} \bullet \frac{1}{Y_t Y_c}$$

See Wu, R.Y. and Stachurski, 2, *Evaluation of the Normal Stress Interaction Parameter in the Tensor Polynomial Strength Theory for Anisotropic Materials*, *Journal of Composite Materials*, Vol. 18, Sept. 1984, pp. 456-463.

6. User Defined Failure Criteria

   Using user subroutine UFAIL, the user may evaluate his own failure criterion as a function of stresses and strains at each integration point.

Interlaminar Shear for Thick Shell and Beam Elements

Another addition made for composite analysis is the calculation of interlaminar shears (a parabolic distribution through the thickness direction) for thick shells, elements 22 and 75, and for thick beam element 45. These interlaminar shears are printed in the local MARC coordinate system above and below each layer selected for printing by PRINT CHOICE or PRINT ELEM. These values will also be available for post-processing. The parameter SHEAR must be used for activating the parabolic shear distribution calculations.

In MARC, the distribution of transverse shear strains through the thickness for thick shell and beam elements was assumed to be constant. From basic strength of materials and the equilibrium of a beam cross-section, it is known that the actual distribution is more parabolic in nature. As an additional option, the formulations for elements 22, 75, and 45 have been modified to include a parabolic distribution of transverse shear strain. The formulation is exact for beam element 45, but is approximate for the thick shell elements 22 and 75. Nevertheless, the approximation is expected to give improved results from the previous constant shear distribution. Furthermore, interlaminar shear stresses for composite beams and shells may now be easily calculated.

With the assumption that the stresses in the V^1 and V^2 direction are uncoupled, the equilibrium condition through the thickness is given by

$$\frac{\partial \tau(z)}{\partial z} + \frac{\partial \sigma(z)}{\partial x} = 0 \qquad \text{(A 6.2-16)}$$

where $\sigma(z)$ is the layer axial stress; $\tau(t)$ is the layer shear stress. From beam theory, we have

$$V + \frac{\partial M}{\partial x} = 0 \qquad \text{(A 6.2-17)}$$

where M is the section bending moment and V is the shear force. Assuming that

$$\sigma(z) = f(z)\,M \qquad \text{(A 6.2-18)}$$

by taking the derivative of Equation A 6.2-18 with respect to x, substituting the result into Equation A 6.2-16, using Equation A 6.2-17 and integrating, we obtain

$$\tau(z) = \int_z f(z)\,dz \bullet V \qquad \text{(A 6.2-19)}$$

The function f(z) is given from beam theory as

$$f(z) = \frac{E_0(z)}{\bar{EI}} (\bar{z} - z) \qquad \text{(A 6.2-20)}$$

where $E_0(z)$ is the layer initial Young's Modulus, $\bar{z}$ is the location of the neutral axis and $\bar{EI}$ is the section bending moment of inertia. Equation A 6.2-20 and Equation A 6.2-18 express the usual bending relation

$$\sigma(z) = -\frac{Mz}{I} \qquad \text{(A 6.2-21)}$$

except that these two equations are written so that the z = 0 axis is not necessarily the neutral axis of bending. With respect to this axis, membrane and bending action is, in general, coupled. Note that

$$\bar{z} = \frac{\int_z E(z)\, dz}{\int_z zE(z)\, dz} \qquad \text{(A 6.2-22)}$$

and stress $\tau(z) = 0$ at the top and bottom surface of the shell.

Progressive Composite Failure

A model has been put into MARC to allow the progressive failure of certain types of composite materials. The aspects of this model are defined below

1. Failure occurs when any one of the failure criteria is satisfied.
2. The behavior up to the failure point is linear elastic.
3. Upon failure, the material moduli for orthotropic materials at the integration points are changed such that all of the moduli have the lowest moduli entered.
4. Upon failure, for isotropic materials, the failed moduli are taken as 10% of the original moduli.
5. If there is only one modulus, such as in a beam or truss problem, the failed modulus is taken as 10% of the original one.
6. There is no healing of the material.

This option is flagged through the FAIL DATA model definition option.

A 6.3 NONLINEAR HYPOELASTIC MATERIAL

The hypoelastic model is able to represent a nonlinear elastic (reversible) material behavior. For this constitutive theory, MARC assumes that

$$\dot{\sigma}_{ij} = L_{ijkl}\dot{\varepsilon}_{kl} + g_{ij} \qquad \text{(A 6.3-1)}$$

where L is a function of the elastic strain and g is a function of the temperature.

When used in conjunction with the large displacement option, Equation A 6.3-1 is expressed as

$$\dot{S}_{ij} = L_{ijkl}\dot{E}_{kl} + g_{ij} \qquad \text{(A 6.3-2)}$$

where E, S are the Green-Lagrangian strain and second Piola-Kirchhoff stress, respectively.

The HYPOELASTIC model definition option is necessary to invoke this model. This model may be used with any stress element, including Herrmann formulation elements.

The tensors L and g are defined by the user in user subroutine HYPELA. In order to provide an accurate solution, L should be a tangent stiffness evaluated at the beginning of the iteration. In addition, the total stress should be defined as its exact value at the end of the increment. This allows the residual load correction to work effectively.

A 6.4 ELASTOMER

An elastomer is a polymer which shows nonlinear elastic stress-strain behavior. The term elastomer is often used to refer to materials which show a rubber-like behavior, even though no rubbers exist which show a purely elastic behavior. Depending upon the type of rubber, elastomers show a more or less strongly pronounced viscoelastic behavior. The MARC program considers both the viscous effects and the elastic aspects of the materials behavior. These materials are characterized by their elastic strain energy function.

Elastomeric materials are elastic in the classical sense. Upon unloading, the stress-strain curve is retraced and there is no permanent deformation. Elastomeric materials are initially isotropic. Figure A 6.4-1 shows a typical stress-strain curve for an elastomeric material.

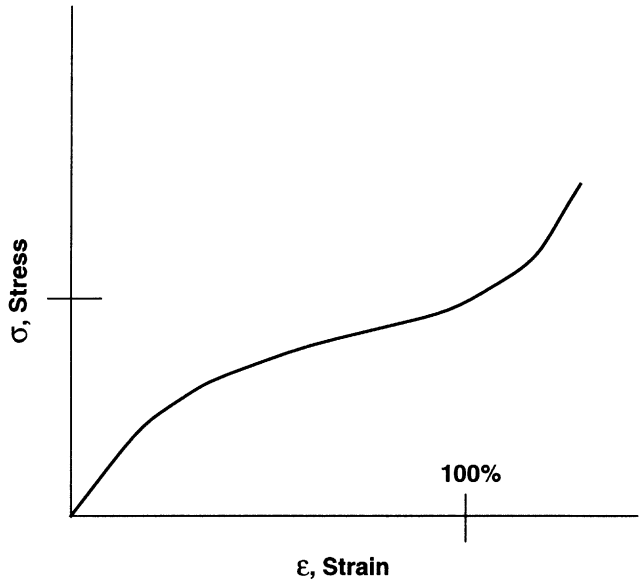


Figure A 6.4-1 A Typical Stress-Strain Curve for an Elastomeric Material

To simulate elastomeric materials, incompressible element(s) are used for plane strain, axisymmetric, and three-dimensional problems. These elements may be used with each other or in combination with other elements in the library. For plane stress, beam, plate or shell analysis, conventional elements may be used.

There are three elastomer modes in MARC. The first is the third-order deformation form, considered by Jamus, Green, and Simpson.

$$W = C_{10}(I_1 - 3) + C_{01}(I_2 - 3) + C_{11}(I_1 - 3)(I_2 - 3) + C_{20}(I_1 - 3)^2 + C_{30}(I_1 - 3)^3 \qquad \text{(A 6.4-1)}$$

where

W	is the strain energy function,
$C_{10}, C_{01}, C_{11}, C_{20}, C_{30}$	are material constants obtained from experimental data
I_1, I_2	are the first and second invariants of the elastic strain.

Simpler forms of the above strain energy function are

$$W = C_{10}(I_1 - 3) + C_{01}(I_2 - 3) \quad \text{Mooney-Rivlin}$$
$$W = C_{10}(I_1 - 3) \quad \text{Neo-Hookean} \qquad \text{(A 6.4-2)}$$

Use the MOONEY model definition option to activate the elastomeric material option in the MARC program and enter the material constants C_{10}, C_{01}, C_{11}, C_{20}, C_{30}.

The TEMPERATURE EFFECTS model definition option can be used to input the temperature dependency of the constants C_{10} and C_{01}. The user subroutine UMOONY can be used to modify all five constants C_{01}, C_{10}, C_{11}, C_{20}, and C_{30}. For viscoelastic, the additional model definition option VISCEL MOON must be included.

The original Ogden strain energy form with N terms is

$$W = \sum_{n=1}^{N} \frac{\mu_n}{\alpha_n}(\lambda_1^{\alpha_n} + \lambda_2^{\alpha_n} + \lambda_3^{\alpha_n} - 3) \qquad \text{(A 6.4-3)}$$

This has been enhanced to allow dilatational behavior.

$$W = \sum_{n=1}^{N} \frac{\mu_n}{\alpha_n} J^{-\alpha_n/3}(\lambda_1^{\alpha_n} + \lambda_2^{\alpha_n} + \lambda_3^{\alpha_n} - 3) + 4.5K(J^{-1/3} - 1)^2 \qquad \text{(A 6.4-4)}$$

where α_i, μ_i are material constants, and K is the bulk modulus. If no bulk modulus is given, it is taken to be virtually incompressible. This model is different from the Mooney model in several respects. The MOONEY option is with respect to the invariants of the right Cauchy-Green strain tensor and implicitly assumes that the material is incompressible. The Ogden formulation is with respect to the eigenvalues of the right Cauchy-Green strain, and the presence of the bulk modulus implies some compressibility. Using a two-term series will result in identical behavior as the Mooney model. The material data is given through the OGDEN model definition option, or user subroutine UOGDEN. For viscoelastic behavior, the additional model definition option VISCELOGDEN must be included.

Additionally, a damage model may be included with the Ogden model.

The rubber foam model which is based on Ogden formulation has a strain energy form as follows:

$$W = \sum_{n=1}^{N} \frac{\mu_n}{\alpha_n}(\lambda_1^{\alpha_n} + \lambda_2^{\alpha_n} + \lambda_3^{\alpha_n} - 3) + \sum_{n=1}^{N} \frac{\mu_n}{\beta_n}(1 - J^{\beta_n}) \qquad \text{(A 6.4-5)}$$

where μ_n, α_n, β_n are material constants. The model reduces to incompressible Ogden model when β_n equals zero.

If the strain energy function expressed by Equation A 6.4-1, Equation A 6.4-3, or Equation A 6.4-5 is not adequate, user subroutine UENERG can be used to define the strain energy function.

The stress may be calculated from the strain energy expressed by Equation A 6.4-6. Since it is a nonlinear relation, you must perform an incremental procedure.

$$s_{ij} = \frac{\partial W}{\partial E_{ij}} \qquad \text{(A 6.4-6)}$$

MARC internally calculates the stress-strain relation

$$\dot{S}_{ij} = L_{ijkl}\dot{E}_{kl} \qquad \text{(A 6.4-7)}$$

Strain invariants in elastomers are discussed below. In the rectangular block in Figure 6.4-2, λ_1, λ_2, and λ_3 are the principal stretch ratios along the edges of the block defined by

$$\lambda_i = (l_i + u_i)/l_i \qquad \text{(A 6.4-8)}$$

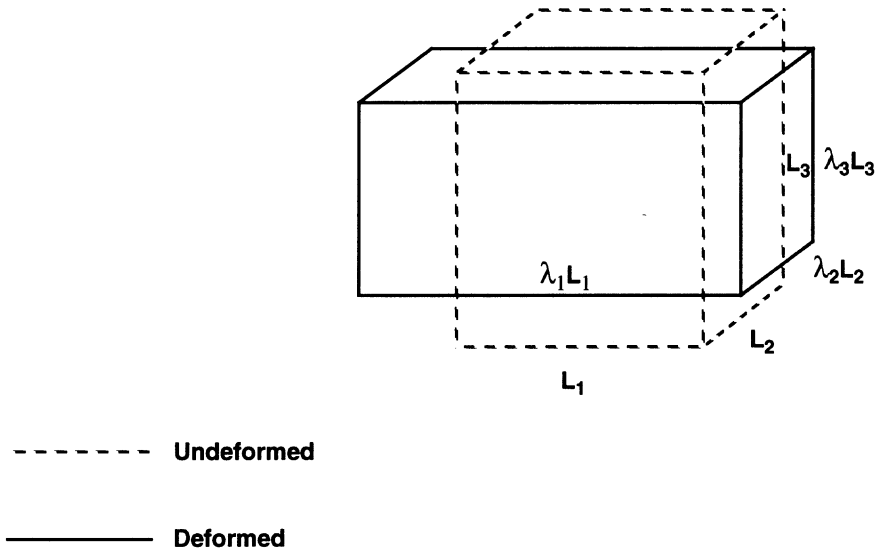


Figure A 6.4-2 Rectangular Rubber Block

In practice, the material behavior is (approximately) incompressible, leading to the constraint equation

$$\lambda_1\lambda_2\lambda_3 = 1 \qquad \text{(A 6.4-9)}$$

the strain invariants are defined as

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

$$I_2 = \lambda_1^2\lambda_2^2 + \lambda_2^2\lambda_3^2 + \lambda_3^2\lambda_1^2 \qquad \text{(A 6.4-10)}$$

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

The material constants for the Mooney-Rivlin form may be obtained from experimental data. The Mooney-Rivlin form of the strain energy density function is

$$W = C_{10}(I_1 - 3) + C_{01}(I_2 - 3) \qquad \text{(A 6.4-11)}$$

For the Mooney-Rivlin model, the force and deformation for a uniaxial test specimen may be related as

$$P = 2A_0\left(1 - \frac{1}{\lambda_1^3}\right)(\lambda C_{10} + C_{01}) \qquad \text{(A 6.4-12)}$$

where P is the force of the specimen, A_0 is the original area of the specimen, and λ_1 is the uniaxial stretch ratio. This equation provides a simple way to determine the Mooney-Rivlin constants. The Mooney-Rivlin constitutive equation is applicable if the plot of

$$P/2A_0\left(1 - \frac{1}{\lambda_1^3}\right) \qquad \text{(A 6.4-13)}$$

versus the stretch ratio results in an approximately straight line, as shown in Figure A 6.4-3.

If only the Young's modulus *E* is supplied, and full uniaxial data are not available then

$$C_1 \cong 0.25C_{10} \qquad \text{(A 6.4-14)}$$

is a reasonable assumption. The constants then follow from the relation:

$$6(C_{10} + C_{01}) \cong E \qquad \text{(A 6.4-15)}$$

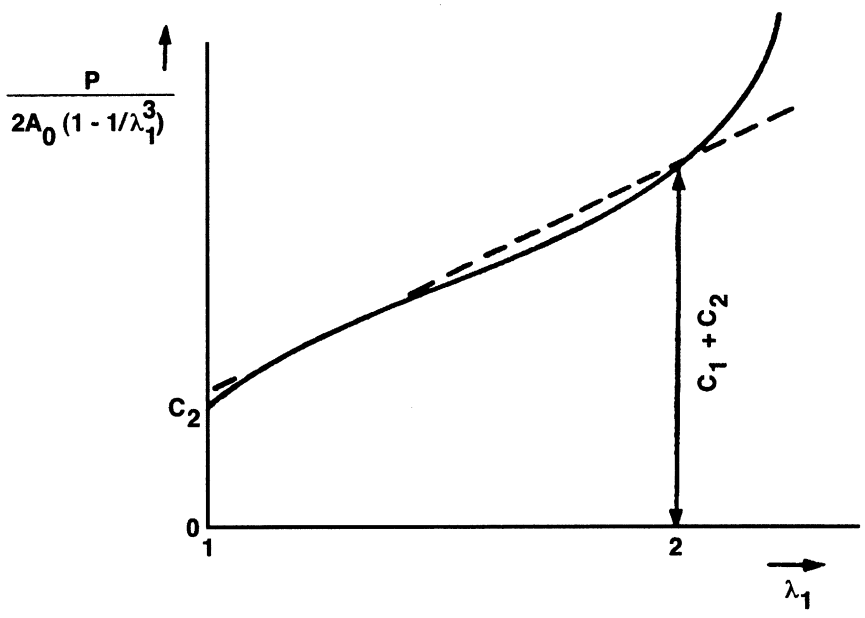
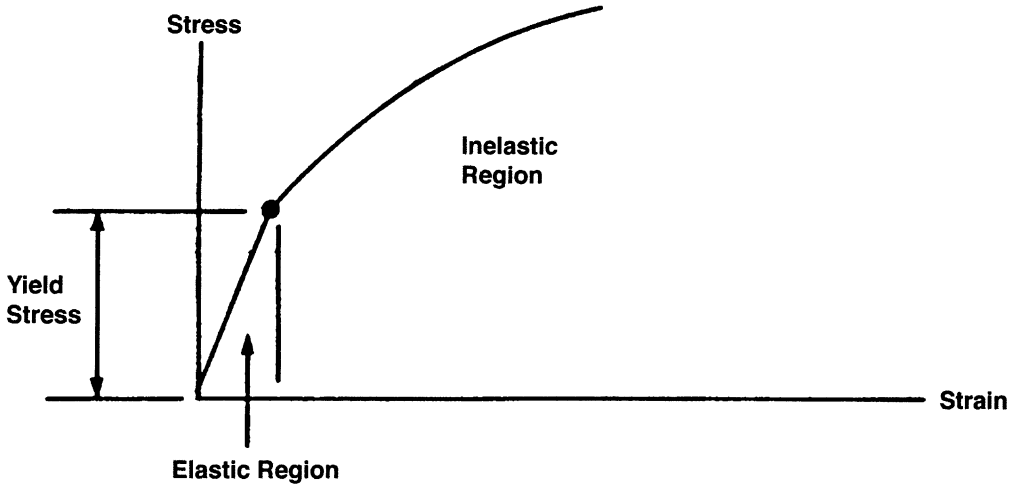


Figure A 6.4-3 An Approximate Straight Line Using the Mooney-Rivlin Constitutive Equation

A 6.5 TIME-INDEPENDENT INELASTIC BEHAVIOR

In uniaxial tension tests of most metals (and many other materials), the following phenomena can be observed. If the stress in the specimen is below the yield stress of the material, the material will behave elastically and the stress in the specimen will be proportional to the strain. If the stress in the specimen is greater than the yield stress, the material will no longer exhibit elastic behavior, and the stress-strain relationship will become nonlinear. Figure A 6.5-1 shows a typical uniaxial stress-strain curve. Both the elastic and inelastic regions are indicated.



Note: Stress and Strain Are Total Quantities

Figure A 6.5-1 Typical Uniaxial Stress-Strain Curve (Uniaxial Test)

Within the elastic region, the stress-strain relationship is unique. Therefore, if the stress in the specimen is increased (loading) from zero (point 0) to σ_1 (point 1), and then decreased (unloading) to zero, the strain in the specimen is also increased from zero to ε_1, and then returned to zero. The elastic strain is completely recovered upon the release of stress in the specimen. Figure A 6.5-2 illustrates this relationship.

The loading-unloading situation in the inelastic region is different from the elastic behavior. If the specimen is loaded beyond yield to point 2, where the stress in the specimen is σ_2 and the total strain is ε_2, upon release of the stress in the specimen the elastic strain, ε_2^e, is completely recovered. However, the inelastic (plastic) strain, ε_2^p, remains in the specimen. Figure A 6.5-2 illustrates this relationship. Similarly, if the specimen is loaded to point 3 and then unloaded to zero stress state, the plastic strain ε_3^p remains in the specimen. It is obvious that ε_2^p is not equal to ε_3^p. We can conclude that in the inelastic region:

- Plastic strain permanently remains in the specimen upon removal of stress.
- The amount of plastic strain remaining in the specimen is dependent upon the stress level at which the unloading starts (path-dependent behavior).

The uniaxial stress-strain curve is usually plotted for total quantities (total stress versus total strain). The total stress-strain curve shown in Figure A 6.5-1 can be replotted as a total stress versus plastic strain curve, as shown in Figure A 6.5-3. The slope of the total stress versus plastic strain curve is defined as the work hardening slope (H) of the material. The work hardening slope is a function of plastic strain.

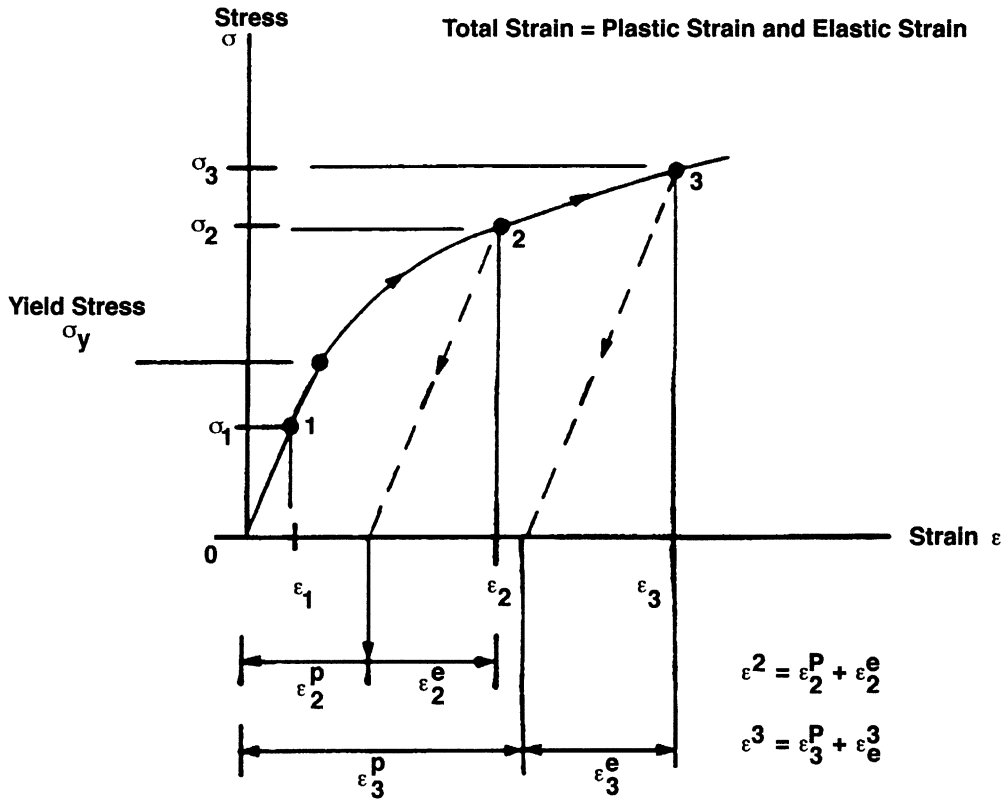


Figure A 6.5-2 Schematic of Simple Loading - Unloading (Uniaxial Test)

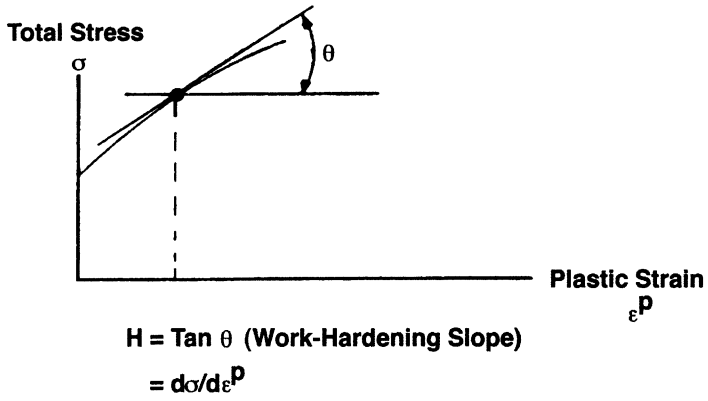


Figure A 6.5-3 Definition of Work Hardening Slope (Uniaxial Test)

The stress-strain curve shown in Figure A 6.5-1 is directly plotted from experimental data. It can be simplified for the purpose of numerical modeling. A few simplifications are shown in Figure A 6.5-4 and are listed below:

1. Bilinear representation – constant work hardening slope
2. Elastic perfectly-plastic material – no work hardening
3. Perfectly-plastic material – no work hardening and no elastic response
4. Piecewise linear representation – multiple constant work hardening slopes
5. Strain-softening material – negative work hardening slope

In addition to elastic material constants (Young's modulus and Poisson's ratio), it is essential to be concerned with yield stress and work hardening slopes in dealing with inelastic (plastic) material behavior. These quantities vary with parameters such as temperature and strain rate, further complicating the analysis. Since the yield stress is generally measured from uniaxial tests, and the stresses in real structures are usually multiaxial, the yield condition of a multiaxial stress state must be considered. The conditions of subsequent yield (work hardening rules) must also be studied.

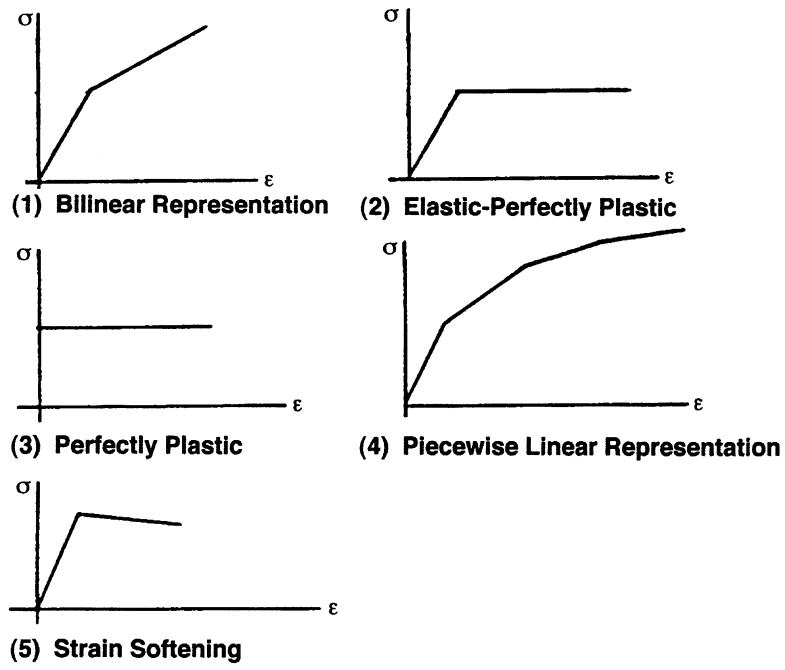


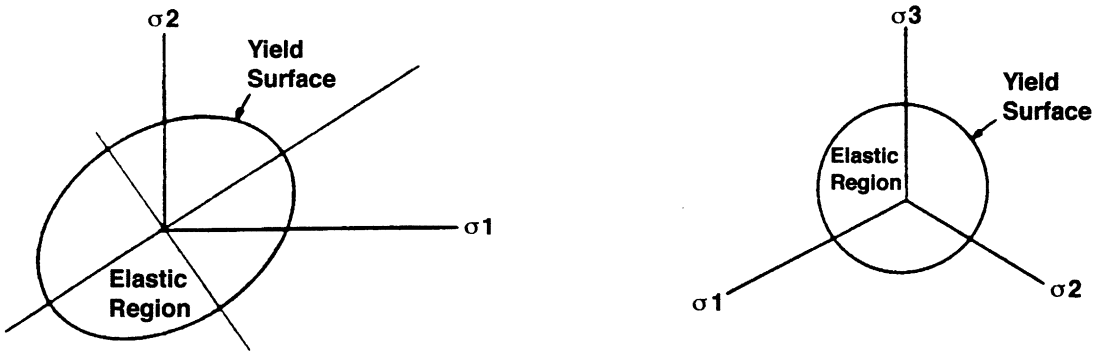
Figure A 6.5-4 Simplified Stress-Strain Curves (Uniaxial Test)

Yield Conditions

The yield stress of a material is a measured stress level that separates the elastic and inelastic behavior of the material. The magnitude of the yield stress is generally obtained from a uniaxial test. However, the stresses in a structure are usually multiaxial. A measurement of yielding for the multiaxial state of stress is called the yield condition. Depending on how the multiaxial state of stress is represented, there can be many forms of yield conditions. For example, the yield condition can be dependent on all stress components, on shear components only, or on hydrostatic stress. A number of yield conditions are available in the MARC program, and are discussed in this section.

von Mises Yield Condition

Although many forms of yield conditions are available, the von Mises criterion is the most widely used. The success of the von Mises criterion is due to the continuous nature of the function that defines this criterion and its agreement with observed behavior for the commonly encountered ductile materials. The von Mises criterion states that yield occurs when the effective (or equivalent) stress (σ_y) equals the yield stress (σ) as measured in a uniaxial test. Figure A 6.5-5 shows the von Mises yield surface in two-dimensional and three-dimensional stress space.



(a) Two-Dimensional Stress Space

(b) π-Plane

Figure A 6.5-5 von Mises Yield Surface

1. For an isotropic material

$$\bar{\sigma} = [(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2]^{1/2}/\sqrt{2} \qquad \text{(A 6.5-1)}$$

where σ_1, σ_2 and σ_3 are the principal stresses.

σ can also be expressed in terms of non-principal stresses.

$$(\bar{\sigma} = [(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2)]^{1/2})/\sqrt{2} \qquad \text{(A 6.5-2)}$$

2. The anisotropic yield function (of Hill) and stress potential are assumed as

$$\bar{\sigma} = [a_1(\sigma_y - \sigma_z)^2 + a_2(\sigma_z - \sigma_x)^2 + a_3(\sigma_x - \sigma_y)^2 + 3a_4\tau_{yz}^2 + 3a_5\tau_{yz}^2 + 3a_6\tau_{xy}^2]^{1/2}/\sqrt{2} \qquad \text{(A 6.5-3)}$$

where σ is the equivalent tensile yield stress for isotropic behavior.

Ratios of actual to isotropic yield (in the preferred orientation) are defined in the array YRDIR for direct tension yielding, and in YRSHR for yield in a shear (the ratio of actual shear yield to $\sigma/\sqrt{3}$ isotropic shear yield). Then the a_1 above are defined by:

$$a_1 = \frac{1}{\text{YRDIR}(2)**\ 2} + \frac{1}{\text{YRDIR}(3)**\ 2} - \frac{1}{\text{YRDIR}(1)**\ 2} \qquad \text{(A 6.5-4)}$$

$$a_2 = \frac{1}{\text{YRDIR}(3)**\ 2} + \frac{1}{\text{YRDIR}(1)**\ 2} - \frac{1}{\text{YRDIR}(2)**\ 2} \qquad \text{(A 6.5-5)}$$

$$a_3 = \frac{1}{\text{YRDIR}(1)**\ 2} + \frac{1}{\text{YRDIR}(2)**\ 2} - \frac{1}{\text{YRDIR}(3)**\ 2} \qquad \text{(A 6.5-6)}$$

$$a_4 = \frac{2}{\text{YRSHR}(3)**\ 2} \qquad \text{(A 6.5-7)}$$

$$a_5 = \frac{2}{\text{YRSHR}(2)**\ 2} \qquad \text{(A 6.5-8)}$$

$$a_6 = \frac{2}{\text{YRSHR}(1)**\ 2} \qquad \text{(A 6.5-9)}$$

For isotropic material, the von Mises yield condition is the default condition in MARC. The yield stress σ_y is defined in the ISOTROPIC and ORTHOTROPIC options.

For anisotropic material, use model definition options ORTHOTROPIC or ANISOTROPIC to indicate the anisotropy. Use the ORTHOTROPIC option or the user subroutine ANPLAS for the specification of anisotropic yield condition (constants a_1 through a_6, as defined above), and the model definition option ORIENTATION or, the user subroutine ORIENT, if necessary to specify preferred orientations.

Mohr-Coulomb Material (Hydrostatic Yield Dependence)

The MARC program includes options for elastic-plastic behavior based on a yield surface that exhibits hydrostatic stress dependence. Such behavior is observed in a wide class of soil and rock-like materials. These materials are generally classified as Mohr-Coulomb materials (generalized von Mises materials). Ice is also thought to be a Mohr-Coulomb material. The generalized Mohr-Coulomb behavior implemented in MARC was developed by Drucker and Prager. There are two types of Mohr-Coulomb materials: linear and parabolic. Each is discussed on the following pages.

Linear Mohr-Coulomb Material

The deviatoric yield function, as shown in Figure A 6.5-6, is assumed to be a linear function of the hydrostatic stress.

$$f = \alpha J_1 + J_2^{1/2} - \bar{\sigma}/3 = 0 \qquad \text{(A 6.5-10)}$$

where

$$J_1 = \sigma_{ii} \qquad \text{(A 6.5-11)}$$

$$J_2 = \frac{1}{2}\sigma'_{ij}\sigma'_{ij}, \qquad \sigma'_{ij} = \sigma_{ij} - \frac{1}{3}\delta_{ij}\sigma_{kk} \qquad \text{(A 6.5-12)}$$

Analysis of linear Mohr-Coulomb material based on the constitutive description above is available in MARC through the ISOTROPIC model definition option. Through the ISOTROPIC option, the values of σ and α are entered. Note that, throughout the program, the convention that the tensile direct stress is positive is maintained, contrary to its use in many soil mechanics texts .

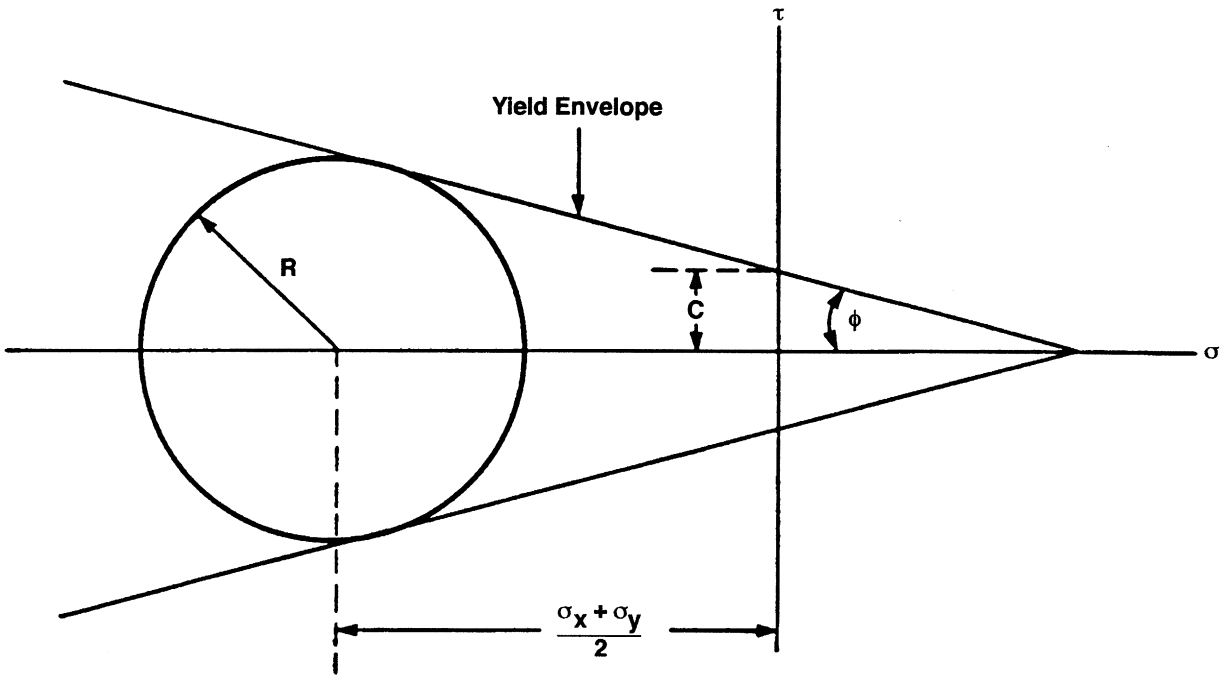


Figure A 6.5-6 Yield Envelope of Plane Strain (Linear Mohr-Coulomb Material)

The constants α and σ may be related to c and ϕ by

$$c = \frac{\bar{\sigma}}{3\,(1 - 12\alpha^2)^{1/2}}; \qquad \frac{3\alpha}{(1 - 3\alpha^2)^{1/2}} = \sin\phi \qquad \text{(A 6.5-13)}$$

where c is the cohesion and ϕ is the angle of friction.

Parabolic Mohr-Coulomb Material

The hydrostatic dependence is generalized to give a yield envelope which is parabolic in the case of plane strain (see Figure A 6.5-7).

$$f = (3J_2 + \sqrt{3}\beta\bar{\sigma}J_1)^{1/2} - \bar{\sigma} = 0 \qquad \text{(A 6.5-14)}$$

The parabolic yield surface is obtained in MARC through the ISOTROPIC definition option. Enter the values $\bar{\sigma}$ and β through the model definition option ISOTROPIC.

where

$$\bar{\sigma}^2 = 3\left(c^2 - \frac{\alpha^2}{3}\right) \qquad \beta = \frac{\alpha}{\left(3\left(3c^2 - \alpha^2\right)\right)^{1/2}} \qquad \text{(A 6.5-15)}$$

where c is the cohesion.

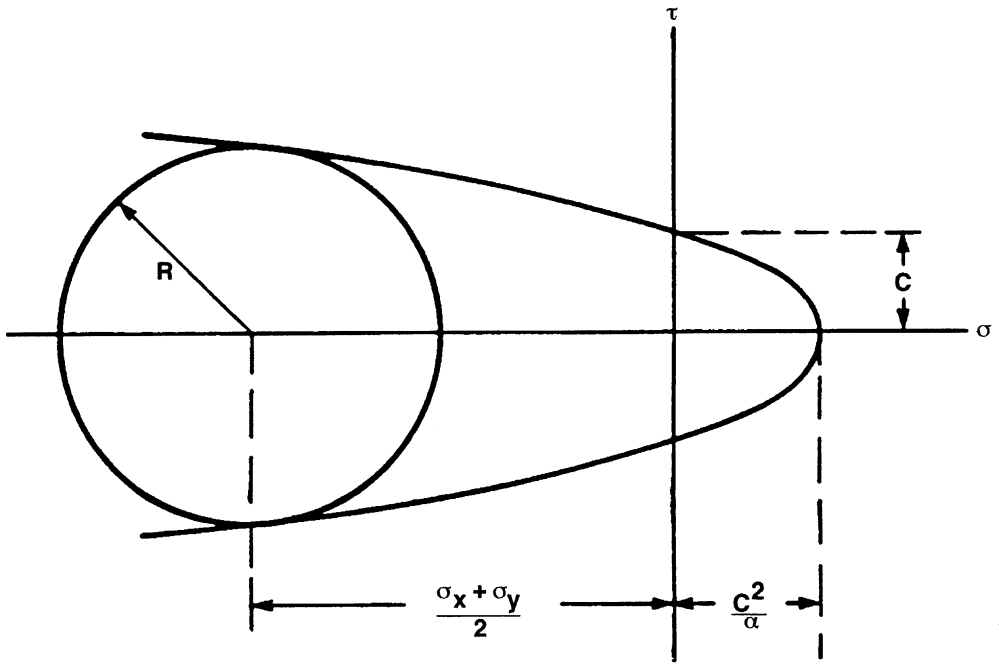


Figure A 6.5-7 Resultant Yield Condition of Plane Strain (Parabolic Mohr- Coulomb Material)

Powder Material

Some materials, during certain stages of manufacturing are granular in nature. In particular, powder metals are often used in certain forging operations and during hot isostatic pressing (HIP). These material properties are now functions of both the temperature and the densification. It should be noted that the soil model discussed in Chapter 4 also exhibits some of these characteristics. In the model incorporated into MARC, a unified viscoplastic approach is used. The yield function is

$$F = \frac{1}{\gamma}\left(\frac{3}{2}\sigma_{ij}'\sigma_{ij}' + \frac{p^2}{\beta^2}\right)^{1/2} - \sigma_y \qquad \text{(A 6.5-16)}$$

where σ_y is the uniaxial yield stress, σ' is the deviatoric stress, and p is the hydrostatic pressure. γ, β are material parameters.

σ_y may be a function of temperature and relative density, γ, β are functions only of relative density.

Typically, we allow:

$$\beta = (q_1 + q_2\rho^{q_3})^{q_4}$$
$$\gamma = (b_1 + b_2\rho^{b_3})^{b_4} \qquad \text{(A 6.5-17)}$$

where ρ is the relative density.

As the powder becomes more dense, ρ approaches 1 and the classical von Mises model is recovered.

It should be noted that the elastic properties are also functions of relative density. In particular, as the material becomes fully dense the Poisson's ration approaches .5.

As most processes involving powder materials are both pressure and thermally driven, it may be necessary to perform a coupled analysis. The MARC program also allows you to specify density effects for the thermal properties, conductivity and specific heat. The basic input data is entered through the POWDER option. In addition to the TEMPERATURE EFFECTS option, there is also a DENSITY EFFECTS option. The initial relative density is entered through the RELATIVE DENSITY option.

Note that this material may undergo both large shear and volumetric strains. The LARGE DISP, UPDATE, and FINITE OPTIONS should also be used. In HIP processes, the can would typically be modelled using a conventional elastic plastic material law. Additionally, the FOLLOW FOR option would be used to ensure that the pressure remain normal to the deformed material.

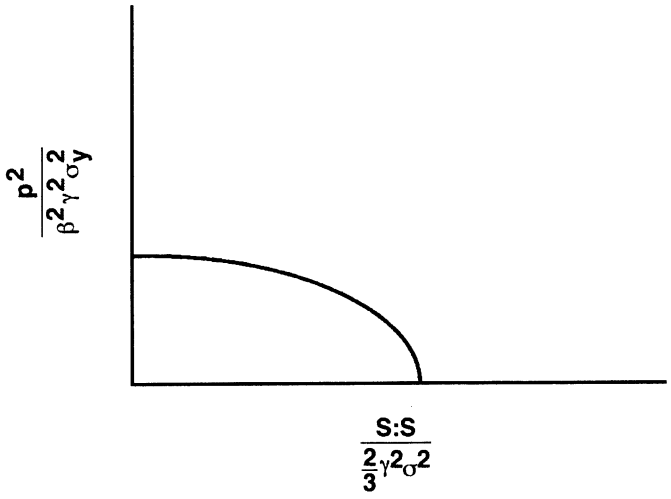


Figure A 6.5-8 Yield Function of Shima Model

Oak Ridge National Laboratory Options

In the MARC program, the ORNL options are based on the definitions of ORNL-TM- 3602 [1] for stainless steels and ORNL recommendations [2] for 2 1/4 Cr-1 Mo steel.

The initial yield stress should be used for the initial inelastic loading calculations for both the stainless steels and 2 1/4 Cr-1 Mo steel. The 10th- cycle yield stress should be used for the hardened material. The 100th-cycle yield stress must be used in the following circumstances:

1. To accommodate cyclic softening of 2 1/4 Cr-1 Mo steel after many load cycles
2. After a long period of high temperature exposure
3. After the occurrence of creep strain

To enter initial and 10th-cycle yield stresses, use the model definition option ISOTROPIC or ORTHOTROPIC.

Effects on Yield Stress

This section describes the MARC program capabilities with respect to the effect of temperature and strain rate.

The MARC program allows you to input a temperature-dependent yield stress. To enter the yield stress at a reference temperature, use the model definition options ISOTROPIC or ORTHOTROPIC. To enter variations of yield stress with temperatures, use the model definition options TEMPERATURE EFFECTS and ORTH TEMP. Repeat the model definition options TEMPERATURE EFFECTS and ORTHO TEMP for each material, as necessary. The effect of temperatures on yielding is discussed further in Section 6.5, in "Constitutive Relations."

The program allows you to enter a strain rate dependent yield stress, for use in dynamic and flow (e.g., extrusion) problems. To use the strain rate dependent yield stress in static analysis, enter a fictitious time using the TIME STEP option. The zero-strain-rate yield stress is given on

the ISOTROPIC or ORTHOTROPIC options. Repeat the model definition option STRAIN RATE for each different material where strain rate data are necessary. Refer to Section 6.5 in "Constitutive Relations," for more information on the strain-rate effect on yielding.

Work Hardening Rules

In a uniaxial test, the work hardening slope is defined as the slope of the stress-plastic strain curve. The work hardening slope relates the incremental stress to incremental plastic strain in the inelastic region and dictates the conditions of subsequent yielding. A number of work hardening rules (isotropic, kinematic and combined) are available in MARC. A description of these work hardening rules is given below. The uniaxial stress-plastic strain curve may be represented by a piecewise linear function through the WORK HARD option. As an alternative, you can specify work hardening through the user subroutine WKSLP.

There are two methods to enter this information, using the WORK HARD option. In the first method, you must enter work hardening slopes for uniaxial stress data as a change in stress per unit of plastic strain (see Figure A 6.5-9) and the plastic strain at which these slopes become effective (breakpoint).

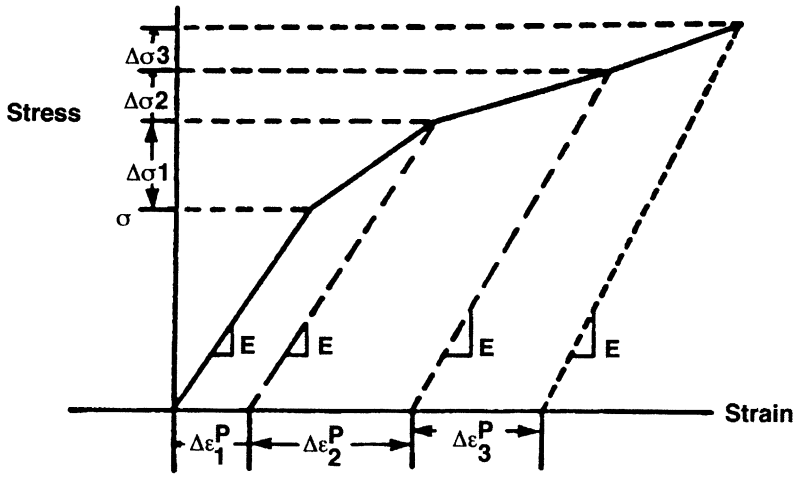


Figure A 6.5-9 Work Hardening Slopes

Slope	Breakpoint
$\frac{\Delta\sigma_1}{\Delta\varepsilon_1^p}$	0.0
$\frac{\Delta\sigma_2}{\Delta\varepsilon_2^p}$	$\Delta\varepsilon_1^p$
$\frac{\Delta\sigma_3}{\Delta\varepsilon_3^p}$	$\Delta\varepsilon_1^p + \Delta\varepsilon_2^p$

and so on.

NOTE

The slopes of the work hardening curves should be based on a plot of the stress versus plastic strain for a tensile test. The elastic strain components of the stress-strain curve should not be included. The first breakpoint of the work hardening curve should be 0.0.

In the second method, you enter a table of yield stress, plastic strain points. This option is flagged by adding the word DATA to the WORK HARD statement.

NOTE

The data points should be based on a plot of the stress versus plastic strain for a tensile test. The elastic strain components should not be included. The first plastic strain should equal 0.0 and the first stress should agree with that given as the yield stress in the ISOTROPIC or ORTHOTROPIC options.

Isotropic Hardening

The isotropic work hardening rule assumes that the center of the yield surface remains stationary in the stress space, but that the size (radius) of the yield surface expands, due to work hardening. The change of the von Mises yield surface is plotted in Figure A 6.5-10b.

A review of the load path of a uniaxial test that involves both the loading and unloading of a specimen will assist in describing the isotropic work hardening rule. The specimen is first loaded from stress free (point 0) to initial yield at point 1, as shown in Figure A 6.5-10a. It is then continuously loaded to point 2. Then, unloading from 2 to 3 following the elastic slope E (Young's modulus) and then elastic reloading from 3 to 2 takes place. Finally, the specimen is plastically loaded again from 2 to 4 and elastic unloaded from 4 to 5. Reverse plastic loading occurs between 5 and 6.

It is obvious that the stress at 1 is equal to the initial yield stress σ_y and stresses at points 2 and 4 are higher than σ_y, due to work hardening. During unloading, the stress state can remain elastic (e.g., point 3) or it can reach to a subsequent (reversed) yield point (e.g., point 5). The isotropic work hardening rule states that the reverse yield occurs at current stress level in the reversed direction. Let σ_4 be the stress level at point 4. Then, the reverse yield can only take place at a stress level of $-\sigma_4$ (point 5).

The isotropic work hardening model (with a work slope of 0) is the default option in MARC. To explicitly specify the isotropic hardening option in MARC, use model definition options ISOTROPIC or ORTHOTROPIC. To input work hardening slope data, use the WORK HARD option or user subroutine WKSLP.

For many materials, the isotropic work hardening model is inaccurate if unloading occurs (as in cyclic loading problems). For these problems, the kinematic hardening model or the combined hardening model represents the material better.

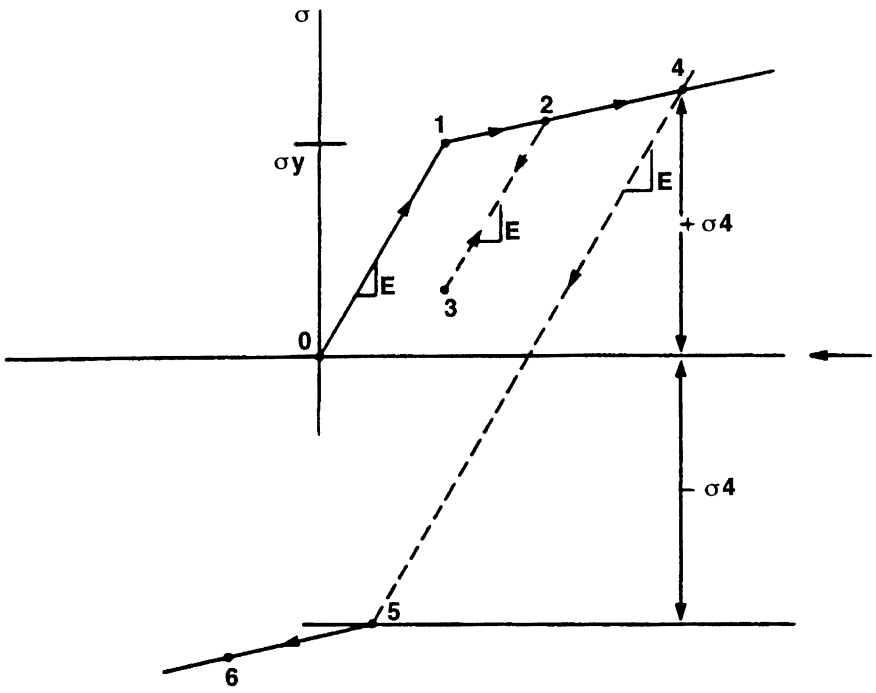
Kinematic Hardening

Under the kinematic hardening rule, the von Mises yield surface does not change in size or shape, but the center of the yield surface can move in stress space. Figure A 6.5-11b illustrates this condition.

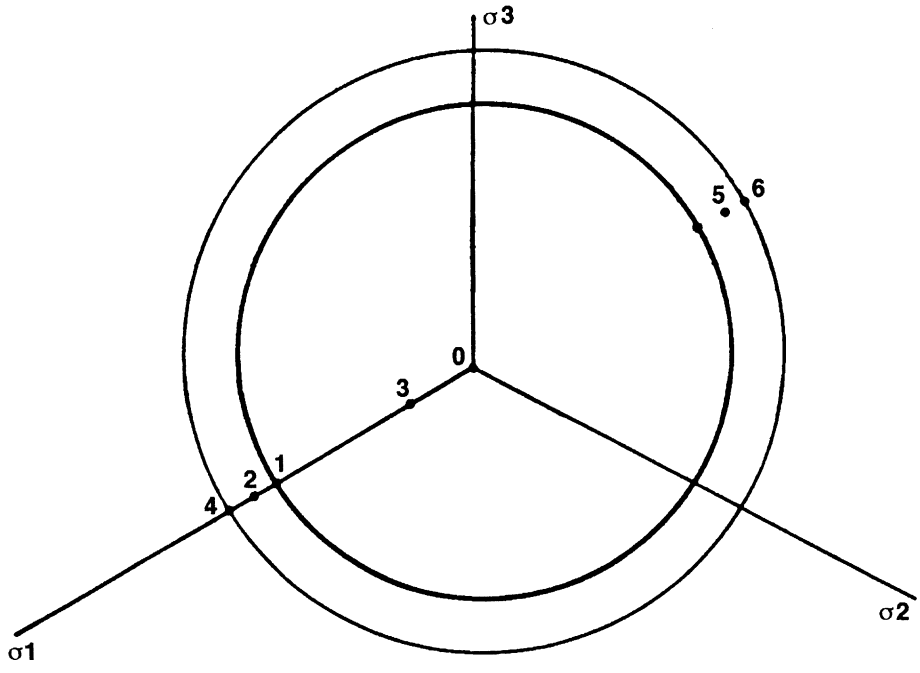
The loading path of a uniaxial test is shown in Figure A 6.5-11a. The specimen is loaded in the following order: from stress free (point 0) to initial yield (point 1), 2 (loading), 3 (unloading), 2 (reloading), 4 (loading), 5 and 6 (unloading). As in isotropic hardening, stress at 1 is equal to the initial yield stress σ_y, and stresses at 2 and 4 are higher than σ_y, due to work hardening. Point 3 is elastic, and reverse yield takes place at point 5. Under the kinematic hardening rule, the reverse yield occurs at the level of $\sigma_5 = (\sigma_4 - 2\sigma_y)$, rather than at the stress level of $-\sigma_4$. Similarly, if the specimen is loaded to a higher stress level σ_7 (point 7), and then unloaded to the subsequent yield point 8, the stress at point 8 is $\sigma_8 = (\sigma_7 - 2\sigma_y)$. If the specimen is unloaded from a (tensile) stress state (such as point 4 and 7), the reverse yield can occur at a stress state in either the reverse (point 5) or the same (point 8) direction.

To invoke the kinematic hardening in MARC, use the model definition options ISOTROPIC or ORTHOTROPIC. To input work hardening slope data, use the WORK HARD option or user subroutine WKSLP.

For many materials, the kinematic hardening model gives a better representation of loading/ unloading behavior than the isotropic hardening model. For cyclic loading, however, the kinematic hardening model can represent neither cyclic hardening nor cyclic softening.



(a) Loading Path



(b) von Mises Yield Surface

Figure A 6.5-10 Schematic of Isotropic Hardening Rule (Uniaxial Test)

Combined Hardening

Figure A 6.5-12 shows a material with highly nonlinear hardening. Here, the initial hardening is assumed to be almost entirely isotropic, but after some plastic straining, the elastic range attains an essentially constant value (i.e., pure kinematic hardening). The basic assumption of the combined hardening model is that such behavior is reasonably approximated by a classical constant kinematic hardening constraint, with the superposition of initial isotropic hardening. The isotropic hardening rate eventually decays to zero as a function of the equivalent plastic strain measured by

$$\bar{\varepsilon}^p = \int \dot{\bar{\varepsilon}}^p dt = \int \left(\frac{2}{3}\dot{\varepsilon}^p_{ij}\dot{\varepsilon}^p_{ij}\right)^{1/2} dt \qquad \text{(A 6.5-18)}$$

This implies a constant shift of the center of the elastic domain, with a growth of elastic domain around this center until pure kinematic hardening is attained. In this model, there is a variable proportion between the isotropic and kinematic contributions that depends on the extent of plastic deformation (as measured by ε^p).

Use model definition options ISOTROPIC or ORTHOTROPIC to activate the combined work hardening option in MARC. Use the WORK HARD option or user subroutine WKSLP to input work hardening slope data.

The work hardening data at small strains governs the isotropic behavior, and the data at large strains governs the kinematic hardening behavior. If the last work hardening slope is zero, the behavior will be the same as the isotropic hardening model.

Flow Rule

Yield stress and work hardening rules are two experimentally related phenomena that characterize plastic material behavior. The flow rule is also essential in establishing the incremental stress-strain relations for plastic material. The flow rule describes the differential changes in the plastic strain components $d\varepsilon^p$ as a function of the current stress state.

The Prandtl-Reuss representation of the flow rule is available in MARC.

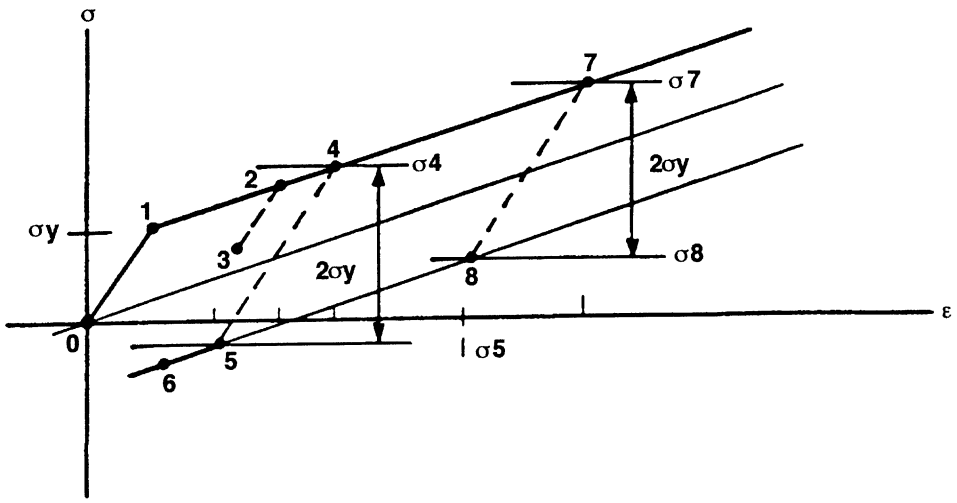
$$d\varepsilon^p = d\bar{\varepsilon}^p \frac{\partial \bar{\sigma}}{\partial \sigma} \qquad \text{(A 6.5-19)}$$

where $d\bar{\varepsilon}^p$ and σ are equivalent plastic strain increment and equivalent stress, respectively.

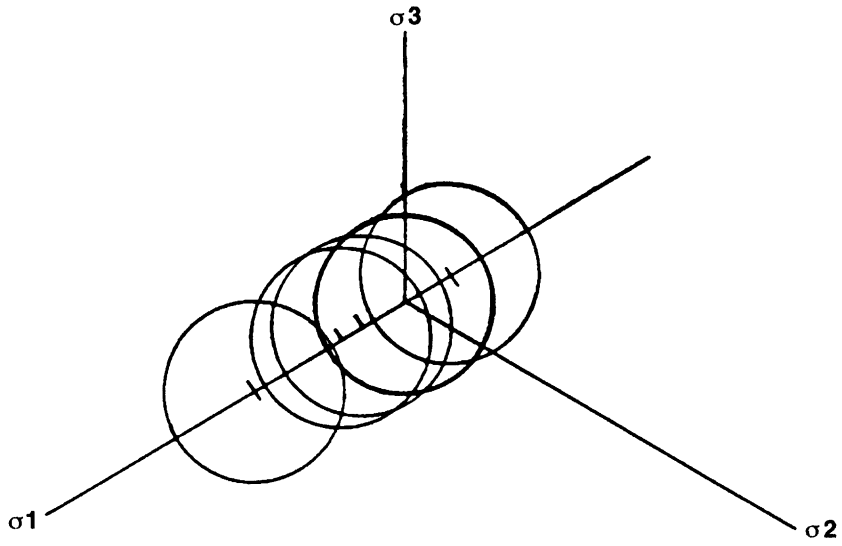
The significance of this representation is illustrated in Figure A 6.5-12. This figure illustrates the "stress-space" for the two-dimensional case. The solid curve gives the yield surface (locus of all stress states causing yield) as defined by the von Mises criterion.

Equation A 6.5-19 expresses the condition that the direction of inelastic straining is normal to the yield surface. This condition is called either the normality condition or the associated flow rule.

If the von Mises yield surface is used, then the normal is equal to the deviatoric stress.



(a) Loading Path



(b) von Mises Yield Surface

Figure A 6.5-11 Schematic of Kinematic Hardening Rule (Uniaxial Test)

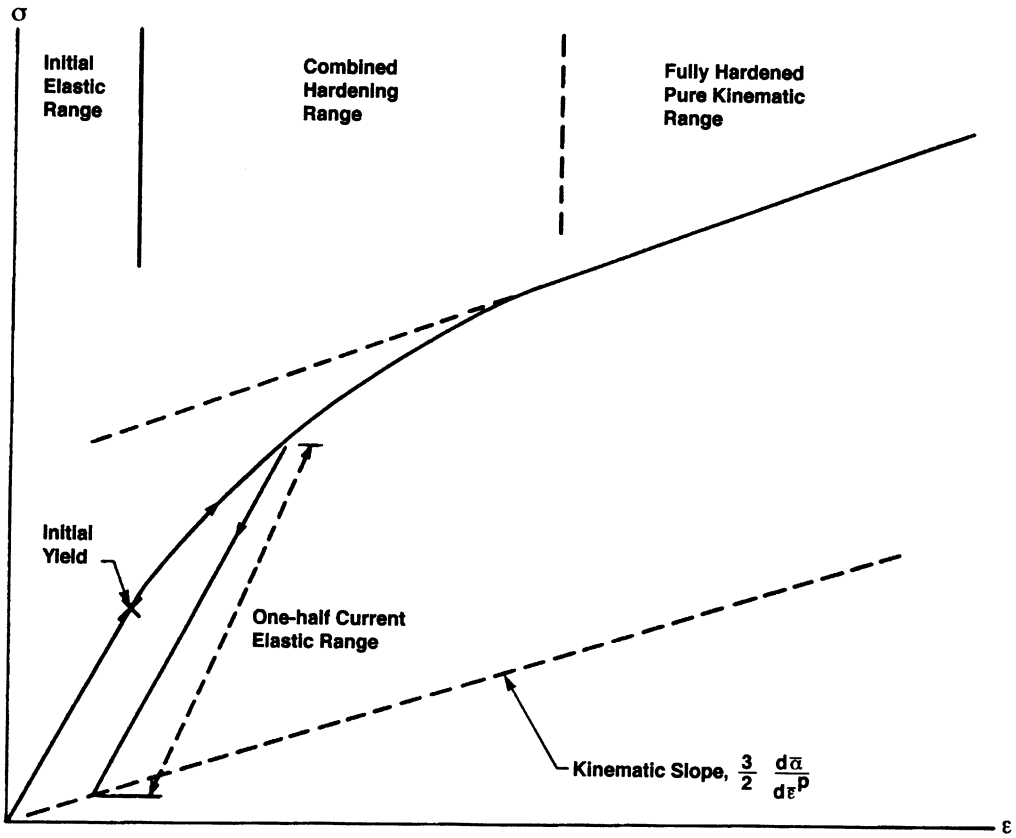


Figure A 6.5-12 Basic Uniaxial Tension Behavior of the Combined Hardening Model

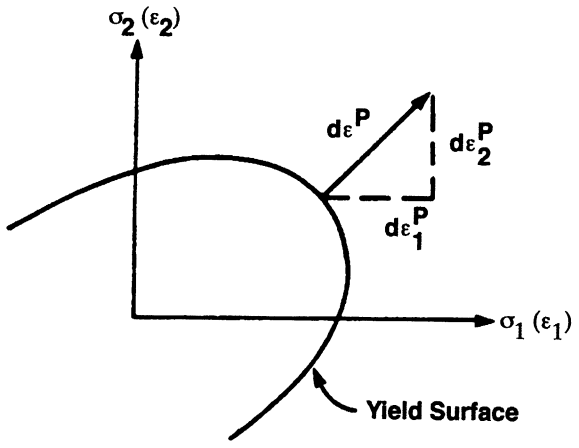


Figure A 6.5-13 Yield Surface and Normality Criterion 2-D Stress Space

Constitutive Relations

This section presents the constitutive relation that describes the incremental stress-strain relation for an elastic-plastic material. The material behavior is governed by the incremental theory of plasticity, the von Mises yield criterion, and the isotropic hardening rule.

Let the work hardening coefficient H be expressed as

$$H = d\bar{\sigma}/d\bar{\varepsilon}^p \qquad \text{(A 6.5-20)}$$

and the flow rule be expressed as

$$d\varepsilon^p = d\bar{\varepsilon}^p \{\frac{d\bar{\sigma}}{d\sigma}\} \qquad \text{(A 6.5-21)}$$

Consider the differential form of the familiar stress-strain law, with the plastic strains interpreted as initial strains

$$d\sigma = Cd\varepsilon - Cd\varepsilon^p \qquad \text{(A 6.5-22)}$$

After substitution of Equation A 6.5-21, this becomes

$$d\sigma = Cd\varepsilon - C\frac{\partial\bar{\sigma}}{\partial\sigma}d\bar{\varepsilon}^p \qquad \text{(A 6.5-23)}$$

Multiplying through by $\frac{d\bar{\sigma}}{d\sigma}$

$$\frac{\partial\bar{\sigma}}{\partial\sigma}d\sigma = \frac{\partial\bar{\sigma}}{\partial\sigma}Cd\varepsilon - \frac{\partial\bar{\sigma}}{\partial\sigma}C\frac{\partial\bar{\sigma}}{\partial\sigma}d\bar{\varepsilon}^p \qquad \text{(A 6.5-24)}$$

and, with use of Equation A 6.5-20 in place of the left-hand side,

$$Hd\bar{\varepsilon}^p = \frac{\partial\bar{\sigma}}{\partial\sigma}Cd\varepsilon - \frac{\partial\bar{\sigma}}{\partial\sigma}C\frac{\partial\bar{\sigma}}{\partial\sigma}d\bar{\varepsilon}^p \qquad \text{(A 6.5-25)}$$

By rearrangement

$$d\bar{\varepsilon}^p = \frac{\frac{\partial\bar{\sigma}}{\partial\sigma}Cd\varepsilon}{H + \frac{\partial\bar{\sigma}}{\partial\sigma}C\frac{d\bar{\sigma}}{d\sigma}} \qquad \text{(A 6.5-26)}$$

Finally, by substitution of this expression into Equation A 6.5-24, we obtain

$$d\sigma = L^{ep} d\varepsilon \tag{A 6.5-27}$$

where

$$L^{ep} = C - L^{p} \tag{A 6.5-28}$$

with $[C^e]$ the conventional elastic material stiffness matrix and

$$L^{p} = \frac{C \dfrac{\partial \bar{\sigma}}{\partial \sigma} \dfrac{\partial \bar{\sigma}}{\partial \sigma} C}{\left[H + \dfrac{\partial \bar{\sigma}}{\partial \sigma} \left(C \dfrac{\partial \bar{\sigma}}{\partial \sigma} \right) \right]} \tag{A 6.5-29}$$

$$L^{ep} = \left[\frac{I - C \dfrac{\partial \bar{\sigma}}{\partial \sigma} \dfrac{\partial \bar{\sigma}}{\partial \sigma}}{H + \dfrac{\partial \bar{\sigma}}{\partial \sigma} C \dfrac{\partial \bar{\sigma}}{\partial \sigma}} \right] C \tag{A 6.5-30}$$

Observe that L^{ep} and L^{p} are symmetric, and the case of perfect plasticity, where $H = 0$, causes no difficulty.

Temperature Effects

This section discusses the effects of temperature-dependent plasticity on the constitutive relation.

The following constitutive relations for thermo-plasticity were developed by Naghdi. Temperature effects are discussed using the isotropic hardening model and the von Mises yield condition.

The stress rate can be expressed in the form

$$\dot{\sigma} = L_{ijkl} \dot{\varepsilon}_{kl} + h_{ij} \dot{T} \tag{A 6.5-31}$$

For elastic-plastic behavior, the moduli L_{ijkl} are

$$L_{ijkl} = C_{ijkl} - (C_{ijmn} \sigma'_{mn} \sigma'_{pq} C_{pqkl}) / d \tag{A 6.5-32}$$

and for purely elastic response

$$L_{ijkl} = C_{ijkl} \tag{A 6.5-33}$$

The term that relates the stress increment to the increment of temperature for elastic-plastic behavior is

$$h_{ij} = X_{ij} - C_{ijkl}\alpha_{kl} - (C_{ijkl}\sigma'_{kl}(\sigma_{pq}X_{pq} - \frac{2}{3}\bar{\sigma}\frac{\partial\bar{\sigma}}{\partial T}))/d \quad \text{(A 6.5-34)}$$

and for purely elastic response

$$H_{ij} = X_{ij} - C_{ijkl}\alpha_{kl} \quad \text{(A 6.5-35)}$$

where

$$d = \frac{4}{9}\bar{\sigma}^2\frac{\partial\bar{\sigma}}{\partial\bar{\varepsilon}^p} + \sigma'_{ij}\ C_{ijkl}\ \sigma'_{kl} \quad \text{(A 6.5-36)}$$

and

$$X_{ij} = \frac{\partial C_{ijkl}}{\partial T}\ \varepsilon^e_{kl} \quad \text{(A 6.5-37)}$$

and α_{kl} are the coefficients of thermal expansion.

Strain Rate Effects

This section discusses the influence of strain rate on the elastic-plastic constitutive relation.

Strain rate effects cause the structural response of a body to change because they influence the material properties of the body. These material changes lead to an instantaneous change in the strength of the material. Strain rate effects become more pronounced for temperatures greater than half the melting temperature (T_m). The following discussion explains the effect of strain rate on the size of the yield surface.

Using the von Mises yield condition and normality rule, we obtain an expression for the stress rate of the form

$$\sigma_{ij} = L_{ijkl}\varepsilon_{kl} + \dot{r}_{ij}\bar{\varepsilon}^p \quad \text{(A 6.5-38)}$$

For elastic-plastic response

$$L_{ijkl} = C_{ijkl} - (C_{ijmn}\sigma'_{mn}\sigma'_{pq}C_{pqkl})/d \quad \text{(A 6.5-39)}$$

and

$$r_{ij} = C_{ijmn}\sigma'_{mn}\ \frac{2}{3}\bar{\sigma}\ \frac{\partial\bar{\sigma}}{\partial\bar{\varepsilon}}\ /d \quad \text{(A 6.5-40)}$$

where

$$d = \frac{4}{9}\bar{\sigma}^2\frac{\partial\bar{\sigma}}{\partial\bar{\varepsilon}^p} + \sigma'_{ij}C_{ijkl}\sigma'_{kl} \quad \text{(A 6.5-41)}$$

A 6.6 TIME-DEPENDENT INELASTIC BEHAVIOR

Force-displacement relationships vary in different material models. A perfectly elastic material and a perfectly viscous material can be represented by a spring and a dashpot, respectively (as shown in Figure A 6.6-1). In a perfectly elastic material, the deformation is proportional to the applied load. In a perfectly viscous material, the rate of change of the deformation over time is proportional to the load.

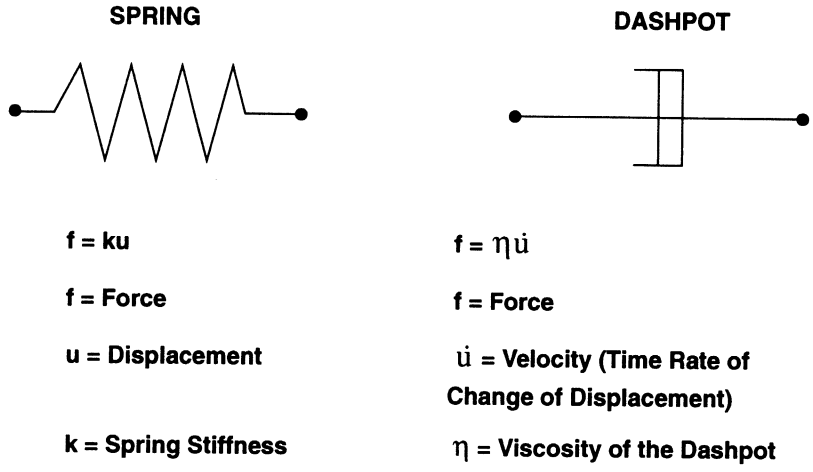


Figure A 6.6-1 Perfectly Elastic (Spring) and Viscous (Dashpot) Materials

In the class of viscoelastic and creeping materials, the application of a constant load is followed by a deformation, which may be made up of an instantaneous deformation (elastic effect) followed by a continual deformation with time (viscous effect). Eventually it may become pure viscous flow. Continued deformation under constant load is termed creep (see Figure A 6.6-2).

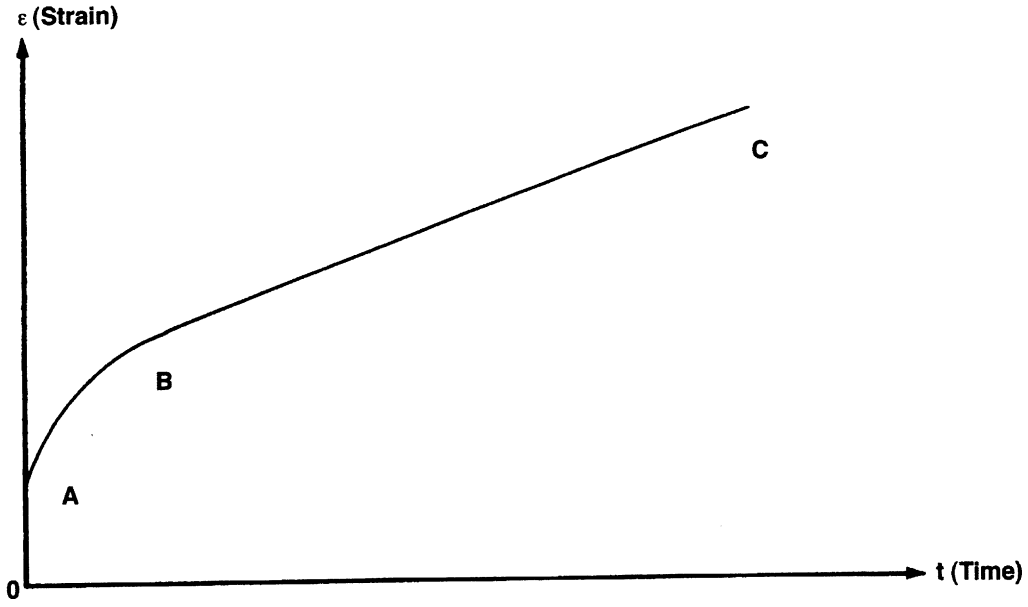


Figure A 6.6-2 The Creep Curve

A viscoelastic material may be subjected to sudden application of a constant deformation. This results in an instantaneous proportional load (elastic effect), followed by a gradual reduction of the required load with time, until a limiting value of the load is attained. The decreasing of load for a constant deformation, is termed relaxation (see Figure A 6.6-3).

Viscoelastic and creeping materials can be represented by models consisting of both springs and dashpots because the material displays both elastic effects and viscous effects. This implies that the material either continues to flow for a given stress, or the stress decreases with time for a given strain. The measured relation between stress and strain is generally very complex.

Two models that are commonly used to relate stress and strain are the Maxwell and Kelvin (Voigt or Kelvin-Voigt) models. A description of these models is given below.

The mathematical relation which holds for the Maxwell solid is

$$\dot{\varepsilon} = \alpha\dot{\sigma} + \beta\sigma \qquad \text{(A 6.6-1)}$$

In the one-dimensional case for normal stress

$$\alpha = \frac{1}{E}$$

$$\beta = \frac{1}{\eta} \qquad \text{(A 6.6-2)}$$

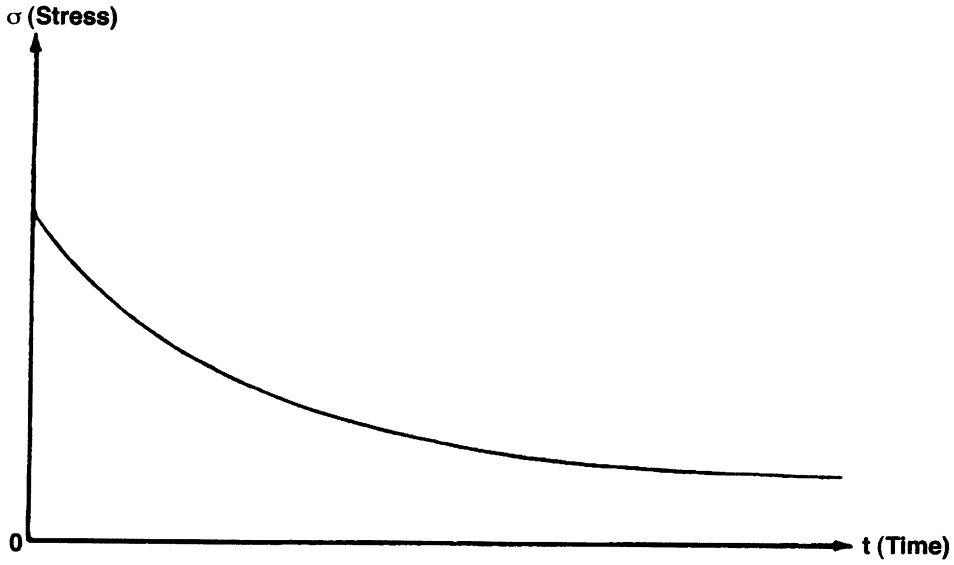


Figure A 6.6-3 The Relaxation Curve

This relation may be depicted as a spring and dashpot in series, as shown in Figure A 6.6-4. The integration of Figure A 6.6.6-1 yields

$$\varepsilon = \frac{\sigma}{E} + \int \frac{\sigma}{\eta} dt \qquad \text{(A 6.6-3)}$$

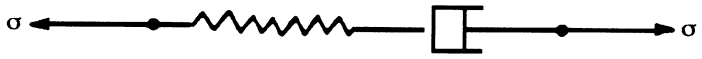
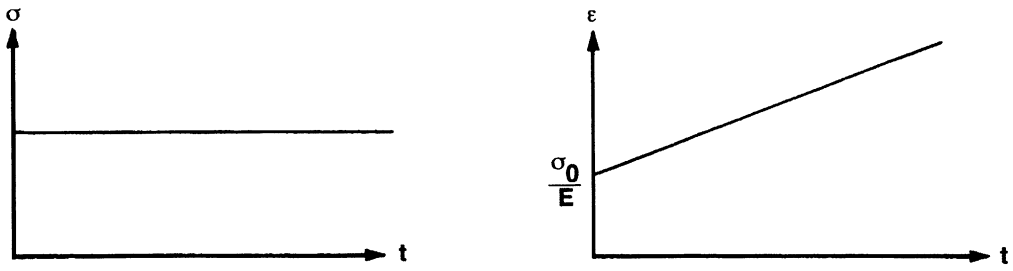


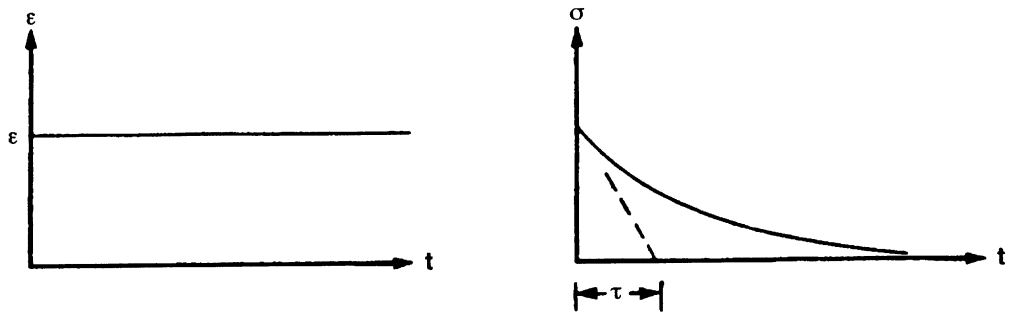
Figure A 6.6-4 Maxwell Solid

The strain and stress responses of the Maxwell Solid model are shown in Figure A 6.6-5 and Figure A 6.6-6, respectively.



Constant Stress Applied **Response to Constant Stress**

Figure A 6.6-5 Strain Response to Applied Constant Stress (Maxwell Solid)



Constant Strain Applied **Response to Constant Strain**

Figure A 6.6-6 Stress Response to Applied Constant Strain (Maxwell Solid)

The mathematical relation which holds for the Kelvin (Voigt or Kelvin-Voigt) solid is

$$\sigma = \alpha\varepsilon + \beta\dot{\varepsilon} \qquad \text{(A 6.6-4)}$$

This equation is depicted as a spring and dashpot in parallel. (See Figure A 6.6-7). When $\beta = 0$ (no dashpot), the system is a linearly elastic system in which $\alpha = E$, the elastic modulus.

When $E = 0$ (no spring), the "solid" obeys Newton's equation for a viscous fluid and $\beta = \eta$, the viscous coefficient. Thus, we can rewrite Equation A 6.6-4 in the form

$$\sigma = E\varepsilon + \eta\dot{\varepsilon} \qquad \text{(A 6.6-5)}$$

In the above relation, we have considered one-dimensional normal stress and strain. The relation holds equally well for shear stress τ and shear strain γ in which $\alpha = G$, the shear modulus, and $\beta = \eta$, the viscous coefficient. Equation A 6.6-4 may be rewritten as

$$\tau = G\gamma + \eta\dot{\gamma} \qquad \text{(A 6.6-6)}$$

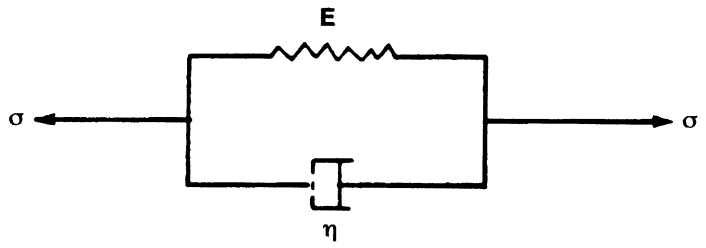


Figure A 6.6-7 Kelvin (Voigt or Kelvin-Voigt) Solid

The strain responses of the Kelvin Solid model are depicted in Figure A 6.6-8. For multiaxial situations, these equations can be generalized to tensor quantities.

To invoke the Maxwell model, use the CREEP parameter. The creep strain can be specified as either deviatoric creep strain (conventional creep) or dilatational creep strain (swelling). To invoke the Kelvin model, also use the CREEP parameter.

Creep (Maxwell Model)

Creep is an important factor in elevated-temperature stress analysis. In MARC, creep is represented by a Maxwell model. Creep is a time-dependent, inelastic behavior, and can occur at any stress level (i.e., either below or above the yield stress of a material). The creep behavior can be characterized as primary, secondary, and tertiary creep, as shown in Figure A 6.6-9. Engineering analysis is often limited to the primary and secondary creep regions. Tertiary creep in a uniaxial specimen is usually associated with geometric instabilities, such as necking. The major difference between the primary and secondary creep is that the creep strain rate is much larger in the primary creep region than it is in the secondary creep region. The creep strain rate is the slope of the creep strain-time curve. The creep strain rate is generally dependent on stress, temperature, and time.

The creep data may be specified in either an exponent form or in a piecewise linear curve. To specify creep data, use model definition option CREEP. Subroutine CRPLAW allows alternative forms of creep behavior to be programmed directly.

$$\dot{\bar{\varepsilon}}^c = \frac{d\bar{\varepsilon}^c}{dt} \qquad \text{(A 6.6-7)}$$

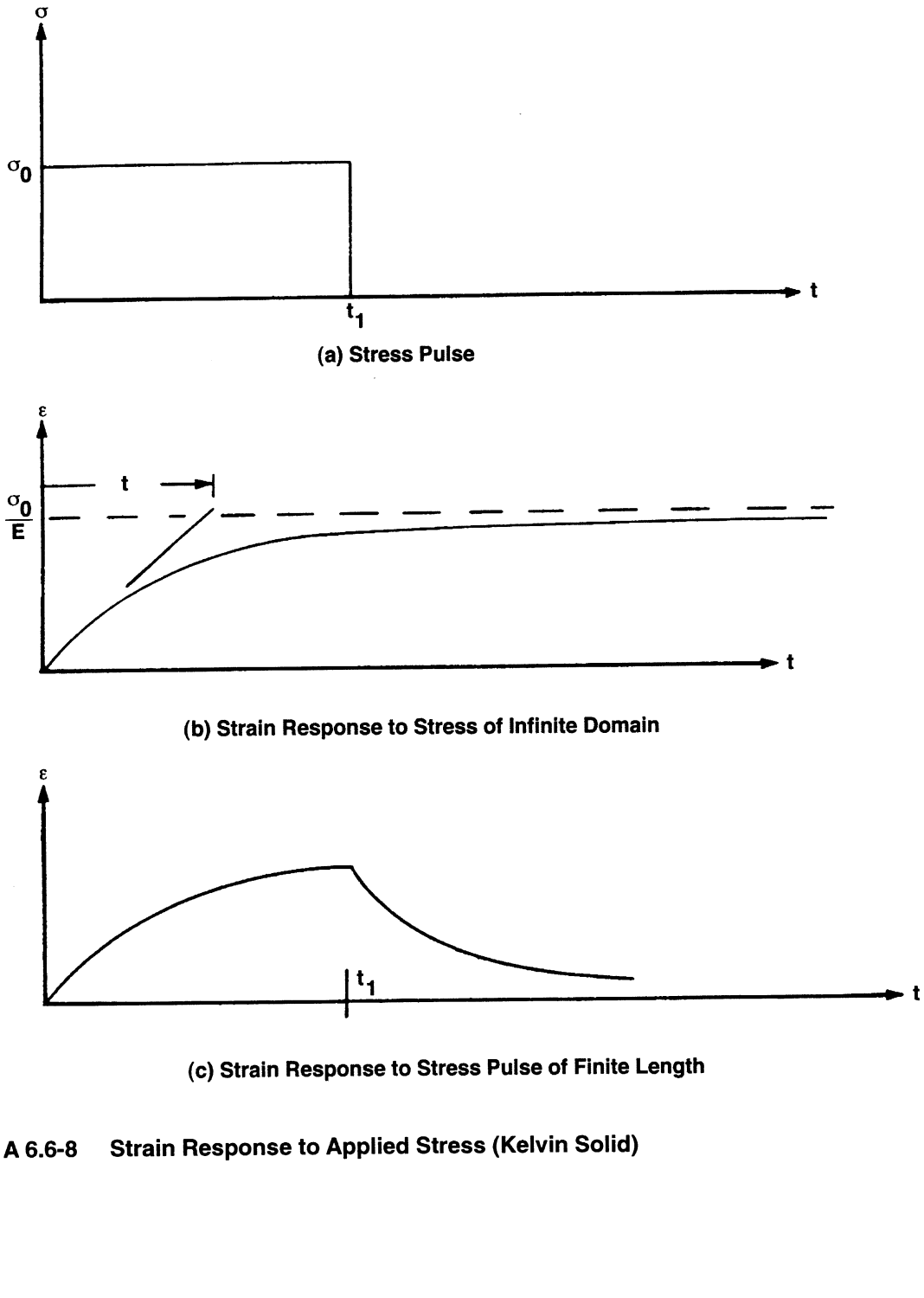
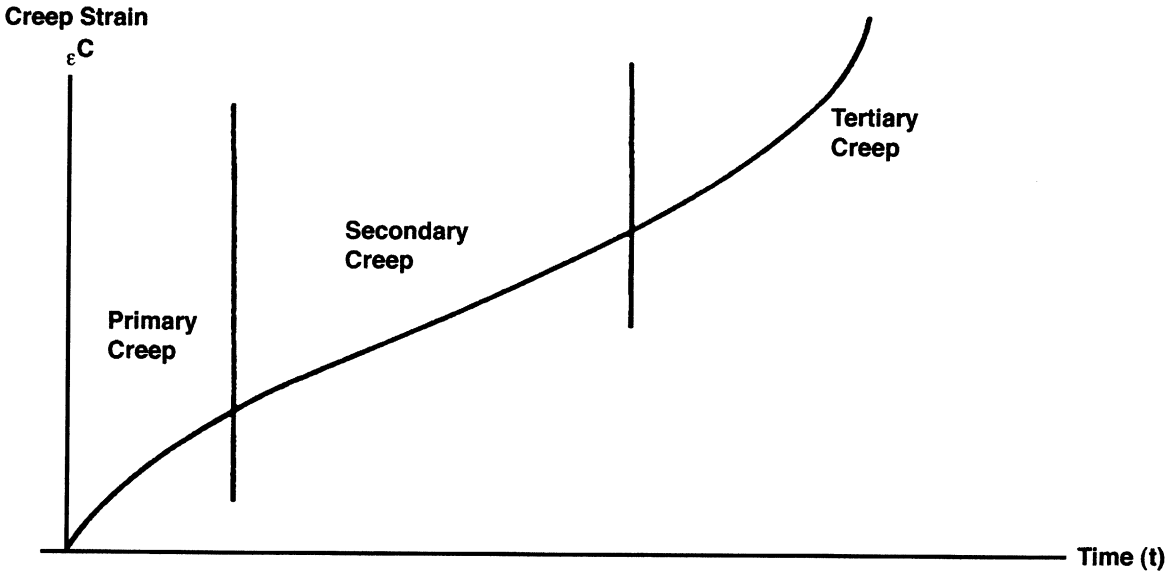


Figure A 6.6-8 Strain Response to Applied Stress (Kelvin Solid)



NOTE

Primary Creep:	Fast decrease in creep strain rate
Secondary Creep:	Slow decrease in creep strain rate
Tertiary Creep:	Fast increase in creep strain rate

Figure A 6.6-9 Creep Strain Versus Time (Uniaxial Test at Constant Stress and Temperature)

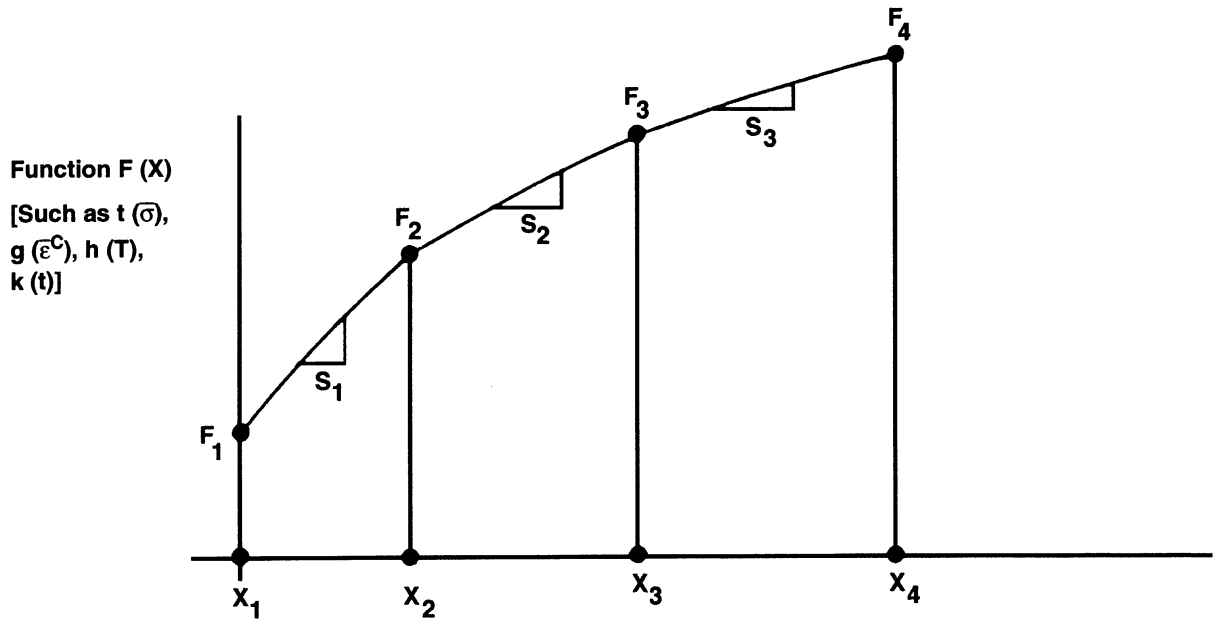
There are four possible modes of input for creep constitutive data.

1. Express the dependence of equivalent creep strain rate on any independent parameter through a piecewise linear relationship. The equivalent creep strain rate is then assumed to be a piecewise linear approximation to

$$\dot{\bar{\varepsilon}}^c = A \bullet f(\bar{\sigma}) \bullet g(\bar{\varepsilon}^c) \bullet h(T) \bullet \frac{dk(t)}{dt} \qquad \text{(A 6.6-8)}$$

where A is a constant; $\dot{\bar{\varepsilon}}^c$ is equivalent creep strain rate; and $\bar{\sigma}$, $\bar{\varepsilon}^c$, T, and t are equivalent stress, equivalent creep strain, temperature and time, respectively. The functions f, g, h, and k are piecewise linear and entered in the form as either slope-break point data or function-variable data. This representation is shown in Figure A 6.6-10. Enter functions f, g, h, and k through model definition option CREEP.

(Any of the functions, f, g, h, or k may be set to unity by setting the number of piecewise linear slopes for that relation to zero on the input data.)



(1) Slope-Break Point Data

Slope	Break Point
S_1	X_1
S_2	X_2
S_3	X_3

(2) Function-Variable Data

Function	Variable
F_1	X_1
F_2	X_2
F_3	X_3
F_4	X_4

Figure A 6.6-10 Piecewise Linear Representation of Creep Data

2. The dependence of equivalent creep strain rate on any independent parameter may be given directly in power law form by the appropriate exponent. The equivalent creep strain rate is

$$\dot{\bar{\varepsilon}}^c = A\bar{\sigma}^m \bullet \bar{\varepsilon}^{c^n} \bullet T^p \bullet (qt^{q-1}) \qquad \text{(A 6.6-9)}$$

Enter the constants A, m, n, p and q directly through model definition option CREEP. This is often adequate for engineering metals at constant temperature where Norton's rule is a good approximation.

$$\dot{\bar{\varepsilon}}^c = A\bar{\sigma}^n \qquad \text{(A 6.6-10)}$$

3. Define the equivalent creep strain rate directly with user subroutine CRPLAW.

4. Use the ISOTROPIC option to activate the ORNL (Oak Ridge National Laboratory rules) capability of the program.

Isotropic creep behavior is based on a von Mises creep potential described by the equivalent creep law

$$\dot{\bar{\varepsilon}} = \bar{f}(\bar{\sigma}, \bar{\varepsilon}^c, T, t) \qquad \text{(A 6.6-11)}$$

The material creep behavior is described by

$$\dot{\varepsilon}^c = \dot{\bar{\varepsilon}}^c \frac{\partial \bar{\sigma}}{\partial \sigma} \qquad \text{(A 6.6-12)}$$

where

$$\frac{\partial \bar{\sigma}}{\partial \sigma}$$

is normal to the yield surface.

During creep, the creep strain rate usually decreases. This effect is called creep hardening and can be a function of time or creep strain. The following section discusses the difference between these two types of hardening.

Consider a simple power law that illustrates the difference between time and strain-hardening rules for the calculation of the creep strain rate.

$$\varepsilon_c = \beta \bullet t^n \qquad \text{(A 6.6-13)}$$

where ε_c is the creep strain, β and n are values obtained from experiments and t is time. The creep rate can be obtained by taking the derivative ε_c with respect to time

$$\dot{\varepsilon}_c = \frac{d\varepsilon_c}{dt} = n \bullet \beta \bullet t^{n-1} \qquad \text{(A 6.6-14)}$$

However, t being greater than 0, we can compute the time t as

$$t = \left(\frac{\varepsilon_c}{\beta}\right)^{1/n} \qquad \text{(A 6.6-15)}$$

Substituting Equation A 6.6-14 into Equation A 6.6-13 we have

$$\dot{\varepsilon}_c = n\beta t^{n-1} = n\beta^{1/n} \bullet \varepsilon_c^{n-1/n} \qquad \text{(A 6.6-16)}$$

Equation A 6.6-14 shows that the creep strain rate is a function of time (time hardening). Equation A 6.6-16 indicates that the creep strain rate is dependent on the creep strain (strain hardening). The creep strain rates calculated from these two hardening rules generally are different. The selection of a hardening rule in creep analysis must be based on data obtained from experimental results. Figure A 6.6-11 and Figure A 6.6-12 show time and strain hardening rules in a variable state of stress. It is assumed that the stress in a structure varies from σ_1 to σ_2 to σ_3; depending upon the model chosen, different creep strain rates are calculated accordingly at points 1, 2, 3 and 4. Obviously, creep strain rates obtained from the time hardening rule are quite different from those obtained by the strain hardening rule.

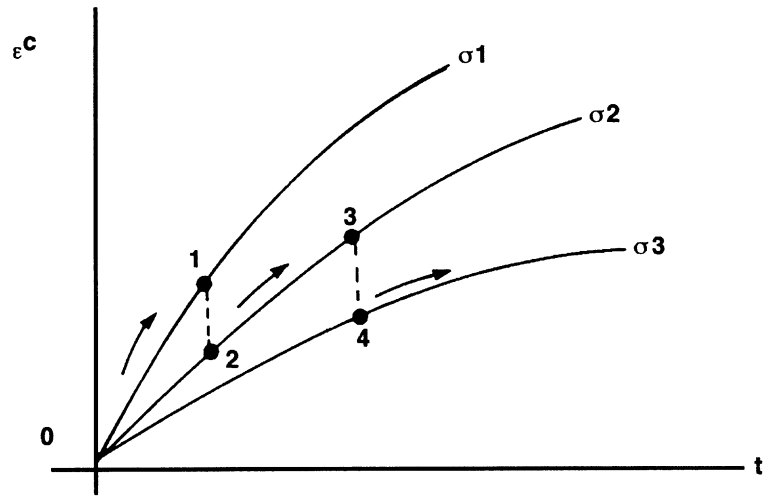


Figure A 6.6-11 TIME Hardening

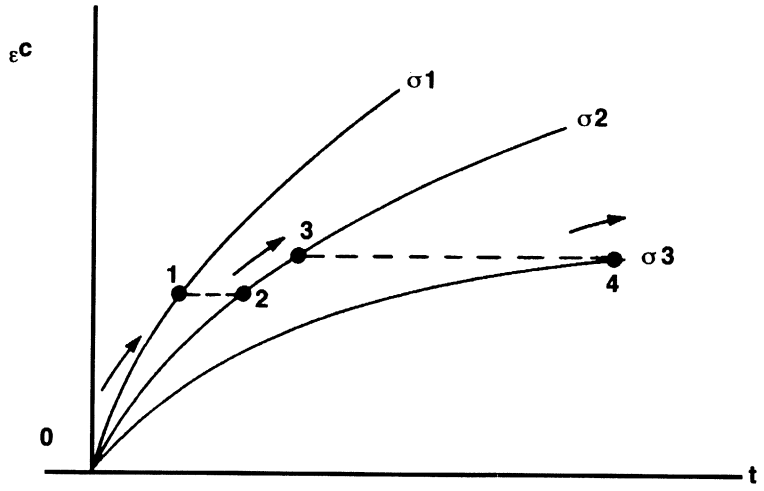


Figure A 6.6-12 STRAIN Hardening

Oak Ridge National Laboratory Laws

Oak Ridge National Laboratory (ORNL) has performed a large number of creep tests on stainless and other alloy steels. It has also set certain rules that characterize creep behavior for application in nuclear structures. A summary of the ORNL rules on creep is given below. The references listed at the end of this section offer a more detailed discussion of the ORNL rules.

1. Auxiliary Rules for Applying Strain-Hardening to Situations Involving Stress Reversals

   The Blackburn Creep Law is required as a user subroutine (CRPLAW). The parameter EQCP (first parameter in CRPLAW) is defined as

$$\bar{\varepsilon}^c = \left(\frac{2}{3}\Sigma\Delta\varepsilon^c_{ij}\Sigma\Delta\varepsilon^c_{ij}\right)^{1/2} \qquad \text{(A 6.6-17)}$$

   when the ORNL constitutive option is flagged through use of the ISOTROPIC option. In all other cases, the definition

$$\bar{\varepsilon}^c = \Sigma\left(\frac{2}{3}\Delta\varepsilon^c{}_{ij}\Delta\varepsilon^c_{ij}\right)^{1/2} \qquad \text{(A 6.6-18)}$$

   is retained. The equivalent primary creep strain passes into CRPLAW in EQCPNC, the second parameter. The second parameter must be redefined in that routine as the equivalent (total) creep strain increment. The first parameter (EQCP) must be redefined as the equivalent primary creep strain increment when the ORNL constitutive option is flagged. During analysis with the ORNL option, equivalent creep strain stores the distance between the two shifted origins in creep strain space (ε in ORNL-TM-3602). The sign on this value indicates which origin is currently active, so that a negative sign indicates use of the "negative" origin ($-\varepsilon_{ij}$).

2. Plasticity Effect on Creep

The effect of plastic strains on creep must be accommodated for the time-dependent creep behavior of 2 1/4 Cr -1 Mo Steel. Since plastic strains in one direction reduce the prior creep strain hardening accumulated in the reverse direction, ORNL recommends that the softening influence due to plastic strains be treated much the same as when reversed creep strain occurs. The following quantities are defined:

$$N_{ij}^{+} = (\varepsilon_{ij}^{I} - \varepsilon_{ij}^{+}) / G^{+} \qquad \text{(A 6.6-19)}$$

$$N_{ij}^{-} = (\varepsilon_{ij}^{I} - \varepsilon_{ij}^{-}) / G^{-} \qquad \text{(A 6.6-20)}$$

where ε_{ij}^{I} is instantaneous creep strain components

ε_{ij}^{+}; ε_{ij}^{-} = positive and negative strain origins (A 6.6-21)

and

$$G^{+} = G(\varepsilon_{ij}^{I} - \varepsilon_{ij})^{+} = [2/3 (\varepsilon_{ij}^{I} - \varepsilon_{ij}^{+}) (\varepsilon_{ij}^{I} - \varepsilon_{ij}^{+})]^{1/2} \qquad \text{(A 6.6-22)}$$

$$G^{-} = G(\varepsilon_{ij}^{I} - \varepsilon_{ij}^{-}) = [2/3 (\varepsilon_{ij}^{I} - \varepsilon_{ij}^{-}) (\varepsilon_{ij}^{I} - \varepsilon_{ij}^{-})]^{1/2} \qquad \text{(A 6.6-23)}$$

Swelling

The MARC program allows pure swelling (dilatational creep) effect in a creep analysis. To use the swelling option, perform a regular creep analysis as discussed in Section 6.6.1. Use the user subroutine VSWELL to define the increment of volumetric swelling (DV/V). The increment of volumetric swelling is generally a function of neutron flux, time, and temperatures.

For example, radiation-induced swelling strain model for 20% C. W. Stainless Steel 316 may be expressed as:

$$\frac{\Delta V}{V}\% = R\phi t + \frac{R}{\alpha} \ln\left[\frac{1 + \exp(\alpha(\tau - \phi t))}{1 + \exp\tau}\right] \qquad \text{(A 6.6-24)}$$

where R, t, and α are functions of temperature, ϕ is neutron flux, and t is time.

Viscoplasticity (Explicit Formulation)

The creep (Maxwell) model can be modified to include a plastic element (as shown in Figure A 6.6-13). This plastic element is inactive when the stress (σ) is less than the yield stress (σ_y) of the material. The modified model is an elasto- viscoplasticity model and is capable of producing some observed effects of creep and plasticity. In addition, the viscoplastic model can be used to generate time-independent plasticity solutions when stationary conditions are reached. At the other extreme, the viscoplastic model can reproduce standard creep phenomena. The model allows the treatment of nonassociated flow rules and strain softening which present difficulties in conventional (tangent modulus) plasticity analyses.

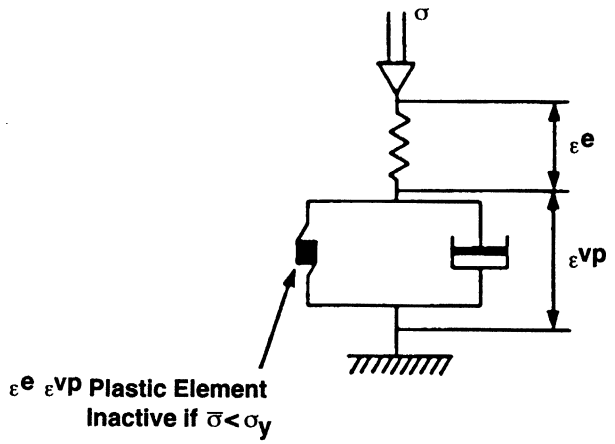


Figure A 6.6-13 Uniaxial Representation of Viscoplastic Material

The viscoplasticity option can be used to implement very general constitutive relations with the aid of the following user subroutines: ZERO, YIEL, NASSOC, and CRPLAW. See Chapter 5.2 for details on how to use these procedures.

It is recommended that users use the implicit formulation described in the following paragraph.

Creep (Implicit Formulation)

This formulation, as opposed to that described in the previous section, is fully implicit. A fully implicit formulation is unconditionally stable for any choice of time step size; hence, allowing a larger time step than permissible using the explicit method. Additionally, this is more accurate than the explicit method. The disadvantage is that each increment may be more computationally expensive. This model is activated using the CREEP parameter option. There are two methods for defining the inelastic strain rate. The CREEP model definition option may be used to define a Maxwell creep model. The back stress must be specified through the field reserved for the yield stress in the ISOTROPIC or ORTHOTROPIC options. There is no creep strain when the stress is less than the back stress. The equivalent creep strain increment is expressed as

$$\Delta\bar{e}^{i} = A\,(\bar{s}_{n+1} - \sigma_y)^{m}\Delta t$$

and the inelastic deviatoric strain components are

$$\Delta e^{i} = \frac{3}{2}\,\frac{\Delta\bar{\varepsilon}^{i}}{\bar{s}}\,S_{n+1}$$

where S_{n+1} is the deviatoric stress at the end of the increment and σ_y is the back stress. A is a function of temperature, time, etc. An algorithmic tangent is used to form the stiffness matrix.

To allow for the implementation of general unified creep-plasticity or viscoplastic models, user subroutine UVSCPL is available. This routine requires the user to define only the inelastic strain rate. The program automatically calculates a tangent stiffness matrix. This option is activated by indicating that the material is VISCO PLAS in the ISOTROPIC or ORTHOTROPIC option.

Based on a parameter defined in the CONTROL option, one of three tangent matrices will be formed. The first is using an elastic tangent, which will require more iterations, but may be computationally efficient because re-assembly may not be required. The second is an algorithmic tangent that provides the best behavior for small strain power law creep. The third is a secant (approximate) tangent that gives the best behavior for general viscoplastic models.

Viscoelastic Material

The MARC program has two models that represent viscoelastic materials. The first can be defined as a Kelvin-Voigt model. The latter is a general hereditary integral approach.

Kelvin-Voigt Model

The Kelvin model allows the rate of change of the inelastic strain to be a function of the total stress and previous strain. To activate the Kelvin model in MARC, use the CREEP parameter option.

The Kelvin material behavior (viscoelasticity) is modeled by assuming an additional creep strain ε^k_{ij}, governed by

$$\frac{d}{dt}\varepsilon^k_{ij} = A_{ijkl}\sigma'_{kl} - B_{ijkl}\varepsilon^k_{kl} \qquad \text{(A 6.6-25)}$$

where [A] and [B] are defined in the user subroutine CRPVIS.

$$\sigma'_{ij} = \sigma_{ij} - \delta_{ij}\sigma_{kk}/3 \qquad \text{(A 6.6-26)}$$

and the total strain is

$$\varepsilon_{ij} = \varepsilon^e_{ij} + \varepsilon^p_{ij} + \varepsilon^c_{ij} + \varepsilon^k_{ij} + \varepsilon^{th}_{ij} \qquad \text{(A 6.6-27)}$$

ε^{th}_{ij} = thermal strain components (A 6.6-28)

ε^e_{ij} = elastic strain components (instantaneous response) (A 6.6-29)

ε^p_{ij} = plastic strain components (A 6.6-30)

ε^c_{ij} = creep strains defined via CRPLAW and VSWELL routines (A 6.6-31)

ε^k_{ij} = Kelvin model strain components as defined above (A 6.6-32)

The CRPVIS routine is called at each integration point of each element when the Kelvin model is used.

Use the AUTO CREEP option to define the time-step and to set the tolerance control for the maximum strain in any increment.

The CREEP option allows Maxwell models to be included in series with the Kelvin model.

Hereditary Integral Model

The stress-strain equations in viscoelasticity are not only dependent on the current stress and strain state (as represented in the Kelvin model), but also on the entire history of development of these states. This constitutive behavior is most readily expressed in terms of hereditary or Duhamel integrals. These integrals are formed by considering the stress or strain built-up of increments applied at successive times. Two equivalent integral forms exist: the stress relaxation form and the creep function form. In the MARC program, the stress relaxation form is used.

The viscoelasticity option in MARC can be used for both the small strain and large strain (stress relaxation in Mooney material) problems. A description of these models is as follows:

Small Strain Viscoelasticity

In the stress relaxation form, the constitutive relation can be written as a hereditary integral formulation

$$\sigma_{ij}(t) = \int_0^t G_{ijkl}(t-\tau)\frac{d\varepsilon_{kl}(\tau)}{d\tau}d\tau \qquad \text{(A 6.6-33)}$$

The functions G_{ijkl} are called stress relaxation functions. They represent the response to a unit applied strain and have characteristic relaxation times associated with them. The relaxation functions for materials with a fading memory can be expressed in terms of Prony or exponential series.

$$G_{ijkl}(t) = G^{\infty}_{ijkl} + \sum_{n=1}^{N} G^{n}_{ijkl}\exp(-t/\lambda^{n}) \qquad \text{(A 6.6-34)}$$

in which G^{n}_{ijkl} is a tensor of amplitudes and λ^{n} is a positive time constant (relaxation time). In the current implementation, it is assumed that the time constant is isotropic. In equation (6.6-34), G^{∞}_{ijkl} represents the long term modulus of the material. The short term moduli (describing the instantaneous elastic effect) are then given by

$$G^{0}_{ijkl} = G_{ijkl}(0) = G^{\infty}_{ijkl} + \sum_{n=1}^{N} G^{n}_{ijkl} \qquad \text{(A 6.6-35)}$$

The stress can now be considered as the summation of the stresses in a generalized Maxwell model (Figure A 6.6-14)

$$\sigma_{ij}(t) = \sigma_{ij}^{\infty}(t) + \sum_{n=1}^{N} \sigma_{ij}^{n}(t) \qquad \text{(A 6.6-36)}$$

where

$$\sigma_{ij}^{\infty} = G_{ijkl}^{\infty} \varepsilon_{kl}(t) \qquad \text{(A 6.6-37)}$$

$$\sigma_{ij}^{n} = \int_{0}^{t} G_{ijkl}^{n} \exp\left[-(t-\tau)/\lambda^{n}\right] \frac{d\varepsilon_{kl}(\tau)}{d\tau} d\tau \qquad \text{(A 6.6-38)}$$

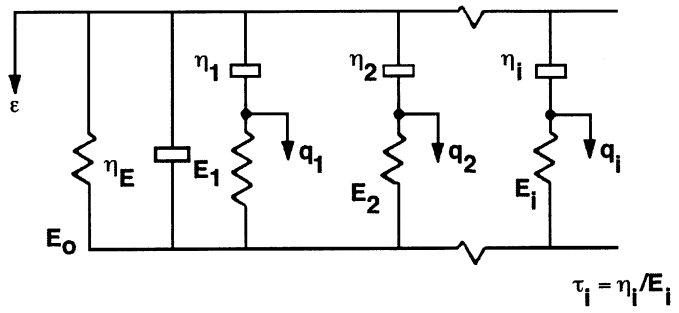


Figure A 6.6-14 The Generalized Maxwell or Stress Relaxation Form

For integration of the constitutive equation, the total time interval is subdivided into a number of subintervals $[t_{m-1}, t_m]$ with time-step $h = t_m - t_{m-1}$. A recursive relation can now be derived expressing the stress increment in terms of the values of the internal stresses σ_{ij}^{n} at the start of the interval. With the assumption that the strain varies linearly during the time interval h, we obtain

$$\sum_{n=1}^{N} \frac{\lambda^{n}}{h} \left[1 - \exp\left(-h/\lambda^{n}\right)\right] G_{ijkl}^{n} \Delta\varepsilon_{kl}(t_m) \qquad \text{(A 6.6-39)}$$

Let

$$\alpha^n(h) = 1 - \exp(-h/\lambda^n) \tag{A 6.6-40}$$

and

$$\beta^n(h) = \alpha^n(h)\,\lambda^n/h$$

The increment total stress-strain relation may then be expressed as

$$\Delta\sigma_{ij}(t_m) = \left[G^{\infty}_{ijkl} + \sum_{n=1}^{N} \beta^n(h)\, G^n_{ijkl} \right] \Delta\varepsilon_{kl} - \sum_{n=1}^{N} \alpha^n(h)\, \sigma^n_{ij}(t_m - h) \tag{A 6.6-41}$$

or the expression for each term is

$$\Delta\sigma^n_{ij}(t_m) = \beta^n(h)\, G^n_{ijkl} \Delta\varepsilon_{kl}(t_m) - \alpha^n(h)\, \sigma^n_{ij}(t_m - h) \tag{A 6.6-42}$$

In the MARC program, the incremental equation for the total stress is expressed in terms of the short term moduli (See Equation A 6.6-35).

$$\begin{aligned} \Delta\sigma_{ij}(t_m) = {} & \{ G^{o}_{ijkl} - \sum_{n=1}^{N} [1 - \beta^n(h)]\, G^n_{ijkl} \}\, \Delta\varepsilon_{kl}(t_m) \\ & - \sum_{n=1}^{N} \alpha^n(h)\, \sigma^n_{ij}(t_m - h) \end{aligned} \tag{A 6.6-43}$$

In this way, the instantaneous elastic moduli can be specified through the ISOTROPIC or ORTHOTROPIC options. Moreover, since the TEMPERATURE EFFECTS option acts on the instantaneous elastic moduli, it is more straightforward to use the short term values instead of the long term ones. Note that the set of equations given by Equation A 6.6-42 and Equation A 6.6-43 can directly be used for both anisotropic and isotropic materials.

Isotropic Viscoelastic Material

For an isotropic viscoelastic material, MARC assumes that the deviatoric and volumetric behavior are fully uncoupled and that the behavior can be described by a time dependent shear and bulk modules. The bulk moduli is generally assumed to be time independent, however this is an unnecessary restriction of the general theory.

Both the shear and bulk moduli can be expressed in a Prony series

$$G(t) = G^{\infty} + \sum_{n=1}^{N} {}^{D}G^{n} \exp(-t/\lambda_D) \qquad \text{(a)}$$

$$K(t) = K^{\infty} + \sum_{n=1}^{N} {}^{v}K^{n} \exp(-t/\lambda_D) \qquad \text{(b)} \qquad \text{(A 6.6-44)}$$

with short term values given by

$$G^{0} = G^{\infty} + \sum_{n=1}^{N} {}^{D}G^{n} \qquad \text{(a)}$$

$$K^{0} = K^{\infty} + \sum_{n=1}^{N} {}^{v}K^{n} \qquad \text{(b)} \qquad \text{(A 6.6-45)}$$

Let the deviatoric and volumetric component matrices π_D and π_V be given by

$$\pi_D = \begin{bmatrix} 4/3 & -2/3 & -2/3 & 0 & 0 & 0 \\ -2/3 & 4/3 & -2/3 & 0 & 0 & 0 \\ -2/3 & -2/3 & 4/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{(A 6.6-46)}$$

$$\pi_V = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The increment set of equations is then given by

$$\Delta\sigma(t_m) = \{G^0 - \sum_{n=1}^{N^D} [1-\beta_D^n(h)]\,G^n\}\,\pi_d\Delta\varepsilon(t_m) + \{K^0 - \sum_{n=1}^{N^v} [1-\beta_v^n(h)]\,K^n\}$$

$$\pi_M\Delta\varepsilon(t_m) - \sum_{n=1}^{N^D} \alpha_D^n(h)\,\sigma_D^n(t_m-h) - \sum_{n=1}^{N^V} \alpha_V^n(h)\,\sigma_V^n(t_m-h) \qquad \text{(A 6.6-47)}$$

and

$$\Delta\sigma_D^n(t_m) = \beta_D^n(h)\,G^n\pi_D\Delta\varepsilon(t_m) - \alpha_D^n(h)\,\sigma_D^n(t_m-h)$$

$$\Delta\sigma_V^n(t_m) = \beta_V^n(h)\,k^n\pi_V\Delta\varepsilon(t_m) - \alpha_V^n(h)\,\sigma_V^n(t_m-h) \qquad \text{(A 6.6-48)}$$

Note that the deviatoric and volumetric response are fully decoupled.

The instantaneous moduli need to be given in the ISOTROPIC option. Time dependent values (shear moduli G^n and time constants λ_D^n; bulk moduli K^n and time constants λ_v^n) need to be entered in the VISCELPROP option.

Time-stepping is performed using the TIME STEP or AUTO TIME option in the history definition block.

Note that the algorithm is exact for linear variations of the strain during the increment. The algorithm is an implicit one and for each change in time-step a new assembly of the stiffness matrix is required.

Anisotropic Viscoelastic Material

The set of equations Equation A 6.6-42 and Equation A 6.6-43 can be used for the analysis of anisotropic viscoelastic materials. The tensor of amplitudes G_{ijkl}^0 must be entered through the ORTHOTROPIC option. However, both the G_{ijkl}^n and the λ^n must be entered using the VISCELORTH option.

Alternatively, a complete set of moduli (21 components) can be specified in the user subroutine HOOKVI.

The ORIENTATION option or user subroutine ORIENT can be used to define a preferred orientation both for the short time moduli G_{ijkl}^0 and the amplitude functions G_{ijkl}^n.

Incompressible Isotropic Viscoelastic Materials

Incompressible elements in MARC allow the analysis of incompressible and nearly incompressible materials in plane strain, axisymmetric and three-dimensional problems. The incompressibility of the element is simulated through the use of an augmented variational principle based on the Herrmann formulation.

The constitutive equation for a material with no time dependence in the volumetric behavior can be expressed as

$$\Delta\sigma_{ij}(t_m) = 2\{G^0 - \sum_{n=1}^{N}[1-\beta^n(h)]G^n\}\left[\Delta\varepsilon_{ij}(t_m) - \frac{1}{3}\Delta\varepsilon_{kk}(t_m)\delta_{ij}\right] - \sum_{n=1}^{N}\alpha^n(h)\,s_{ij}^n(t_m) + \frac{1}{3}\sigma_{kk}\delta_{ij} \qquad \text{(A 6.6-49)}$$

$$\Delta\sigma_{kk}(t_m) = 3K^0\Delta\varepsilon_{kk}(t_m) \qquad \text{(A 6.6-50)}$$

The hydrostatic pressure term will be used as an independent variable in the variational principle. The Herrmann pressure variable will now be defined in the same way as in the formulation for time independent elastic materials.

$$H = \frac{\sigma_{kk}}{2G^0(1+\nu^0)} \qquad \text{(A 6.6-51)}$$

The constitutive Equation A 6.6-49 and Equation A 6.6-50 can then be rewritten

$$\Delta\sigma_{ij}(t_m) = 2G^e(\Delta\varepsilon_{ij} + \nu^* H\delta_{ij}) - \sum_{n=1}^{N}\alpha^n(h)\,s_{ij}^n(t_m - h) \qquad \text{(A 6.6-52)}$$

where

$$G^e = G^0 - \sum_{n=1}^{N}[1-\beta^n(h)G^n] \qquad \text{(A 6.6-53)}$$

$$\nu^* = \frac{G^0(1+\nu^0) - G^e(1-2\nu^0)}{3G^e} \qquad \text{(A 6.6-54)}$$

Thermo-Rheologically Simple Behavior

The rate processes in many viscoelastic materials are known to be highly sensitive to temperature changes. Such temperature-dependent properties cannot be neglected in the presence of any appreciable temperature variation. For example, there is a large class of polymers which are adequately represented by linear viscoelastic laws at uniform temperature. These polymers exhibit an approximate translational shift of all the characteristic response functions with a change of temperature, along a logarithmic time axis. This shift occurs without a change of shape. These temperature-sensitive viscoelastic materials are characterized as thermo-rheologically simple.

A "reduced" or "pseudo" time can be defined for the materials of this type and for a given temperature field. This new parameter is a function of both time and space variables. The viscoelastic law has the same form as one at constant temperature in real time. If the shifted time is used, however, the transformed viscoelastic equilibrium and compatibility equations are not equivalent to the corresponding elastic equations.

In the case where the temperature varies with time, the extended constitutive law implies a nonlinear dependence of the instantaneous stress state at each material point of the body upon the entire local temperature history. In other words, the functionals are linear in the strains but nonlinear in the temperature.

The time scale of experimental data is extended for thermo-rheologically simple materials. All characteristic functions of the material must obey the same property. The shift function is a basic property of the material and must be determined experimentally. As a consequence of the shifting of the mechanical properties data parallel to the time axis (see Figure A 6.6-15), the values of the zero and infinite frequency complex moduli do not change due to shifting. Hence, elastic materials with temperature-dependent characteristics neither belong to nor are consistent with the above hypothesis for the class of thermo-rheologically simple viscoelastic solids.

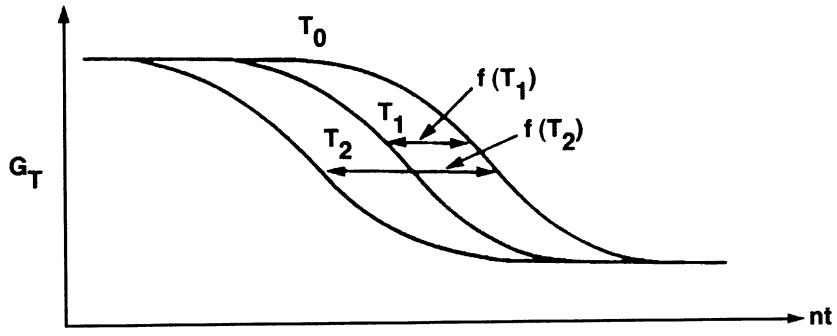


Figure A 6.6-15 Relaxation Modulus vs. Time at Different Temperatures

Let E (*ln t*) be the relaxation modulus as a function of ln t at uniform temperature, T. Then

$$E_T(\ln\ t) = E_{T_o}[\ln\ t + f(T)] * (\rho T/\rho_o T_o) \qquad \text{(A 6.6-55)}$$

where f(T) is measured relative to some arbitrary temperature T. The modulus curve shifts towards shorter times with an increase of temperature; f(T) is a positive increasing function for $T > T_0$. If $G_T(t)$ denotes the relaxation modulus as a function of time at uniform temperature T, so that,

$$G_T(t) = E_T(\ln t) \qquad \text{(A 6.6-56)}$$

then

$$G_T(t) = G_{T_0}(\varepsilon) \qquad \text{(A 6.6-57)}$$

The relaxation modulus (and the other characteristic functions) at an arbitrary uniform temperature is thus expressed by the base temperature behavior related to a new time scale that depends on that temperature.

There is some mapping of the time coordinate for nonuniform, nonconstant temperature, T(x,t), which depends on the position

$$\varepsilon = \varepsilon(x, t) \qquad \text{(A 6.6-58)}$$

and

$$d\varepsilon = a[T(x, t)]\, dt \qquad \text{(A 6.6-59)}$$

The MARC program offers two explicit forms for entering the shift function. The first is based on the familiar Williams-Landel-Ferry (WLF) equation. Rewriting the above expression for reduced time as

$$e(x, t) = \int_0^t \frac{dt'}{A_T[T(x, t')]} \qquad \text{(A 6.6-60)}$$

then the WLF form state that

$$\log_{10} A_T(T) = \frac{-C_1(T - T_0)}{C_2 + (T - T_0)} = -h(T) \qquad \text{(A 6.6-61)}$$

and

$$e(t) = \int_0^t 10^{h[T(t')]} dt' \qquad \text{(A 6.6-62)}$$

Typically, the glassy transition point is taken as the reference temperature in the above relation. The logarithmic shift may also be expressed in a polynomial expansion about the arbitrary reference point as

$$\log_{10} A_T(T) = \sum_{i=o}^{m} a_i (T - T_0)\, i \qquad \text{(A 6.6-63)}$$

Enter the shift function parameters associated with thermo-rheologically simple behavior through model definition option SHIFT FUNCTION. As an alternative to the WLF function, MARC allows use of Series Expansion or specification via the user subroutine TRSFAC.

In addition to the thermo-rheologically simple material behavior variations of initial stress-strain moduli G^0_{ijkl}, the temperature of the other mechanical properties (coefficient of thermal expansion, etc.) due to changes in temperature can be specified via the TEMPERATURE EFFECTS option.

Note, however, that only the instantaneous moduli are effected by the TEMPERATURE EFFECTS option. Hence, the long term moduli given by

$$G^{\infty}_{ijkl} = G^0_{ijkl}(t) - \sum_{n=1}^{N} G^n_{ijkl} \qquad \text{(A 6.6-64)}$$

can easily become negative if the temperature effects are not defined properly.

Large Strain Viscoelasticity

For an elastomeric time independent material, the constitutive equation is expressed in terms of an energy function W. For a large strain viscoelastic material, Simo generalized the small strain viscoelasticity material behavior to a large strain viscoelastic material. The energy functional then becomes

$$\psi(E_{ij} Q^n_{ij}) = \psi^0(E_{ij}) - \sum_{n=1}^{N} Q^n_{ij} E_{ij} + \sum_{n=1}^{N} \psi^n_I(Q^n_{ij}) \qquad \text{(A 6.6-65)}$$

where E_{ij} are the components of the Green-Lagrange strain tensor, Q^n_{ij} internal variables and ψ^0 the elastic strain energy density for instantaneous deformations. In MARC, it is assumed that $\psi^0 = W$, meaning that the energy density for instantaneous deformations is given by the third order James Green and Simpson form or the Ogden form.

The components of the second Piola-Kirchhoff stress then follow from

$$S_{ij} = \frac{\partial \psi}{\partial E_{ij}} = \frac{\partial \psi^0}{\partial E_{ij}} - \sum_{n=1}^{N} Q^n_{ij} \qquad \text{(A 6.6-66)}$$

The energy function can also be written in terms of the long term moduli resulting in a different set of internal variables T_{ij}^n

$$\psi(E_{ij}, T_{ij}^n) = \psi^{\infty}(E_{ij}) + \sum_{n=1}^{N} T_{ij}^n E_{ij} \qquad \text{(A 6.6-67)}$$

where ψ^{∞} is the elastic strain energy for long term deformations. Using this energy definition, the stresses are obtained from

$$S_{ij} = \frac{\partial \psi^{\infty}(E)}{\partial E_{ij}} + \sum_{n=1}^{N} T_{ij}^n \qquad \text{(A 6.6-68)}$$

Observing the similarity with the equations for small strain viscoelasticity, Equation A 6.6-36 and Equation A 6.6-38, the internal variables can be obtained from a convolution expression

$$T_{ij}^n = \int_0^t \dot{S}_{ij}^n(\tau) \exp[-(t-\tau)/\lambda^n] \, d\tau \qquad \text{(A 6.6-69)}$$

where S_{ij}^n are internal stresses obtained from energy functions.

$$S_{ij}^n = \frac{\partial \psi^n}{\partial E_{ij}} \qquad \text{(A 6.6-70)}$$

Let the total strain energy be expressed as a Prony series expansion

$$\psi = \psi^{\infty} + \sum_{n=1}^{N} \psi^n \exp(-t/\lambda^n) \qquad \text{(A 6.6-71)}$$

If, furthermore, in the energy function each term in the series expansion has a similar form, Equation A 6.6-61 can be rewritten

$$\psi = \psi^{\infty} + \sum_{n=1}^{N} \delta^n \psi^0 \exp(-t/\lambda^n) \qquad \text{(A 6.6-72)}$$

where δ^n is a scalar multiplier for the energy function based on the short term values.

The stress-strain relation is now given by

$$S_{ij}(t) = S_{ij}^{\infty}(t) + \sum_{n=1}^{N} T_{ij}^{n}(t) \qquad \text{(A 6.6-73)}$$

$$S_{ij} = \frac{\partial \psi^{\infty}}{\partial E_{ij}} = \left(1 - \sum_{n=1}^{N} \delta^{n}\right) \frac{\partial \psi^{0}}{\partial E_{ij}} \qquad \text{(A 6.6-74)}$$

$$T_{ij}^{n} = \int_{o}^{t} \delta^{n} S_{ij}^{0}(t) \exp\left[-(t-\tau)/\lambda^{n}\right] d\tau \qquad \text{(A 6.6-75)}$$

Analogue to the derivation for small strain viscoelasticity, a recursive relation can be derived expressing the stress increment in terms of values of the internal stresses at the start of the increment.

In the MARC program, the instantaneous values of the energy function are always given on the MOONEY or the OGDEN option, the equations are reformulated in terms of the short time values of the energy function

$$\Delta S_{ij}(t_m) = \{1 - \sum_{n=1}^{N} [1 - \beta^{n}(h)]\,\delta^{n}\}\,\{S_{ij}^{o}(t_m) - S_{ij}^{o}(t_m - h)\} - \sum_{n=1}^{N} \alpha^{n} S_{ij}^{n}(t_m - h) \qquad \text{(A 6.6-76)}$$

$$\Delta S_{ij}(t_m) = \beta^{n}(h)\,\delta^{n}\,[S_{ij}^{o}(t_m) - S_{ij}^{n}(t_m - h)] - \alpha^{n}(h)\,S_{ij}^{n}(t_m - h) \qquad \text{(A 6.6-77)}$$

It is assumed that the viscoelastic behavior in MARC acts only on the deviatoric behavior. The incompressible behavior is taken into account using special Herrmann elements.

Large strain viscoelasticity is invoked by use of the VISCEL MOON or VISCEL OGDEN option in the constitutive option of the model definition deck. The instantaneous energy function is specified through the MOONEY or OGDEN option. The time dependent multipliers δ^{n} and associated relaxation times λ^{n} as defined by Equation A 6.6-62 are given in the VISCEL MOON or VISCEL OGDEN option. For the Ogden model, both deviatoric and dilatational relaxation behavior is allowed.

Time-stepping can be performed using the TIME STEP with AUTO LOAD, or AUTO TIME option of the history definition block.

A 6.7 TEMPERATURE EFFECTS AND COEFFICIENT OF THERMAL EXPANSION

Experimental results indicate that a large number of material properties vary with temperatures. The MARC program accepts the temperature-dependent material properties shown in Table A 6.7-1 in the form of piecewise linear functions for different types of analysis.

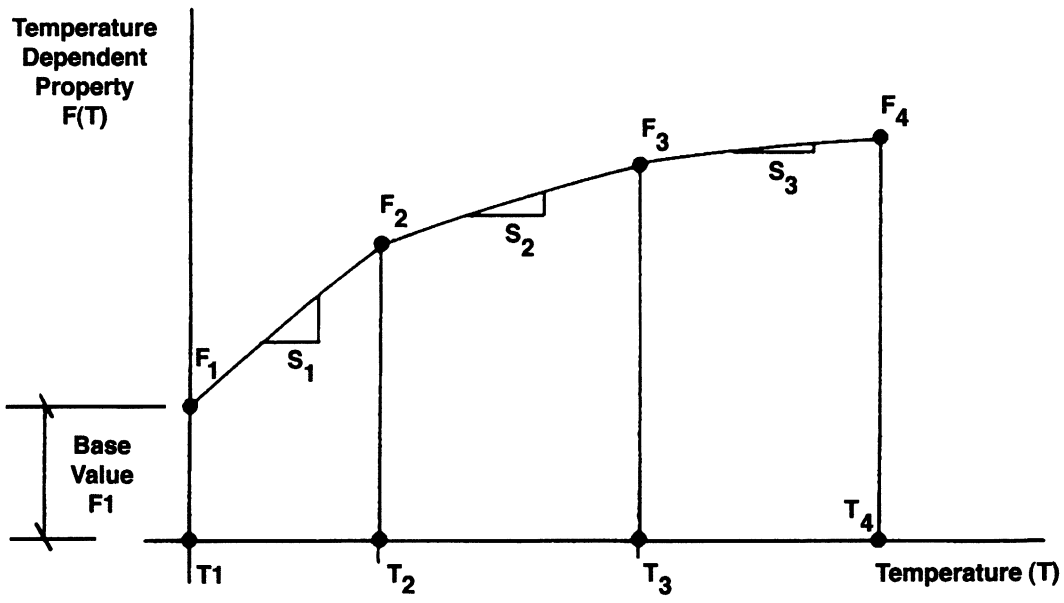
Table A 6.7-1 Temperature-Dependent Material Properties

Analysis Type	Material Properties	
Stress Analysis	Modulus of elasticity (Young's Modulus)	E(T)
	Poisson's Ratio	υ(T)
	Yield Stress	σ_y(T)
	Work Hardening Slope	h(T)
	Coefficient of Thermal Expansion	α(T)
	Mooney Constants	C_{01}(T),C_{10}(T)
Heat Transfer Analysis	Thermal Conductivity	K(T)
	Specific Heat	C(T)
Couples Thermo-Electrical (Joule Heating Analysis)	Electric Resistivity	π(T)
Hydrodynamic Heating	Viscosity	μ(T)

Please note that T is temperature in the above expressions. With the exception of heat transfer or Joule heating, the temperature is a state variable.

Piecewise Linear Representation

In MARC, the temperature variation of a material constant F(T) is assumed to be a piecewise linear function of temperature. Figure A 6.7-1 illustrates this function. Input the piecewise linear function of temperature using model definition options ISOTROPIC, ORTHOTROPIC, TEMPERATURE EFFECTS, and ORTHO TEMP. Enter the base value or value at reference temperature (the lowest temperature that will occur during your analysis), through the ISOTROPIC, and ORTHOTROPIC model definition options. Input either the slope-breakpoint data or the function-temperature data through the model definition options, TEMPERATURE EFFECTS and ORTHO TEMP.



(1) Slope-Break Point Data

Slope	Break Point
$S_1 = (F_2 - F_1)/(T_2 - T_1)$	T_1
$S_2 = (F_3 - F_2)/(T_3 - T_2)$	T_2
$S_3 = (F_4 - F_3)/(T_4 - T_3)$	T_3

(2) Function-Variable Data

Function	Variable
F_1	T_1
F_2	T_2
F_3	T_3
F_4	T_4

Figure A 6.7-1 Piecewise Linear Representation of Temperature-Dependent Material Properties

Temperature-Dependent Creep

In MARC, the temperature dependency of creep strain can be entered in two ways. The creep strain rate may be entered as a piecewise linear function. If the creep strain $\dot{\varepsilon}^c$ can be expressed in the form of a power law

$$\dot{\varepsilon}^c = AT^m \qquad \text{(A 6.7-1)}$$

where A and m are two experimental constants, input the experimental constants through the CREEP model definition option.

For other temperature dependency, you must use the user subroutine CRPLAW to input the variation of creep strain with temperature.

Coefficient of Thermal Expansion

The MARC program always uses an instantaneous thermal expansion coefficient definition

$$d\varepsilon_{ij}^{th} = \alpha_{ij} dT \qquad \text{in general} \tag{A 6.7-2}$$

or

$$d\varepsilon_{ij}^{th} = \alpha dT \qquad \text{for the isotropic case no sum on i} \tag{A 6.7-3}$$

In many cases, the thermal expansion data is given with respect to a reference temperature

$$\varepsilon^{th} = \bar{\alpha}(T - T^0) \tag{A 6.7-4}$$

where α is a function of temperature:

$$\bar{\alpha} = \bar{\alpha}(T) \tag{A 6.7-5}$$

Clearly, in this case

$$d\varepsilon^{th} = \left[\bar{\alpha} + \frac{d\bar{\alpha}}{dT}(T - T^0)\right] dT \tag{A 6.7-6}$$

so the necessary conversion procedure is

1. Compute and plot Equation A 6.7-4 in the form Equation A 6.7-6

$$\alpha = \bar{\alpha} + \frac{d\bar{\alpha}}{dT}(T - T^0) \tag{A 6.7-7}$$

   as a function of temperature.

2. Model Equation A 6.7-7 in user subroutine ANEXP, or with piecewise linear slopes and breakpoints in the TEMPERATURE EFFECTS option.

   The anisotropic coefficient of thermal expansion can be input through either the model definition option ORTHOTROPIC or the user subroutine ANEXP.

A 6.8 TIME-TEMPERATURE-TRANSFORMATION

Certain materials, such as carbon steel, exhibit a change in mechanical or thermal properties when quenched or air cooled from a sufficiently high temperature. At any stage during the cooling process, these properties are dependent on both the current temperature and the previous thermal history. The properties are influenced by the internal microstructure of the material, which in turn depends on the rate at which the temperature changes. Only in instances where the temperature is changed very gradually does the material respond in equilibrium, where properties are simply a function of the current temperature. In addition, during the cooling process certain solid-solid phase transformations may occur. These transformations represent another form of change in the material microstructure which can influence the mechanical or thermal properties. These transformations may be accompanied by changes in volume.

The occurrence of phase change is also dependent on the rate of cooling of the material. This relationship is shown in a typical cooling diagram (see Figure A 6.8-1). The curves A, B, and C in Figure A 6.8-1 represent the temperature history of a structure that has been subjected to a different cooling rate. It is obvious that the structural material experiences phase changes at different times and temperatures, depending on the rate of cooling. Under cooling rate A, the material changes from phase 1 to phase 4 directly. The material undergoes three phase changes (phase 1 to phase 2 to phase 3 to phase 4) for both cooling rates B and C. However, the phase changes take place at different times and temperatures.

The Time-Temperature-Transformation (T-T-T) option allows you to account for the time-temperature-transformation interrelationships of certain materials during quenching or casting analyses.

Use the T-T-T parameter to invoke the T-T-T option. Input all the numerical data required for this option through the model definition option TIME-TEMP.

In a transient heat transfer analysis, the thermal properties which can be defined as a function of time and temperature are the thermal conductivity and the specific heat per unit reference mass. Here, the effects of latent heat or phase transformation can be included through the definition of the specific heat.

In a thermal stress analysis, the mechanical properties which can be defined as a function of time and temperature are the Young's modulus, Poisson's ratio, yield stress, work hardening slope, and coefficient of thermal expansion. The effects of volumetric change due to phase transformation can be included through the definition of the thermal coefficient of expansion.

Test data must be available in a tabular form for each property of each material group. For a given cooling rate, the value of a property must be known at discrete points over a range of temperatures. There can be several sets of these discrete points corresponding to measurements at several different cooling rates. The cooling tests must be of a specific type known as Newton Cooling, i.e., the temperature change in the material is controlled such that

$$T(t) = A \exp(-at) + B \qquad \text{(A 6.8-1)}$$

In addition, a minimum and a maximum temperature that bracket the range over which the T-T-T option is meant to apply must also be given.

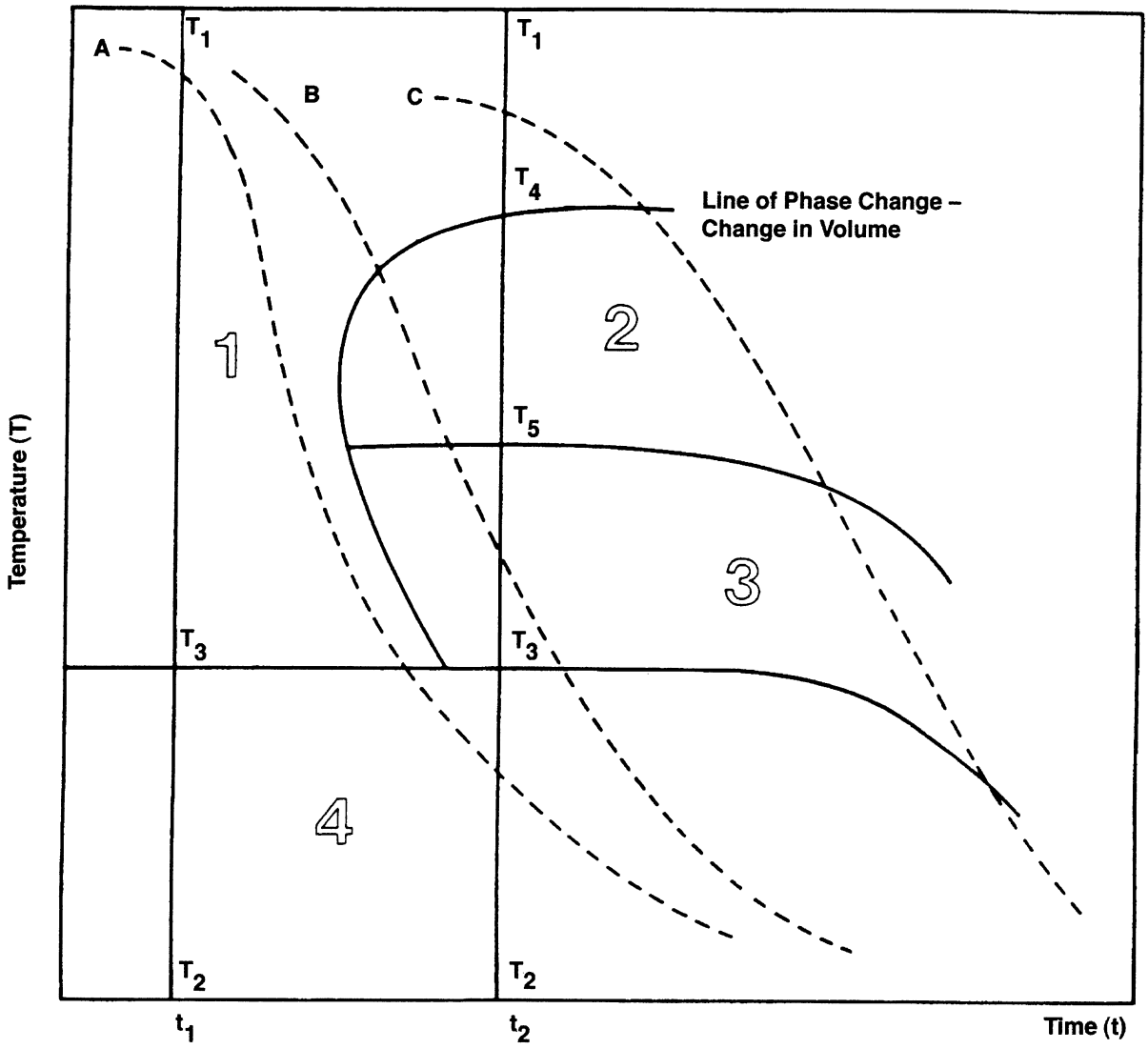


Figure A 6.8-1 Simplified Cooling Transformation Diagram

For the simulation of the cooling rate effect in finite element analysis, material properties of a structure can be assumed as a function of two variables: time and temperature. Two-dimensional interpolation schemes are used for the interpolation of properties.

Interpolation is based on making the time variable discrete. Stress analysis is carried out incrementally at discrete time stations and material properties are assumed to vary piecewise linearly with temperature at any given time. These temperature-dependent material properties are updated at each increment in the analysis. For illustration, at time t_1, the material is characterized by the phase 1 and phase 4 behaviors at temperature ranges T_1 to T_3, and T_3 to T_2, respectively (see Figure A 6.8-1). Similarly, at time t_2, the material behavior must be characterized by all four phases, each in a different temperature range (i.e., phase 1, T_1 to T_4; phase 2, T_4 to T_5; phase 3, T_5 to T_3; phase 4, T_3 to T_2). The selection of an interpolation scheme

is generally dependent on the form of the experimental data. A linear interpolation procedure can be effectively used where the properties are expressed as a tabulated function of time and temperature.

During T-T-T, the change in volume in a stress analysis is assumed to take place in a temperature range ΔT. The change in volume is also assumed to be uniform in space, such that the effect of the volume change can be represented by a modification of the coefficient of thermal expansion. For a triangular distribution of $\alpha(T)$ in the temperature range ΔT, the value of the modified coefficient of thermal expansion is

$$\alpha_m = \frac{2}{\Delta T}\left[\sqrt[3]{1+\phi} - 1\right] \qquad \text{(A 6.8-2)}$$

where ϕ is the change in volume. A schematic of the modified $\alpha(T)$ is shown in Figure A 6.8-2.

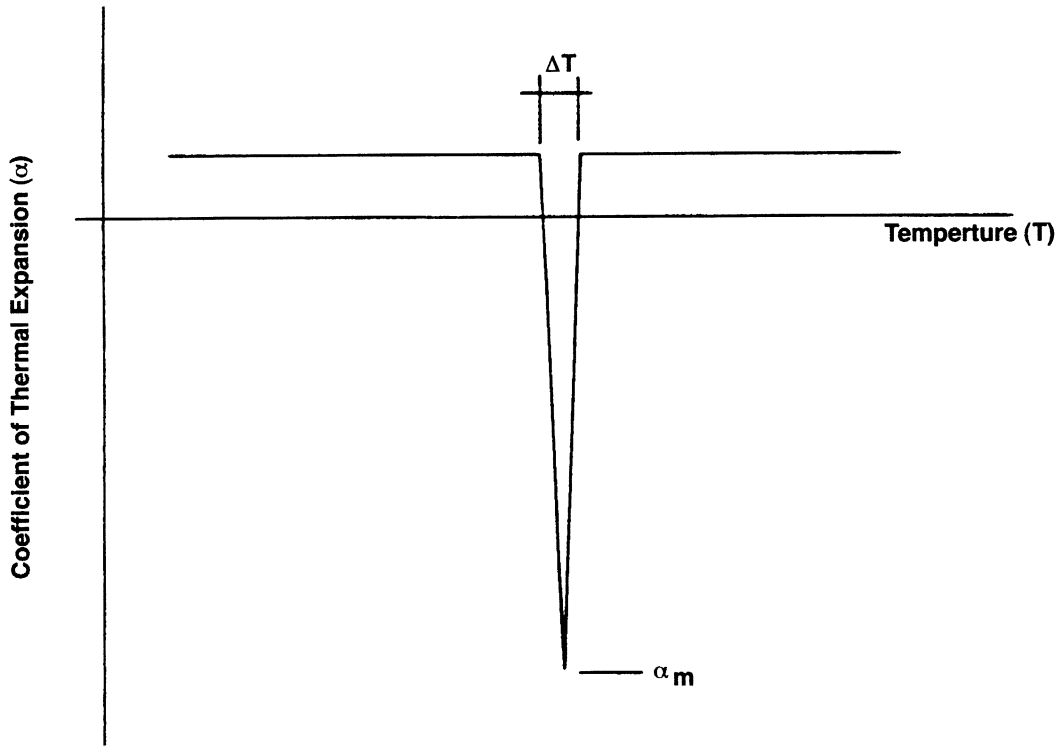


Figure A 6.8-2 Modified Coefficient of Thermal Expansion for Short-Time Change in Volume

A 6.9 LOW TENSION MATERIAL

The MARC program can handle concrete and other low tension material. The CRACKING option assists in predicting crack initiation and in simulating tension softening, plastic yielding and crushing. This option can be used for the following:

- Elements with a one-dimensional stress-strain relation (beam and truss elements)
- Elements with a two-dimensional stress-strain relation (plane stress, plane strain, and shell elements)
- Three-dimensional elements (bricks)

Analytical procedures that accurately determine stress and deformation states in concrete structures are complicated by several factors. Two such factors are the following:

- The low strength of concrete in tension that results in progressive cracking under increasing loads
- The nonlinear load-deformation response of concrete under multiaxial compression

Because concrete is mostly used in conjunction with steel reinforcement, an accurate analysis requires consideration of the components forming the composite structure. Steel reinforcement bars are introduced as REBAR elements.

Each REBAR element must be input with a separate element number, and the same connectivity as the filler element. User subroutine REBAR is used to define the orientation of the reinforcement rods.

Uniaxial Cracking Data

The cracking option is accessed through the ISOTROPIC option. Uniaxial cracking data can be specified using the CRACK DATA option or the user subroutine UCRACK. When the CRACK DATA option is used to specify uniaxial cracking data, the following must be specified: the critical cracking stress, the modulus of the linear strain softening behavior, and the strain at which crushing occurs. Material properties, such as Young's modulus and Poisson's ratio, are entered using the ISOTROPIC block and the WORK HARD data block. This model is for a material which is initially isotropic; if the model is initially orthotropic, see FAIL DATA for an alternative cracking model. A typical uniaxial stress-strain diagram is shown in Figure A 6.9-1.

Low Tension Cracking

A crack develops in a material perpendicular to the direction of the maximum principal stress if the maximum principal stress in the material exceeds a certain value (see Figure A 6.9-2). After an initial crack has formed at a material point, a second crack may form perpendicular to the first. Likewise, a third crack may form perpendicular to the first two. The material loses all load-carrying capacity across the crack, unless tension softening is included.

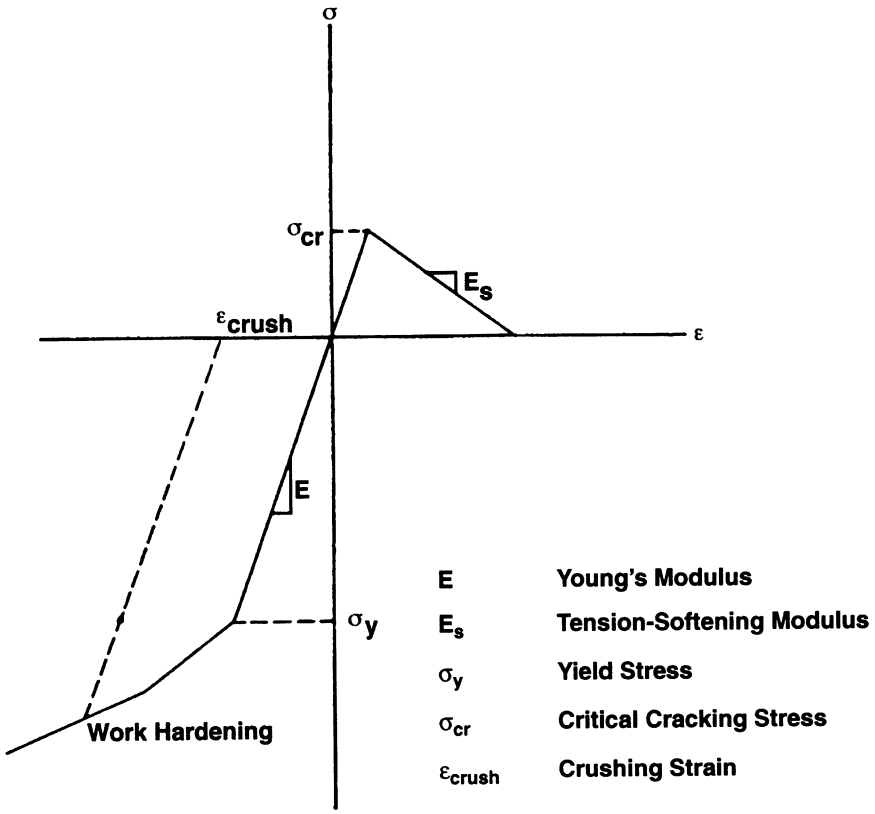


Figure A 6.9-1 Uniaxial Stress-Strain Diagram

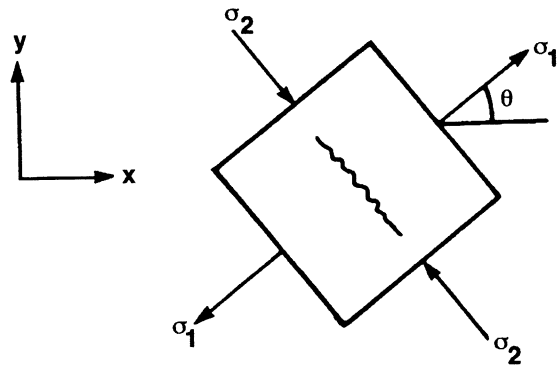


Figure A 6.9-2 Crack Development

Tension Softening

If tension softening is included, the stress in the direction of maximum stress does not go immediately to zero; instead the material softens until there is no stress across the crack. At this point, no load-carrying capacity exists in tension (see Figure A 6.9-1). The softening behavior is characterized by a descending branch in the tensile stress-strain diagram, and it may be dependent upon the element size.

Crack Closure

After a crack forms, the loading may be reversed; therefore, the opening distance of a crack must be considered. In this case, the crack may close again, and partial mending occurs. When mending occurs, it is assumed that the crack has full compressive stress-carrying capability and that shear stresses are transmitted over the crack surface, but with a reduced shear modulus.

Crushing

As the compressive stress level increases, the material eventually loses its integrity, and all load-carrying capability is lost; this is referred to as crushing. Crushing behavior is best described in a multiaxial stress state by a crushing surface having the same shape as the yield surface. The failure criterion can be used for a two-dimensional stress state with reasonable accuracy. For many materials, experiments indicate that the crushing surface is roughly three times larger than the initial yield surface.

A 6.10 DAMAGE MODELS

In many structural applications, the finite element method is used to predict failure. This is often performed by comparing the calculated solution to some failure criteria, or by using classical fracture mechanics. Previously, we discussed two models where the actual material model changed due to some failure, see PROGRESSIVE COMPOSITE FAILURE and the previous section on LOW TENSION MATERIALS. In this section, the damage models appropriate for ductile metals and rubber materials will be discussed.

Ductile Metals

In ductile materials given the appropriate loading conditions, voids will form in the material, grow, then coalesce, leading to crack formation and potentially, failure. Experimental studies have shown that these processes are strongly influenced by the presence of a hydrostatic stress. Gurson studied microscopic voids in materials and derived a set of modified constitutive equations for elastic-plastic materials. Tvergaard and Needleman modified the model with respect to the behavior at small void volume fractions and at void coalescence.

In the modified Gurson model, the amount of damage is indicated with a scalar parameter called the void volume fraction f. The yield criterion for the macroscopic assembly of voids and matrix materials is given by:

$$F = \left(\frac{\bar{\sigma}}{\sigma_y}\right)^2 + 2q_1 f^* \cosh\left(\frac{q_2 \sigma_{kk}}{2\sigma_y}\right) - [1 - (q_1 f^*)^2] = 0 \qquad \text{(A 6.10-1)}$$

as seen in Figure A 6.10-1.

The parameter q_1 was introduced by Tvergaard to improve the Gurson model at small values of the volume fraction. The parameter f* is introduced to model the rapid decrease in load carrying capacity if void coalescence occurs.

$$\begin{aligned} f^* &= f && \text{if } f \le f_c \\ f^* &= f_c + \left(\frac{f_u^* - f_c}{f_F - f_c}\right)(f - f_c) && \text{if } f > f_c \end{aligned} \qquad \text{(A 6.10-2)}$$

where f_c is the critical void volume fraction, f_F is the void volume at failure, and $f_u = 1/q$.

The existing value of the void volume fraction changes due to the growth of existing voids and due to the nucleation of new voids.

$$\dot{f} = \dot{f}_{growth} + \dot{f}_{nucleation} \qquad \text{(A 6.10-3)}$$

The growth of voids can be determined based upon incompressibility of the matrix material surrounding the void.

$$\dot{f}_{growth} = (1 - f)\,\dot{\varepsilon}^p_{kk} \qquad \text{(A 6.10-4)}$$

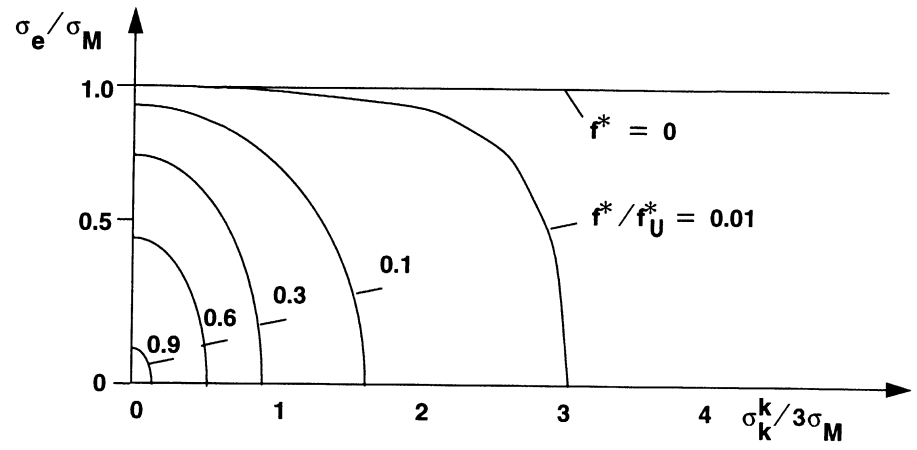


Figure A 6.10-1 Plot of Yield Surfaces in Gurson Model

One can define the nucleation of new voids to be either strain or stress controlled. Both follow a normal distribution about a mean value.

In the case of strain controlled nucleation, this is given by

$$\dot{f}_{\text{nucleation}} = \frac{f_N}{S\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\varepsilon_m^p - \varepsilon_n}{S}\right)^2\right]\dot{\varepsilon}_m^p \qquad \text{(A 6.10-5)}$$

where f_N is the volume fraction of void forming particles, ε_n the mean strain for void nucleation and S the standard deviation.

When the material reaches 90 percent of f_F, the material is considered to be failed. At this point, the stiffness and the stress at this element are reduced to zero.

Rubber-like Materials

In filled rubber-like materials, a softening of the instantaneous response can occur under repeated loading (Figure A 6.10-2).

This phenomenon, when combined with visco-elastic behavior, is called the Mullin's effect. The time-independent component can be described as having a dependence on the current stress and the previous maximum strain. The model implemented in MARC is a modification of one first proposed by Simo and assumes that the current value of stress is affected by the largest value of the instantaneous strain energy attained during the entire loading history.

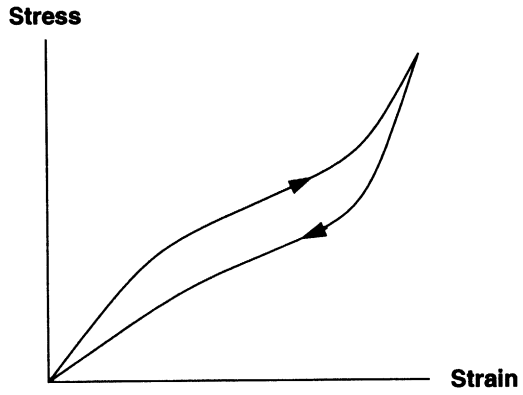


Figure A 6.10-2 Hysteresis Loop in Damaged Rubber Material

A scalar function representing the current amount of damage is now postulated to exist in the form (Figure A 6.10-3).

$$F(E_{MAX}) = \beta + (1-\beta)\frac{1-e^{-\sqrt{2E_{MAX}}/D}}{\sqrt{2E_{MAX}}/D} \qquad \text{(A 6.10-6)}$$

in which E_{MAX}, D, and β, respectively, denote the maximum previously attained value of the instantaneous strain energy, the damage rate, and the maximum damage factor. E_{MAX} is a nondecreasing variable whose magnitude is described by the Ogden strain energy function of Equation A 6.4-3. It is to be further noted that a value of β equal to zero eventually will produce complete failure of the material in the sense that no stiffness against additional deformation will be available.

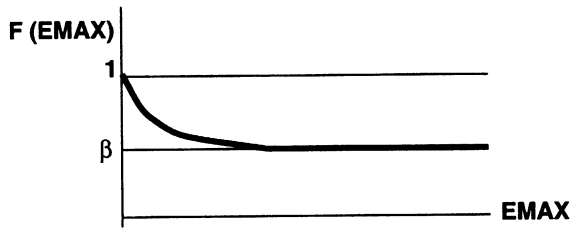


Figure A 6.10-3 Cumulative Damage Function

To capture the experimental observation that the stress softens under cyclic deformation, the instantaneous strain energy W^o of Equation A 6.4-3 is now modified to become

$$W = F(E_{MAX}) \bullet W^o \qquad \text{(A 6.10-7)}$$

The stress then follows as

$$S = F(E_{MAX}) \frac{\partial W^o}{\partial E} = F(E_{MAX}) \bullet S^o \qquad \text{(A 6.10-8)}$$

The above damage model is available for both deviatoric as well as dilatational behavior and is flagged by means of the OGDEN and DAMAGE model definition options. If, in addition, viscoelastic behavior is desired, the VISCEL OGDEN option can be included. Finally, a user subroutine UELDAM can be used to define damage functions different from Equation A 6.10-6.

A 6.11 NON-STRUCTURAL MATERIALS

In addition to stress analysis, the MARC program can be used for heat transfer, coupled thermo-electrical heating (Joule heating), hydrodynamic bearing, fluid/solid interaction, electrostatic, magnetostatic, electromagnetic and acoustic problems. Material properties associated with these analyses and the MARC program options that control these analyses are described below.

Heat Transfer Analysis

In heat transfer analysis, use model definition options ISOTROPIC, and ORTHOTROPIC to input values of thermal conductivity, specific heat, and mass density. If the latent heat effect is to be included in the analysis, the value of latent heat and associated solidus and liquidus temperatures must be entered through model definition option TEMPERATURE EFFECTS. Both the thermal conductivity and specific heat can be dependent on temperatures. The mass density must be constant throughout the analysis. In addition, the user subroutine ANKOND can be used for the input of anisotropic thermal conductivity.

Thermo-electrical Analysis

In addition to thermal conductivity, specific heat and mass density, the value of electric resistivity must be entered using the model definition options ISOTROPIC, and ORTHOTROPIC in a coupled thermo-electrical analysis. Input the variation of electric resistivity with temperature through the TEMPERATURE EFFECTS option.

Hydrodynamic Bearing Analysis

In a hydrodynamic bearing analysis, the model definition options ISOTROPIC and ORTHOTROPIC can be used for entering both the viscosity and specific mass. Define the temperature dependency of viscosity through the TEMPERATURE EFFECTS option.

Fluid/Solid Interaction Analysis

In a fluid/solid analysis, the density of a fluid is given in the first field of the PROPERTY option. The fluid is assumed to be nonviscous and incompressible.

Electrostatic Analysis

In an electrostatic analysis, the permittivity may be defined through either the ISOTROPIC or ORTHOTROPICS options. In addition, the user subroutine UEPS can be used for the input of anisotropic permittivity.

Magnetostatic Analysis

In a magnetostatic analysis, the permeability may be defined through either the ISOTROPIC or ORTHOTROPIC options. A nonlinear relationship may be entered via the B-H RELATION option. The user subroutine UMU can be used for the input of isotropic permeability.

Electromagnetic Analysis

In an electromagnetic analysis, the permittivity, permeability and conductivity can be defined using either the ISOTROPIC or ORTHOTROPIC options. An assumption made is that the permittivity is a constant (does no vary with time) in the analysis. A nonlinear permeability may be entered via the B-H RELATION option. The user subroutines UEPS, UMU, and USIGMA may be used for the input of anisotropic permittivity, permeability and conductivity respectively.

Acoustic Analysis

In an acoustic analysis, the bulk modulus and the relative density of the medium may be entered through the ISOTROPIC option.

References

1. "Current Recommended Constitutive Equations for Inelastic Design Analysis of FFTF Components." ORLN-TM-360Z, October 1971.
2. Mooney, M. J. *Appl Phys.*, Vol. II, p. 582, 1940.
3. Naghdi, P. M. "Stress-Strain Relations in Plasticity and Thermoplasticity." *In Plasticity, Proceedings of the Second Symposium on Naval Structural Mechanics*, edited by E. H. Lee and P. S. Symonds. Pergamon Press, 1960.
4. Prager, W. *Introduction to Mechanics to Continua*. New York: Dover Press, 1961.
5. "Proposed Modification to RDT Standard F9-5T Inelastic Analysis Guidelines." ORNL, October 1978.
6. Rivlin, R. S. *Phil Trans Roy Soc (A)*, Vol. 240, 459, 1948.
7. Rivlin, R. S. *Phil Trans Roy Soc (A)*, Vol. 241, 3-79, 1948.
8. Timoshenko, S. P., and J. N. Goodier. *Theory of Elasticity.* Third Ed. New York: McGraw-Hill, 1970.
9. Simo, J. C. "On a Fully Three-dimensional Finite Strain Viscoelastic Damage Model: Formulation and Computational Aspects," *Computer Methods in Applied Mechanics and Engineering*, Vol. 60, 1987, pp. 153 -173.

Chapter 7 ELEMENT LIBRARY

A 7.1 OVERVIEW OF THE ELEMENT LIBRARY

The MARC program includes an extensive element library. The element library allows the user to model various types of one-, two- and three-dimensional structures, such as plane stress and plane strain structures, axisymmetric structures, full three-dimensional solid structures, and shell-type structures. Table A 7.1-1 lists the element types in the library. Table A 7.1-2 shows the structural classification of elements. (See Volume B of the MARC User's Manual for a detailed description of each element, a reference to the use of each element, and recommendations concerning the selection of element types for analysis).

After you select an element type or a combination of several element types for your analysis with the ELEMENTS or SIZING parameter options, you must prepare the necessary input data for the element(s). In general, the data consist of element connectivity, thickness for two-dimensional beam, plate and shell elements, cross-section for three-dimensional beam elements, coordinates of nodal points, and face identifications for distributed loadings.

You can choose different element types to represent various parts of the structure in an analysis. If there is an incompatibility between the nodal degrees-of-freedom of the elements, you have to provide appropriate tying constraints to ensure the compatibility of the displacement field in the structure. The MARC program assists you by providing many standard tying constraint options, but you are responsible for the consistency of your analysis.

You can use almost all of the MARC elements for both linear and nonlinear analyses, with the following exceptions.

- No plasticity or cracking is allowed in the elastic beams, the shear-panel, the FOURIER, and the Herrmann (incompressible) elements.
- The updated Lagrange and finite strain plasticity features are not available for all elements.
- Only heat links and two-dimensional heat transfer elements can be used for hydrodynamic bearing analysis.
- Fourier analysis can be carried out only for a limited number of axisymmetric elements.

The program defines all continuum elements in the global coordinate system. Truss, beam, plate, and shell elements are defined in a local coordinate system and the resulting output must be interpreted accordingly. You should give special attention to the use of these elements if the material properties have preferred orientations. The ORIENTATION option is available to define the preferred directions.

All MARC elements are numerically integrated. Element quantities, such as stresses, strains, and temperatures, are calculated at each integration point of the element if you use the ALL POINTS option. This is the default in MARC. These quantities are computed only at the centroid of the element, if the CENTROID parameter is used.

Table A 7.1-1 Element Library

Element	Updated Lagrange Available	Code
2-node axisymmetric shell element	Yes	(1)
Axisymmetric triangular ring element	Yes	(2)
Two-dimensional (plane stress) 4-node, isoparametric quadrilateral	Yes	(3)
Curved quadrilateral thin-shell element	Yes	(4)
Beam column	No	(5)
Two-dimensional plane strain, constant stress triangle	Yes	(6)
8-node isoparametric three-dimensional hexahedron	Yes	(7)
3-node triangular arbitrary shell	Yes	(8)
Three-dimensional truss element	Yes	(9)
Axisymmetric quadrilateral element (isoparametric)	Yes	(10)
Plane strain quadrilateral element (isoparametric)	Yes	(11)
Gap and friction element	No	(12)
Open-section beam	No	(13)
Closed-section beam	Yes	(14)
Isoparametric 2-node axisymmetric shell	Yes	(15)
Isoparametric 2-node curved beam	Yes	(16)
Pipe-bend element	No	(17)
4-node isoparametric membrane	No	(18)
Generalized plane-strain quadrilateral	Yes	(19)
Axisymmetric torsional quadrilateral	Yes	(20)
Three-dimensional 20-node brick	Yes	(21)
Curved quadrilateral thick-shell element	Yes	(22)
Three-dimensional, 20-node rebar element	No	(23)
Curved quadrilateral shell element	Yes	(24)
Closed-section beam in three dimensions	Yes	(25)
Plane stress, 8-node distorted quadrilateral	Yes	(26)
Plane strain, 8-node distorted quadrilateral	Yes	(27)
Axisymmetric, 8-node distorted quadrilateral	Yes	(28)
Generalized plane strain distorted quadrilateral	Yes	(29)

Table A 7.1-1 Element Library (Continued)

Element	Updated Lagrange Available	Code
Membrane 8-node distorted quadrilateral	No	(30)
Elastic Curved Pipe (Elbow)/Straight Beam Element	No	(31)
Plane strain, 8-node distorted quadrilateral, for incompressible behavior	No	(32)
Axisymmetric 8-node distorted quadrilateral, for incompressible behavior	No	(33)
Generalized plane strain 8-node distorted quadrilateral for incompressible behavior	No	(34)
Three-dimensional 20-node brick for incompressible behavior	No	(35)
Heat transfer element (three-dimensional link)	No	(36)
Heat transfer element (arbitrary planar triangle)	No	(37)
Heat transfer element (arbitrary axisymmetric triangle)	No	(38)
Heat transfer element (planar bilinear quadrilateral)	No	(39)
Heat transfer element (axisymmetric bilinear quadrilateral	No	(40)
Heat transfer element (8-node planar biquadratic quadrilateral	No	(41)
Heat transfer element (8-node axisymmetric biquadratic quadrilateral)	No	(42)
Heat transfer element (three-dimensional 8-node brick)	No	(43)
Heat transfer element (three-dimensional 20-node brick)	No	(44)
Curved Timoshenko beam element in a plane	Yes	(45)
Plane strain rebar element	No	(46)
Generalized plane strain rebar element	No	(47)
Axisymmetric rebar element	No	(48)
Triangular thin shell element	Yes	(49)
Triangular heat transfer shell element	No	(50)
Cable Element	No	(51)
Elastic beam	Yes	(52)
Plane stress 8-node quadrilateral with reduced integration	Yes	(53)
Plane strain 8-node distorted quadrilateral with reduced integration	Yes	(54)
Axisymmetric 8-node distorted quadrilateral with reduced integration	Yes	(55)
Generalized plane strain 10-node distorted quadrilateral with reduced integration	Yes	(56)
Three-dimensional 20-node brick with reduced integration	Yes	(57)

Table A 7.1-1 Element Library (Continued)

Element	Updated Lagrange Available	Code
Plane strain 8-node distorted quadrilateral for incompressible behavior with reduced integration	No	(58)
Axisymmetric 8-node distorted quadrilateral for incompressible behavior with reduced integration	No	(59)
Generalized plane strain 10-node distorted quadrilateral incompressible behavior with reduced integration	No	(60)
Three-dimensional 20-node brick for incompressible behavior with reduced integration	No	(61)
Axisymmetric 8-node quadrilateral for arbitrary loading	No	(62)
Axisymmetric 8-node quadrilateral for arbitrary loading, and incompressible behavior	No	(63)
Isoparametric 3-node truss element	Yes	(64)
Heat transfer element 3-node truss (three-dimensional link)	No	(65)
8-node axisymmetric with twist, for incompressible behavior	No	(66)
8-node axisymmetric with twist	No	(67)
Elastic 4-node shear panel	No	(68)
Heat transfer element (8-node, planar biquadratic quadrilateral with reduced integration)	No	(69)
Heat transfer element (8-node, axisymmetric biquadratic quadrilateral with reduced integration)	No	(70)
Heat transfer element (three-dimensional 20-node brick with reduced integration)	No	(71)
Bilinear discrete Kirchhoff	Yes	(72)
Axisymmetric 8-node quadrilateral for arbitrary loading with reduced integration	No	(73)
Axisymmetric 8-node quadrilateral for arbitrary loading, for incompressible behavior with reduced integration	No	(74)
Bilinear thick shell	Yes	(75)
Thin-walled beam in three dimensions without warping	Yes	(76)
Thin-walled beam in three dimensions including warping	Yes	(77)
Thin-walled beam in three dimensions without warping	Yes	(78)
Thin-walled beam in three dimensions including warping	Yes	(79)
Plane strain 7-node quadrilateral for incompressible behavior	No	(80)
Generalized plane strain 7-node quadrilateral for incompressible behavior	No	(81)

Table A 7.1-1 Element Library (Continued)

Element	Updated Lagrange Available	Code
Axisymmetric 5-node quadrilateral for incompressible behavior	No	(82)
Axisymmetric 5-node quadrilateral with twist for incompressible behavior	No	(83)
8-node three-dimensional hexahedron for incompressible behavior	No	(84)
Heat transfer element (bilateral shell)	No	(85)
Heat transfer element (curved quadrilateral shell)	No	(86)
Heat transfer element (curved axisymmetric shell)	No	(87)
Heat transfer element (linear axisymmetric shell)	No	(88)
Thick curved axisymmetric shell	Yes	(89)
Thick curved axisymmetric shell for arbitrary loading	No	(90)
Linear plane strain semi-infinite element	No	(91)
Linear axisymmetric semi-infinite element	No	(92)
Quadratic plane strain semi-infinite element	No	(93)
Quadratic axisymmetric semi-infinite element	No	(94)
Quadrilateral axisymmetric element with bending	No	(95)
8-node distorted quadrilateral axisymmetric element with bending	No	(96)
Double gap and friction element for use with Element Types 95 or 96	No	(97)
Elastic beam with transverse shear	Yes	(98)
Heat transfer link	No	(99)
Heat transfer link	No	(100)
6-node planar semi-infinite heat transfer	No	(101)
6-node axisymmetric semi-infinite heat transfer	No	(102)
9-node planar semi-infinite heat transfer	No	(103)
9-node axisymmetric semi-infinite heat transfer	No	(104)
12-node 3-D semi-infinite heat transfer	No	(105)
27-node 3-D semi-infinite heat transfer	No	(106)
12-node 3-D semi-infinite	No	(107)
27-node 3-D semi-infinite	No	(108)
8-node 3-D magnetostatic	No	(109)
12-node 3-D semi-infinite magnetostatic	No	(110)

Table A 7.1-1 Element Library (Continued)

Element	Updated Lagrange Available	Code
4-Node Quadrilateral Planar Electromagnetic	No	(111)
4-Node Quadrilateral Axisymmetric Electromagnetic	No	(112)
8-Node Quadrilateral Three-Dimensional Electromagnetic	No	(113)
4-Node Quadrilateral Plane Stress, Reduced Integration with Hourglass Control	Yes	(114)
4-Node Quadrilateral Plane Strain, Reduced Integration with Hourglass Control	Yes	(115)
4-Node Quadrilateral Axisymmetric, Reduced Integration with Hourglass Control	Yes	(116)
8-Node Three-Dimensional Brick, Reduced Integration with Hourglass Control	Yes	(117)
Incompressible 4+1-Node, Quadrilateral, Plane Strain, Reduced Integration with Hourglass Control	No	(118)
Incompressible 4+1-Node, Quadrilateral, Axisymmetric, Reduced Integration with Hourglass Control	No	(119)
Incompressible 8+1-Node, Three-Dimensional Brick, Reduced Integration with Hourglass Control	No	(120)
4-Node, Heat Transfer Planar, Reduced Integration with Hourglass Control	No	(121)
4-Node, Heat Transfer Axisymmetric, Reduced Integration with Hourglass Control	No	(122)
8-Node, Heat Transfer Three-Dimensional Brick, Reduced Integration with Hourglass Control	No	(123)
6-Node, Plane Stress Triangle	Yes	(124)
6-Node, Plane Strain Triangle	Yes	(125)
6-Node, Axisymmetric Triangle	Yes	(126)
10-Node, Tetrahedron	Yes	(127)
Incompressible, 6-Node Triangle	No	(128)
Incompressible, 6-Node Triangle	No	(129)
Incompressible, 10-Node Tetrahedral	No	(130)
6-Node, Heat Transfer Planar	No	(131)
6-Node, Heat Transfer Plane Axisymmetric	No	(132)
10-Node, Heat Transfer Tetrahedral	No	(133)

Table A 7.1-2 Structural Classification of Elements

Element Structural Type	Element Number	Function	Remarks
Three-dimensional truss	9	Linear	2-node straight
	12	Linear	4-node straight gap and friction
	64	Quadratic	3-node curved
	97	Linear	4-node straight double gap and friction
	51	Analytic	2-node cable element
Two-dimensional beam column	5	Linear/cubic	2-node straight
	16		2-node curved
	45		3-node curved Timoshenko theory
Three-dimensional beam column	13	Cubic	2-node curved open section
	14	Linear/cubic	2-node straight closed section
	25	Cubic	2-node straight closed section
	31	Analytic	2-node elastic
	52	Linear/cubic	2-node straight elastic
	76	Linear/cubic	2 + 1-node straight closed section; use with Element 72
	77	Linear/cubic	2 + 1-node straight open section; use with Element 72
	78	Linear/cubic	2-node straight closed section; use with Element 75
	79	Linear/cubic	2-node straight open section
	98	Linear	2-node straight Timoshenko theory
Axisymmetric shell	1	Linear/cubic	2-node straight
	15	Cubic	2-node curved
	89	Quadratic	3-node curved thick shell theory
	90	Quadratic	3-node curved with arbitrary loading; thick shell theory

Table A 7.1-2 Structural Classification of Elements (Continued)

Element Structural Type	Element Number	Function	Remarks
Plane stress	3	Linear	4-node quadrilateral
	26	Quadratic	8-node quadrilateral
	53	Quadratic	8-node reduced integration quadrilateral
	114	Linear/ Assumed strain	4-node quadrilateral, reduced integration, with hourglass control
	124	Quadratic	6-node triangle
Plain strain	6	Linear	3-node triangle
	11	Linear	4-node quadrilateral
	27	Quadratic	8-node quadrilateral
	54	Quadratic	8-node reduced integration quadrilateral
	91	Linear/special	6-node semi-infinite
	93	Quadratic/ special	9-node semi-infinite
	115	Linear/ Assumed strain	4-node quadrilateral, reduced integration with hourglass control
	125	Quadratic	6-node triangle
Generalized plane strain	19	Linear	4 + 2-node quadrilateral Node 5-6; transverse strain
	29	Quadratic	8 + 2-node quadratic Node 9-10; transverse strain
	56	Quadratic	8 + 2-node quadrilateral reduced integration

Table A 7.1-2 Structural Classification of Elements (Continued)

Element Structural Type	Element Number	Function	Remarks
Axisymmetric solid	2	Linear	3-node triangle
	10	Linear	4-node quadrilateral
	28	Quadratic	8-node quadrilateral
	20	Linear	4-node quadrilateral with twist
	55	Quadratic	8-node quadrilateral reduced integration
	62	Quadratic	8-node quadrilateral with arbitrary loading
	67	Quadratic	8-node quadrilateral with twist
	73	Quadratic	8-node quadrilateral with reduced integration and arbitrary loading
	92	Linear/special	6-node semi-infinite
	94	Quadratic/ special	9-node semi-infinite
	95	Linear	4-node quadrilateral with bending
	96	Quadratic	8-node quadrilateral with bending
	116	Linear/ Assumed strain	4-node quadrilateral, reduced integration with hourglass control
	126	Quadratic	6-node triangle
Membrane three-dimensional	18	Linear	4-node quadrilateral
	30	Quadratic	8-node quadrilateral
Doubly-curved thin shell	4	Cubic	4-node curved quadrilateral
	8	Fractional cubic	3-node curved triangle
	24	Cubic patch	4 + 4-node curved quadrilateral
	49	Linear	3 + 3-node curved triangular discrete Kirchhoff
	72	Linear	4 + 4-node twisted quadrilateral discrete Kirchhoff
Doubly-curved thick shell	22	Quadratic	8-node curved quadrilateral with reduced integration
	75	Linear	4-node twisted quadrilateral

Table A 7.1-2 Structural Classification of Elements (Continued)

Element Structural Type	Element Number	Function	Remarks
Three-dimensional solid	7	Linear	8-node hexahedron
	21	Quadratic	20-node hexahedron
	57	Quadratic	20-node hexahedron reduced integration
	107	Linear/special	12-node semi-infinite
	108	Quadratic/ special	27-node semi-infinite
	117	Linear/ Assumed strain	8-node hexahedron, reduced integration with hourglass control
	127	Quadratic	10-node tetrahedron
Incompressible plane strain	32	Quadratic	8-node quadrilateral
	58	Quadratic	8-node quadrilateral with reduced integration
	80	Linear	4 + 1-node quadrilateral with Node 5: pressure
	118	Linear/ Assumed strain	4 + 1-node quadrilateral with Node 5: pressure, reduced integration with hourglass control
	128	Quadratic	6-node triangle
Incompressible generalized plan strain	34	Quadratic	8- and 2-node; Node 9-10 transverse strain
	60	Quadratic	8 + 2-node quadrilateral with reduced integration
	81	Linear	4 + 3-node quadrilateral, Node 5-6: transverse strain; Node 7: pressure

Table A 7.1-2 Structural Classification of Elements (Continued)

Element Structural Type	Element Number	Function	Remarks
Incompressible axisymmetric	33	Quadratic	8-node quadrilateral
	59	Quadratic	8-node quadrilateral with reduced integration
	63	Quadratic	8-node quadrilateral with arbitrary loading
	66	Quadratic	8-node quadrilateral with twist
	74	Quadratic	8-node quadrilateral with arbitrary loading and reduced integration
	82	Linear	4 + 1-node quadrilateral Node 5: pressure
	83	Linear	4 + 1-node quadrilateral with twist Node 5: pressure
	119	Linear/ Assumed strain	4 + 1 node quadrilateral; Node 5: pressure; reduced integration with hourglass control
	129	Quadratic	6-node triangle
Incompressible three-dimensional solid	35	Quadratic	20-node hexahedron
	61	Quadratic	20-node hexahedron with reduced integration
	84	Linear	8 + 1-node hexahedron; Node 9: pressure
	120	Linear/ Assumed strain	8 + 1-node hexahedron; Node 9: pressure; reduced integration with hourglass control
	130	Quadratic	10-node tetrahedron
Pipe bend	17	Cubic	2-nodes in-section; 1-node out-of-section
	31	Special	2- node elastic
Rebar Elements	23	Quadratic	20-node hexahedron
	46	Quadratic	8-node quadrilateral plane strain
	47	Quadratic	8 + 2-node quadrilateral generalized plane strain
	48	Quadratic	8-node quadrilateral axisymmetric
Three-dimensional shear panel	68	Linear	4-node quadrilateral
Head conduction 3D link	36	Linear	2-node straight
	65	Quadratic	3-node curved

Table A 7.1-2 Structural Classification of Elements (Continued)

Element Structural Type	Element Number	Function	Remarks
Heat conduction planar	37	Linear	3-node triangle
	39	Linear	4-node quadrilateral
	41	Quadratic	8-node quadrilateral
	69	Quadratic	8-node quadrilateral reduced integration
	101	Linear/special	6-node semi-infinite
	103	Linear/special	9-node semi-infinite
	121	Linear	4-node quadrilateral, reduced integration with hourglass control
	131	Quadratic	6-node triangular
Heat conduction axisymmetric	38	Linear	3-node triangle
	40	Linear	4-node quadrilateral
	42	Quadratic	8-node quadrilateral
	70	Quadratic	8-node quadrilateral with reduced integration
	102	Linear/special	6-node semi-infinite
	104	Linear/special	9-node semi-infinite
	122	Linear	4-node quadrilateral, reduced integration with hourglass control
	132	Quadratic	6-node triangular
Heat conduction solids	43	Linear	8-node hexahedron
	44	Quadratic	20-node hexahedron
	71	Quadratic	20-node hexahedron with reduced integration
	105	Linear/special	12-node semi-infinite
	106	Quadratic/ special	27-node semi-infinite
	123	Linear	8-node hexahedron; reduced integration with hourglass control
	133	Quadratic	10-node tetrahedron

Table A 7.1-2 Structural Classification of Elements (Continued)

Element Structural Type	Element Number	Function	Remarks
Heat conduction shell	50	Linear	3-node triangular
	85	Linear	4-node quadrilateral
	86	Quadratic	8-node quadrilateral
Heat conduction axisymmetric shell	87	Quadratic	3-node curved
	88	Linear	2-node straight
Magnetostatic three-dimensional solids	109	Linear	8-node hexahedron
	110	Linear/special	12-node semi-infinite
Electromagnetic planar	111	Linear	4-node quadrilateral
Electromagnetic axisymmetric	112	Linear	4-node quadrilateral
Electromagnetic solid	113	Linear	8-node hexahedron

Distributed loads may be applied along element edges, over element surfaces, or in the volume of the element. MARC automatically evaluates consistent nodal forces through numerical integration. (See Volume B of the MARC User's Manual for details on this process). Concentrated forces must be applied at the nodal points.

All plate and shell elements may be used in a composite analysis. You can have a variable thickness shell, and control the thickness and material property and orientation for each layer. For the thick shell elements (types 22, 45, or 75), the interlaminar shear may also be calculated.

Five concepts differentiate the various elements types. These concepts are summarized in Tables 7.1-1 and 7.1-2 and are listed below.

1. The type of geometric domain that the element is modeling. These geometric domains are:

   Truss
   Membrane
   Beam
   Plate
   Shell
   Plane stress
   Plane strain
   Generalized plane strain
   Axisymmetric
   Three-dimensional solid
   Special

2. The type of interpolation (shape) functions used in the element. These functions are:

   Linear
   Quadratic
   Cubic
   Hermitian
   Special

   The interpolation function is used to describe the displacement at an arbitrary point in the body.

$$u_i(x) = N_i(x)\,\bar{u}_i \qquad \text{(A 7.1-1)}$$

   where u (x) is the displacement at x, N are the interpolation (shape) functions, and $\bar{u}$ are the generalized nodal displacements.

   Engineering strain is

$$\varepsilon_{ij}(x) = 1/2\left(\frac{\partial u_i(x)}{\partial x_j} + \frac{\partial u_j(x)}{\partial x_i}\right) \qquad \text{(A 7.1-2)}$$

   Therefore, the computational evaluation is

$$e_{ij}(x) = 1/2\left(\frac{\partial N_i \bar{u}_i}{\partial x_j} + \frac{\partial N_j \bar{u}_j}{\partial x_i}\right) = \beta_{ij}\bar{u}_i \qquad \text{(A 7.1-3)}$$

   Hence,

$$\beta_{ij} = \frac{\partial N_i}{\partial x_j} \qquad \text{(A 7.1-4)}$$

3. The number of nodes in a particular element.
4. The number of degrees-of-freedom associated with each node, and the type of degrees-of-freedom.
5. The integration method used to evaluate the stiffness matrix. MARC contains elements which use full integration and reduced integration.

A 7.2 TRUSS ELEMENTS

MARC contains 2- and 3-node isoparametric truss elements that may be used in three dimensions. These elements have only displacement degrees-of-freedom.

Since truss elements have no shear resistance, ensure that there are no rigid body modes in the system.

A 7.3 MEMBRANE ELEMENTS

MARC contains 4- and 8-node isoparametric membrane elements that may be used in three dimensions. These elements have only displacement degrees-of-freedom.

Since membrane elements have no bending resistance, ensure that there are no rigid body modes in the system.

Membrane elements are often used in conjunction with beam or truss elements.

A 7.4 CONTINUUM ELEMENTS

MARC contains continuum elements that may be used to model plane stress, plane strain, generalized plane strain, axisymmetric and three-dimensional solids. These elements have only displacement degrees-of-freedom. As a result, solid elements are not efficient for modeling thin structures dominated by bending. Either beam or shell elements should be used in these cases.

The solid elements that are available in MARC have either linear or quadratic interpolation functions.

They include

- 4-, 6-, and 8-node plane stress elements
- 3-, 4-, 6-, and 8-node plane strain elements
- 6 (4 plus 2)- and 10 (8 plus 2)-node generalized plane strain elements
- 3-, 4-, 6-, and 8-node axisymmetric ring elements
- 8-, 10-, and 20-node brick elements

In general, the elements in MARC use a full-integration procedure. Some elements use reduced integration. The lower-order reduced integration elements include an hourglass stabilization procedure to eliminate the singular modes.

Continuum elements are widely used for thermal stress analysis. For each of these elements, there is a corresponding element available for heat transfer analysis in the MARC program. As a result, you can use the same mesh for the heat transfer and thermal stress analyses.

The program has no singular element for fracture mechanics analysis. However, the simulation of stress singularities can be accomplished by moving the midside nodes of 8-node quadrilateral and 20-node brick elements to quarter-point locations near the crack tip. Many fracture mechanics analyses have used this quarter-point technique successfully.

The 4- and 8-node quadrilateral elements can be degenerated into triangles, and the 8- and 20-node solid brick elements can be degenerated into wedges and tetrahedra by collapsing the appropriate corner and midside nodes. The number of nodes per element is not reduced for degenerated elements. The same node number is used repeatedly for collapsed sides or faces. When degenerating incompressible elements, exercise caution to ensure that a proper number of Lagrange multipliers remain. Users are advised to use the higher-order triangular or tetrahedron elements wherever possible, as opposed to using collapsed quadrilaterals and hexahedra.

A 7.5 BEAM ELEMENTS

MARC's beam elements are 2- and 3-node, two- and three-dimensional elements that can model straight and curved-beam structures and framed structures and can serve as stiffeners in plate and shell structures. A straight beam of a circular cross-section can be used for modeling the straight portion of piping systems. Translational and rotational degrees-of-freedom are included in beam elements. The cross-section of the beam may be standard rectangular, circular sections, or arbitrarily closed or open sections. The BEAM SECT option is used to define an arbitrary closed or open cross-section.

The beam elements are numerically integrated along their axial direction using Gaussian integration. The stress strain law is integrated through the cross-section using Simpson's rule. Stresses and strains are evaluated at each integration point through the thickness. This allows an accurate calculation if nonlinear material behavior is present. In elastic beam elements, only the total axial force and moments are computed at the integration points.

A 7.6 PLATE ELEMENTS

The linear shell elements (Types 49, 72 or 75) or the quadratic element (Type 22) may be used effectively to model plates and have the advantage that tying is unnecessary. For element type 49, one may indicate on the GEOMERTY option that the flat plate formulation is to be used. This reduces computational costs. To further reduce computational costs for linear elastic plate analysis, the number of points through the thickness may be reduced to one by use of the SHELL SECT option.

A 7.7 SHELL ELEMENTS

MARC contains three isoparametric, doubly curved, thin shell elements: 3-, 4-, and 8-node elements (Types 4, 8, and 24, respectively) based on Koiter-Sanders shell theory. They fulfill continuity requirements and exactly represent rigid-body modes. The program defines a mesh of these elements with respect to a surface curvilinear coordinate system. You can use the FXORD model definition option to generate the nodal coordinates. Tying constraints must be used at shell intersections.

The 6-node bilinear constrained shell element (Type 49), the 8-node bilinear constrained shell element (Type 72), the 4-node bilinear thick shell element (Type 75), and the 8-node quadradic thick shell element (Type 22) are recently developed shell elements that are useful in the analysis of both plate and shell structures. The global coordinate system defines the nodal degrees-of-freedom of these elements. These elements are convenient for modeling intersecting shell structures since tying constraints at the shell intersections are not needed.

The MARC program contains three axisymmetric shell elements: 2-node straight, 2-node curved, and 3-node curved. You can use these elements to model axisymmetric shells; combined with axisymmetric ring elements, they may be used to simulate the thin and thick portions of the structure. The program provides standard tying constraints for the transition between shell and axisymmetric ring elements.

A 7.8 HEAT TRANSFER ELEMENTS

The heat transfer elements in MARC consist of the following:

- 2- and 3-node three-dimensional links
- 3-, 4-, 6-, and 8-node planar and axisymmetric elements
- 6- and 9-node planar and axisymmetric semi-infinite elements
- 8-, 10-, and 20-node solid elements
- 12- and 27-node semi-infinite brick elements
- 2-, 3-, 4-, 6- and 8-node shell elements.

For each heat transfer element, there is at least one corresponding stress element, enabling you to use the same mesh for both the heat transfer and thermal stress analyses. Heat transfer elements are also employed to analyze coupled thermo-electrical (Joule heating) problems.

In heat transfer elements, temperature is the only nodal degree-of-freedom. In heat transfer shell elements, two or three temperatures through the thickness are used as nodal degrees-of-freedom. In Joule heating, the voltage and temperature are the nodal degrees-of-freedom.

Acoustic Analysis

Heat transfer elements are also used to model the compressible media in acoustic analysis. In this case, the pressure is the single nodal degree-of-freedom.

Electrostatic Analysis

Heat transfer elements are also used for electrostatic analysis. The scalar potential is the degree-of-freedom.

Fluid/Solid Interaction

Heat transfer elements are used to model the inviscid, incompressible fluid/solid interaction problems. The hydrostatic pressure is the degree-of-freedom.

Hydrodynamic Bearing Analysis

The three-dimensional heat transfer links and planar elements are used to model the lubricant film. As no variation occurs through the thickness of the film, two-dimensional problems are reduced to one-dimensional, and three-dimensional problems are reduced to two-dimensional. The pressure is the degree-of-freedom.

Magnetostatic Analysis

For two-dimensional problems, a scalar potential may be used; hence, the heat transfer planar and axisymmetric elements are employed. The single degree-of-freedom is the potential. For three-dimensional analyses, magnetostatic elements are available. In such cases, the degrees-of-freedom are the full (3) vector potential.

Electromagnetic Analysis

In electromagnetic problems, a vector potential, augmented with a scalar potential, is used. There are lower-order elements available for these analyses.

Soil Analysis

There are three types of soil/pore pressure analysis. If a pore pressure analysis only is performed, "heat transfer" elements (41, 42, or 44) are used. If an uncoupled soil analysis is performed, the standard elements (21, 27, or 28) are used. If a coupled analysis is performed, the Herrmann elements (32, 33, or 35) are used. In this case, the last degree-of-freedom is the pore pressure.

A 7.9 SPECIAL ELEMENTS

The MARC program contains a gap-and-friction element, a pipe-bend element, a shear-panel element, and several rebar elements. All of these special elements are unique features in the program.

The gap-and-friction element (12, 97) is based on the imposition of gap closure constraint and frictional-stick or frictional-slip through Lagrange multipliers. This element provides frictional and gapping connection between any two nodes of a structure; it can be used in several variations, depending on the application. In the default formulation, the element simulates a gap in a fixed direction, such that a body does not penetrate a given flat surface. Using the optional formulation, you can constrain the true distance between two end-points of the gap to be greater than some arbitrary distance. This ability is useful for an analysis in which a body does not penetrate a given two-dimensional circular or three-dimensional spherical surface. Finally, you can update the gap direction and closure distance during analysis for the modeling of sliding along a curved surface.

The pipe-bend (3-node) element 17 is designed for linear and nonlinear analysis of piping systems. It is a modified axisymmetric shell element for modeling the bends in a piping system. The element has a beam mode superposed on the axisymmetric shell modes so that ovalization of the cross-section is admitted. The twisting of a pipe-bend section is ignored because the beam has no flexibility in torsion. Built-in tying constraints are used extensively for coupling the pipe-bend sections to each other and to straight beam elements.

The 3 pipe-bend (2-node) element 31 is designed for linear analysis only. The stiffness matrix is based upon the analytic elastic solution of a curved pipe.

The shear panel (element 68) is a 4-node, elastic element of arbitrary shape. It is an idealized model of an elastic sheet. This element will only provide shearing resistance. It must be used with beam stiffeners to ensure any normal or bending resistance. The shear-panel element is restricted to linear material and small displacement analysis.

The cable element (51) is an element which exactly represents the catenary behavior of a cable. It is an elastic element only.

The rebar elements (46, 47, and 48) are hollow elements in which you can place single-direction strain members (reinforcing cords or rods). A rebar element is used in conjunction with another solid element (filler) to represent a reinforced material such as reinforced concrete. The reinforcing members and the filler are accurately represented by the superposition of rebar and solid elements. You can use separate stress-strain relations, such as concrete in the filler and elastic-plastic behavior in the rebars.

A 7.10 INCOMPRESSIBLE ELEMENTS

Incompressible and nearly incompressible materials are modeled by using a special group of elements in the program. These elements, based on a modified Herrmann variational principle, are assumed to be elastic and capable of handling large displacement effects as well as creep and thermal strains. You may use them in conjunction with elastic-plastic elements in the same mesh. The incompressibility constraint is imposed by using Lagrange multipliers. There are two types of incompressible elements:

The lower (linear) order elements which have a single additional node that contains the Lagrange multiplier.

The higher (quadratic) order elements which have Lagrange multipliers at each corner node.

Rigid-Plastic Flow

In rigid-plastic flow analysis, with effectively no elasticity, incompressible elements must be used. For such analyses, the user has a choice of using either the elements with Lagrange multipliers discussed above, or standard displacement continuum elements. In the latter case, a penalty function will be used to satisfy the incompressible constraint.

A 7.11 CONSTANT DILATATION ELEMENTS

You can choose an integration scheme option, which makes the dilatational strain constant throughout the element. This can be accomplished by setting the second field of the GEOMETRY option to one. Constant dilatational elements are recommended for use in approximately incompressible, inelastic analysis, such as large strain plasticity, because conventional elements result in computational results too stiff for nearly incompressible behavior. This option is only available for elements of lower order (Types 7, 10, 11, 19 and 20). For the lower-order reduced integration elements (114 to 117) with hourglass control, as only one integration point is used, these elements do not lock, and, effectively, a constant dilatation formulation is used.

A 7.12 REDUCED INTEGRATION ELEMENTS

The program uses a reduced integration scheme to evaluate the stiffness matrix or the thermal conductivity matrix for a number of isoparametric elements. The mass matrix and the specific heat matrix of the element are always fully integrated.

For lower-order, 4-node quadrilateral elements, the number of numerical integration points is reduced from 4 to 1; in 8-node solid elements, the number of numerical integration points is reduced from 8 to 1. For 8-node quadrilateral elements, the number of numerical integration points is reduced from 9 to 4; for 20-node solid elements, the number of numerical integration points is reduced from 27 to 8.

The energy due to the higher-order deformation mode(s) associated with high-order elements is not included in the analysis. Reduced integration elements and fully integrated elements can be used together in an analysis. Reduced integration elements are more economical than fully integrated elements and they can improve analysis accuracy. However, with near singularities and in regions of high-strain gradients, the use of reduced integration elements can lead to oscillations in the displacements and produce inaccurate results. Using reduced integration elements results in zero-energy modes, or breathing nodes. In the lower-order elements, an additional stabilization stiffness is added which eliminates these so-called "hourglass" modes.

A 7.13 FOURIER ELEMENTS

A special class of elements exists which allows the analysis of axisymmetric structures with nonaxisymmetric loads. The geometry and material properties of these structures do not change in the circumferential direction, and the displacement may be represented by a FOURIER series. This representation allows a three-dimensional problem to be decoupled into a series of two-dimensional problems. Both solid and shell FOURIER elements exist in MARC. These elements may only be used for linear elastic analyses.

A 7.14 SEMI-INFINITE ELEMENTS

This group of elements may be used to model unbounded domains. In the semi-infinite direction, the interpolation functions are exponential, such that the function (displacement) is zero at the far domain. The rate of decay of the function is dependent upon the location of the midside nodes. The interpolation function in the non semi-infinite directions are either linear or quadratic.

These elements can be used for static plane strain, axisymmetric, or 3D solid analysis. They may also be used for heat transfer, electrostatic and magnetostatic analyses.

A 7.15 ASSUMED STRAIN FORMULATION

Conventional isoparametric four-node plane stress and plane strain, and eight-node brick elements behave poorly in bending. The reason is that these elements do not capture a linear variation in shear strain which is present in bending when a single element is used in the bending direction. For three elements – 3, 7, and 11 – the interpolation functions have been modified such that shear strain variation can be better represented. For the lower-order reduced integration elements (114 to 123), an assumed strain formulation written with respect to the natural coordinates is used. For elastic isotropic bending problems, this will allow the exact displacements to be obtained with only a single element through the thickness. This option is invoked by setting the third field of the GEOMETRY option to one.

A 7.16 FOLLOW FORCE STIFFNESS CONTRIBUTION

When activating the FOLLOW FORCE option, the distributed loads are calculated based upon the current deformed configuration. It is possible to activate an additional contribution which goes into the stiffness matrix. This improves the convergence. This capability is available for element types 3, 7, 10, 11, 18, 72, 75, 80, 81, 82, 83, 84, 114, 115, 116, 117, 118, and 119.

A 7.17 EXPLICIT DYNAMICS

The explicit dynamics formulation IDYN=5 model is restricted to the following elements:

2, 3, 5, 6, 7, 9, 11, 18, 19, 20, 52, 64, 75, 98, 114, 115, 116, 117, 118, 119, and 120.

When using this formulation, the mass matrix is defined semi-analytically; i.e., no numerical integration is performed. In addition, a quick method is used to calculate the stability limit associated with each element. For these reasons, this capability has been limited to the elements mentioned above.

A 7.18 ADAPTIVE MESH REFINEMENT

The MARC program has a capability to perform adaptive mesh refinement to improve the accuracy of the solution. This capability is invoked by using the ADAPTIVE parameter and model definition option. The adaptive meshing is available for the following 2D and 3D elements:

2, 3, 6, 7, 10, 11, 18, 19, 20, 37, 38, 39, 40 43, 75, 80, 81, 82, 83, 84, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123

References

1. Herrmann, L.R. "Elasticity Equations for Incompressible and Nearly Incompressible Materials by a Variational Theorem." *AIAA Journal*, 3 (10): 1896-2000, 1965.

2. Nagtegaal, J.C., Parks, D.M., and Rice, J.R. "On Numerically Accurate Finite Element Solutions in the Fully Plastic Range." *Computer Methods in Applied Mechanics and Engineering*, 4, 153-178, 1974.

3. Zienkiewicz, O.C. *The Finite Element Method.* 3rd ed., London: McGraw-Hill, 1977.

Chapter 8 BOUNDARY CONDITIONS

The MARC program is based on the stiffness method and deals primarily with force-displacement relations. In a linear elastic system, force and displacement are related through the constant stiffness of the system; the governing equation of such a system can be expressed as

$$Ku = f \qquad \text{(A 8.0-1)}$$

where K is the stiffness matrix and u and f are nodal displacement and nodal force vectors, respectively. Equation 8.0-1 can be solved either for unknown displacements subjected to prescribed forces or for unknown forces (reactions) subjected to prescribed displacements. In general, the system is subjected to mixed (prescribed displacement and force) boundary conditions, and the program computes both the unknown displacements and reactions. Obviously, at any nodal point, the nodal forces and nodal displacements cannot be simultaneously prescribed as boundary conditions for the same degree-of-freedom.

NOTE

The user must prescribe at least a minimum number of boundary conditions to insure that rigid body motion does not occur.

The prescribed force boundary conditions are often referred to as loads and the prescribed displacement boundary conditions as boundary conditions.

NOTE

Boundary conditions can be prescribed in either the global or a local coordinate system. A nodal transformation between the global and the local coordinate systems must be carried out if the boundary condition will be prescribed in a local system.

In a nonlinear stress analysis problem, the program carries out the analysis incrementally and expresses the governing equation in an incremental form in terms of the incremental displacement vector du and the incremental force vector df.

$$Kdu = df \qquad \text{(A 8.0-2)}$$

Consequently, you must also define both the loads and the prescribed nodal displacements incrementally.

In addition to the prescribed displacement boundary conditions, constraint relations may exist among the nodal displacements. For example, the first degree-of-freedom of node i is equal to that of node j at all times. The expression of this constraint relation is

$$u_i = u_j \qquad \text{(A 8.0-3)}$$

Generally, a homogenous linear constraint equation can be expressed as

$$u_t = a_1u_1 + a_2u_2 + ... + a_nu_n \qquad \text{(A 8.0-4)}$$

where u represents the degrees-of-freedom to be constrained, $u_1,...,u_n$ are other retained degrees-of-freedom in the structure, and $a_1,...,a_n$ are constraints.

You can enter constraints through either the TYING option or the SERVO LINKS option.

You can use linear/nonlinear springs and foundations to provide special support to the structure and the gap and friction element (or CONTACT option) to simulate the contact problem.

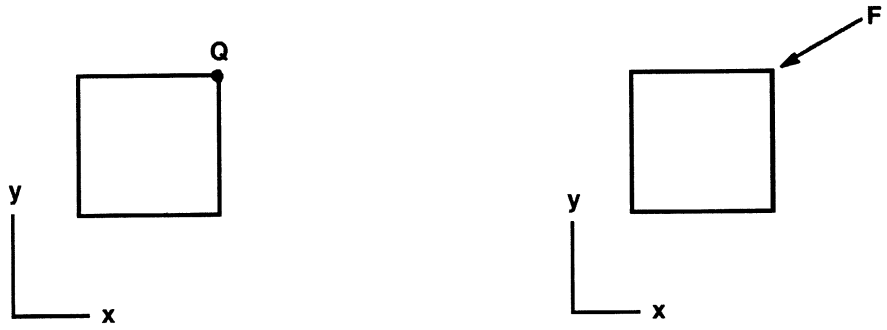
A 8.1 LOADING

Different types of analyses require different kinds of loading. For example, the loads in stress analysis are forces; those in heat transfer analysis are heat fluxes. Force is a vector quantity defined by magnitude and direction; heat flux is a scalar quantity defined by magnitude only. Loading can be time invariant (constant value) or time dependent.

Loading Types

You can categorize a particular type of load as either a point (concentrated) load or surface/volumetric (distributed) load, depending on application conditions. The spatial distribution of the load can be uniform or nonuniform. Special loading types also exist in various analyses. For example, centrifugal loading exists in stress analysis, and convection and radiation exist in heat transfer analysis. You can add point loads directly to the nodal force vector, but equivalent nodal forces first must be calculated by MARC from distributed loads and then added to the nodal force vector. These distinguishing features are described below.

A point (or nodal) load of either a vector (force, moment) or a scalar (heat flux) quantity is a concentrated load that is applied directly to a nodal point (see Figure A 8.1-1). In a global or local coordinate system, a force must be defined in terms of vector components (see Figure A 8.1-2). If the force vector is defined in a local coordinate system, then a global-to-local coordinate transformation matrix must be defined for the nodal point (see Figure A 8.1-3 and Figure A 8.1-4). For axisymmetric elements, the magnitude of the point load must correspond to the ring load integrated around the circumferences.



(a) Point Heat Flux Q (Scalar)

(b) Point Force F (Vector)

Figure A 8.1-1 Schematic of a Point Load

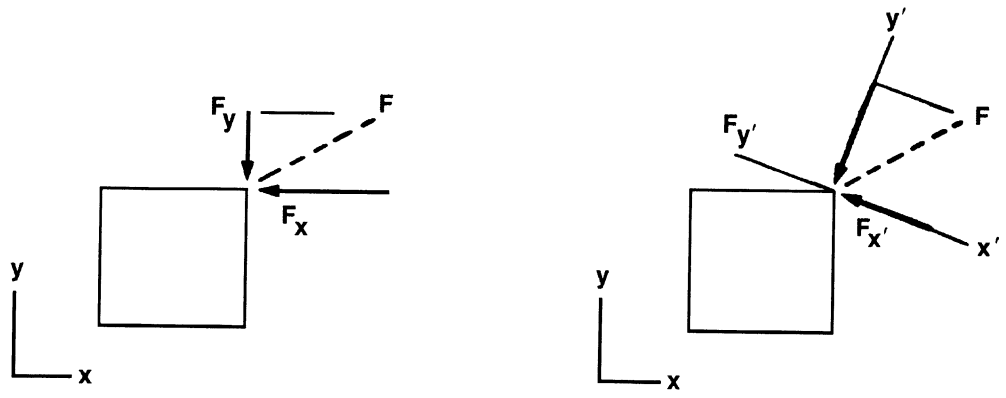


Figure A 8.1-2 Force Components

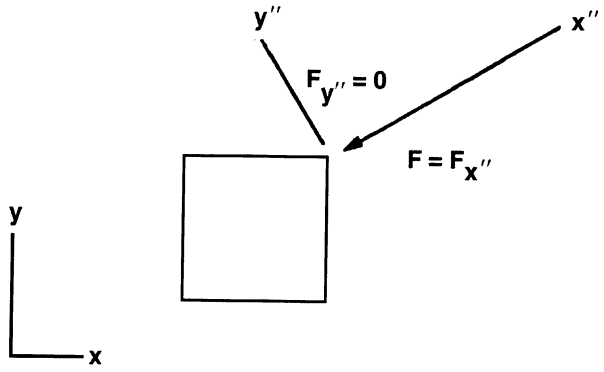
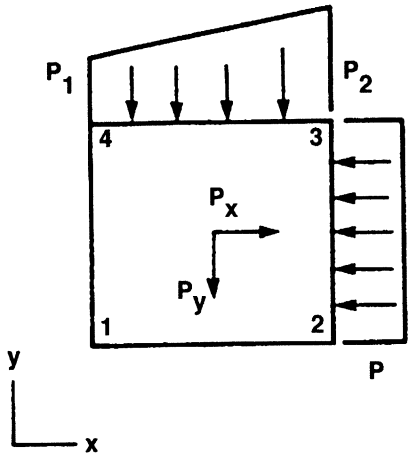


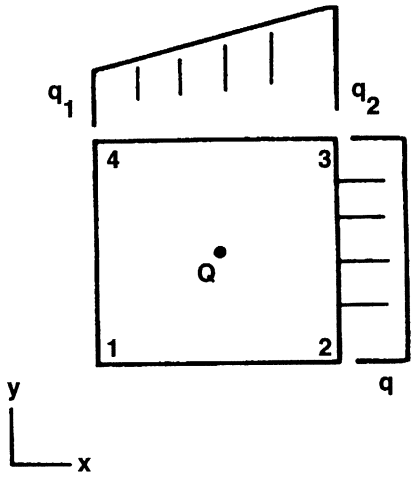
Figure A 8.1-3 Special Selection of Local (x", y") Coordinate System Force Components: F_x, $F_y = 0$

Surface/volumetric loads, such as pressure, distributed heat flux, and body force, are distributed loads that are applied to the surface (volume) of various elements (see Figure A 8.1-4). A surface/volumetric load is characterized by the distribution (uniform/nonuniform) and the magnitude of the load, as well as the surface to which the load is applied (surface/volume identification). The total load applied to the surface (volume) is, therefore, dependent on the area (contents) of the surface (volume).

Equivalent nodal forces first must be calculated from surface/volumetric loads and then added to the nodal force vector. MARC carries out this computation through numerical integration. (See Volume B of the MARC User's Manual for the numbers and locations of these integration points for different elements). The calculated equivalent nodal forces for lower-order elements are the same as those obtained by simple lumping of the uniform distribution. However, for high-order elements, the lumping is no longer simple (see Figure A 8.1-5). As a result, the surface/volumetric loads should not be lumped arbitrarily.



(a) Distributed Mechanical Load



(b) Distributed Heat Flux

Surface 2-3:

Uniform Normal Pressure p

Uniform Heat Flux q

Surface 3-4:

Nonuniform Normal Pressure p_1 - p_2

Nonuniform Heat Flux $q_1 = q_2$

Whole Volume:

Volumetric Loads P_x,P_y

Volumetric Heat Flux Q

Figure A 8.1-4 Schematic of Surface/Volumetric Load

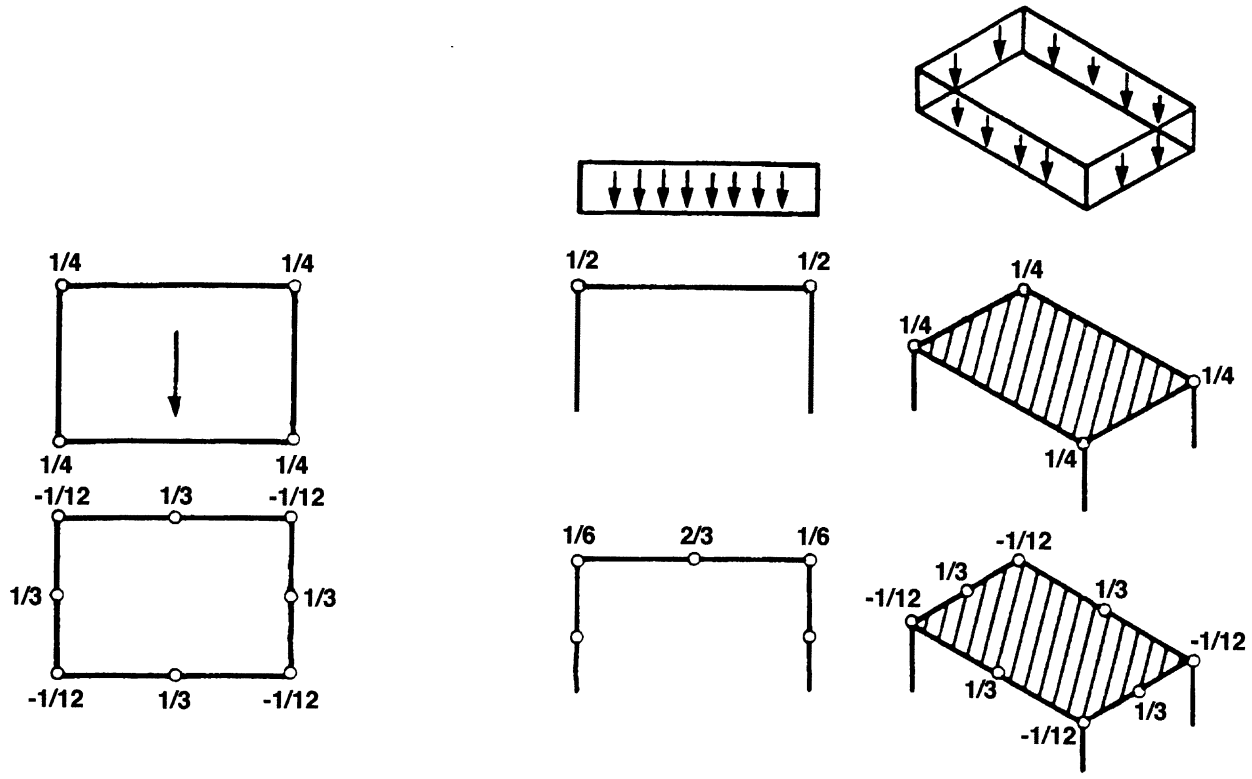


Figure A 8.1-5 Allocation of a Uniform Body Force to Nodes for a Rectangular Element Family

Fluid Drag and Wave Loads

MARC provides a fluid drag and wave load capabilities that can be applied on beam type structures that are partially or fully submerged in fluid (see Figure A 8.1-6). Morison's equation is used to evaluate the fluid drag loads that are associated with steady currents. Only distributed drag and buoyancy effects are considered. MARC employs Airy wave theory to evaluate wave velocities that can be invoked for dynamic analysis option.

Fluid drag and wave loads are invoked using DIST LOADS model definition with an IBODY load type of 11. These loads also require FLUID DRAG model definition to input the relevant information regarding the fluid elevation and its flow. Table A 8.1-2 lists input options for fluid drag and wave loads.

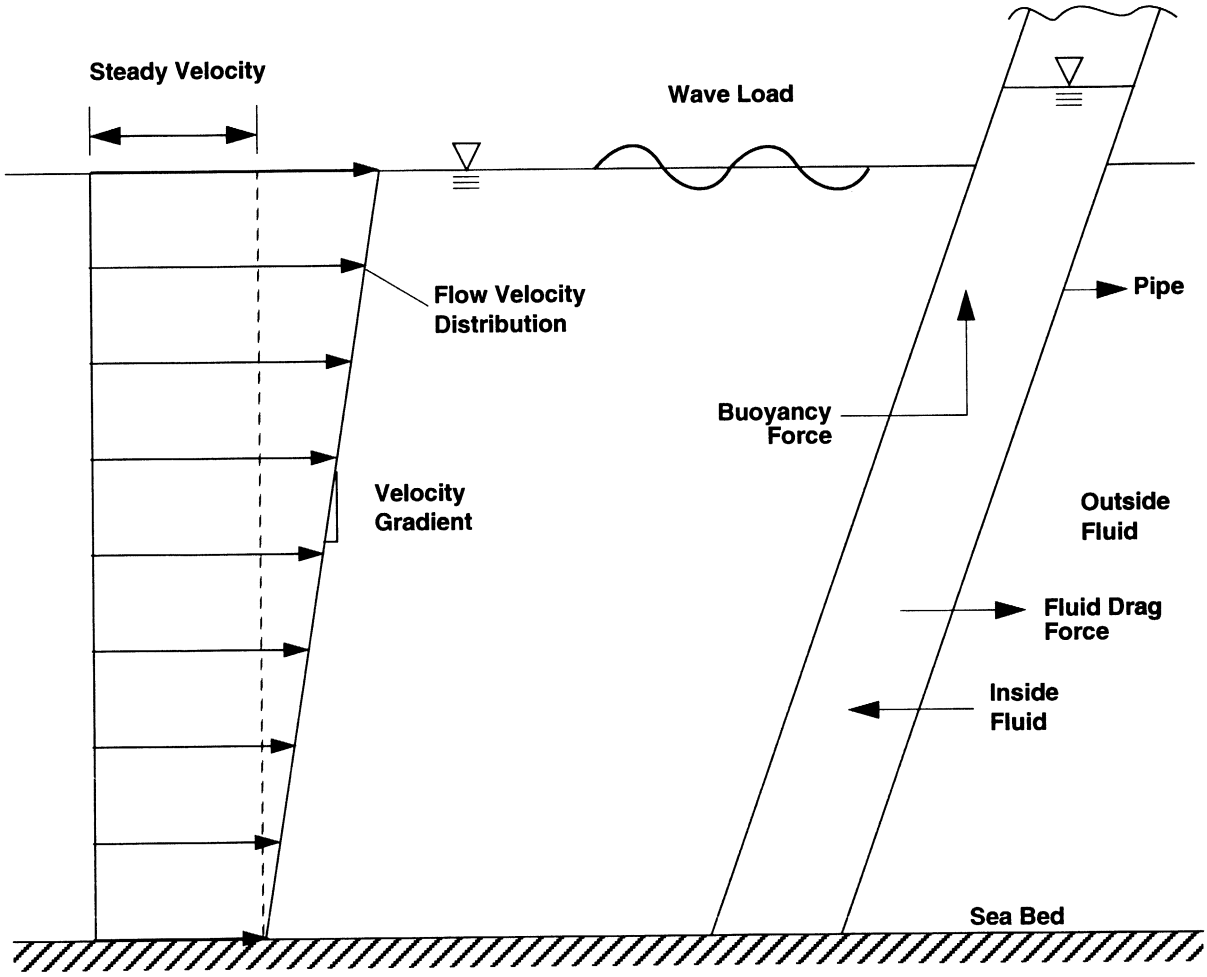


Figure A 8.1-6 Fluid Drag and Wave Loads

Table A 8.1-1 Input Options for Fluid Drag and Wave Loads

Load Description	Input Options		
	Model Definition	History Definition	User Subroutine
Fluid Drag Load	DIST LOADS FLUID DRAG	DIST LOADS	
Wave Load	DIST LOADS FLUID DRAG	DIST LOADS	

Mechanical Loads

The MARC program allows you to enter mechanical loads in various forms for stress analysis. These loads can be concentrated forces and moments, uniformly and nonuniformly distributed pressures, body forces, gravity or centrifugal loads. Table A 8.1-2 lists input options for mechanical loads.

Table A 8.1-2 Input Options for Mechanical Loads

Load Description	Input Options		
	Model Definition	History Definition	User Subroutine
Point Load Concentrated and Force/Moment	POINT LOAD	POINT LOAD	FORCDT
Surface Load Pressure Shearing Forces, and Distributed Moment (Uniform/Nonuniform Distribution)	DIST LOADS	DIST LOADS	FORCEM
Volumetric Load Body Forces and Acceleration Forces	DIST LOADS	DIST LOADS	FORCEM
Centrifugal Loading	DIST LOADS ROTATION A	DIST LOADS	
Coriolis Loading	DIST LOADS ROTATION A	DIST LOADS	
Fluid Loading	DIST LOADS FLUID DRAG		

Application of centrifugal and Coriolis loadings require the model definition ROTATION A, which defines the data corresponding to the axis of rotation. The actual load can be invoked by specifying an IBODY load types 100 or 103 for centrifugal and coriolis loadings, respectively. The square of rotation speed, ω^2, is entered in radians per time, for the magnitude of the distributed load. The mass density must also be defined in the PROPERTY, ISOTROPIC, or ORTHOTROPIC options.

Application of gravity load is achieved by using IBODY load type 102. The mass density must be defined in the PROPERTY, ISOTROPIC, or ORTHOTROPIC options. The acceleration can be given independently in the x, y, and z direction through the DIST LOADS option.

Use of the FOLLOW FORCE option results in the equivalent nodal loads due to pressures being calculated based on the current geometry. When this option is used, any change of surface area or orientation results in a change of load. FOLLOW FORCE automatically invokes the LOAD CORRECTION option, so stresses should be stored at all integration points. The FOLLOW FORCE option is typically used when a shell structure, which may undergo large deformation and rotations, is subjected to a pressure load. One can also specify that the follower force stiffness matrix may be included.

A number of history definition options are available for input of multiple load increments. For example, the AUTO LOAD option generates a specified number of increments, all having the same load increment, and is useful for nonlinear analysis with proportional loads; the PROPORTIONAL INCREMENT option allows the previous load increment to be scaled up or down for use in the current load increment. The AUTO INCREMENT option allows automatic load stepping in a quasi-static analysis and is useful for both geometric and material nonlinear problems.

Thermal Loads

Element temperatures are used in a thermal stress analysis to generate thermal load. The history definition AUTO THERM allows automatic application of temperature increments based on a set of temperatures defined throughout the mesh as a function of time. The CHANGE STATE option presents the temperatures to the program, and the program then creates its own set of temperature steps based on a temperature change tolerance provided through this option. Table A 8.1-3 lists input options for thermal loads. You can input either the incremental temperature or the total temperatures as thermal loads.

The thermal strain increment is defined as

$$\Delta\varepsilon^{th} = \int \alpha \Delta T \qquad \text{(A 8.1-1)}$$

where $\Delta\varepsilon^{th}$ is thermal strain increment, α is the coefficient of thermal expansion, and ΔT is the temperature increment. Equivalent nodal forces are calculated from the thermal strain increment and then added to the nodal force vector for the solution of the problem.

You can input the coefficient of thermal expansion through the model definition options PROPERTY, ISOTROPIC, ORTHOTROPIC and the temperature increment through various options.

Table A 8.1-3 Input Options for Thermal Loads

Load Description	Input Options			
	Parameter	Model Definition	History Definition	User Subroutine
Incremental Temperatures*	THERMAL	THERMAL LOADS	THERMAL LOADS	CREDE
Total Temperatures*	THERMAL	CHANGE STATE	CHANGE STATE	NEWSV
Initial Temperature*		INITIAL STATE		INITSV
Total Nodal Temperatures		POINT TEMP	POINT TEMP	
Initial Nodal Temperature		INITIAL TEMP		USINC

* Temperatures must be specified at each integration point (or at the centroid if the CENTROID option is used) of each element in the mesh.

Initial Stress and Initial Plastic Strain

The MARC program allows you to enter a set of initial stresses that simulate the stress state in the structure at the beginning of an analysis. A typical example is pre-stress in a tensioned fabric roof. The set of initial stresses must be self-equilibrating and should not exceed the yield stress of the material. Table A 8.1-4 shows the input options for initial stress.

Table A 8.1-4 Input Options for Initial Stress and Initial Plastic Strain

Load Description	Input Options			
	Parameter	Model Definition	History Definition	User Subroutine
Initial Stress Prestress	ISTRESS	INIT STRESS		ITRESS
Initial Plastic Strain		INITIAL PL		INITPL

The MARC program also provides various ways of initializing the equivalent plastic strain throughout the model. This is useful in metal forming analysis in which the previous amount of equivalent plastic strain is often required. This history dependent variable represents the amount of plastic deformation that the model was subjected to, and is used in the work (strain) hardening model. The input option is shown in Table A 8.1-4.

Heat Fluxes

In a heat transfer analysis, you can enter heat fluxes in various forms. Unlike stress analysis, heat transfer analysis requires entering the total values of flux. Table A 8.1-5 lists the input options for heat fluxes.

Table A 8.1-5 Input Options for Heat Fluxes

Load Description	Input Options		
	Model Definition	History Definition	User Subroutine
Point Heat Flux (Sink or Source)	POINT FLUX	POINT FLUX	FORCDT
Surface Heat Flux, Convection, Radiation	DIST FLUX FILMS	DIST FLUX FILMS	FLUX* FILM*
Volumetric Flux Load Body Flux	DIST FLUXES	DIST FLUXES FILMS	FLUX*

* Can be used for complicated FLUX loadings, convection, and radiation, allowing the input of nonuniform temperature- and time-dependent boundary conditions.

There are three special heat flux conditions representing insulation, convection, and radiation.

1. Insulation

   $$q = 0 \qquad \text{(A 8.1-2)}$$

   No input is required for the insulated case.

2. Convection

   $$q = H(T_S - T_\bullet) \qquad \text{(A 8.1-3)}$$

   You must enter the film coefficient H and ambient temperature $T_\bullet$ through the FILMS option or the user subroutine FILM. You can also directly input the heat flux q using user subroutine FLUX.

3. Radiation

$$q = \sigma \bullet \varepsilon (T_s^4 - T_\infty^4) \qquad \text{(A 8.1-4)}$$

You must enter either the heat flux q using user subroutine FLUX or the temperature dependent film coefficient $H(T_s, T_\bullet, \sigma, \varepsilon)$ and ambient temperature $T_\bullet$ using subroutine FILM. These relationships are shown in Equation A 8.1-5. The use of FILM is recommended to ensure a stable solution.

$$\begin{aligned} q &= \sigma\varepsilon (T_s^4 - T_\infty^4) \\ &= \sigma \bullet \varepsilon (T_s^3 + T_s^2 T_\infty + T_s T_\infty^2 + T_\infty^3)\ (T_s - T_\infty) \\ &= H (T_s, T_\infty)\ (T_s - T_\infty) \end{aligned} \qquad \text{(A 8.1-5)}$$

where σ is the Stefan-Boltzmann constant, ε is emissivity, T_s and T_∞ are unknown surface and ambient temperatures, respectively.

Mass Fluxes and Restrictors

In a hydrodynamic bearing analysis, you can enter mass fluxes, restrictors, and pump pressures as loads. Table A 8.1-6 lists input options for these quantities.

Table A 8.1-6 Input Options for Mass Fluxes and Restrictors

Load Description	Input Options			
	Parameter	Model Definition	History Definition	User Subroutine
Nodal Mass Fluxes	BEARING	POINT FLUX	POINT FLUX	
Distributed Mass Fluxes	BEARING	DIST FLUXES	DIST FLUXES	FLUX*
Restrictors	BEARING RESTRICTOR	RESTRICTOR		URESTR**

* Can be used for nonuniform mass fluxes
** Can be used for nonuniform restrictions or pump pressures.

Electrical Currents

In coupled thermo-electrical (Joule heating) analysis, you can prescribe electrical currents as loads for the calculation of unknown nodal voltages. In magnetostatic analysis, one can also define the current. In such cases, as a steady state analysis is performed, there is no time variation of the currents. Table A 8.1-7 lists input options for electrical currents.

Table A 8.1-7 Input Options for Electrical Currents

Load Description	Input Options		
	Model Definition	History Definition	User Subroutine
Nodal Current	POINT CURRENT	POINT CURRENT	FORCDT
Surface and Body Currents	DIST CURRENT	DIST CURRENT	FLUX

Electrostatic Charges

In an electrostatic analysis, the charge can be entered, noting that a steady state analysis is performed so there is no time variation of charge. Table A 8.1-8 summarized the input option.

Table A 8.1-8 Input Options for Electrostatic Charge

Load Description	Input Options		
	Model Definition	History Definition	User Subroutine
Nodal Charge	POINT CHARGE	POINT CHARGE	
Distributed and Body Charges	DIST CHARGE	DIST CHARGE	FLUX

Acoustic Sources

In acoustic analysis, you can enter a source pressure if a transient analysis by modal superposition is being performed. Table A 8.1-9 summarized the input options.

Table A 8.1-9 Input Options for Acoustic Sources

Load Description	Input Options		
	Model Definition	History Definition	User Subroutine
Nodal Source	POINT SOURCE	POINT SOURCE	FORCDT
Distributed Source	DIST SOURCE	DIST SOURCE	FLUX

Magnetostatic Currents

In a magnetostatic analysis, the current can be entered, noting that a steady state analysis is performed so there is no time variation of current. Table A 8.1-10 summarizes the input options.

Table A 8.1-10 Input Options for Magnetostatic Current

Load Description	Input Options		
	Model Definition	History Definition	User Subroutine
Nodal Current	POINT CURRENT	POINT CURRENT	
Distributed and Body Current	DIST CURRENT	DIST CURRENT	FLUX

Electromagnetic Currents and Charges

In an electromagnetic analysis, the current can be entered. These values can have time variation if a transient analysis is performed or a harmonic excitation may be applied. Table A 8.1-11 summarizes the input options.

Table A 8.1-11 Input Options for Currents and Charges

Load Description	Input Options		
	Model Definition	History Definition	User Subroutine
Nodal Current	POINT CURRENT	POINT CURRENT	
Distributed and Body Current	DIST CURRENT	DIST CURRENT	FLUX
Nodal Charge	POINT CURRENT-CHARGE	POINT CURRENT-CHARGE	
Distributed and Body Charge	DIST CHARGE	DIST CHARGE	

A 8.2 KINEMATIC CONSTRAINTS

The MARC program allows you to input kinematic constraints through various options that include

- Boundary conditions (prescribed nodal values)
- Transformation of degrees-of-freedom
- Shell transformation
- Tying constraints
- Support

Boundary Conditions

The program allows you to specify the nodal value for a particular degree-of-freedom. If you do not give a nodal value when you specify the boundary condition, MARC sets the fixed nodal value to zero. An option allows boundary conditions to be specified at the time of two-dimensional mesh generation with MESH2D. You can apply a different set of boundary conditions for each load increment.

Table A 8.2-1 gives the nodal values for the various analyses, and Table A 8.2-2 lists the input options for boundary conditions in different analyses.

Table A 8.2-1 Analyses with Corresponding Nodal Values

Analysis	Nodal Values
Stress	Displacements
Heat transfer	Temperature
Rigid Perfectly-plastic flow	Velocity
Coupled thermo-electrical	Voltage and Temperature
Hydrodynamic bearing	Pressure
Coupled thermo-mechanical	Displacement and Temperature
Acoustics	Pressure
Electrostatic	Potential
Magnetostatic	Potential

Table A 8.2-2 Input Options for Boundary Conditions

Load Description	Input Options		
	Model Definition	History Definition	User Subroutine
Displacement	FIXED DISP	DISP CHANGE	FORCDT
Temperature	FIXED TEMP	TEMP CHANGE	FORCDT
Voltage	FIXED VOLTAGE	VOLTAGE CHANGE	
Pressure	FIXED PRESSURE	PRESS CHANGE	FORCDT
Potential	FIXED POTENTIAL		

Transformation of Degree-of-Freedom

The program allows transformation of individual nodal degrees-of-freedom from the global direction to a local direction through an orthogonal transformation that facilitates the application of boundary conditions and the tying together of shell and solid elements.

Transformations are assumed to be orthogonal. Once you invoke a transformation on a node, you must input all loads and kinematic conditions for the node in the transformed system. Nodal output will be in the transformed system. This option is invoked using the TRANSFORMATION option.

The UTRANFORM option allows transformations to be entered via user subroutine UTRANS. This allows you to transform the degrees-of-freedom at an individual node from global directions to a local direction through an orthogonal transformation. UTRANFORM allows you to change the transformation with each increment. When you invoke this option, the nodal output will be in both the local and the global system.

Shell Transformation

The SHELL TRANSFORMATION option allows you to transform the global degrees-of- freedom of doubly curved shells to shell degrees-of-freedom in order to facilitate application of forces in the shell directions, edge moments, and clamped or simply supported boundary conditions. There are four types of shell transformations.

The model definition block SHELL TRANSFORMATION specifies information on the shell transformation. For Types 1 and 3, only the node number has to be specified. For Types 2 and 4, a boundary direction (the direction cosine of t as shown in Figure A 8.2-1) also has to be specified in the θ_1, θ_2 surface.

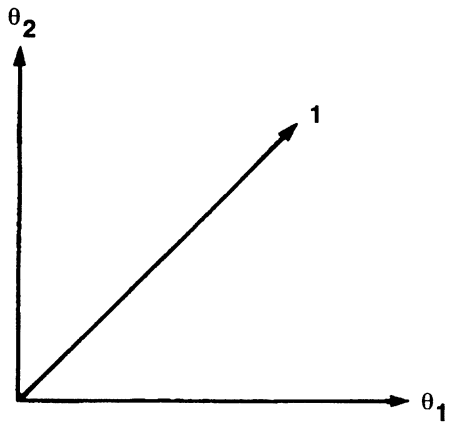


Figure A 8.2-1 Boundary Directions in Shell Transformation

After transformation, the following definitions apply.

Type 1: Transformation for two-dimensional beams and shell nodes (Element Types 15, 16 and 17). The transformation defines the degrees-of- freedom with respect to a local coordinate system (s, n) (see Figure A 8.2-2). The four degrees-of-freedom after transformation are:

1 = u_s tangential displacement
2 = u_n normal displacement
3 = ϕ rotation
4 = ε Meridional stretch

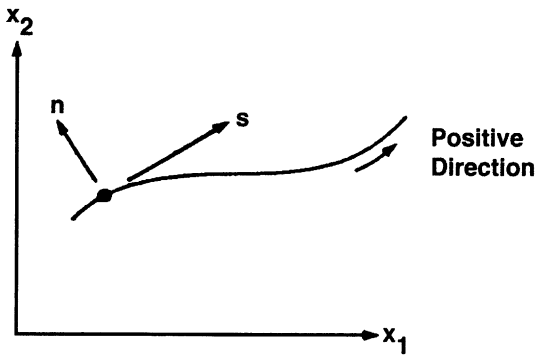


Figure A 8.2-2 Type 1: Shell Transformation

Type 2: Transformations for doubly curved shell nodes with nine degrees-of- freedom (Element Type 4, Element Type 24, corner nodes). The transformation defines a local coordinate system (t,s,n). (See Figure A 8.2-3).

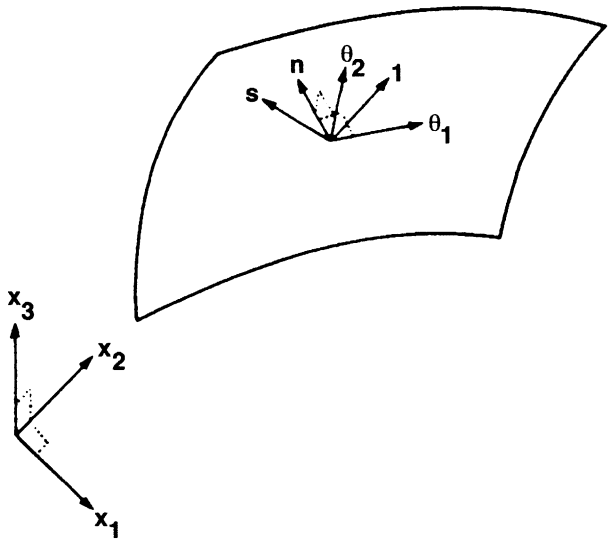


Figure A 8.2-3 Types 2 and 4: Shell Transformations

The nine degrees-of-freedom after transformation are:

1 = u_t displacement in specified (boundary) direction
2 = u_s displacement normal to (boundary) direction but tangential to shell surface
3 = u_n displacement normal to shell surface
4 = ϕ_t rotation of shell around boundary
5 = ϕ_s rotation of shell around normal to boundary tangential to the shell surface
6 = ϕ_n rotation of boundary around normal to shell surface
7 = ε_t stretch tangential to specified (boundary) direction
8 = ε_s stretch normal to specified (boundary) direction
9 = γ_{ts} shear stretch in t-s direction

Type 3: Transformations for doubly curved shell nodes with 3 degrees-of- freedom (Element Type 24, midside nodes). The transformation defines a local coordinate system (t,s,n), (see Figure A 8.2-4). The degrees-of-freedom after transformation are:

1 = $\partial u_t/\partial s$ rotation of normal to boundary around normal to shell
2 = $\partial u_s/\partial s$ stretch normal to boundary
3= $\partial u_n/\partial s$ rotation around boundary

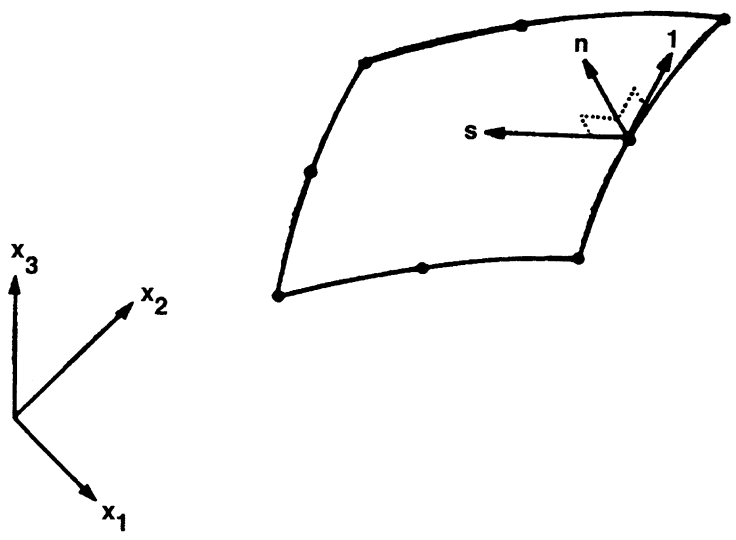


Figure A 8.2-4 Type 3: Shell Transformation

Type 4: Transformation for doubly curved nodes with 12 degrees-of-freedom (Element Type 4). The transformation defines a local coordinate system (t,s,n). The first nine degrees-of-freedom after transformation are the same as 1 through 9 in Type 2 (Figure A 8.2-3), and the remaining three are:

$10 = \partial^2 u_t / \partial\Theta_1 \partial\Theta$ variation of g along the boundary (A 8.2-1)

$11 = \partial^2 u_s / \partial\Theta_1 \partial\Theta_2$ variation of e along the boundary (A 8.2-2)

$12 = \partial^2 u_n / \partial\Theta_1 \partial\Theta_2$ variation of f along the boundary (A 8.2-3)

When using the SHELL TRANSFORMATION option, the displacement increments and reaction forces are output in the local directions immediately after solution of the equations. At the end of the increment, you can print out the global displacement increments and total displacements in the global coordinate directions.

If you invoke the FOLLOW FOR option, the shell transformation option defines a local coordinate system in the current (updated) geometry of the structure. This additional feature is especially useful with the UPDATE option, because you can then specify edge moments and/or large edge rotations of shells and beams.

****WARNING****

If you apply a shell transformation to a node, do not apply a standard transformation or shell tying type to that node.

Tying Constraint

The MARC program contains a generalized tying (constraint) condition option. Any constraint involving linear dependence of nodal degrees-of-freedom may be included in the stiffness equations.

A tying constraint involves one tied node and one or more retained nodes, and a tying (constraint) condition between the tied and retained nodes. The degrees-of-freedom (for example, displacements, temperatures) of the tied node are dependent on the degrees-of-freedom of the retained nodes through the tying condition. In some special tying conditions, the tied node can also be a retained node. The tying condition can be represented by a tying (constraint) matrix. Note that if the tying constraint involves only one retained node, the choice of which node is to be tied to retained is arbitrary.

As a simple example, impose the constraint that the first degree-of-freedom of node I be equal to that of node J at all times (see Figure A 8.2-5).

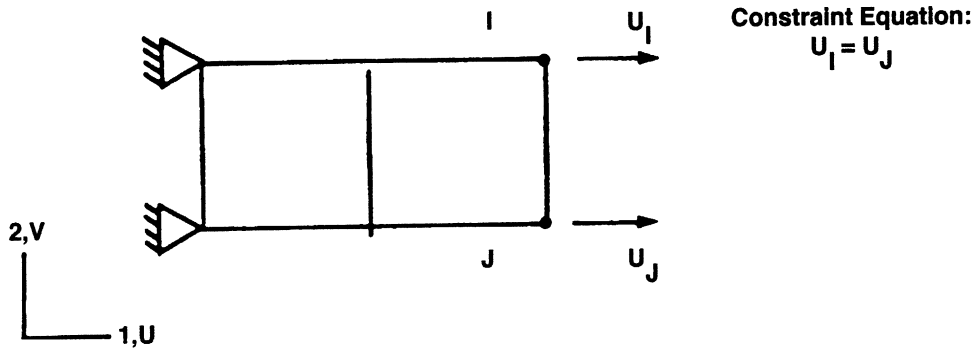


Figure A 8.2-5 Simple Tying Constraint

As a second example, the simulation of a sliding boundary condition requires the input of both the boundary conditions and the tying constraints (see Figure A 8.2-6).

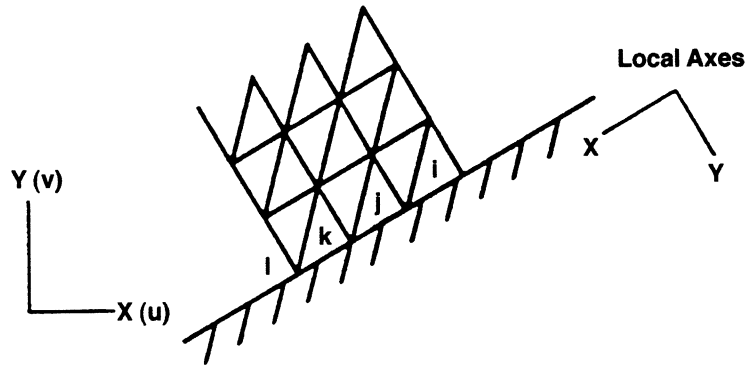


Figure A 8.2-6 Tying Constraint Illustration (Sliding Boundary Conditions)

The example illustrated in Figure A 8.2-6 enforces rigid sliding on the boundary in the local coordinates defined above.

$$v_i = v_j = v_k = v_l = 0 \qquad \text{(A 8.2-4)}$$

$$u_i = u_j = u_k = u_l \qquad \text{(A 8.2-5)}$$

The first equation is a set of fixed boundary conditions. The second equation is a constraint equation and can be rewritten as three constraint equations:

$$u_i = u_j \qquad \text{(A 8.2-6)}$$

$$u_k = u_j \qquad \text{(A 8.2-7)}$$

$$u_l = u_j \qquad \text{(A 8.2-8)}$$

These equations express all the u displacements in terms of u_j. In this example, node j is chosen to be the retained node; nodes i, k, and l are tied nodes. You can use the TYING option to enter this information.

The MARC program has a number of standard tying constraints that can be used for mesh refinement, shell-to-shell, shell-to-beam, beam-to-beam, and shell-to-solid intersections. Table A 8.2-3 through Table A 8.2-6 describe these options. Table A 8.2-7 through Table A 8.2-9 show the tying constraints for pipe elements, shell stiffeners, and nodal degrees-of-freedom. Table A 8.2-10 summarizes the rigid link constraint.

The SERVO LINK option uses the homogeneous linear constraint capability (TYING) to input simple constraints of the form

$$u_t = a_1 u_{r1} + a_2 u_{r2} + \ldots \qquad \text{(A 8.2-9)}$$

where u_t is a degree-of-freedom to be constrained; u_{r1}, u_{r2},..., are the other retained degrees-of-freedom in this structure; and a_1, a_2,..., are constants provided in this option.

You can use the model definition option TYING or SERVO LINK to enter standard tying constraint information, and the TYING CHANGE option to change tying constraints during load incrementation.

User subroutine UFORMS is a powerful method to specify a user-defined constraint equation. This constraint may be nonlinear, i.e., it may be dependent on time or previous deformation.

Table A 8.2-3 Tying Constraints for Mesh Refinement

Tying Code	Number of Retained Nodes	Purpose	Remarks
31	2	Refine mesh of first order (linear displacement) elements in two dimensions	Tie interior nodes on refined side to corner nodes in coarse side (see Figure A 8.2-7)
32	3	Refine mesh of second order (quadratic displacement in two dimensions	Tie interior nodes on refined side to edge of element of coarse side (see Figure A 8.2-8)
33	4	Refine mesh of 8-node bricks	Tie interior node on refined side to four-corner nodes of an element face on coarse side (see Figure A 8.2-9)
34	8	Refine mesh of 20-node bricks	Tie interior nodes on refined side to eight (four-corner, four midside) nodes of element face on coarse side (see Figure A 8.2-10)

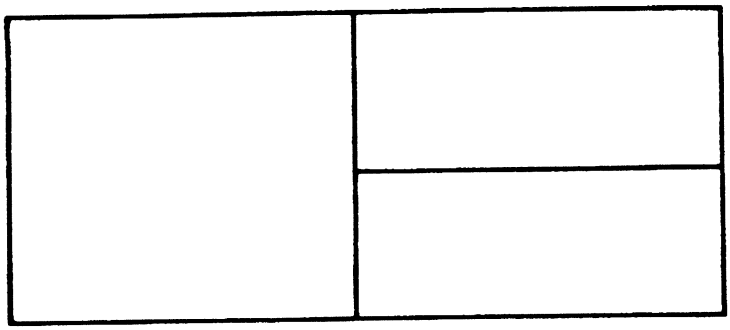


Figure A 8.2-7 Mesh Refinement for 4-Node Quad

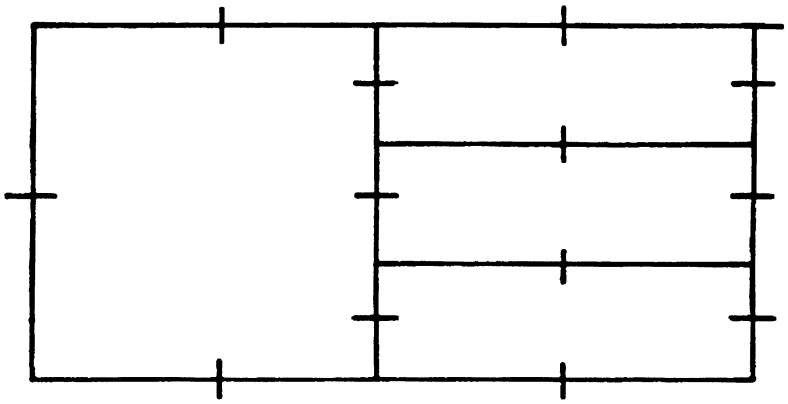


Figure A 8.2-8 Mesh Refinement for 8-Node Quad

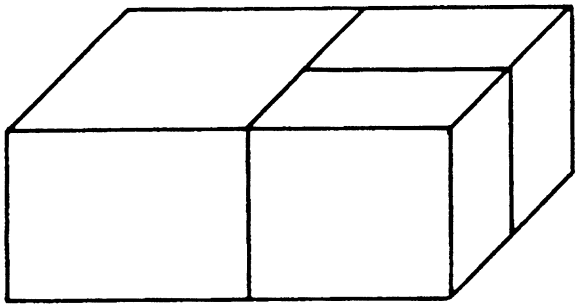


Figure A 8.2-9 Mesh Refinement for 8-Node Brick

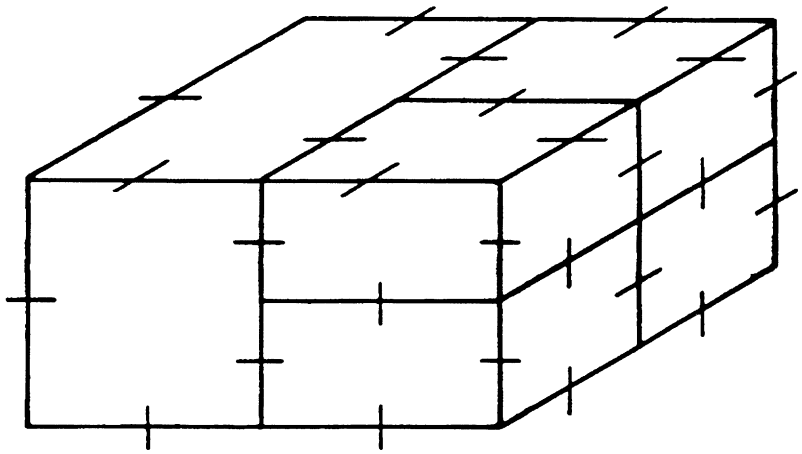


Figure A 8.2-10 Mesh Refinement for 20-Node Brick

Table A 8.2-4 Tying Constraints for Shell-to-Shell Intersection

Tying Code	Number of Retained Nodes	Purpose	Remarks
22	2	Join intersecting shells, Element Type 4, 8, or 24; fully moment-carrying joint	Tied node is also the second retained node*
18	2	Join intersecting shells, Element Type 4, 8, or 14; fully moment-carrying joint	Tied node is also the second retained node
28	2	Join intersecting shells, Element Type 4, 8, or 24; pinned point	Tied node is also the second retained node
24**	2	Join intersecting shells or beams, Element Types 15-17	Tied node is also the second retained node

* Thickness vector must be specified at tied nodes and retained nodes.
** See Figure A 8.2-11.

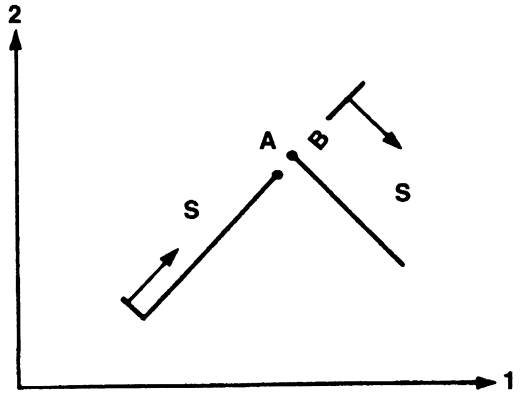


Figure A 8.2-11 Standard Tying Type 24, Tie Shell-to-Shell or Beam-to- Beam; Moment-Carrying (24)

Table A 8.2-5 Tying Constraints for Beam-to-Beam Intersections

Tying Code	Number of Retained Nodes	Purpose	Remarks
13	2	Join Two Elements Type 13 under an arbitrary angle; full moment-carrying joint	Tied node also the second retained node
52	1	Pin joint for Beam Typed 14, 25, 52, 76-79, 98	
53	1	Full moment-carrying joint for Beam Types 14, 25, 52, 76-79, 98	Tie interior node on refined side to four-corner nodes of an element face on coarse side (see Figure A 8.2-9)

Table A 8.2-6 Tying Constraints for Shell-to-Solid Intersections

Tying Code	Number of Retained Nodes	Purpose	Remarks
23		Tie axisymmetric solid node to axisymmetric shell (Element Type 2 node)	Tied and retained nodes must be transformed to local system and TRANSFORMATION option invoked.
25	2	Join solid mesh to shell or beam (Type 15 or 16)	Similar to 23 but no transformation needed. Tied node also second retained node. (see Figure A 8.2-12)

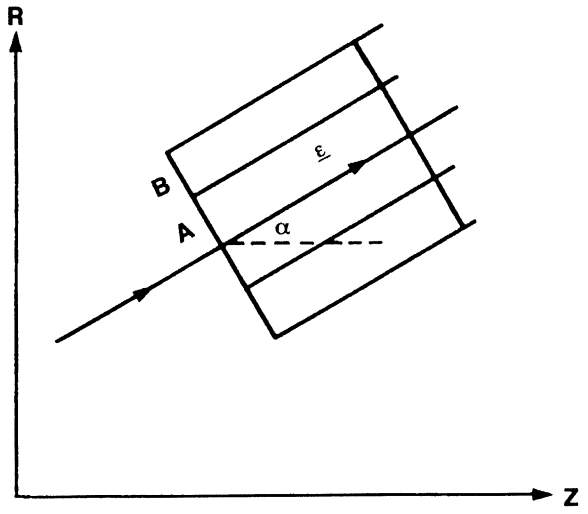


Figure A 8.2-12 Standard Tying Type 25, Tie Solid-to-Shell (Element Type 15)

Table A 8.2-7 Tying Constraints for Pipe Bend Element (Elements 14 and 17)

Tying Code	Number of Retained Nodes	Purpose	Remarks
15	One less than the number of shell nodes in the z-r plane of the section	Special tying types for pipe bend Element 17 to remove rigid body modes	
16	Number of shell nodes in the z-r plane of the section	Special tying types for pipe bend Element 17 to remove rigid body modes	
17	2	Special tying types for pipe bend Element 17 to couple bend section into pipe bend	

Table A 8.2-8 Tying Constraints for Shell Stiffener (Element 13 as a Stiffener on Shell Elements 4 or 8)

Tying Code	Number of Retained Nodes	Purpose	Remarks
19	2	Use beam Element 13 as a stiffener on shell Elements 4 or 8. Tied node is beam node; first retained node is shell node, second is beam node again. Beam node should be on, or close to, the normal to the shell at the shell node	Tied node also second retained node
20	3	Create an extra node in a shell Type 8 Element tied to the interpolated shell displacements with tying Type 21 to tie a beam Element 13 or a stiffener across a shell element	Always used after tying Type 21
21	2	Same as Type 19, but tying beam to an interpolated shell node not a vertex of an element (Element Type 8 only) must be followed by Type 20 to tie interpolated shell node into shell mesh	Must be followed by tying Type 20

Table A 8.2-9 Tying Constraints for Nodal Degrees-of-Freedom

Tying Code	Number of Retained Nodes	Purpose	Remarks
I≤NDEG	1	Tie the Ith degree of freedom at the tied node to the Ith degree of freedom at the retained node	NDEG = number of degrees of freedom per node
100	1	Tie all degrees of freedom at the tied node to the corresponding degrees of freedom at the retained node	
>100	1	Generate several tyings of type ≤NDEG	Tying code is first degree of freedom multiplied by 100 added to last degree of freedom (209 means tie second through ninth degree to freedom at tied node to respective second through ninth degree of freedom at retained node)
<0 (negative integer)		User-defined	User-generated tying type through subroutine UFORMS

Rigid Link Constraint

Tying type 80 may be used to define a rigid link between nodes. This capability may be used for both small deformation and large deformation, large rotation problems. In small deformations, a linear constraint equation is used. In addition to the end points of the link, a second retained node must be given. This node will be used to calculate and store the rigid body rotations. For two-dimensional problems, this will represent a rotation about the Z-axis.

A complete rigid region may be modelled by using multiple tying type 80's. In this case the same two nodes will be used for the retained nodes in all of the constraints.

The rigid links may be used with all elements except types 4, 8, 24, 15, 16 and 17. In addition, it should not be used in rigid plastic analysis.

Table A 8.2-10 Rigid Link Constraint

Tying Code	Number of Retained Nodes	Purpose	Remarks
80	2	Define a rigid link between nodes	The second retained node is an unattached node which will contain the rotation

Shell-to-Solid Tying

In many problems, a region exists that is modelled with both brick elements and shell elements. A particular case of this is shown is Figure A 8.2-13 and Figure A 8.2-14 below.

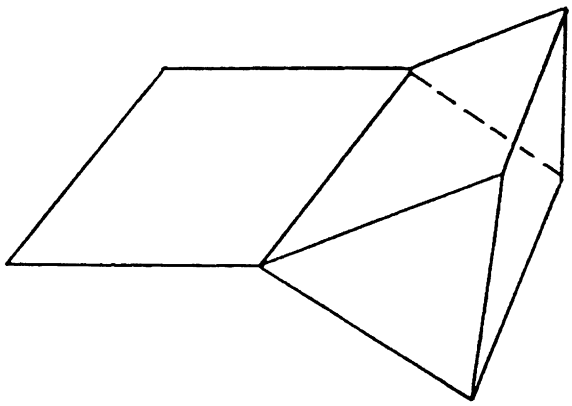


Figure A 8.2-13 4-Node Shell-to-Solid Automatic Constraint

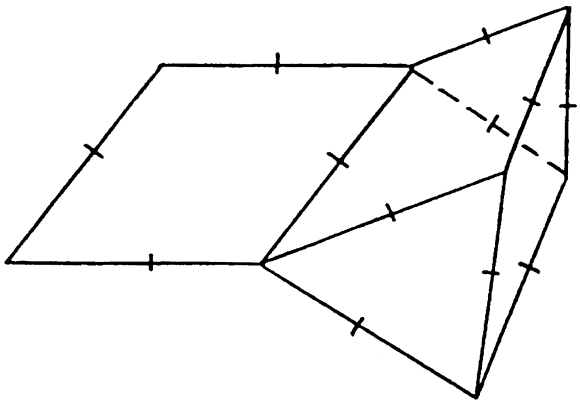


Figure A 8.2-14 8-Node Shell-to-Solid Automatic Constraint

In the first case, an 8-node brick which has been reduced to a triangular prism is connected to a 4-node shell. In the second case, a 20-node brick is connected to an 8-node shell. An automatic constraint equation is developed between the elements. Note that the thickness of the shell must be entered as the brick thickness.

Support Conditions

The MARC program provides linear and nonlinear springs and foundations for the modeling of support conditions. For dynamic analysis, a dashpot may also be included.Table A 8.2-11 lists input options for linear springs and elastic foundations.

The force in the spring is

$$F = K(u_2 - u_1) + D(v_2 - v_1) \qquad \text{(A 8.2-10)}$$

where K is the spring stiffness, u_2 is the displacement of the degree-of-freedom at the second end of the spring, and u_1 is the displacement of the degree-of-freedom at the first end of the spring. In a dynamic analysis, D is the damping factor and v_2 and v_1 are the velocities of the nodes.

You can specify the elements in MARC to be supported on a frictionless, elastic foundation. The foundation supports the structure with a force per unit area (force per unit length, for beams) given by

$$p_n = Ku_n \qquad \text{(A 8.2-11)}$$

where K is the equivalent spring stiffness of the foundation, and u_n is the normal displacement of the surface at a point in the same direction as p_n.

The same conventions apply to the elastic foundation specification as for pressure specification, in terms of the face of the element that will be used. The force will be applied whether the displacement is tensile or compressive.

Subroutine USPRNG allows the user to supply a nonlinear spring stiffness or a nonlinear foundation stiffness as a function of prior displacement and force.

Table A 8.2-11 Input Options for Linear Springs and Elastic Foundations

Load Description	Input Options		
	Model Definition	History Definition	User Subroutine
Linear Springs	SPRING		USPRING
Elastic Foundation	FOUNDATION	FOUNDATION	USPRING

Chapter 9 OUTPUT RESULTS

This chapter summarizes the information that MARC provides in the output. In addition to reviewing your input, MARC provides information about the procedures the program uses and the workspace allocation. All calculated results are automatically written to the output file unless the user specifically requests otherwise.

A 9.1 WORKSPACE INFORMATION

The MARC program reports several aspects of workspace information, specifically, the allocation of memory workspace and the size of the work files.

First, the program gives the workspace needed for the input and stiffness assembly. This number tells how much memory workspace is needed to store the user-supplied data, the program-calculated data, and two-element stiffness matrices. Each set of data comprises three parts: overhead, element information, and nodal information. Element information consists of properties, geometries, strains, and stresses, and nodal information consists of coordinates, displacements, and applied forces.

NOTE

You must input a number (an amount of memory) of the SIZING option greater than the number (amount of memory) requested by the program.

Next, the program specifies the internal core allocation parameters. These values are useful in user subroutines and are often provided to the user subroutines. The values provided here are:

- the number of degrees-of-freedom per node (NDEG)
- the number of coordinates per node (NCRD)
- the number of generalized strains per integration point (NGENS)
- the number of stresses per integration point (NSTRMX)
- the number of invariants per integration point (NEQST)

Note the following:

- The number of degrees-of-freedom per node indicates the number of boundary conditions necessary to eliminate rigid body modes.
- The number of generalized strains per integration point for beams and shells include both membrane and curvature strains.
- The number of stresses per integration point is the number of stresses per layer multiplied by the number of layers.
- The number of invariants per integration point is the number of layers for either shell or beam elements.

The ELSTO option stores element information on an auxiliary storage device, rather than in main memory. You can invoke this option, or MARC will invoke it automatically. In this option, the program:

- sets a flag for element storage (IELSTO)
- specifies how many words are used per element (NELSTO)
- specifies the number of elements per buffer (MXELS)
- specifies the total amount of space needed to store this information

The number of elements per buffer should be at least two, and you can change the buffer size in the ELSTO parameter option to increase the number of elements which will be in the buffer.

The program states how much workspace is needed for the MESH2D option and any of the bandwidth optimizers. In each case, the number you prescribe in the SIZING option must be larger than the number given by the program.

Direct Solution Procedure

MARC offers two direct solver options. The default direct solver uses a variable bandwidth (also known as profile) storage scheme. Since the required storage space and the solution time depend on the cumulative profile length, a nodal renumbering that minimizes the profile length can give rise to considerable savings in storage and solution time. Several renumbering schemes are available in MARC and they can be requested using OPTIMIZE. These optimization procedures are described in Section 4.9. The other direct solver option available in MARC uses a sparse storage scheme. A nodal renumbering based on minimum degree algorithm is done automatically to minimize the fill after factorization.

After the program has determined the total profile length or the total fill after factorization depending on the requested solver option, it reports the total workspace needed for in-core matrix storage. If the workspace specified on SIZING is larger than the total workspace needed for the in-core matrix solution, the in-core matrix storage will be employed. In the case when the required storage exceeds the workspace specified in SIZING, the out-of-core solver is invoked. In which case, the program reports the required workspace for each nodal row, the number of nodal rows per buffer, and the size of each auxiliary file. The more nodal rows per buffer, the more efficient the solution procedure is. The buffer size can be increased only by increasing the allocation in SIZING. The program terminates unless the number of nodal rows per buffer is two or greater. If the out-of-core option is invoked, the program reports the structure of the global stiffness matrix indicating the nonzero values in the stiffness matrix.

EBE Iterative Solution Procedure

The program divides the mesh into groups of similar element types; on machines that have parallel processing capabilities, these groups are further divided to improve performance. The program first attempts to put all element groups in core, which results in a very efficient technique, requiring no input/output. If it is not possible to put all of the groups into memory, the program will put one group in memory at a time. This will result in additional input/output.

A 9.2 INCREMENT INFORMATION

A variety of information is given for each increment, and often for each iteration, of the analysis. This information is useful in determining the accuracy and the stability of the analysis.

Summary of Loads

The printout entitled "Load Increments Associated with Each Degree-of-Freedom" allows you to check the load input quickly. It represents the sum, over all nodes in the mesh, of the point loads and the equivalent forces obtained after distributed loads (pressures, body forces) are applied. For example, you can easily check a pressure because the total force in a global coordinate direction would be the projected area normal to that direction of the surface upon which the pressure is applied, multiplied by the pressure magnitude.

Timing Information

The amount of CPU necessary to reach the given location in the analysis is indicated by the following output:

- start of increment
- start of assembly
- start of matrix solution
- end of matrix solution
- end of increment

As an example, subtract the time associated with the start of the matrix solution from the time associated with the end of the matrix solution to determine how much time was spent in the equation solver.

Singularity Ratio

The singularity ratio is a measure of the conditioning of the matrix (see the Appendix for further details). This ratio is printed each time there is a solution of the matrix equations. You can measure the influence of the nonlinearities in the structure by examining the change in singularity ratios between increments.

Convergence

Several messages are printed that concern the convergence of the solution. These messages indicate the displacement, velocity, or residual error and are very important because they provide information concerning the accuracy of the solution procedure. These messages also indicate the ratio of the error and its relative quantity. This ratio must be less than that given in the CONTROL option for convergence to occur. See the Appendix for details on convergence testing.

When the iterative solver is used, additional messages are printed regarding the convergence of the solution procedures.

A 9.3 SELECTIVE PRINTOUT

The MARC program gives you several options and user subroutines for the control of program output.

Options

PRINT CHOICE allows you to select how much of the element and nodal information is to be printed. The possible selections are

- group of elements
- group of nodes which layers (form beam and shell elements)
- which integration points
- increment frequency between printouts

The data entered through this option remain in control until you insert a subsequent PRINT CHOICE set. You can include such a set with either the model definition or with the history definition card set. To obtain the default printout after a previous PRINT CHOICE, invoke PRINT CHOICE using blank entries.

NOTE

The PRINT CHOICE option has no effect on the RESTART or POST file.

You can use the PRINT ELEM capability as a replacement for, or in conjunction with, the PRINT CHOICE option. The enhancements this option offers over the PRINT CHOICE option are the following:

1. You have a choice of integration and layer points for each element to be printed, which is especially useful when several different element types are used in one analysis.
2. You have a choice of the type of quantity to be printed, for instance, stresses could be printed for all elements, but strains for only a few.
3. The types of output quantities that can be selected are stresses, strains, creep strain, thermal strain, cracking strain, Cauchy stress, state variables, strain energy, or all nonzero quantities. These values are printed in the
4. The output may be placed in a file other than the standard output file.
5. The PRINT ELEM NODE option may be used to obtain element quantities such as stress and strains at the nodal points of each element. These are obtained by extrapolating the integration point values to the nodes.

The PRINT NODE option is an alternative to the PRINT CHOICE option for controlling the output of nodal quantities. The additional capabilities of this option compared to the PRINT CHOICE are as follows:

1. You can choose which of the following nodal quantities are to be printed:

   incremental displacements
   total displacements velocities
   accelerations
   reaction forces
   generalized stresses

2. Different quantities can be printed for different node points.
3. The output may be placed in a file other than the standard output file.

NOTE

All quantities are saved on the Restart tape, and you can obtain them at a later time by using the RESTART option. Quantities that were not printed out are available for later use.

The SUMMARY option allows you to get a quick summary of the results obtained in the analysis. This option prints the maximum and minimum quantities in tabular form. The table is designed for direct placement into reports. You control the increment frequency of summary information and the file unit to which the information is written.

The SUMMARY feature reports on the physical components and the Tresca, von Mises, and mean values of stress, plastic strain, and creep strain. It also reports on such nodal quantities as displacements, velocities, accelerations, and reaction forces.

The SORT options (ELEM SORT and NODE SORT) allow you to sort calculated quantities in either ascending or descending order. These quantities may either be sorted by their real magnitude or their absolute magnitude. The SORT options also allow you to control the type of quantity to be sorted, for example, equivalent stress and the number of items to be sorted. The SORT options print the sorted values in tabular form. The table is designed for direct placement into reports. You may control the increment frequency of sorted values and the file unit to which these values are written.

The PRINT VMASS option allows you to selectively choose which elements and associated volumes and masses to be printed. In order to have correct mass computations, mass density for each element must be given. The volumes and masses can be written on either standard output file unit 6, or user specified unit.

User Subroutines

User subroutines may be used to obtain additional output, which can be accessed at each time/load increment. The available subroutines are the following:

- IMPD – obtains nodal quantities, such as displacements, coordinates, reaction forces, velocities, and accelerations
- ELEVAR – obtains element quantities such as strains, stresses, state variables, and cracking information
- ELEVEC – outputs element quantities during harmonic subincrements
- INTCRD – obtains the integration point coordinates used for forming the stiffness matrix

A 9.4 RESTART

One capability of the RESTART option allows you to recover the output at increments where printout was suppressed in previous runs. This option can be used to print time/load increments for a number of consecutive increments. Under this option, the program will not do any analysis.

In conjunction with the RESTART option, you can use either the PRINT CHOICE or PRINT ELEM, or PRINT NODE option to select a region of the model for which you want to obtain results.

A 9.5 ELEMENT INFORMATION

The main output from an increment includes element information followed by nodal information. The system provides the element data at each integration point. If you use the CENTROID parameter option, the system provides the element data at the centroidal point. You can control the amount of printed output by using either the PRINT CHOICE or PRINT ELEM option. All quantities are total values at the current state (at the end of the current increment), and the physical components are printed for each tensor quantity (stress, strain, and generalized stress and strain). The orientation of these physical components is generally in the global coordinate system; however, their orientation depends on the element type. (Volume B of the User's Manual explains the output for each particular element type.) Also, the physical components may be printed with respect to a user defined preferred system.

In addition to the physical components, certain invariants are given, as follows:

1. Tresca intensity - the maximum difference of the principal values, and the measure of intensity usually required for ASME code analysis

   von Mises intensity - defined for stress as

$$\bar{\sigma} = \sqrt{\frac{3}{2} S_{ij} S_{ij}} \qquad S_{ij} = \sigma - \frac{1}{3}\delta_{ij}\sigma_{kk} \tag{A 9.5-1}$$

   where σ is a stress tensor and s is the deviatoric stress tensor. For beam and truss elements, $\sigma_{22} = \sigma_{33} = 0$, and additional shears are zero; for plane stress elements, including plates and shells, $\sigma_{33} = 0$.

2. von Mises intensity - calculated for strain type quantities as

$$\bar{\varepsilon} = \sqrt{\frac{2}{3} \varepsilon_{ij} \varepsilon_{ij}} \tag{A 9.5-2}$$

   The program uses these measures in the plasticity and creep constitutive theories. For example, incompressible metal creep and plasticity are based on the equivalent von Mises stress. For beam, truss, and plane stress elements, an incompressibility assumption is made regarding the noncalculated strain components.

   a. For beam and truss elements, this results in

$$\varepsilon_{22} = \varepsilon_{33} = -\frac{1}{2}\varepsilon_{11} \tag{A 9.5-3}$$

   b. For plane stress elements

$$\varepsilon_{33} = -(\varepsilon_{11} + \varepsilon_{22}) \tag{A 9.5-4}$$

   c. For beam and plane stress elements taken as a whole

$$\varepsilon_{kk} = 0 \tag{A 9.5-5}$$

4. Mean normal intensity - calculated as

$$(\sigma_{11} + \sigma_{22} + \sigma_{33})/3$$

or (A 9.5-6)

$$(\varepsilon_{11} + \varepsilon_{22} + \varepsilon_{33})/3$$

Equation A 9.5-6 represents the negative hydrostatic pressure for stress quantities. For strain quantities, the equation gives the dilatational magnitude. This measurement is important in hydrostatically dependent theories (Mohr-Coulomb or extended von Mises materials), and for materials susceptible to void growth.

5. The principal values are calculated from the physical components. The program solves the eigenvalue problem for the principal values using the Jacobi transformation method. Note that this is an iterative procedure and may give slightly different results from those obtained by solving the cubic equation exactly.

6. State variables are given at any point where they are nonzero.

Solid (Continuum) Elements

For solid (continuum) elements, stress, strain, and state variables are the only element quantities given. The program prints out stresses and total strain for each integration point. In addition, it prints out thermal, plastic, creep, and cracking strains, if they are applicable. Note that the total strains include the thermal contribution.

Shell Elements

The program prints generalized stresses and generalized total strains for each integration point. For thick shell (Type 22, 72, 75) elements interlaminar shear stresses are printed at the interface of two laminated layers.

The generalized stresses printed out for shell elements are

$$\int_{-t/2}^{+t/2} \sigma_{ij} dy \qquad \text{(A 9.5-7)}$$

(Section Force)

$$\int_{-t/2}^{+t/2} y\sigma_{ij} dy \qquad \text{(A 9.5-8)}$$

(Section moment)

The generalized strains printed are

$$E_{\alpha\beta;} \qquad \alpha, \beta = 1, 2 \qquad \text{(A 9.5-9)}$$

(Stretch)

$$\kappa_{\alpha\beta;} \qquad \alpha, \beta = 1, 2 \qquad \text{(A 9.5-10)}$$

(Curvature)

Physical stress values are output only for the extreme layers unless you invoke either PRINT CHOICE or PRINT ELEM. In addition, thermal, plastic, creep, and cracking strains are printed for values at the layers, if applicable. Although the total strains are not output for the layers, you can calculate them using the following equations.

$$\varepsilon_{11} = E_{11} + h\kappa_{11} \qquad \text{(A 9.5-11)}$$

$$\varepsilon_{22} = E_{22} + h\kappa_{22} \qquad \text{(A 9.5-12)}$$

$$\gamma_{12} = E_{12} + hk\kappa_{12} \qquad \text{(A 9.5-13)}$$

where h is the directed distance from the midsurface to the layer; E_{ij} are the stretches; and κ_{ij} are the curvatures as printed. You can obtain these quantities by using the subroutine ELEVAR.

Beam Elements

The printout for beam elements is similar to shell elements, except that the section values are force, bending and torsion moment, and bimoment for open section beams. For beam element type 45, interlaminar shear stresses are printed at the interface of laminated layers if the TSHEAR parameter option is used.

These values are given relative to the section axes (X, Y, Z) which are defined in the COORDINATES or the GEOMETRY option.

Heat Transfer Elements

The program prints the element temperatures and temperature gradients. In the coupled thermo-electric (Joule heating) analysis, the program also prints the element voltage, current, and heat fluxes.

Gap Elements

The program prints the gap contact force, friction forces, and the amount of slip.

Linear Springs

The program prints the spring forces.

Hydrodynamic Bearing

The program prints the pressure in the lubricant.

A 9.6 NODAL INFORMATION

The program also prints out the following quantities at each nodal point.

Stress Analysis

- Incremental displacements - the amount of deformation that occurred in the last increment
- Total displacements - the summation of the incremental displacements
- Total equivalent nodal forces (distributed plus point loads) - the total force applied to the model through distributed loads (pressures) and point loads
- Reaction forces at fixed boundary conditions
- Residual loads at nodal points that are not fixed by boundary conditions

NOTE

Each value listed above is given at each nodal point, unless you invoke the PRINT CHOICE or PRINT NODE option. If you invoke the TRANSFORMATION option, MARC prints the nodal information relative to the user-defined system, rather than the global coordinate system.

Reaction Forces

The program computes reaction forces based on the integration of element stresses. This is the only way to compute total reaction forces in a nonlinear analysis. Since such integration is only exact if the stresses are known at each integration point, the reaction forces are not printed if you use the CENTROID parameter.

In a nonlinear analysis, you should check that the reaction forces are in equilibrium with the external forces. If they are not in equilibrium, the analysis will be inaccurate, usually due to excessively large incremental steps. In most cases, the equilibrium is automatically ensured due to the convergence testing in MARC.

Residual Loads

The residual loads are a measure of the accuracy of the equilibrium in the system during analysis. This measure is very important in a nonlinear analysis and should be several orders of magnitude smaller than the reaction forces.

Dynamic Analysis

In a dynamic analysis, the program prints out the total displacement, velocity, and acceleration at each time increment.

Heat Transfer Analysis

In a heat transfer analysis, the program prints out nodal temperatures and nodal heat fluxes. If a thermo-electrical analysis is performed, the program prints nodal voltages as well.

Rigid Plastic Analysis

In steady-state rigid plastic analysis, the program prints nodal velocities.

Hydrodynamic Bearing Analysis

In a hydrodynamic bearing analysis, the program gives the mass flux at the nodal points.

Electrostatic Analysis

In an electrostatic analysis, the program prints the scalar potential and the charge.

Magnetostatic Analysis

In a magnetostatic analysis, the program prints the vector potential and the current.

Electromagnetic Analysis

In an electromagnetic analysis, the program prints out the vector and scalar potential, the current, and the charge.

Acoustic Analysis

In an acoustic analysis, the program prints the pressure and the source.

Contact Analysis

The standard output includes at the end of each increment a summary of information regarding each body. This information reports the increment's rigid body velocity, the position of the center of rotation, and the total loads on the body. These last values are obtained by adding the contact forces of all nodes in contact with the rigid body in cause. Deformable bodies being in equilibrium have no load reported.

Additional information can be obtained by means of the parameter PRINT,5. In such cases, all the contact activity is reported. Namely, every time a new node touches a surface, or separates from a surface, a corresponding message is issued.

Contact with rigid surfaces entails an automatic transformation. Displacement increments and reactions in the transformed coordinates - tangent and normal to the contact interface - are also reported for every node that is in contact in the usual MARC manner.

On the POST file, fictitious beam elements are added to represent the profiles of rigid surfaces, allowing finite element post processors to display rigid bodies both in the initial and subsequent positions in time.

A 9.7 PROGRAM MESSAGES

The messages provided by the program at various points in the output show the current status of the problem solution. Several of these messages are listed below.

- START OF INCREMENT x indicates the start of increment number x.
- START OF ASSEMBLY indicates the program is about to enter the stiffness matrix assembly.
- START OF MATRIX SOLUTION indicates the start of the solution of the linear system.
- SINGULARITY RATIO prints yyy, where yyy is an indication of the conditioning of the matrix. This value is typically in the range 10^{-6} to 1. If yyy is of the order of machine accuracy (10^{-6} for most machines), the equations may be considered singular and the solution unreliable.
- END OF MATRIX SOLUTION indicates the end of matrix decomposition.
- END OF INCREMENT x indicates the completion of increment number x.
- RESTART DATA at INCREMENT x ON TAPE 8 indicates that the restart data for increment number x has been written (saved) on Tape 8.
- POST DATA at INCREMENT x on TAPE y indicates that the post data for increment number x has been written (saved) on tape y.

In addition to these program messages, exit messages indicate normal and abnormal exists from the program. Table A 9.7-1 shows the most common exit messages.

Table A 9.7-1 MARC Exit Messages

Message	Meaning
MARC EXIT 3001	Normal exit.
MARC EXIT 3004	Normal exit.
MARC EXIT 12	End of plotting (normal exit).
MARC EXIT 13	Input data errors were detected by the program.
MARC EXIT 2004	Operator matrix (for example, stiffness matrix in stress analysis) has become non-positive definite and the analysis terminated.
MARC EXIT 3002	Convergence has not occurred within the allowable number of recycles.

Chapter 10 PLOTTING CAPABILITIES

The MARC program features extensive input and output plotting capabilities for checking data and displaying results. To invoke the plotting capability, insert a MESH PLOT parameter in the input deck. A flag on this parameter allows you to select pre-plotting, post-plotting, or both. It is recommended that both pre- and post-plots be created interactively using Mentat.

A 10.1 PLOT LAYOUT

Several options control the placement of the figure on the screen or paper and the labeling of the figure. These options are described in the sections below.

Viewpoint

To obtain plots of three-dimensional structures, it is necessary to enter a viewpoint. If this viewpoint is chosen close to the structure, a perspective projection will be obtained. If the viewpoint is far from the structure, an isometric projection will be obtained. This data is given in the PLOT TYPE option.

Scaling

The SCALING option controls the size of the physical plot.

Two SCALING options are available. Absolute scale factors can be input for the width and height of the plot, in inches-per-user dimensions. Also, a plot can be scaled to fit a frame by specifying first the width and then the height of the frame (in inches).

If a frame size is prescribed that is larger than is allowed by the plotting device, the result will be an incomplete or scrambled plot.

Frame Set

The FRAME SET option enables you to position different features of the plot within the plot frame. If this option is not included in your input, the defaults position these features. Table A 10.1-1 and Figure A 10.1-1 show the plot parameters of the plot. MARC provides default values for these dimensions.

Table A 10.1-1 Plot Parameters

Parameter	Variable
Left margin	BMARGN(1)
Bottom margin	BMARGN(2)
Right margin	TMARGIN(1)
Top margin	TMARGN(2)
X coordinate of axis	AXX(1)
Y coordinate	AXX(2)
X coordinate of first contour descriptor	CONTX(1)
Y coordinate of first contour descriptor	CONTX(2)
X coordinate of title	TITX(1)
Y coordinate of title	TITX(2)

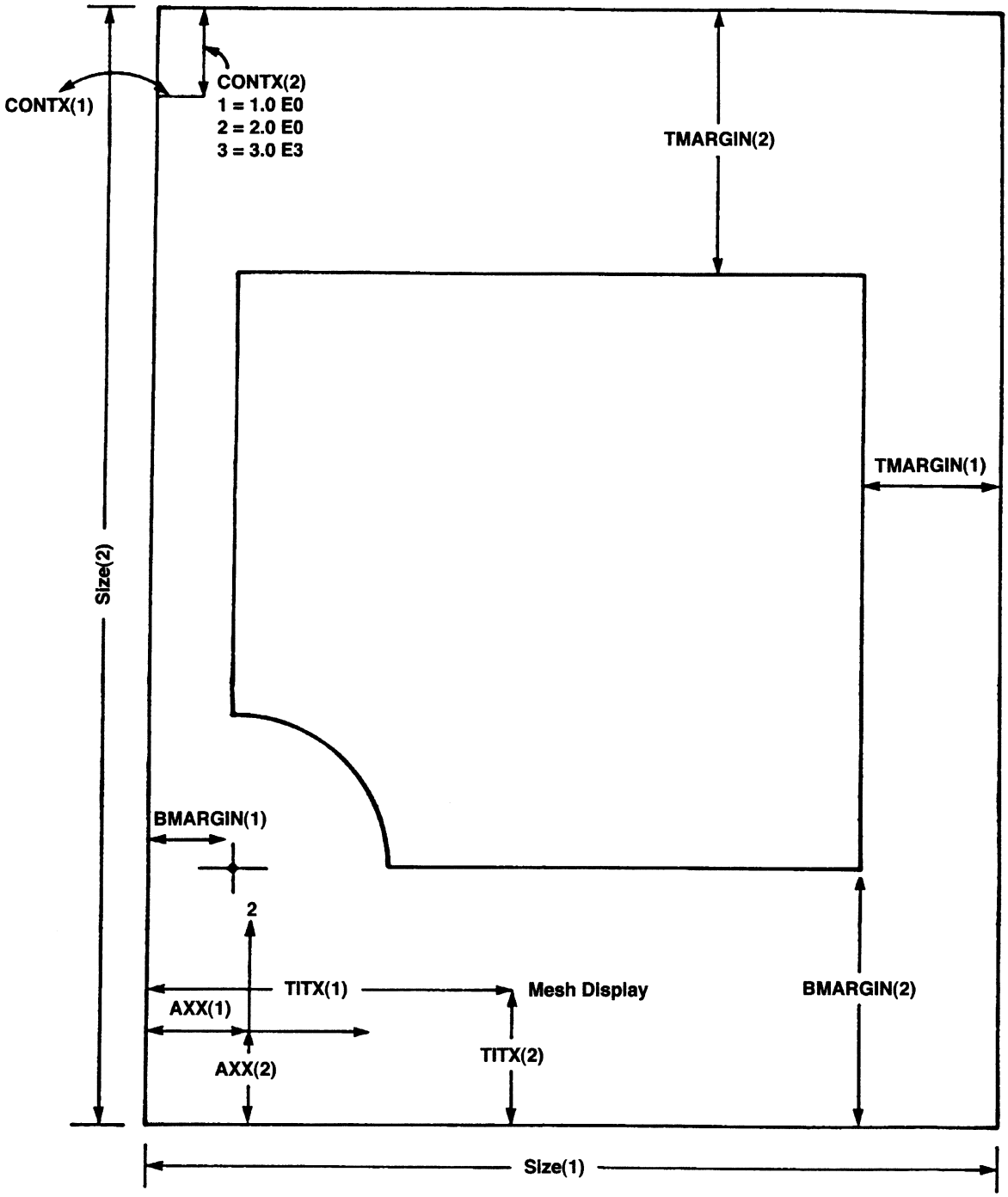


Figure A 10.1-1 Frame Set Parameters

Sectioning

Sectioning allows you to view a portion of the mesh and to zoom in on an area of the mesh to observe greater detail. There are three SECTIONING options; only one can be used in any one plot.

1. Plots the elements included in a list of element numbers you have selected.
2. Allows you to select plots by element type and plots all elements of the type you select.
3. Allows you to plot elements within a defined physical region. With this option, you define a frame by giving the minimum and maximum coordinates of the region to be plotted. Any element whose centroid lies in this region will be plotted (see Figure A 10.1-1, Figure A 10.1-2, and Figure A 10.1-3).

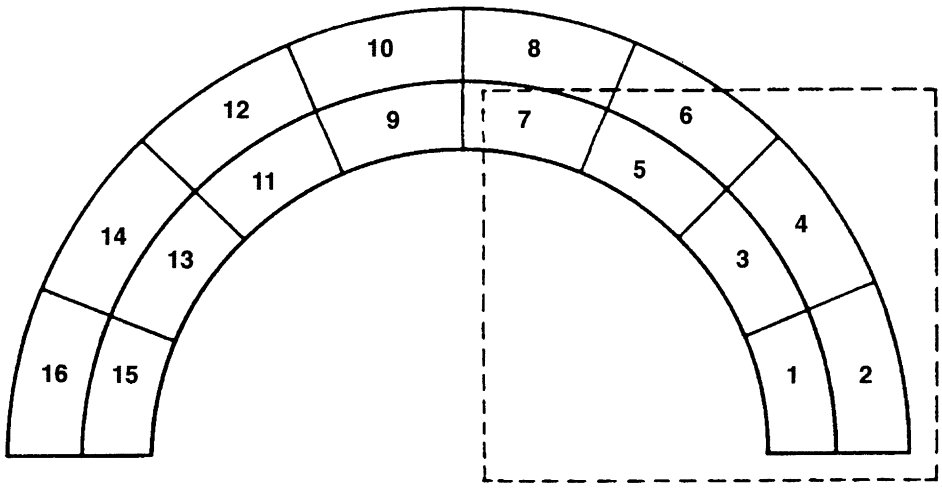


Figure A 10.1-2 Mesh Plot Before Sectioning

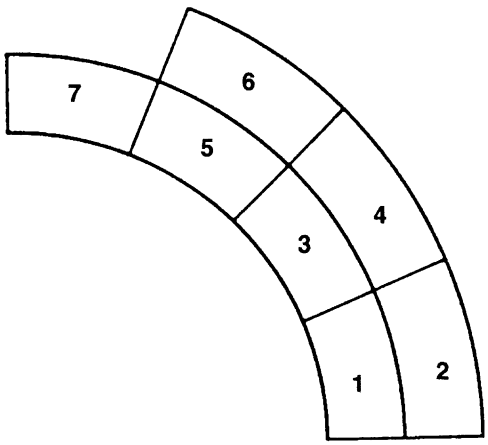


Figure A 10.1-3 Mesh Plot After Sectioning

ELEMENT IDENT, NODE IDENT, and TITLE Options

The ELEMENT IDENT option allows you to plot the element numbers on the mesh and prescribe the height of the characters to be used as element numbers. The NODE IDENT option allows you to plot the nodal numbers on the mesh and to prescribe the height of the characters to be used as nodal number. In the TITLE option, you can give the title with the mesh and prescribe the height of the characters for the title.

HIDDEN Option

The HIDDEN option is designed to remove internal element boundaries. You can use it for two-dimensional mesh plots and all contour plots (see Figure A 10.1-4 and Figure A 10.1-5), but should not use if for displaying three-dimensional meshes. You can display three-dimensional meshes with hidden lines removed using the MESH3D program.

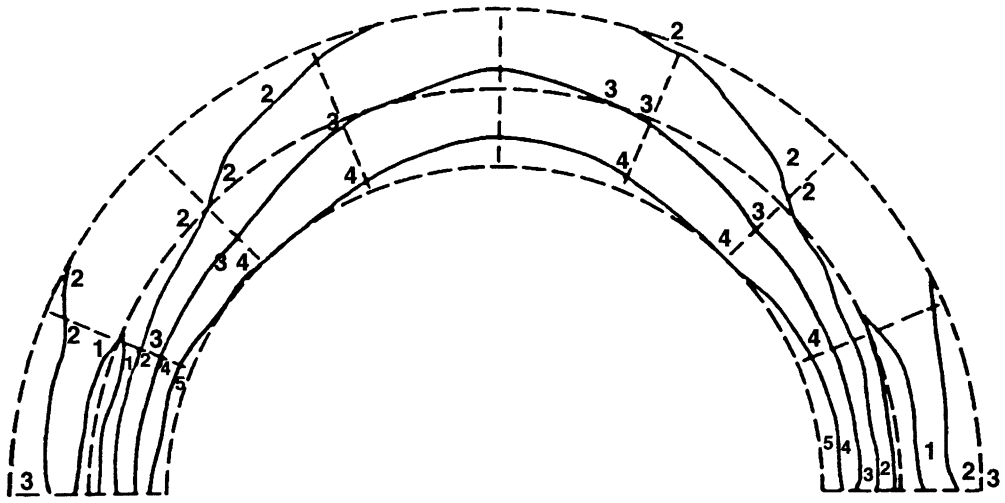


Figure A 10.1-4 Contour Plot without HIDDEN Option

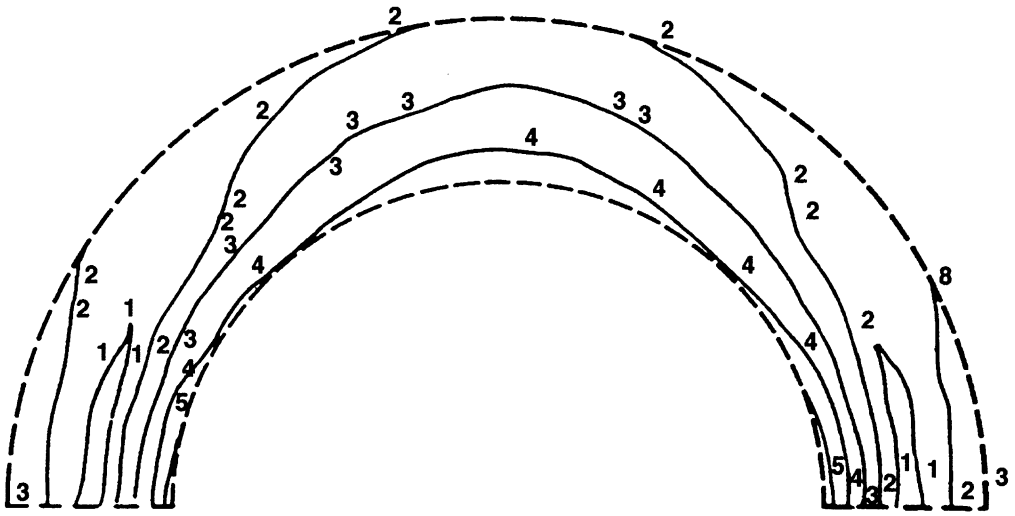


Figure A 10.1-5 Contour Plot with HIDDEN Option

A 10.2 PRE-PLOTTING

The program provides pre-plotting for mesh display so that you can view the mesh and check the accuracy and suitability of your model (see Figure A 10.2-1 and 10.2-2). As discussed above, the SECTIONING option allows you to plot various parts of the model separately, making it easy to examine modeling details, which is particularly important for shells or three-dimensional solid elements. You also have the option to display element and/or node numbers for identification. For curved elements such as the higher order solid elements or shell elements, the actual curve fitted by the program through your nodal coordinate data is displayed. You can also plot curved elements with straight lines by invoking the STRAIGHT option. The SHRINK option reduces the size of each element so that the connectivity may be verified.

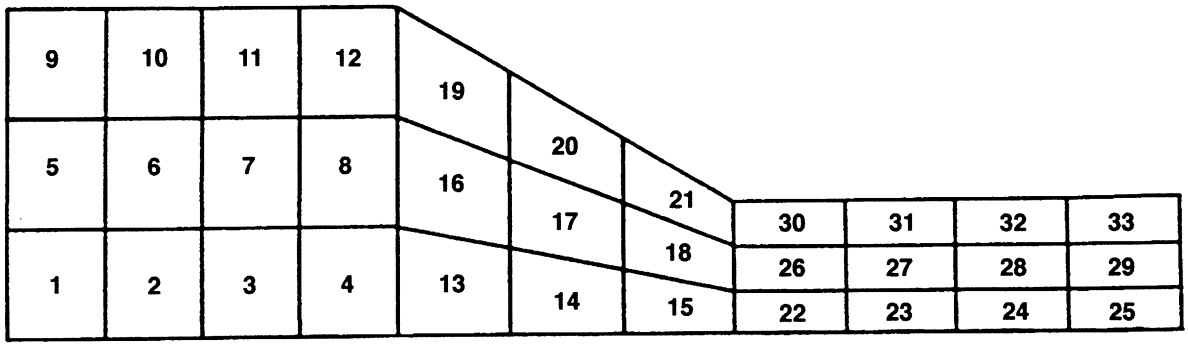


Figure A 10.2-1 Mesh Plot with Element Identification

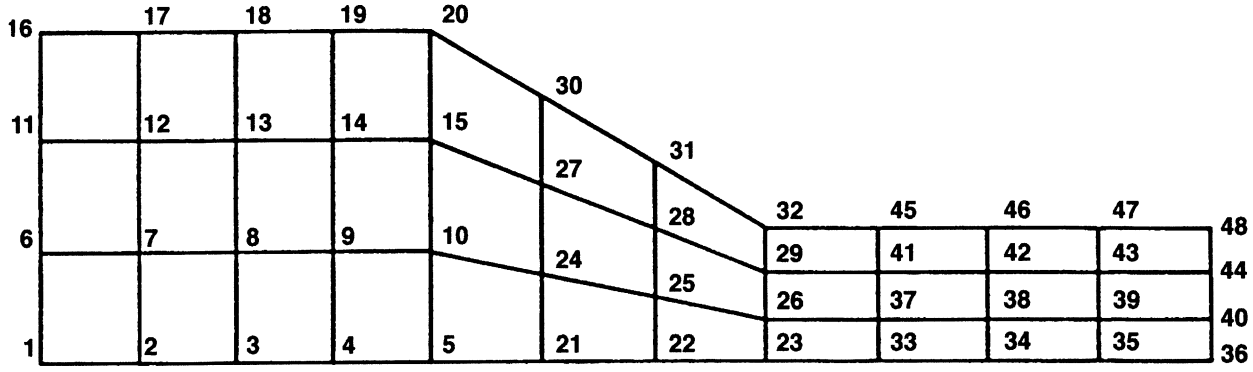


Figure A 10.2-2 Mesh Plot with Node Identification

A 10.3 POST-PLOTTING

MARC provides post-plotting capabilities so that you may display the results of the analysis for quick interpretation. This graphical display is easier to review than the printed output, which may be very lengthy.

The post-plotting capability has six main features:

- displaced position plots
- contour plots
- principal value plots
- vector plots
- linear plots
- time/history plots in MARC PLOT

The first five concepts are discussed below. For a discussion of time/history plots with MARC PLOT, see Section 10.5.

Displaced Position Plots

By superimposing a (magnified) displaced position plot in solid lines over a plot of the original mesh in dashed lines, displaced position plots allow you to compare the structure after deformation to the structure as it appeared before deformation. Figure A 10.3-1 shows an example of a deformed mesh plot. This feature allows you to examine buckling modes or dynamic mode shapes and velocity or acceleration profiles. You can prescribe the magnification factor; if you do not provide the magnification factor, the program will automatically choose one so that the maximum displacement will correspond to about 0.05 of the size of the plot. You may use this option in combination with any other options except the CONTOURS, LINEAR, or VECTOR options.

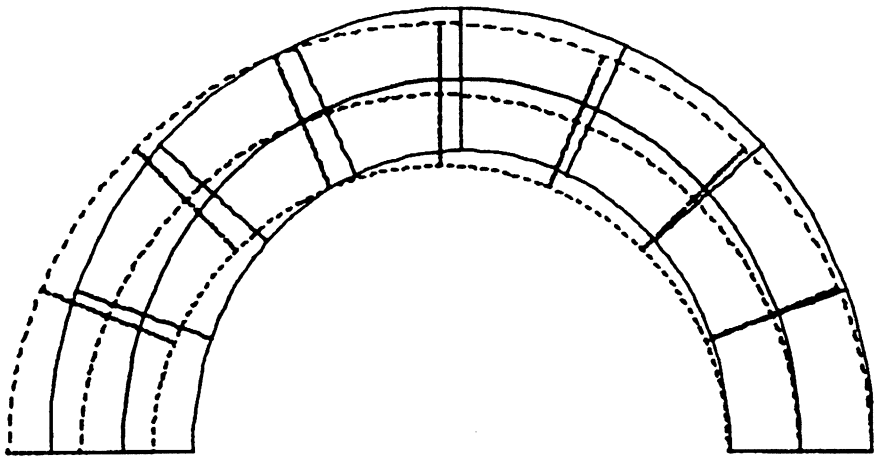


Figure A 10.3-1 Deformed Mesh Plot

Contour Plots

Contour plots allow you to view the variation in a particular parameter over the structure. The program enables contouring of any variable over any surface or solid element type. You can choose the variable to be plotted, the contour range, and the number of contours used to span the range. You can plot all contours, but eliminate from the plot the numbering of certain contour levels, and you can also eliminate all numbering by choosing a starting contour level number that is greater than the total number of contour levels. You can plot contours either relative to the original mesh geometry or relative to the current (displaced) geometry. For solid, three-dimensional elements, you can plot the contours on a slice through the element. For shell, plate, or three-dimensional elements, you must define the viewpoint (see PLOT TYPE).

The MARC program produces a contour plot with the mesh shown in dashed lines. Because straight-line contouring is used, meshes containing higher-order (curved-edge) elements are stylized to show straight-line element edges for consistency. You may use the HIDDEN LINE option to eliminate the internal element boundaries. An example of a contour plot with the internal lines eliminated can be seen in Figure A 10.3-2.

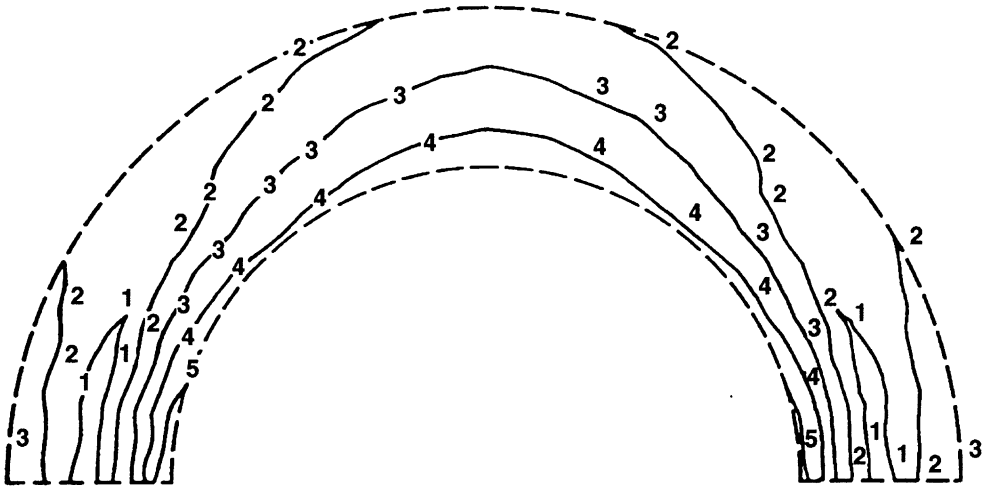
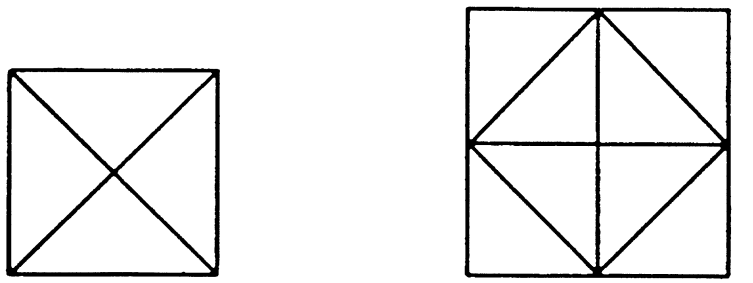


Figure A 10.3-2 Contour of von Mises Stress

Once a section is defined, the contour plot is generated as follows:

Nodal values of element quantities are computed by extrapolating the integration point values of the chosen variable to the nodes of the element. A weighted average is calculated at each node, the weight being the angle subtended at the node by each element contributing to the node. Values are averaged only for elements in the chosen section if you invoke the SECTIONING option. This averaging is physically unreasonable if there are discontinuities in the material or element type. If a discontinuity is present, you should use SECTIONING to avoid averaging across the discontinuity. You can print the average nodal values with the CONTOUR PRINT option.

Contouring is performed by subdividing elements into linear triangles (see Figure A 10.3-3). First-order quadrilaterals (Element Types 3, 10, 11, etc.) are divided into four triangles. Second-order quadrilaterals (Element Types 22, 26, 27, 28, etc.) are divided into eight triangles. Contours are drawn in each triangle, assuming linear variation between values of the variable at the corners of the triangle obtained by the averaging process discussed above.



First Order Quadrilateral

Second Order Quadrilateral

Figure A 10.3-3 Sub-triangles Used for Contouring Quadrilateral Elements

To allow contouring of user-defined variables, you should invoke user subroutine PLOTV in conjunction with either code 19 or a negative code in the CONTOURS option of the mesh display plotting. Volume D of the MARC User's Manual describes this routine.

Layer or Slice Specification

To generate a contour plot for doubly curved shell elements, you must choose one of the layers the program uses through the thickness of the shell as the layer for which the variable is to be contoured. For a solid element, you must choose a slice in the element on which contours will be plotted. Figure A 10.3-4 and Figure A 10.3-5 show the possible slices for the two types of three-dimensional solid elements in the program. Slice specification is related to the face (or node) ordering of the element. This procedure is convenient particularly for generated meshes. The slices pass through the integration points of the elements. You can use either the CONTOUR option or the SLICE option to specify the slice number. Once you define by choosing a section, or a layer, the contour plot is generated.

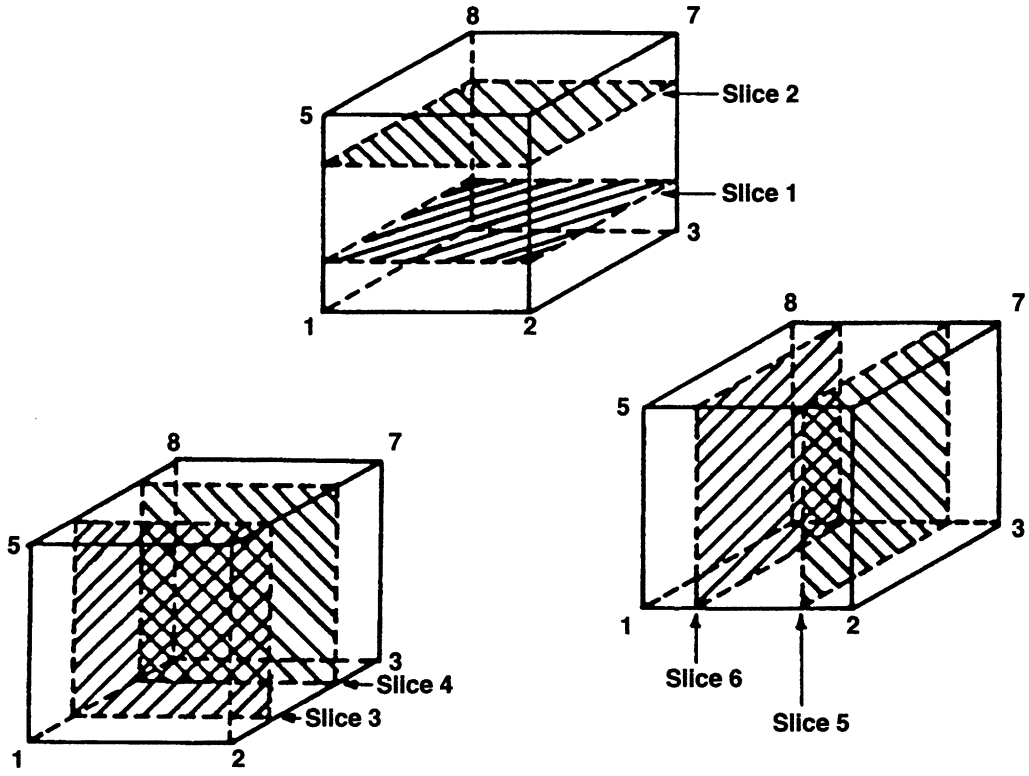


Figure A 10.3-4 Slice Specification for the 8-Node Brick (Element 7) and the 20-Node Bricks with Reduced Integrations (Element Types 57, 61)

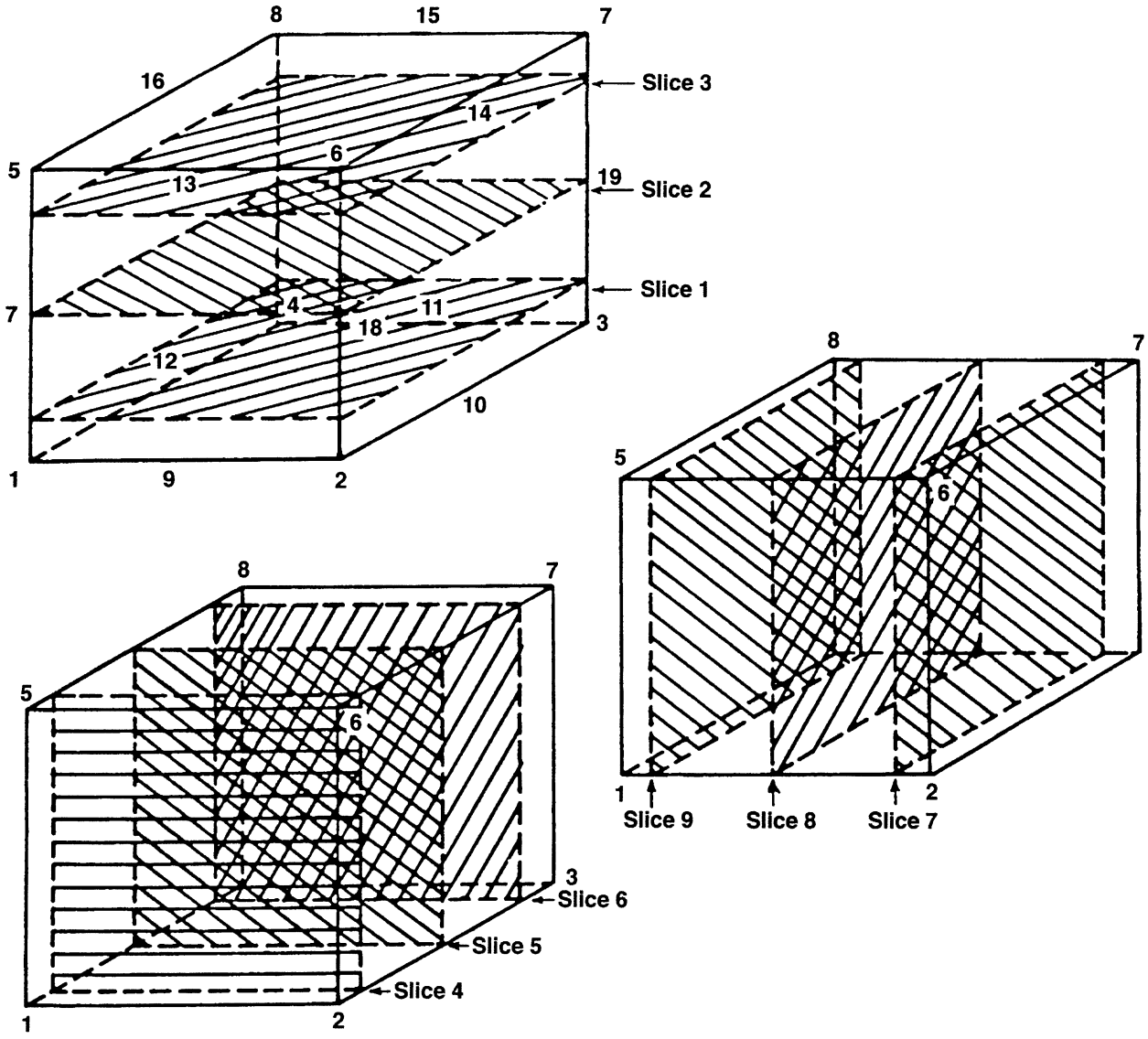


Figure A 10.3-5 Slice Specification for the 20-Node Brick (Elements 21, 35)

Principal Value

You can obtain principal value plots for planar elements. Principal stresses are computed from the stress components and are plotted as straight lines. The magnitude of the principal stress is represented by the length of the displayed line. The direction of the principal stress in the global coordinate system is represented by the orientation of the line. Figure A 10.3-6 gives an example of principal value plots.

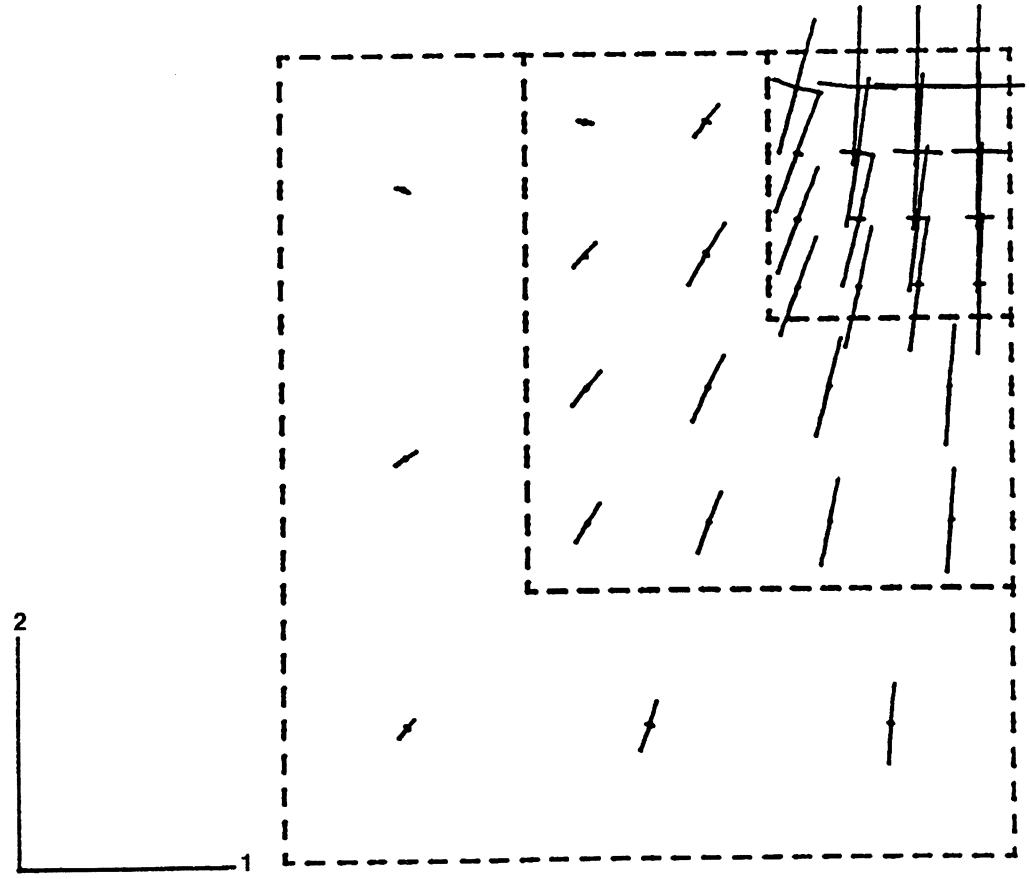


Figure A 10.3-6 Principal Stress Directions and Magnitudes

Vector Plots

The VECTOR PLOTS option provides a method for displaying a calculated quantity as a vector superimposed on the original mesh. Figure 10.3-7 shows a vector plot of mass flow. The vectors are drawn at the nodes of the mesh so that the size of the vector represents the magnitude of the quantity plotted. You can plot all nodal and element quantities with the VECTOR PLOT option, which can be used in combination with any other option except the CONTOURS, DISPLACED, or LINEAR options.

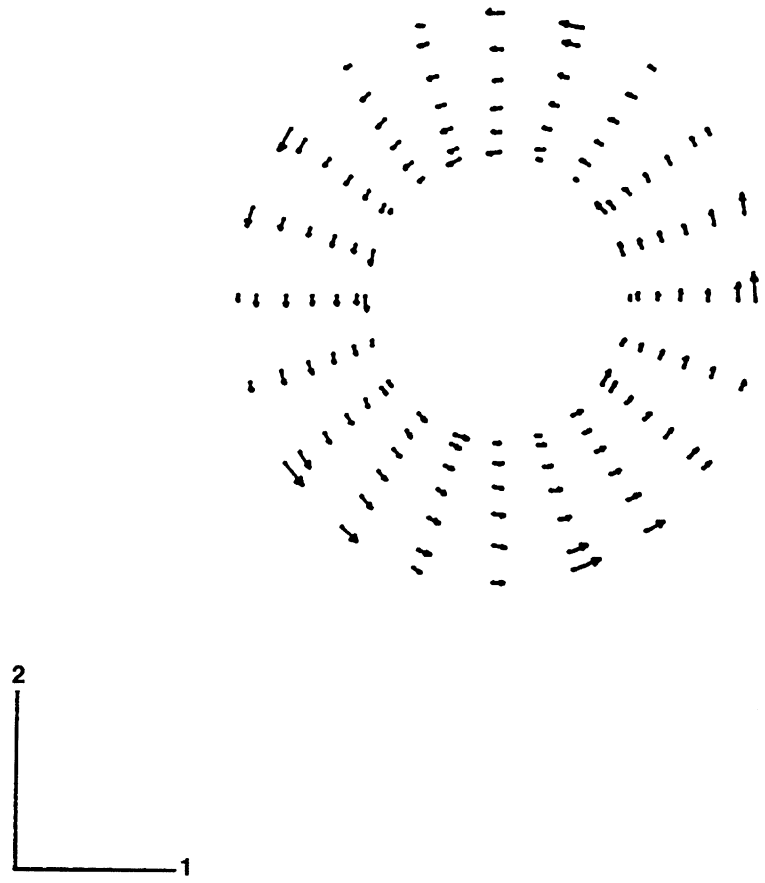


Figure A 10.3-7 Vector Plot of Mass Flow

Linear Plots

The LINEAR PLOT option provides a graphic display of output quantities of linear elements such as beams and trusses. This option allows the construction of beam moment diagrams. See Figure A 10.3-8. An X-Y plot is made where the X-axis represents the distance along the beam and the Y-axis gives the magnitude of the variable. To construct the plot, the X and Y values are calculated at every integration point of each active element. You can use the LINEAR PLOT option in combination with other options, except CONTOURS, DISPLACED, VECTOR, SLICE, or HIDDEN.

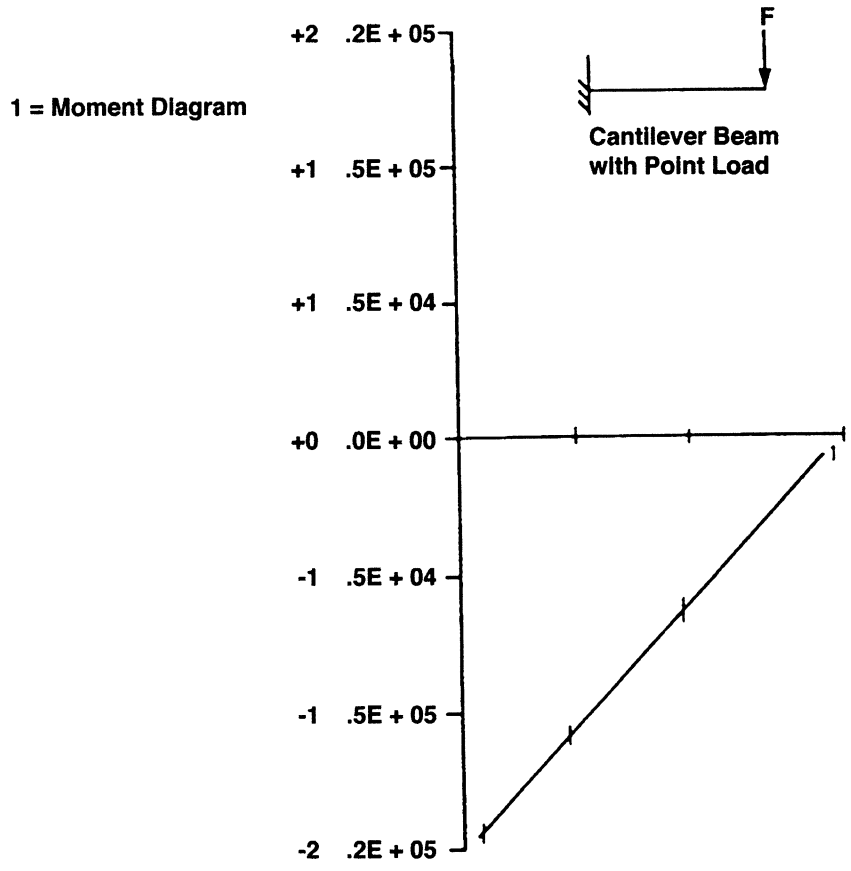


Figure A 10.3-8 Beam Moment Diagram

Restart

You can generate plots from a previous analysis which has been stored on a RESTART file by using the MESH PLOT, 0 option. This option indicates that you only intend to do post-processing. Use the RESTART, 2 option to indicate that you intend to read a restart tape and the first increment number that is to be read from the tape. Place your plot data cards immediately following the END OPTION.

You can use the POSITION option to read and display additional increments.

A 10.4 MESH3D PROGRAM

The MESH3D program can display a three-dimensional structure with the hidden lines removed in either a job that creates the mesh or a separate job. If all the element edges are to be shown, you should use the plot facilities in the stress program.

The MESH3D program can make hidden-line plots of three-dimensional brick, shell, and membrane structures (see Figure A 10.4-1). The plot can be scaled and a title can be included. To activate the display capability, you must use the PLOT option.

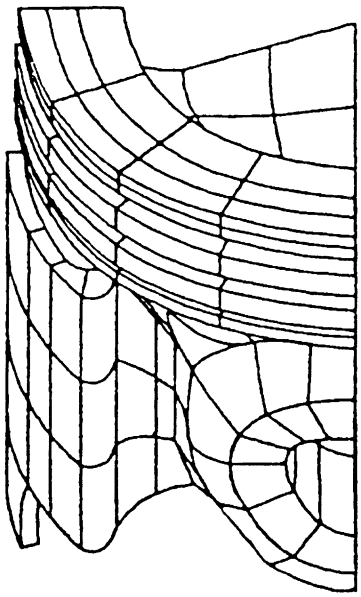


Figure A 10.4-1 Hidden Line Plot Using MESH3D

Plot Block

The PLOT BLOCK operation produces an isometric hidden line plot of the previously generated block(s) or mesh. You may want to plot after a MAP, MERGE, or JOIN operation. You can specify the orientation and the size of the plot. The hidden line algorithm can take considerable machine time, and you should allow for this in your estimates of computer processing time.

Stand-alone Plotting Back

The stand-alone plotting (MESH) option plots either a previously generated mesh or joins it to other blocks created in a MESH3D run. Any blocks that have been written on tape using the DATA operation can be plotted in a later run if you give the exact node and element totals as those printed in the listing of the corresponding DATA block.

A 10.5 MARC-PLOT PROGRAM

MARC-PLOT is a post-processing program that provides plots of history and variable versus variable. As input, MARC-PLOT uses the binary POST tape written by MARC. Any number of variables can be plotted together as functions of time, increment number, or another variable. You can generate several plots in a single job. A CONTINUE option completes each plot definition, and an END PLOT option ends the entire job. If the program detects data errors in any plot definition, it ignores that plot but still produces all other plots.

The HISTORY option controls the plot type. Depending on which parameter you enter in the option, all variables are plotted in the same frame against time, increment number, or against the first variable.

You can select nodal quantities, such as displacement and velocity, using the NODE options and you can select element quantities, such as stress and strain, using the ELEMENT option.

The plots produced by the MARC-PLOT program appear in a frame. If the plotted quantities exceed the frame size, true line-clipping is performed. The default scaling procedure uses the available space most effectively. However, you can also maintain complete control of the scaling procedure. Figure A 10.5-1 shows a plot generated with the MARC-PLOT program.

The input/output unit allocations for MARC-PLOT are:

4 = binary neutral plot file
5 = input data
6 = printed output
16 = POST tape (binary) as written by MARC
19 = POST tape (formatted) as written by MARC
34 = formatted neutral plot file

Create the POST tape using the POST model definition option. You can write different nodal quantities to the tape, depending on your analysis. Generally, write this tape to unit 16 if you choose binary format, or unit 19 if you choose formatted (ASCII) format.

Use the RESTART option if you wish to make a continuous post tape. In this case, the old post tape is unit 17 or 20.

MARC-PLOT can use either binary or formatted post tapes. Mentat can use either binary or formatted post tapes.

The PLDUMP program, discussed in Volume D, can be used to read this file.

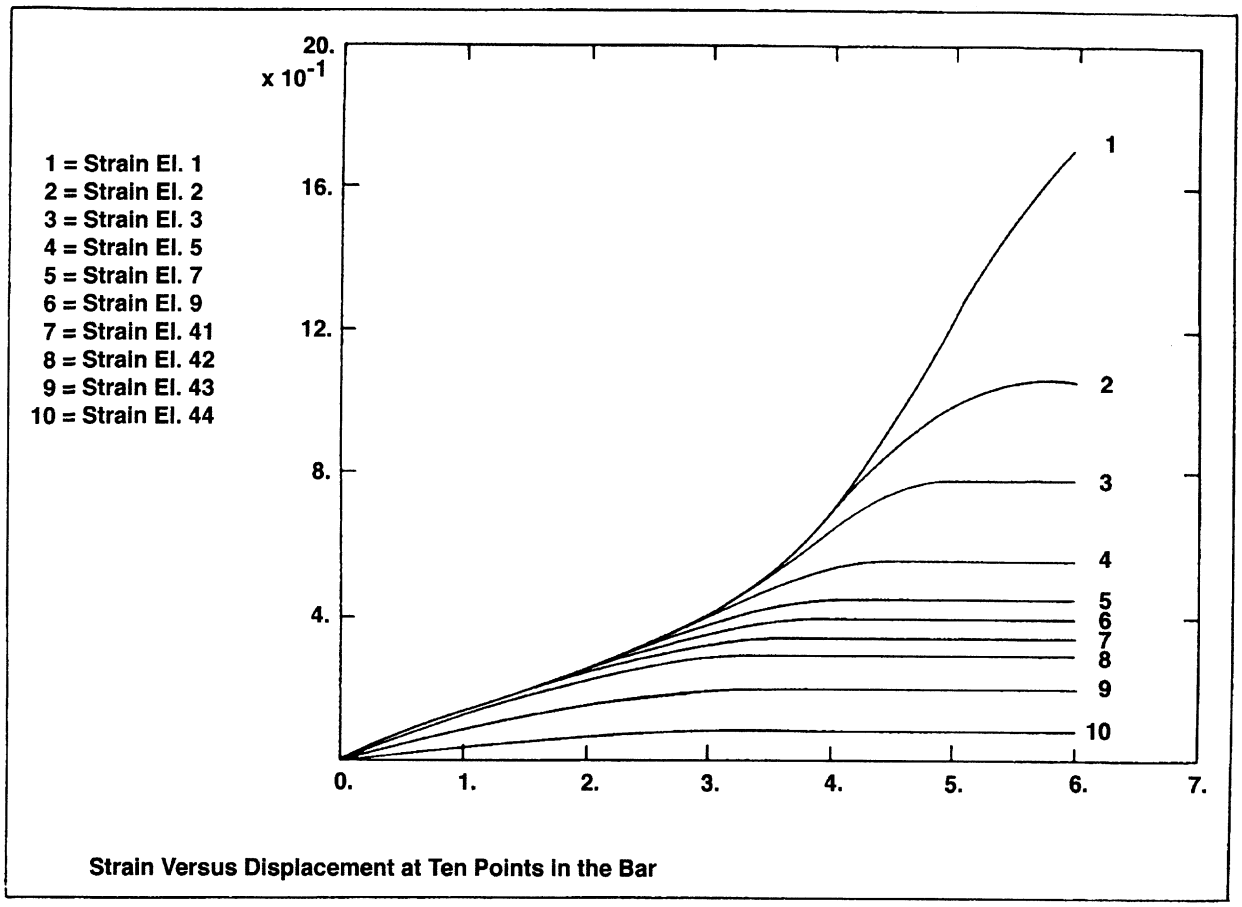


Figure A 10.5-1 Example of MARC-PLOT

A 10.6 NEUTRAL PLOT

Neutral plot refers to a file, as well as to a program that uses that file. In many installations where the plot is displayed by local terminals or plotters, a plot file is created in two steps. The generating program (MARC, MESH3D, or MARC-PLOT) writes the neutral file. Subsequently, the NEUTRAL PLOT program converts this neutral file to an actual plot. There are two advantages to creating the plot file this way.

1. You can display a single plot on a variety of terminals and plotters.
2. You can display the plot with a different computer than the one that generated the neutral plot file.

Figure A 10.6-1 illustrates how a plot is created and then displayed. For example, MARC could be run on a CRAY computer while the neutral plot program is run on a CDC or VAX.

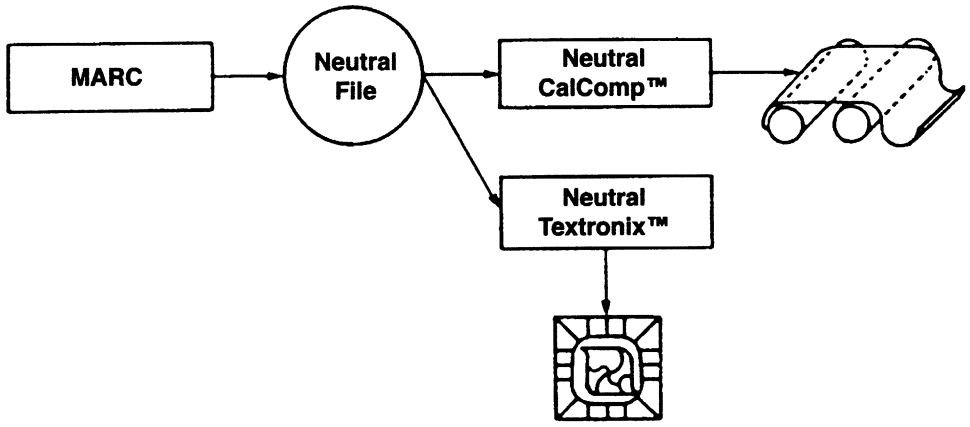


Figure A 10.6-1 Schematic of Neutral Plot Operations

A 10.7 MENTAT PROGRAM

To perform post-processing using Mentat, you must first use the MARC program to create a POST tape. Only those element quantities written to the POST tape can be processed with Mentat. Nodal quantities are automatically written to the POST tape.

The PLOT subprocessor generates the graphics output. You can make pre- and post-processing plots of the model or of a subset of the model. (For further information on the Mentat program, see the *Mentat Reference Manual*).

Appendix INTRODUCTION TO THE FINITE ELEMENT TECHNOLOGY IN MARC

A A.1 GOVERNING EQUATIONS OF VARIOUS STRUCTURAL PROCEDURES

The MARC program was developed on the basis of the displacement method. The stiffness methodology used in the MARC program addresses force-displacement relations through the stiffness of the system. The force-displacement relation for a linear static problem can be expressed as

$$Ku = f \tag{A A.1-1}$$

where K is the system stiffness matrix, u is the nodal displacement, and f is the force vector.

Assuming that the structure has prescribed boundary conditions both in displacements and forces, the governing Equation A A.1-1 can be written as

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} \tag{A A.1-2}$$

u_1 is the unknown displacement vector, f_1 is the prescribed force vector, u_2 is the prescribed displacement vector, and f_2 is the reaction force. After solving for the displacement vector u, the strains in each element can be calculated from the strain-displacement relation in terms of element nodal displacement as

$$\varepsilon_{el} = \beta u_{el} \tag{A A.1-3}$$

The stresses in the element are obtained from the stress-strain relations as

$$\sigma_{el} = L\varepsilon_{el} \tag{A A.1-4}$$

where σ_{el} and ε_{el} are stresses and strains in the elements, and u_{el} is the displacement vector associated with the element nodal points; β and L are strain-displacement and stress-strain relations, respectively.

In a dynamic problem, the effects of mass and damping must be included in the system. The equation governing a linear dynamic system is

$$M\ddot{u} + D\dot{u} + Ku = f \tag{A A.1-5}$$

where M is the system mass matrix, D is the damping matrix, (Equation A A.1-6) is the acceleration vector, and u is the velocity vector. The equation governing an undamped dynamic system i

$$M\ddot{u} + Ku = f \tag{A A.1-6}$$

The equation governing undamped free vibration is

$$M\ddot{u} + Ku = 0 \qquad \text{(A A.1-7)}$$

Natural frequencies and modal shapes of the structural system are calculated using this equation.

$$K\phi - \omega^2 M\phi = 0 \qquad \text{(A A.1-8)}$$

The equations governing some other procedures are similar. For example, the governing equation of transient heat transfer analysis is

$$C\dot{T} + KT = Q \qquad \text{(A A.1-9)}$$

where C is the heat capacity matrix, K is the thermal conductivity matrix, Q is the thermal load vector (flux), T is the nodal temperature vector, and t is the time derivative of the temperature. Equation A A.1-9 reduces to

$$KT = Q \qquad \text{(A A.1-10)}$$

for the steady-state problem. Note that the equation governing steady-state heat transfer (Equation A A.1-10) and the equation of static stress analysis (Equation A A.1-1) take the same form. Similarly, the hydrodynamic bearing problem is analogous to a steady-state heat transfer problem. This problem is governed by an equation similar to Equation A A.1-10.

The matrix equation of the electrical problem in coupled thermo-electrical analysis is

$$\rho(T)\, v = I \qquad \text{(A A.1-11)}$$

The equation governing the thermal problem is:

$$C(T)\dot{T} + K(T)\, T = Q + Q^{\varepsilon} \qquad \text{(A A.1-12)}$$

in both Equation A A.1-11 and Equation A A.1-12, $\rho(T)$ is the temperature-dependent electrical-conductivity matrix, I is the nodal-current vector, C(T) is the temperature-dependent, heat-capacity matrix, and $\kappa(T)$ is the thermal-conductivity matrix. T is the nodal temperature vector, Q is the heat-flux vector, and Q^e is the internal heat-generation vector that results from the electrical current. Electrical and thermal problems are coupled through $\rho(T)$ and Q^e.

The matrix equations for the thermal-mechanical problem are as follows:

$$M\ddot{u} + D(T)\,\dot{u} + K(T)\, u = f \qquad \text{(A A.1-13)}$$

$$C(T)\dot{T} + \kappa(T)\, T = Q + Q^{I} \qquad \text{(A A.1-14)}$$

In Equation A A.1-13 and Equation A A.1-14 the damping matrix D, stiffness matrix K, heat-capacity matrix C and thermal-conductivity matrix κ are all dependent on temperature. Q^I is the internal heat generated due to inelastic deformation. The coupling between the heat

transfer problem and the mechanical problem is due to the temperature-dependent mechanical properties and the internal heat generated. If an updated Lagrangian analysis is performed, $\mathbf{K}$ and κ are dependent upon prior displacement.

The governing equations described above are either sets of algebraic equations (Equation A A.1-1, Equation A A.1-10, and Equation A A.1-11), or sets of ordinary differential equations (Equation A A.1-5 through Equation A A.1-8 and all equations through Equation A A.1-4). The time variable is a continuous variable for the ordinary differential equations. Select an integration operator (e.g., Newmark-Beta, Houbolt, or central difference for dynamic problems, and backward difference for heat transfer) to reduce the set of differential equations to a set of algebraic equations. The final form of governing equations of all analysis procedures is, therefore, a set of algebraic equations.

A A.2 SYSTEM AND ELEMENT STIFFNESS MATRICES

The previous section presented system matrices assembled from element matrices in the system. For example, the system stiffness matrix K is expressed in terms of the element stiffness matrix K_i^{el} as

$$K = \sum_{i=1}^{N} K_i^{el} \qquad \text{(A A.2-1)}$$

where N is the number of elements in the system. The system stiffness matrix is a symmetric-banded matrix. See Figure A A.2-1 for a schematic of the assemblage of element matrices. When the element-by-element iterative solver is used, the total stiffness matrix is never assembled.

The element stiffness matrix can be expressed as

$$K^{el} = \int_{v^{el}} \beta^T L \beta dv^{el} \qquad \text{(A A.2-2)}$$

where v^{el} is the volume of the element, β is the strain-displacement relation, and L is the stress-strain relation.

$$\varepsilon = \beta u \qquad \text{(A A.2-3)}$$

$$\sigma = L\varepsilon \qquad \text{(A A.2-4)}$$

The mass matrix can be expressed as

$$M^{el} = \int_{v^{el}} N^T \rho N dv \qquad \text{(A A.2-5)}$$

The integration in Equation A A.2-2 is carried out numerically in MARC, and is dependent on the selection of integration points. The element stiffness matrix can be fully or under integrated. The mass matrix is always fully integrated.

NOTE

The β matrix is directly associated with the shape functions and geometry of each element. Shape functions associated with different element types are explained in Volume B. Stress-strain relations L are discussed in Chapter 6 of this manual.

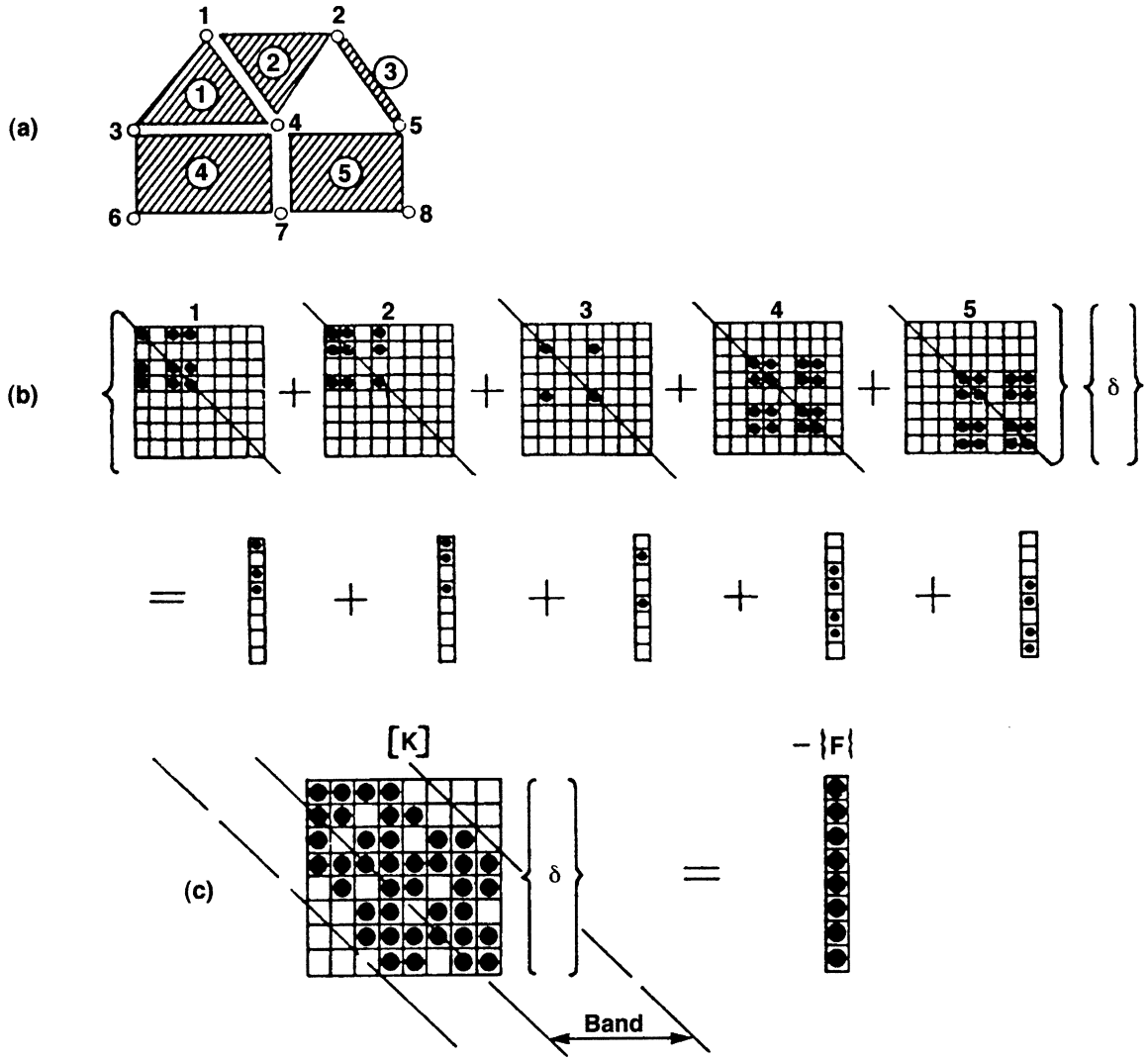


Figure A A.2-1 Schematic of Matrix Assemblage

A A.3 LOAD VECTORS

The nodal force vector f in Equation A A.1-1 includes the contributions of various types of loading.

$$f = f_{point} + f_{surface} + f_{body} + f^* \qquad \text{(A A.3-1)}$$

where f_{point} is the point load vector, $f_{surface}$ is the surface load vector, f_{body} is the body (volumetric) load vector, and f* represents all other types of load vectors (for example, thermal and creep strains, and initial stress).

The point load is associated with nodal degrees-of-freedom and can be added to the nodal force vector directly. Equivalent nodal force vectors $f_{surface}$, f_{body} must be calculated from the distributed (surface/volumetric) load first and then added to the nodal force vector. In MARC, the computation of equivalent nodal forces is carried out through numerical integration of the distributed load over the surface area of volume to which the load is applied. This may be expressed as

$$f_{surface} = \int_A N^T p dA \qquad \text{(A A.3-2)}$$

$$f_{body} = \int_v N^T p dV \qquad \text{(A A.3-3)}$$

where p is the pressure.

Figure A A.2-1 shows the assemblage of the nodal force vector.

A A.4 NONLINEAR EFFECTS

This section addresses three aspects of nonlinear effects.

- Material nonlinearities
- Geometric nonlinearities
- Nonlinear boundary conditions

Material Nonlinearities

P. V. Marcal and I. P. King first documented an accurate, nonlinear, finite-element solution technique for modeling plasticity. This formulation accounts for the transition region, where the material response is elastic at the start of the incremental step and plastic at the end of the step. Refinements of the formulation were based on numerically integrated elements that allow different material responses to be present for each integration point.

The mean stiffness first described by J. R. Rice and D. M. Tracy is employed in the MARC program for elastic-plastic problems. The mean stiffness approach guarantees that the final stress state remains on the yield surface. This approach also guarantees unconditional stability. The conventional von Mises and Mohr-Coulomb yield surfaces are available in MARC. In addition, the damage model may be used to modify the von Mises yield criterion, or the powder metallurgy, or soil model may be used.

Concepts similar to those in plasticity analysis apply to the modeling of creep behavior. Conventional creep behavior exhibits creep-strain increments in the direction of the surface normal to the von Mises stress surface. In addition, purely volumetric creep may be modeled.

The MARC code allows nonlinear elasticity through the Mooney-Rivlin, Ogden, or Foam formulation. The code includes a special group of elements with which incompressible behavior can be modeled.

Material models are discussed in detail in Chapter 6.

Geometric Nonlinearities

Virtually all elements in the MARC code can be used for geometrically nonlinear analysis. You can choose either the total or updated Lagrangian formulation for geometrically nonlinear problems. The large displacement formulation can also be used to obtain buckling estimates.

A formulation for large strains and large rotations is also available, which can be applied to metal-forming problems.

Nonlinear Boundary Conditions

Nonlinear boundary conditions include contact problems, as well as nonlinear support conditions. The gap-friction element, based on the imposition of kinematic constraint, allows the modeling of contact and friction problems. The CONTACT option allows both contact between deformable bodies and between a deformable-body and a rigid surface. The nonlinear SPRING and FOUNDATION options are used to simulate nonlinear supports.

A A.5 SOLUTION PROCEDURES

In a nonlinear problem, the system equation must be solved incrementally. The governing equation of the system can be expressed, in an incremental form, as

$$Kdu = df \qquad \text{(A A.5-1)}$$

where du and df are incremental displacement and force vectors, respectively.

There are five solution procedures available in MARC for the solution of nonlinear equations: full Newton-Raphson method, modified Newton-Raphson method, strain correction method, a secant method, and a direct substitution method.

The full Newton-Raphson method is the default in the MARC program. This method has quadratic convergence properties. This means that, in subsequent iteration, the relative error decreases quadratically. If material nonlinearities are present, some approximations slow down convergence. The full Newton-Raphson method provides good results for most nonlinear problems, but is expensive for large, three-dimensional problems, when the direct solver is used. This computational problem is less significant when the iterative solvers are used.

The modified Newton-Raphson method is similar to the full Newton-Raphson method, but does not reassemble the stiffness matrix during iteration. This method costs less per iteration, but the number of iterations may increase substantially over that of the full Newton-Raphson method. If gap-friction nonlinearities occur, reassembly cannot be avoided. The modified Newton-Raphson method is effective for large-scale, mildly nonlinear problems. When the iterative solver is employed, simple back substitution is not possible, making this process ineffective. Use the full Newton-Raphson method instead.

The strain correction method is a variant of the full Newton method. This method uses a linearized strain calculation, with the nonlinear portion of the strain increment applied as an initial strain increment in subsequent iterations and recycles. This method is appropriate for shell and beam problems in which rotations are large, but membrane stresses are small.

The secant method is similar to the modified Newton-Raphson method in that the stiffness matrix is calculated only once per increment. The residual is modified to improve the rate of convergence. When the iterative solver is employed, simple back substitution is not possible, making this process ineffective. Use the full Newton-Raphson method instead.

The direct substitution method is only used in Eulerian (R-P Flow) analysis. Here the last solution is used as a trial solution for the next iteration.

The basis of the Newton-Raphson method in structural analysis is the requirement that equilibrium must be satisfied. Consider the following set of equations:

$$I(u) = f \qquad \text{(A A.5-2)}$$

$$K(u)\,du = f - I(u) \qquad \text{(A A.5-3)}$$

where u is the nodal-displacement vector, f is the external nodal-load vector, I is the internal nodal-load vector (following from the internal stresses), and K is the tangent-stiffness matrix. The internal nodal-load vector is obtained from the internal stresses as

$$I = \int_v \beta^T \sigma dv \qquad \text{(A A.5-4)}$$

In this set of equations, both I and I< are functions of u. In many cases, f is also a function of u (for example, if f follows from pressure loads, the nodal load vector is a function of the orientation of the structure).

The equations suggest that use of the full Newton-Raphson method is appropriate.

Suppose that the last obtained approximate solution is termed $\delta u^{(i)}$, where (i) indicates the iteration number. Equation A A.5-2 and Equation A A.5-3 may then be written as

$$K(u^{(i)})\,\delta u = f - I(u^{(i)}) \qquad \text{(A A.5-5)}$$

This equation is solved for σu and the next appropriate solution is obtained by

$$u^{(i+1)} = u^{(i)} + \delta u \qquad \text{(A A.5-6)}$$

Solution of this equation completes one iteration, and the process can be repeated.

The full Newton-Raphson method requires recalculation and refactorization of the stiffness matrix K(u) for every iteration. The modified Newton-Raphson method uses the same stiffness matrix for all iterations. This means that Equation A A.5-5 is replaced by

$$K(u^{(0)})\,\delta u = f - I(u^{(i)}) \qquad \text{(A A.5-7)}$$

If the load is applied incrementally, the MARC program recalculates the stiffness matrix at the start of each increment or at selected increments, as specified.

The strain correction method is akin to the Newton-Raphson method; however, the treatment of geometric nonlinearities is different. After each displacement update, the new strains $E_{\alpha\beta}^{(i+1)}$ are calculated from u^i and δ^u, which yield

$$E_{\alpha\beta}^{(i+1)} = E_{\alpha\beta}^{(i)} + \frac{1}{2}\left(\delta u_{a,\beta} + \delta u_{\beta,\alpha} + u_{\kappa,\alpha}^{(i)}\delta u_{\kappa,\beta} + \delta u_{\kappa,\alpha}u_{\kappa,\beta}^{(i)} + \delta u_{\kappa,\alpha}\delta u_{\kappa,\alpha}\right) \qquad \text{(A A.5-8)}$$

This expression is linear except for the last term. In situations where the rotations are large compared to the strain changes, the last term becomes significant and may cause considerable error in the strains and stresses. The errors in the stresses may in turn cause errors in the stiffness matrix K, particularly in slender structures where the stress-dependent terms are important. Convergence problems in the standard Newton-Raphson procedure result from these errors.

The linear and nonlinear portions of E are separated to overcome these problems.

$$\bar{E}_{\alpha\beta}^{(i+1)} = E_{\alpha\beta}^{(i)} + \frac{1}{2}(\delta u_{\alpha,\beta} + \delta u_{\beta,\alpha} + u_{\kappa,\alpha}^{(i)}\delta u_{\kappa\beta} + \delta u_{\kappa,\alpha}u_{\kappa,\beta}^{(i)}) \quad \text{(A A.5-9)}$$

$$E_{\alpha\beta}^{c} = 1/2\delta u_{\kappa,\alpha}\delta u_{\kappa,\beta} \quad \text{(A A.5-10)}$$

The barred quantities represent linearized quantities. The program uses the nonlinear strain change E as a "strain correction" term in the next iteration or increment. This strain correction term takes the form

$$K(u^{(i)}) \bullet \delta u = f - I(u^{(i)}) - I^{c} \quad \text{(A A.5-11)}$$

The Secant Method

The secant method used by MARC is based on the Davidon-rank one quasi-Newton update. The advantage of these methods is that the stiffness matrix is calculated and factorized only once per increment. This method is not effective if the iterative solver is used.

The quasi-Newton requirement is that a stiffness matrix for iteration i could be found based on the right-hand sides of iterations i and i-1 as follows

$$K_i\delta u^i = [f - I(u^{(i)})] - [f - I(u^{(i-1)})] = R^i - R^{i-1} \quad \text{(A A.5-12)}$$

This problem does not uniquely determine K_i. The Davidon-rank one update uses an additive form on the inverse of the tangent stiffness matrix as follows:

$$K_i^{-i} = K_{i-1}^{-1} + \frac{[\delta\Delta u_{i-i} - K_{i-1}^{-1}(R^i - R^{i-1})][\delta u^{i-1} - K_{i-1}^{-1}(R^i - R^{i-1})]^T}{[\delta u^{i-1} - K_{i-1}^{-1}(R^i - R^{i-1})]^T(R^i - R^{i-1})} \quad \text{(A A.5-13)}$$

At every iteration a modified Newton-Raphson solution is obtained first:

$$\delta\Delta u_i^{*} = -K_1^{-1}R_i \quad \text{(A A.5-14)}$$

Substitution in Equation A A.5-12 allows the direct calculation of $\delta\Delta u_i^{*}$ by means of several vector operations. First a coefficient C is obtained

$$C = \frac{(\delta u^{(i-1)} + \delta u^i - \delta u^{(i-1)})^T R^i}{[\delta u^{(i-1)} + \delta u^i - \delta u^{(i-1)}]^T(R - R^{i-1})} \quad \text{(A A.5-15)}$$

then

$$\delta u^i = (1 - C)\,\delta u^i - C\delta u^{i-1} + C\delta u^{i-1} \quad \text{(A A.5-16)}$$

In the Eulerian formulation (R-P FLOW option), the governing equation of the system can be expressed as

$$Kv = f \qquad \text{(A A.5-17)}$$

where v is a velocity vector, and f is a force vector.

This equation is very nonlinear because K is a nonlinear function of v. By default, a direct substitution method is used to solve the problem. If v^i is the velocity at iteration i, the result of iteration i + 1 is

$$K(v^i) \bullet v^{(i+1)} = f \qquad \text{(A A.5-18)}$$

If this method does not converge in 10 iterations, it is possible to switch into a full Newton-Raphson method.

A A.6 CONVERGENCE CONTROLS FOR STRESS ANALYSIS

The default procedure for convergence criterion in MARC is based on the magnitude of the maximum residual load compared to the maximum reaction force. This method is appropriate since the residuals measure the out-of-equilibrium force, which should be minimized. This technique is also appropriate for Newton methods, where zero-load iterations reduce the residual load. The method has the additional benefit that convergence can be satisfied without iteration.

The default procedure is outlined below.

1. RESIDUAL CHECKING

$$\frac{\|F_{residual}\|_\infty}{\|F_{resident}\|_\infty} < TOL_1 \qquad \text{(A A.6-1)}$$

$$\frac{\|F_{residual}\|_\infty}{\|F_{reaction}\|_\infty} < TOL_1 \text{ and } \frac{\|M_{residual}\|_\infty}{\|M_{reaction}\|_\infty} < TOL_2 \qquad \text{(A A.6-2)}$$

$$\|F_{residual}\|_\infty < TOL_1 \qquad \text{(A A.6-3)}$$

$$\|F_{residual}\|_\infty < TOL_1 \text{ and } \|M_{residual}\|_\infty < TOL_2 \qquad \text{(A A.6-4)}$$

Where F is the force vector, and M is the moment vector. TOL_1 and TOL_2 are control tolerances. $\|F\|_\infty$ indicates the component of F with the highest absolute value. Residual checking has two drawbacks. First, if the CENTROID option is used, the residuals and reactions are not calculated accurately. Second, in some special problems, such as free thermal expansion, there are no reaction forces. The program uses displacement checking in either of these cases.

2. DISPLACEMENT CHECKING

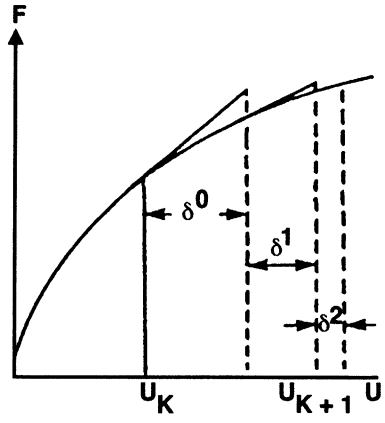
$$\frac{\|\delta u\|_\infty}{\|du\|_\infty} < TOL_1 \qquad \text{(A A.6-5)}$$

$$\frac{\|\delta u\|_\infty}{\|du\|_\infty} < TOL_1 \text{ and } \frac{\|\delta \phi\|_\infty}{\|d\phi\|_\infty} < TOL_2 \qquad \text{(A A.6-6)}$$

$$\|\delta u\|_\infty < TOL_1 \qquad \text{(A A.6-7)}$$

$$\|\delta u\|_\infty < TOL_1 \text{ and } \|\delta \phi\|_\infty < TOL_2 \qquad \text{(A A.6-8)}$$

where du is the displacement increment vector, δu is the displacement iteration vector, dϕ is the rotation increment vector, and δ ϕ is the rotation iteration vector. With this method, convergence is satisfied if the maximum displacement of the last iteration is small compared to the actual displacement change of the increment. A disadvantage of this approach is that it results in at least one iteration, regardless of the accuracy of the solution.



δ^i **Displacement of ith iteration**

U_K **Displacement at increment K**

$$\left| \frac{\delta^i}{\sum_{j=0}^{i} \delta_j} \right| \leq \text{Tolerance}$$

Figure A A.6-1 Displacement Control

3. STRAIN ENERGY CHECKING

This is similar to displacement testing where a comparison is made between the strain energy of the latest iteration and the strain energy of the increment. With this method, the entire model is checked.

$$\frac{\delta E}{dE} < TOL_1 \qquad \text{(A A.6-9)}$$

where dE is the strain energy of the increment and δE is the strain energy of the iteration. These energies are the total energies, integrated over the whole volume. A disadvantage of this approach is that it results in at least one iteration, regardless of the accuracy of the solution. The advantage of this method is that it evaluates the global accuracy as opposed to the local accuracy associated with a single node.

A A.7 SINGULARITY RATIO

The singularity ratio, R, is a measure of the conditioning of the system of linear equations. R is related to the conditioning number, C, which is defined as the ratio between the highest and lowest eigenvalues in the system. The singularity ratio is an upper bound for the inverse of the matrix conditioning number.

$$1/R \leq C \qquad \text{(A A.7-1)}$$

C and R establish the growth of errors in the solution process. If the errors on the right-hand side of the equation are less than E prior to the solution, the errors in the solution will be less than δ, with

$$\delta \leq CE \qquad \text{(A A.7-2)}$$

The singularity ratio is a measure that is computed during the Crout elimination process of the MARC program using the direct solver. In this process, a recursive algorithm redefines the diagonal terms

$$A_{kk}^{(k)} = A_{kk}^{(k-1)} - \sum_{l=i}^{\kappa-1} A_{lk}A_{kl}, \; 1 \leq i \leq k-1 \qquad \text{(A A.7-3)}$$

where i is a function of the matrix profile. A_{kk} is a diagonal of the k^{th} degree-of-freedom. The singularity ratio is defined as

$$R = \min A_{kk}^{(k)} / A_{kk}^{(k-1)} \qquad \text{(A A.7-4)}$$

If all $A_{kk}^{(k)}$ and $A_{kk}^{(k-1)}$ are positive, the singularity ratio indicates loss of accuracy during the Crout elimination process. This loss of accuracy occurs for all positive definite matrices. The number of digits lost during the elimination process is approximately equal to

$$n_{lost} = -\log_{10} R \qquad \text{(A A.7-5)}$$

The singularity ratio also indicates the presence of rigid body modes in the structure. In that case, the elimination process produces zeros on the diagonal $A_{kk}^{(k)} \cong 0$. Exact zeros never appear because of numerical error; therefore, the singularity ratio is of the order

$$R = 0\,(10^{-n\ digit}) \qquad \text{(A A.7-6)}$$

where n_{digit} is the accuracy of floating-point numbers used in the calculation. For most versions of MARC, $n_{digit} > 12$. If rigid body modes are present, $A_{kk}^{(k)}$ will be very small or negative. If either a zero or a negative diagonal is encountered, execution of the program is terminated because the matrix is diagnosed as being singular.

You can force the solution of a nonpositive definite or singular matrix. In this case, the program does not stop when it encounters a negative or small term $A_{kk}^{(k)}$ on the diagonal. If you use Lagrangian multiplier elements, the matrix becomes nonpositive definite and the program automatically disables the test on the sign of $A_{kk}^{(k)}$. However, it still tests on singular behavior.

When the EBE solver is used, an approximation to the condition number is calculated. As the iterative process reaches convergence, the value of the condition number becomes more accurate. If the condition number is less than a minimum value, the iteration process is terminated. This represents either a poorly conditioned system or rigid body modes in the system.

NOTE

A correct solution of a nonpositive definite system of linear equations does not mean that a solution for a nonlinear problem with such an equation system can be obtained.

A A.8 SOLUTION OF LINEAR EQUATIONS

The finite element formulation leads to a set of linear equations. The solution is obtained through numerically inverting the system. Because of the wide range of problems encountered with MARC, there are several solution procedures available.

Most analyses result in a system which is real, symmetric, and positive definite. While this is true for linear structural problems, assuming adequate boundary conditions, it is not true for all analyses. The MARC system has two main modes of solvers – Direct and Iterative. Each of these modes has two families of solvers, based upon the storage procedure. While all of these solvers may be used if there is adequate memory, only a subset use spill logic for an out-of-core solution. Finally, there are classifications based upon non-symmetric and complex systems. This is summarized below:

	Direct Profile	Direct Sparse	Iterative Sparse	Iterative EBE	Vendor Provided
Solver Option	0	4	2	1	3
Real Symmetric	Yes	Yes	Yes	Yes	Yes
Real Nonsymmetric	Yes	No	No	No	No
Complex Symmetric	Yes	No	No	No	No
Complex nonsymmetric	No	No	No	No	No
Out-of-core	Yes	Yes	No	Yes	Maybe
Possible problem with poorly conditioned systems	No	No	Yes	Yes	No

The choice of the solution procedure is made through the SOLVER option.

Direct Methods

Traditionally, the solution of a system of linear equations has been accomplished using direct solution procedures, such as Cholesky decomposition and the Crout reduction method. These methods are usually reliable, in that they give accurate results for virtually all problems at a predictable cost. For positive definite systems, there are no computational difficulties. For poorly conditioned systems, however, the results may degenerate but the cost remains the same. The problem with these direct methods is that a large amount of memory (or disk space) is required, and the computational costs become very large.

Iterative Methods

The K5 release of MARC introduced iterative solvers as a viable alternative for the solution of large systems. These iterative methods are based on *preconditioned conjugate gradient methods*. The single biggest advantage of these iterative methods is that they allow the solution

of very large systems at a reduced computational cost. This is true regardless of the hardware configuration. The disadvantage of these methods is that the solution time is dependent not only upon the size of the problem, but also the numerical conditioning of the system. A poorly conditioned system will lead to slow convergence – hence increased computation costs.

When discussing iterative solvers, two related concepts are introduced: *fractal dimension*, and *conditioning number*. Both are mathematical concepts, although the fractal dimension is a simpler physical concept. The *fractal dimension*, the range of which is between 1 and 3, is a measure of the "chunkiness" of the system. For instance, a beam has a fractal dimension of 1, while a cube has a fractal dimension of 3.

The *conditioning number* is related to the ratio of the lowest to the highest eigenvalues of the system. This number is also related to the *singularity ratio*, which has been traditionally reported in the MARC output when using a direct solution procedure. In problems involving beams or shells, the conditioning number is typically small, because of the large differences between the membrane and bending stiffnesses.

Preconditioners

The choice of preconditioner may substantially improve the conditioning of the system, which in turn reduces the number of iterations required. While all positive definite systems with N degrees of freedom will converge in N iterations, a well conditioned system will typically converge in less than the square root of N iterations.

The available preconditioners available in the iterative solvers are

Preconditioner	Sparse	EBE
None	No	Yes
Diagonal	Yes	No
Scaled Diagonal	Yes	No
Incomplete Cholesky	Yes	No
Element Cholesky	No	Yes

The iterative solvers require an error criteria to determine when convergence occurs. The default is to use an error criteria based upon the ratio between the residuals in the solution and the reaction force. After obtaining the solution of the linear equations u^C evaluate:

$$Ku^C = F^C \qquad \text{(A A.8-1)}$$

The residual from the solution procedure is:

$$R^{es} = F^A - F^C = F^A - Ku^C \qquad \text{(A A.8-2)}$$

If the system is linear (K does not change) and exact numerics are preformed, then $R^{es} = 0$.

Because this is an iterative method R is nonzero, but reduces in size with further iterations. Convergence is obtained when

$$\text{Res/Reac} < \text{tolerance} \qquad \text{(A A.8-3)}$$

The tolerance is specified through the SOLVER option.

Storage Methods

In general, a system of linear equations with N unknowns is represented by a matrix of size N by N, or N^2 variables. Fortunately, in finite element or finite difference analyses, the system is "banded" and not all of the entries need to be stored. This substantially reduces the memory (storage) requirements as well as the computational costs.

In the finite element method, additional zeroes often exist in the system, which results in a partially full bandwidth. Hence, the *profile* (or *skyline*) method of storage is advantageous. This profile storage method is used in MARC to store the stiffness matrix. When many zeroes exist within the bandwidth, the *sparse storage methods* may be quite advantageous. Such techniques do not store the zeroes, but require additional memory to store the locations of the nonzero values. The user can determine the "sparsity" of the system (before decomposition) by examining the statements:

"Number of nodal entries excluding fill in" **x**
"Number of nodal entries including fill in" **y**

If the ratio (x/y) is large, then the sparse matrix storage procedure will be advantageous.

Iterative Solvers

In MARC, two different iterative solvers are available. The first uses an *element-by-element (EBE) method*, while the second uses a *sparse matrix technique*. When using the EBE method, the total stiffness matrix is never assembled, but the element stiffnesses are stored.

Each of these methods is advantageous for different classes of problems. In the EBE method, the preconditioner is applied on an element level, as opposed to applying to the global stiffness matrix. This allows additional flexibility when working with elements with multiple degree of freedom types, such as shell elements or hybrid elements. The EBE method also offers advantages on parallel processing computers if the number of processors is greater than two and may be appropriate for massively parallel processing.

There exist certain types of analyses for which neither iterative solver (EBE or sparse iterative solver) is appropriate. These include:

elastic analysis
explicit creep analysis
complex harmonic analysis
substructures
central difference techniques
eigenvalue analysis
use of gap elements

Elastic or explicit creep analysis involves repeated solutions using different load vectors. When a direct solver is used, this is performed very efficiently using back substitution. However, when an iterative solver is used. the stiffness matrix is never inverted, and the solution associated with a new load vector requires a complete re-solution.

The EBE iterative solver has been implemented in MARC for all analyses *except*:

Joule heating
hydrodynamic bearing
contact involving friction

In addition, the EBE iterative solver should *not* be used with generalized plane strain elements and higher order Herrmann incompressible elements. Also, the sparse iterative solver may exhibit poor convergence when Herrmann incompressible elements are present

Basic Theory

A linear finite element system is expressed as:

$$Ku = F \tag{A A.8-4}$$

And a nonlinear system is expresses as:

$$K^N \Delta u = R \tag{A A.8-5}$$

where K is the elastic stiffness matrix, K^N is the nonlinear tangent stiffness matrix, Δu is the displacement vector, F is the applied load vector, and R is the residual.

The linearized system is converted to a minimization problem expressed as:

$$\phi(u) = 1/2\ u^T K u - u^T F \tag{A A.8-6}$$

For linear structural problems, this process may be considered as the minimization of the potential energy. The minimum is achieved when

$$u = K^{-1} F \tag{A A.8-7}$$

The function ϕ decreases most rapidly in the direction of the negative gradient.

$$\nabla\phi(u) = F - Ku = R \tag{A A.8-8}$$

The objective of the iterative techniques is to minimize ϕ without inverting the stiffness matrix. In the simplest methods,

$$u_{K+1} = u_K + \alpha_K R_K \tag{A A.8-9}$$

where

$$\alpha_k = R_K^T \bullet R_K / R_K^T K R_k \tag{A A.8-10}$$

The problem is that the gradient directions are too close, which results in poor convergence.

An improved method led to the *conjugate gradient method*, in which

$$u_{K+1} = u_K + \alpha_K P_K \tag{A A.8-11}$$

$$\alpha_k = P_K^T \bullet R_{K-1} / P_K^T K P_k \tag{A A.8-12}$$

The trick is to choose P_K to be K conjugate to P_1, P_2, ..., P_{K-1}. Hence, the name "conjugate gradient methods. Note the elegance of these methods is that the solution may be obtained through a series of matrix multiplications and the stiffness matrix never needs to be inverted.

Certain problems which are ill-conditioned may lead to poor convergence. The introduction of a preconditioner has been shown to improve convergence. The next key step is to choose an appropriate preconditioner which is both effective as well as computationally efficient. The easiest is to use the diagonal of the stiffness matrix. The incomplete Cholesky method has been shown to be very effective in reducing the number of required iterations.

A A.9 FLOW DIAGRAM

Figure A A.9-1 is a diagram showing the flow sequence of the MARC program. This diagram shows the input phase, equivalent nodal load vector calculation, matrix assembly, matrix solution, stress recovery, and output phase. It also indicates load incrementation and iteration within a load increment.

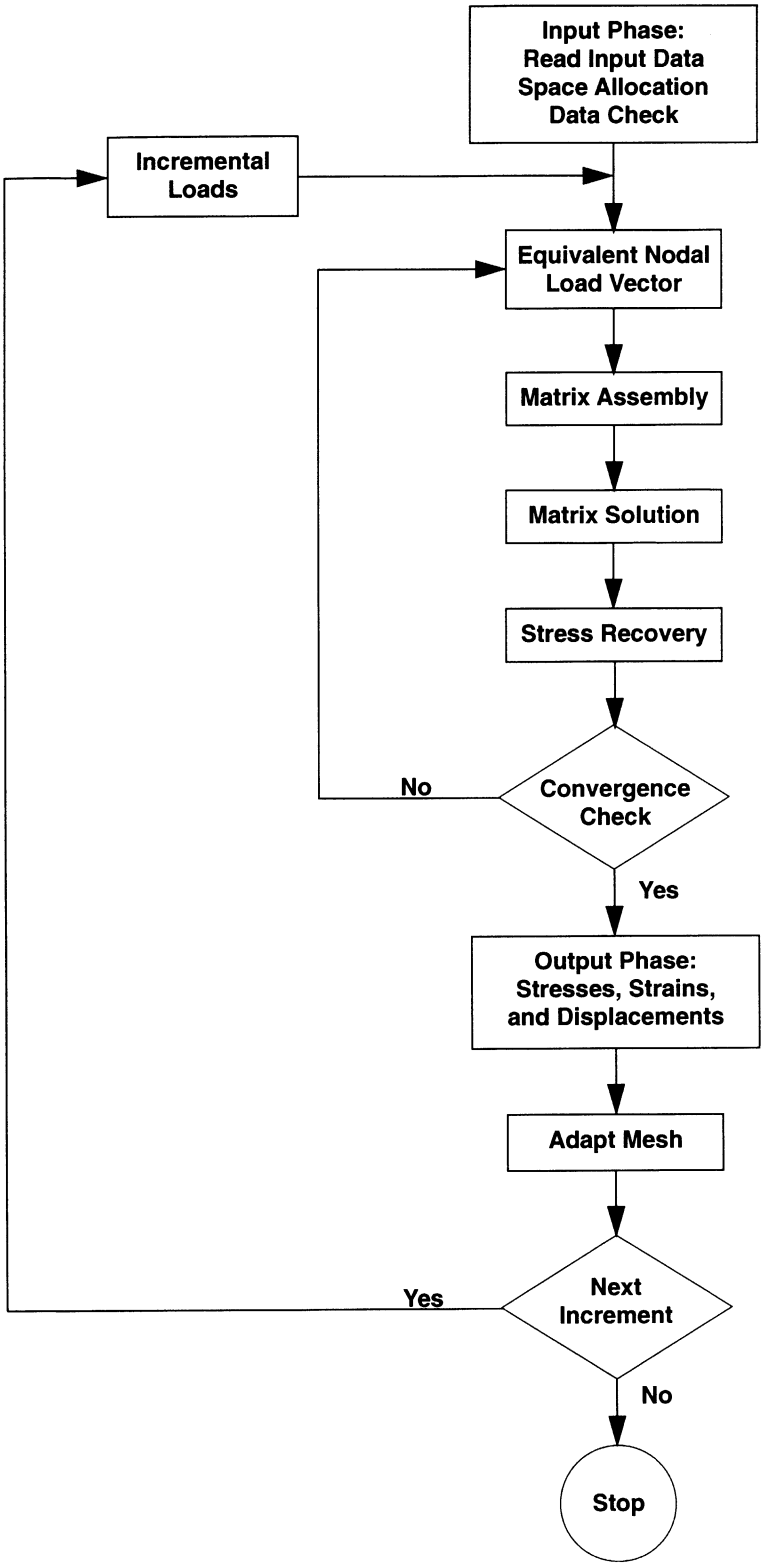


Figure A A.9-1 MARC Program Flow Diagram

References

1. Hibbitt, H. D., Marcal, P. V. and Rice, J. R. "A Finite Element Formulation for Problems of Large Strain and Large Displacement." *Int. J. Solids Struct.* 6, 1069-1086, 1970.

2. Marcal, P. V. "Effect of Initial Displacements on Problems of Large Deflection and Stability." Brown University Engineering Report, ARPA E54, November, 1967. Presented at Proc. ASCE Joint Specialty Conference, Optimization and Nonlinear Problems, April, 1968.

3. Marcal, P. V., and King, I. P. "Elastic-Plastic Analysis of Two-Dimensional Stress Systems by the Finite Element Method." *Int. J. Mech.*, 9, 3, 1967.

4. Nagtegaal, J. C., and deJong, J. E. "Some Computational Aspects of Elastic-Plastic Large Strain Analysis." *Int J. Num. Meth. Engrg.* 17, 15-41, 1981.

5. Rice, J. R., and Tracy, D. M. "Computational Fracture Mechanics" in *Proc. Symp. on Numerical and Computer Methods in Structural Mechanics*, edited by S. J. Fenves. New York: Academic Press, 585, 1973. Presented at Symp. on Numerical and Computer Methods in Structural Mechanics, Urbana, IL, 1967.

6. Zienkiewicz, O. C. *The Finite Element Method.* McGraw-Hill, 1977.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Y

Z