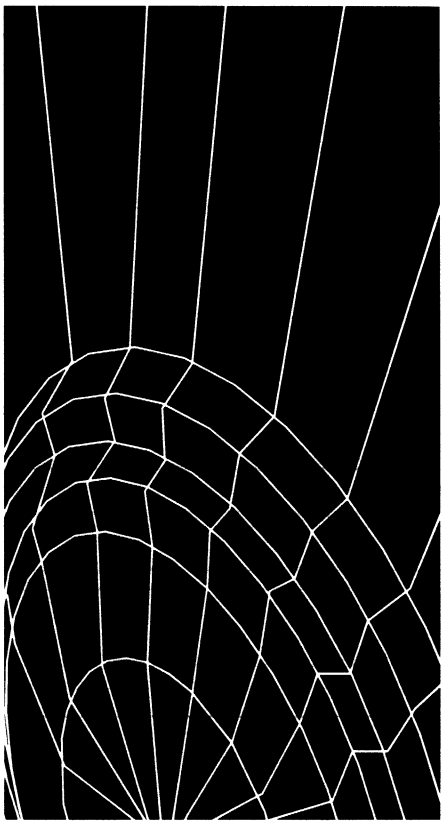
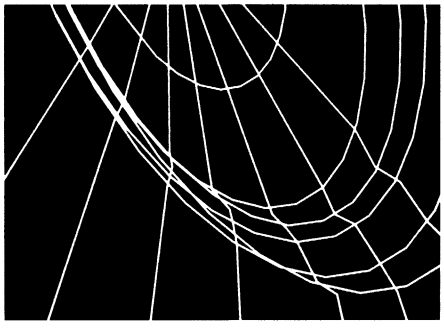


MARC

Analysis
Research
Corporation



Volume D

User Subroutines and Special Routines

Version K7

MARC Analysis Research Corporation
260 Sheridan Avenue, Suite 309
Palo Alto, CA 94306 USA

Phone: (650) 329-6800
FAX: (650) 323-5892

Document Title: **MARC Volume D: User Subroutines and Special Routines, Version K7**
Part Number: RF-3004-07
Revision Date: August, 1997

Proprietary Notice

Restricted Rights Notice

Trademarks

Contents

1 Introduction

Common Blocks Description 1-1
UNIX. 1-1
Note on Double Precision 1-2
Format 1-2
ELMVAR Utility Routine 1-5
MATDAT Common Block. 1-6
CONCOM Common Block 1-7
ELMCOM Common Block 1-10

2 List of User-defined Loading, Boundary Conditions, and State Variables Subroutines

2 User-defined Loading, Boundary Conditions, and State Variables Subroutines

FORCEM – Input of Nonuniform Distributed Loads. 2-4
FLUX – Input of Nonuniform Fluxes. 2-8
CUPFLX – Coupling of Inelastic Energy and Internal Heat Generation 2-10
UINSTR – Input of Initial State of Stresses. 2-11
UFOUR – Input of a User-defined Function F(**Q**) for Fourier Analysis 2-12
FORCDT – Input of Time Dependent Nodal Loads, Displacements or Time Dependent Nodal Fluxes, Temperatures for Heat Transfer 2-14
FORCDF – Input of Frequency Dependent Loads or Displacements in Harmonic Analysis 2-18
FILM – Input of Nonuniform Film Coefficients 2-20
FLOW – Input of Mass Flow Rate and Inlet Temperature 2-22
GAPT – Input of Thermal Contact (Conrad) Gap Temperature 2-24
UFORMS – Definition of Constraint Conditions 2-25
CREDE – Input of Pre-specified State Variables 2-30

INITSV – Initialize State Variable Values 2-32
NEWSV – Input New State Variable Values 2-33
USSD – Input of Spectral Response Density 2-34
USINC – Input of Initial Conditions 2-35
USDATA – Input of Initial Data 2-36
UVELOC – Generation or Modification of Nodal Velocity Vectors 2-37
MOTION (2D) – Definition of Rigid Surface Motion for 2D Contact 2-38
MOTION (3D) – Definition of Rigid Surface Motion for 3D Contact 2-40
UFRIC – Definition of Friction Coefficients 2-42
DIGEOM – Definition of 3D Rigid Surface Patch 2-44
SEPFOR – Definition of Separation Force 2-45
SEPSTR – Definition of Separation Stress 2-47
UHTCOE – Definition of Environment Film Coefficient 2-48
UHTCON – Definition of Contact Film Coefficient 2-51
UNORST – Definition of Normal Stress, Flow Stress and Temperature at Contact Node . . 2-53
UCONTACT – User-defined Contact Condition 2-56
INITPL – Initialize Equivalent Plastic Strain Values 2-59
INITPO – Initialize Pore Pressure in an Uncoupled Fluid-Soil Analysis 2-60
NEWPO – Modify Pore Pressure in an Uncoupled Fluid-Soil Analysis 2-61

3 List of User-defined Anisotropy and Constitutive Relations Subroutines

3 User-defined Anisotropy and Constitutive Relations Subroutines

ANELAS – Elastic Anisotropy 3-4
HOOKLW – Anisotropic Elastic Law 3-12
ANPLAS – Anisotropic Yield Surface and Creep Potential 3-14
UFAIL – User-defined Failure Criterion 3-16
ORIENT – Specification of Preferred Orientation 3-17
ANEXP – Anisotropic Thermal Expansion 3-19
ANKOND – Input of Anisotropic Thermal Conductivity Matrix 3-21
UEPS – Input of Anisotropic Permittivity Matrix 3-23
UMU – Input of Anisotropic Permeability Matrix 3-24
USIGMA – Input of Anisotropic Electric Conductivity Matrix 3-26
USPCHT – Definition of Specific Heat 3-27
CRPLAW – Input of Special Creep Law 3-28
VSWELL – Input of Special Swelling Law 3-33

WKSLP – User Subroutine for Work-hardening Slope Definition 3-35
USPRNG – Input of Nonlinear Spring, Dashpot and Foundation Stiffness 3-38
UCRACK – Input of Ultimate Stress for Cracking Analysis 3-40
TENSOF – Input of Tension Softening Modulus for Cracking Analysis. 3-42
USHRET – Input of Shear Retention Factor for Cracking Analysis 3-43
UVOID – Definition of the Initial Void Volume Fraction 3-44
UVOIDN – Definition of the Void Nucleation Rate. 3-45
UPOWDR – Definition of Material Data for Powder Metallurgy Model. 3-47
UPERM – Definition of Soil Permeability 3-49
UMOONY – Mooney-Rivlin Material 3-50
UENERG – Strain Energy Function 3-51
UOGDEN – Definition of Ogden Material Parameters 3-52
UELDAM – Definition of Damage Parameters in Ogden Model 3-54
HYPELA – Hypoelastic Material 3-56
HYPELA2 – User-defined Material Behavior. 3-58
UFINITE – Finite Deformation Isotropic Material Models. 3-62
UPSTRECH – Definition of Generalized Principal Stretch-based Elasticity Models 3-64
GENSTR – Generalized Stress Strain Law (Shells & Beams) 3-67
UBEAM – Input for Nonlinear Beam 3-69
UPHI – Input of PHI Function in Harmonic Analysis 3-71
UCOMPL – Input of Viscous Stress Strain Relationship 3-73
GAPU – Input of Gap Direction And Closure Distance 3-75
USELEM – User-Defined Element 3-77
UNEWTN – Input of Viscosity in Flow Analysis 3-80
URPFLO – User Routine for Rigid-Plastic Flow 3-81
References 3-83

4 List of Viscoplasticity and Generalized Plasticity Subroutines

4 Viscoplasticity and Generalized Plasticity Subroutines

UVSCPL – Definition of the Inelastic Strain Rate 4-2
CRPLAW – Input of Special Viscoplastic Strain Rate Law. 4-4
NASSOC – Input of a Nonassociated Flow Law 4-6
ZERO – Calculation of Equivalent Stress 4-8
YIEL – Calculation of Current Yield. 4-9
ASSOC – Input of Associated Flow Law. 4-10
SINCER – User Subroutine for Improving Accuracy 4-11

5 List of Viscoelasticity Subroutines

5 Viscoelasticity Subroutines

CRPVIS – Viscoelasticity – Generalized Kelvin Material Behavior 5-2
TRSFAC – Define a Shift Function for Thermo-Rheologically Simple (T.R.S.) Material Behavior 5-5
HOOKVI – User Defined Anisotropic Viscoelasticity 5-7

6 List of Geometry Modifications Subroutines

6 Geometry Modifications Subroutines

UFXORD – Coordinate Generation or Modification 6-3
UFCONN – Connectivity Generation or Modification 6-4
MAP2D – Boundary Node Coordinates Modification in Mesh2D 6-5
UPNOD – Update Nodal Positions in Flow Solutions 6-6
UACTIVE – Activate or Deactivate Elements. 6-7
REBAR – Input of Rebar Positions, Areas and Orientations 6-8
UFRORD – Rezoning Coordinate Generation or Modification 6-9
URCONN – Rezoning Connectivity Generation or Modification 6-10
UCOORD – Relocate Nodes Created During Adaptive Meshing 6-11
UADAP – User Defined Error Criteria 6-12
UTRANS – Implement Local Coordinate System 6-13
USHELL – Modify Thickness of Shell Elements. 6-14
SSTRAN – Transformation of Substructures 6-15
UTHICK – User-Specified Nodal Thicknesses 6-16
UACTUAT – Prescribe the Length of an Actuator. 6-17

7 List of Output Quantities Subroutines

7 Output Quantities Subroutines

PLOTV – User-selected Postprocessing of Element Variables 7-2
UPOSTV – User-selected Postprocessing of Nodal Variables 7-4
IMPD – Output of Nodal Quantities 7-7

ELEVAR – Output of Element Quantities 7-16
ELEVEC – Output of Element Quantities in Harmonic Analysis 7-18
INTCRD – Output of Integration Point Coordinates 7-20
UBGINC – Beginning of Increment 7-21
UEDINC – End of Increment 7-22
UBGITR – Beginning of Iteration 7-23
UELOOP – Beginning of Element Loop 7-24

8 List of Hydrodynamic Lubrication Subroutines

8 Hydrodynamic Lubrication Subroutines

UBEAR – Input of Spatial Orientation of Lubricant Thickness 8-2
UGROOV – Input of Groove Depths 8-3
URESTR – Input of Nonuniform Restrictor Coefficients 8-4
UTHICK – Generation or Modification of Nodal Thickness or Thickness Change Field . . 8-5
UVELOC – Generation or Modification of Nodal Velocity Vectors 8-6

9 List of Special Routines

9 Special Routines – MARC Post File Processor PLDUMP

PLDUMP 9-2
MARC Post File Processor 9-2
Marc Post File Layout 9-3

A User Subroutines and Special Routines

Contents

Figures

2-1 Coarse to Fine Mesh Example 2-29

Tables

2-1 User-defined Loading, Boundary Conditions, State Variables Subroutine Requirements 2-1

3-1 User-defined Anisotrophy and Constitutive Relations Subroutine Requirements . 3-1

3-2 Allowable Anisotropy 3-5

4-1 Viscoplasticity and Generalized Plasticity User Subroutine Requirements 4-1

5-1 Viscoplasticity Subroutines Requirements 5-1

6-1 Geometry Modifications Subroutines Requirements 6-1

7-1 Output Quantities Subroutines Requirements 7-1

8-1 Hydrodynamic Lubrication Subroutines Requirements 8-1

1 Introduction

In MARC, the user subroutine feature constitutes one of the real strengths of the program, allowing you to substitute your own subroutines for several existing in the program. This feature provides you with a wide latitude for solving nonstandard problems. These routines are easily inserted into the program. When such a routine is supplied, you are simply replacing the one which exists in the program file using appropriate control setup. A description of each of the available user subroutines is given in this manual. In addition, discussions of special routines are also included.

Note: The reading of data is not recommended in most of the user subroutines since many of these routines are in the recycling loop for nonlinear analysis, and hence, you cannot know how many times per increment the routine is called.

Common Blocks Description

Often, when using a user subroutine, you would like more information than is provided through the call arguments. Almost all information is available through common blocks. Much of the information provided below is already available but occasionally, especially in older subroutines, it is not.

All common blocks can be accessed by you by "including" them in the user subroutine. The syntax used to perform this is machine dependent:

UNIX

On machines running under UNIX, or variations of machines on UNIX, the common blocks are in a directory whose path is `xxx/version/common`, where `xxx` is the complete path to the directory where MARC resides, and `version` is the release of MARC; for example, K7.

The syntax to use in the user subroutine is:

```
include 'xxx/version/common/yyy'
```

where `yyy` is the name of the common block. Note that the word `include` must begin after column 6, and the complete pathname must be typed exactly like it is on the disk with no blanks and within quotation marks.

Note on Double Precision

In the K5 and subsequent versions, the MARC program is written completely in double precision. Hence, on all machines (excluding the Cray), an `IMPLICIT REAL *8 (A-H, O-Z)` statement is required in the user subroutines. This is to ensure that variables passed between the MARC program and the user routine are compatible and to ensure that any common blocks included will be correct.

Format

The following quantities are available in all user subroutines:

```
TIME AT BEGINNING OF INCREMENT:     CPTIM
TIME INCREMENT:                     TIMINC
AVAILABLE THROUGH
      include 'path/common/creeps'

INCREMENT NUMBER:           INC
SUBINCREMENT NUMBER:        INCSUB
AVAILABLE THROUGH
      include 'path/common/concom'

NUMBER OF ELEMENTS IN MESH:      NUMEL
NUMBER OF NODES IN MESH:         NUMNP
MAXIMUM NUMBER OF DEGREES OF FREEDOM PER NODE:          NDEG
MAXIMUM NUMBER OF COORDINATE DIRECTIONS:                NCRD
AVAILABLE THROUGH
      include 'path/common/dimen'
```

In a coupled analysis, reference variable `IPASS` to determine if the current iteration is a stress or heat transfer iteration:

```
IPASS = 1                   STRESS
IPASS = 2                   HEAT TRANSFER
AVAILABLE THROUGH
      include 'path/common/concom'
```

The following quantities are available in user subroutines which are in an element loop:

```
ELEMENT NUMBER:     M
AVAILABLE THROUGH
      include 'path/common/far'

ELSTO ELEMENT NUMBER:         N
INTEGRATION POINT NUMBER:     NN
LAYER NUMBER:                 KC
AVAILABLE IN
      include 'path/common/lass'
```

```
NUMBER OF NODES IN ELEMENT:  NNODE
AVAILABLE IN
      include 'path/common/elmcom'

NUMBER OF DIRECT COMPONENTS OF STRESS:  NDI
NUMBER OF SHEAR COMPONENTS OF STRESS:   NSHEAR
SIZE OF STRESS STRAIN LAW:              NGENS
ELEMENT TYPE:                           JTYPE
AVAILABLE THROUGH
      include 'path/common/elmcom'

MAXIMUM NUMBER OF LAYERS PER ELEMENT:    NEQST
MAXIMUM NUMBER OF INT. PTS PER ELEMENT:  NSTRES
AVAILABLE THROUGH
      include 'path/common/nzro1'
```

To determine the coordinates of integration point NN of element M and to place these coordinates in array CCINT, use the following procedure:

```
        include 'path/common/lass'
        include 'path/common/dimen'
        include 'path/common/space'
        include 'path/common/heat'
        include 'path/common/array4'
        DIMENSION CCINT(12)
        LA1 = ICRXPT + (NN-1)*NCRD + LOFR
        D0 1 II = 1, NCRD
        CCINT(II) = VARS(LA1)
      1 LA1 = LA1 + 1
```

Note: This is only available after the first stiffness matrix assembly.

To obtain the array of internal node numbers of an element, use variable LM:

```
include 'path/common/blnk'
```

The first NNODE numbers of LM are the internal node numbers.

To determine the internal node number LINT of user (external) node number LEXT, use the following procedure:

```
include 'path/common/dimen'
include 'path/common/arrays'
include 'path/common/develp'
include 'path/common/space'
LINT = IBSRCH (LEXT, INTS (INOIDS), NUMNP,1)
LA2 = INPNUM + LINT - 1
IF (JOPTIT. NE. 0) LINT = IGETSH (INTS(LA2), 0)
```

To determine the external (LEXT) node number from the internal (LINT) node number, use the following procedure:

```
include 'path/common/prepro'
include 'path/common/arrays'
include 'path/common/develp'
include 'path/common/space'
LEXT = LINT
LA2 = INPNUM + LINT - 1
IF (JOPTIT. NE. 0)  LEXT = IGETSH(INTS(LA2), 1)
IF (NNOIDS. NE. 0)  LEXT = INTS(INOIDS + LEXT-1)
```

To determine the coordinate of internal node number LINT and place these coordinates in array CCNODE, use the following procedure:

```
include 'path/common/dimen'
include 'path/common/array2'
include 'path/common/space'
DIMENSION CCNODE(12)
JRDPRE = 0
CALL VECFTC (CCNODE, VARS(IXORD), NCRDMX, NCRD, LINT,JRDPRE, 2, 1)
```

To determine the total displacement of internal node number LINT and place this data in array DDNODE, use the following procedure:

```
include 'path/common/dimen'
include 'path/common/array2'
include 'path/common/space'
DIMENSION DDNODE(12)
JRDPRE = 0
CALL VECFTC (DDNODE, VARS(IDSXT), NDEGMX, NDEG, LINT, JRDPRE, 2, 5)
```

Note: Displacements are given in transformed system if the TRANSFORMATION option is used at this node.

ELMVAR Utility Routine

To facilitate extraction of solution results, it is possible to use subroutine ELMVAR. This routine can be called from any user subroutine that is within an element loop. This routine is used in conjunction with the MARC post element post codes to return the calculated values to you.

This routine is called with the following header:

```
CALL ELMVAR (ICODE,M,NN,KC,VAR)
```

where:

ICODE	is the post code.
M	is your element number.
NN	is the integration point number.
KC	is the layer number.
VAR	is the current value(s) of the items requested.

Note: If you are requesting a tensor, you must make VAR a local array in your user subroutine.

The values of ICODE are given in *Volume C: Program Input* in the model definition section in the POST option.

If subroutine ELMVAR is called from a routine within the element assembly or stress recovery stage, the values of VAR are the current ones for this iteration. They are not necessarily the converged values.

This subroutine can be called from user subroutines:

ANELAS	FILM	NEWSV	UCRACK	UNEWTN	USIGMA
ANEXP	FLUX	ORIENT	UELDAM	UOGDEN	USPCHT
ANKOND	FORCEM	PLOTV	UELOOP	UPERM	UVOIDN
ANPLAS	GENSTR	REBAR	UENERG	UPOWDR	UVSCPL
ASSOC	HOOKLW	SINCER	UEPS	UPSTRECH	VSWELL
CRPLAW	HOOKVI	TENSOF	UFAIL	URESTR	WKSLP
CRPVIS	HYPELA	TRSFAC	UHTCOE	URPFLO	YIEL
CUPFLX	HYPELA2	UACTIVE	UHTCON	USELEM	ZERO
ELEVAR	INTCRD	UADAP	UMOONY	USHELL	
ELEVEC	NASSOC	UCOMPL	UMU	USHRET	

Example

Suppose you would like the plastic strain tensor from within user subroutine UADAP for a user-defined adaptive meshing criteria. In this example, there are no shell elements, so `KC=1` and the number of integration points per element = 4, so `INTEL=4`. The plastic strain tensor is code 321. The plastic strains are stored in a local array EPTEN. You could create the following routine:

```
SUBROUTINE UADAP (M,XORD,DSXT,NCRDMX,NDEGMX,LM,NNODE,USER)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION XORD(NCRDMX, *),DSXT(NDEGMX, *),LM(*)
DIMENSION EPTEN (6,28)
KC=1
INTEL=4
ICODE=321
DO NN=1,INTEL
   CALL ELMVAR(ICODE,M,NN,KC,EPTEN(1,NN))
ENDDO
   :
   USER CODE TO DEFINE USER
   :
RETURN
END
```

MATDAT Common Block

The material identification numbers (1,2,3, etc.) for cross-referencing to various quantities such as TEMPERATURE EFFECTS, WORK HARD, etc., must be used in user subroutines ANELAS, HOOKLW, ANPLAS, ANEXP, ANKOND, ORIENT, CRPLAW, VSWELL, etc. The common block `elmcom` contains the material identification number `"MATS"` for each material type.

In addition, the reference values of the material properties as given in the model definition section can be obtained in common block `matdat` where:

`ET(3)`	Young's moduli
`XU(3)`	Poisson's ratios
`RHO`	mass density
`SHRMOD(3)`	shear moduli
`COED(3)`	coefficient of thermal expansion
`YIELD(1)`	yield stress
`YIELD(2)`	ORNL 10th cycle yield stress
`YIELD(3)`	ORNL reversed plasticity yield stress

YRDR(3)	direct ratio's for Hill anisotropic plasticity
YRSR(3)	shear ratio's for Hill anisotropic plasticity
CONDU(3)	conductivities
SPHT	specific heat
CONDV(3)	resistivity
RHOHT	mass density for heat transfer
EMISV	emissivity
COSTPV	cost per unit volume
COSTPM	cost per unit mass

CONCOM Common Block

Two common blocks might be particularly useful for advanced usage in MARC. Common block concom contains most of the program controls in MARC, the variables and their meaning are given below:

1	iacous	acoustic analysis
2	iasmbl	reassemble stiffness matrix
3	iautth	auto therm or auto therm creep
4	ibear	hydrodynamic bearing
5	icompl	complex harmonic analysis
6	iconj	EBE iterative solver
7	icreep	explicit creep
8	ideva(15)	debug print flags
9	idyn	dynamic analysis type
10	idynt	permanent dynamic analysis type
11	ielas	elastic reanalysis or Fourier
12	ielcma	electromagnetic analysis (0,1,2)
13	ielect	electrostatic analysis
14	iform	contact
15	ifour	Fourier
16	iharm	harmonic analysis
17	ihcps	coupled analysis

18	iheat	heat transfer
19	iheatt	permanent heat transfer flag
20	ihresp	indicate that currently harmonic subincrement
21	ijoule	Joule heating
22	ilem	indicates in which part of element assembly
23	ilnmom	indicates whether a coupled soil analysis (0,1,2)
24	iloren	DeLorenzi calculation required
25	inc	increment number
26	incext	creep extrapolation
27	incsub	subincrement number
28	ipass	indicates if stress or heat transfer pass
29	iplres	dynamic, buckling or heat transfer second global matrix required
30	ipois	Poisson analysis
31	ipoist	permanent Poisson flag
32	irpflo	Eulerian - rigid plastic flow
33	ismall	small displacement analysis
34	ismalt	permanent small displacement flag
35	isoil	soil analysis
36	ispect	spectrum response
37	ispnow	perform spectrum response now
38	istore	update stress strain information.
39	iswep	currently performing eigenvalue extraction
40	ithcrp	thermal creep analysis
41	itherm	temperature dependent properties
42	iupblg	follower force
43	iupdat	update Lagrange
44	jacflg	Lanczos eigenmethod
45	jel	elastic increment
46	jparks	Fracture mechanics by Park method
47	largst	finite strain
48	lfond	distributed vs foundation flag
49	loadup	nonlinearity has occurred

50	loaduq	nonlinearity has occurred
51	lodcor	load correction
52	lovl	overlay indicator
53	lsub	flag to indicate which part of calculation
54	magnet	magnetostatic
55	ncycle	cycle number
56	newtnt	permanent Newton flag
57	newton	Newton flag
58	noshr	transverse shears included
59	linear	storage of betas, etc.
60	ivscpl	viscoplastic
61	icrpim	implicit creep
62	iradrt	radial return
63	ipshft	radiating cavities
64	itshr	transverse shear
65	iangin	orientation angle
66	iupmdr	update-anisotropy flag
67	iconjf	sparse conjugent gradient solver
68	jincfl	not used
69	jpermg	indicates that permanent magnets are included
70	jhour	indicates that there are some reduced integration with hourglass control elements
71	isolvr	solver flag
72	jritz	indicates that Ritz vectors will be used in eigenvalue analysis
73	jtable	not used
74	jshell	indicates presence of shell elements
75	jdoubl	indicates that double eigenvalue extraction will be used with Inverse Power Sweep method
76	jform	not used
77	jcentr	internal flag
78	imini	reduced storage flag for ELASTIC option
79	kautth	flag used in AUTO THERM option

80	idummy	not used
81	ibukty	convergence problem with buckling flag
82	iassum	assumed strain flag
83	icdnstd	constant dilatation flag
84	icnstt	not used
85	kmakmas	recalculate mass matrix flag
86	imethvp	implicit viscoplastic procedure

ELMCOM Common Block

In subroutines that are within an element loop, information about a particular element can be found in common block elmcom. These variables are also placed in an array IPROPS, that is used in those versions that support parallelization on an element level. The variables in common block elmcom and their meanings are given below.

1	ianels	anisotropy flag
2	ianiso	anisotropy flag
3	irebar	rebar element flag
4	icolps	indicates collapsed element
5	icomps	composite
6	icrack	cracking
7	ictrns	no longer used
8	idamag	damage
9	ianmat	anisotropic elastic constants given in input
10	igenpl	generalized plasticity
11	iherr	Hermann element (0,1,2)
12	intel	number of integration points
13	intin	integration point number if centroid
14	intpre	number of integration points for distributed
15	iort	curvilinear coordinates
16	ipela	hypoelastic
17	irheol	thermal rheologically simple
18	ishell	shell
19	isnte	integration point number if centroid

20	isotrp	elastic material
21	ityp	internal element type
22	iupcls	update Lagrange class (-3,-2,-1,0,1,2,3)
23	ivisc	Kelvin viscoelastic flag
24	ivisel	Hereditary integral viscoelastic flag
25	jcamcl	cam clay model
26	jhip	powder model
27	joakr	Oak Ridge model
28	joakrm	Type of Oak Ridge model
29	jogden	Ogden
30	jsoil	Soil
31	jtype	element type
32	jviscp	viscoplastic
33	jvisel	hereditary integral viscoelastic
34	kinhrd	kinematic hardening
35	lbend	pipe bend
36	lclass	element class
37	lheat	heat transfer element
38	lnoint	no integration points
39	lrebar	rebar element
40	Magna	material or composite id
41	mats	material id
42	march	Moor Coulomb
43	mooney	Mooney
44	mroz	Mroz - not supported
45	uncurled	number of coordinates
46	ndegel	number of degrees of freedom
47	ndi	number of direct components
48	ngenel	number of generalized strains
49	nnode	number of nodes
50	nomid	mid-increment not used
51	noniso	anisotropic
52	kkdum1	dummy
53	nregs	pointer to transverse shear

54	nshear	number of shears
55	nstran	number of strains
56	ntshr	number of transverse shears
57	ipgrcr	progressive cracking
58	ngens	number of generalized strains
59	jparel	element running in parallel mode
60	jhoure	this element is a reduced integration element with hourglass control
61	jfoam	foam model
62	nnodg	number of nodes per element, excluding extra nodes for Hermann and generalized plane strain

2 List of User-defined Loading, Boundary Conditions, and State Variables Subroutines

Subroutines	Page
CREDE	2-30
CUPFLX	2-10
DIGEOM	2-44
FILM	2-20
FLOW	2-22
FLUX	2-8
FORCDF	2-18
FORCDT	2-14
FORCEM	2-4
GAPT	2-24
INITPL	2-59
INITPO	2-60
INITSV	2-32
MOTION (2D)	2-38
MOTION (3D)	2-40
NEWPO	2-61
NEWSV	2-33
SEPFOR	2-45
SEPSTR	2-47
UCONTACT	2-56
UFORMS	2-25
UFOUR	2-12
UFRIC	2-42

Subroutines

	Page
UHTCOE	2-48
UHTCON	2-51
UINSTR.	2-11
UNORST	2-53
USDATA	2-36
USINC	2-35
USSD	2-34
UVELOC	2-37

2 User-defined Loading, Boundary Conditions, and State Variables Subroutines

The user subroutines described in this chapter provide an alternative to the standard input file for providing data in the analysis. Many problems have complex boundary conditions due to their spatial variation (such as wind loads) or due to their temporal variation. These routines provide a powerful mechanism to define this behavior in a simple manner. Table 2-1 summarizes these routines and indicates what parameters or model definition options are required to invoke the user subroutine.

Table 2-1 User-defined Loading, Boundary Conditions, State Variables Subroutine Requirements

User Subroutine	Required Parameters or Model Definition Options	Purpose
CREDE	THERMAL LOADS	Definition of state variable including temperature.
CUPLFX	COUPLE DIST FLUXES (flux type 101)	Heat generated due to inelastic behavior in coupled analysis.
DIGEOM	CONTACT (2D) CONTACT (3D) Table 3-8, "User Subroutines for Contact Problems"	Definition of rigid surface.
FILM	HEAT or COUPLE FILMS (Model Definition) FILMS (History Definition)	Definition of convective heat transfer coefficient and sink temperature.
FLOW	HEAT CHANNEL	Definition of mass flow rate.
FLUX	HEAT or COUPLE DIST FLUXES (Model Definition) DIST FLUXES (History Definition)	Definition of distributed flux.
FORCDF	FORCDT FIXED DISP or DISP CHANGE	Definition of point load or kinematic boundary condition in a harmonic analysis.

Table 2-1 User-defined Loading, Boundary Conditions, State Variables Subroutine Requirements (Continued)

User Subroutine	Required Parameters or Model Definition Options	Purpose
FORCDT	FORCDT FIXED DISP or DISP CHANGE FIXED TEMPERATURE or TEMP CHANGE	Definition of point load or prescribed displacement in stress analysis. Definition of point flux or prescribed temperature in heat transfer analysis.
FORCEM	DIST LOAD (Model Definition) DIST LOAD (History Definition)	Definition of distributed load.
GAPT	HEAT CONRAD GAP	Definition of thermal contact gap temperature.
INITPL	INITIAL PLASTIC STRAIN	Definition of initial plastic strain.
INITPO	PORE INITIAL PORE	Definition of initial pore pressure in a uncoupled soil analysis.
INITSV	INITIAL STATE	Definition of initial values of state variables.
MOTION	CONTACT (2D) CONTACT (3D) Table 3-8, "User Subroutines for Contact Problems" UMOTION MOTION CHANGE (History Definition)	Definition of velocity of rigid surfaces.
NEWPO	PORE CHANGE PORE (Model Definition) CHANGE PORE (History Definition)	Change pore pressure in an uncoupled soil analysis.
NEWSV	CHANGE STATE (Model Definition) CHANGE STATE (History Definition)	Change value of the state variable.
SEPFOR	CONTACT (2D) CONTACT (3D) Table 3-8, "User Subroutines for Contact Problems"	Definition of force required for separation.
SEPSTR	CONTACT (2D) CONTACT (3D) Table 3-8, "User Subroutines for Contact Problems"	Definition of stress required for separation.
UCONTACT	CONTACT (2D) CONTACT (3D) Table 3-8, "User Subroutines for Contact Problems" UCONTACT	Allow user-defined contact procedure.

Table 2-1 User-defined Loading, Boundary Conditions, State Variables Subroutine Requirements (Continued)

User Subroutine	Required Parameters or Model Definition Options	Purpose
UFORMS	TYING	Definition of user-defined constraint matrices.
UFOUR	FOURIER	Definition of function giving nonuniform variation about the circumference in Fourier analysis.
UFRIC	CONTACT (2D) CONTACT (3D) Table 3-8, "User Subroutines for Contact Problems" UFRICTION	Definition of friction coefficient.
UHTCOE	CONTACT (2D) CONTACT (3D) Table 3-8, "User Subroutines for Contact Problems" UHTCOEF	Definition of heat transfer coefficient to environment for coupled contact analysis.
UHTCON	CONTACT (2D) CONTACT (3D) Table 3-8, "User Subroutines for Contact Problems" UHTCON	Definition of heat transfer coefficient between bodies in contact in coupled analysis.
UINSTR	ISTRESS	Definition of initial stress.
UNORST	CONTACT (2D) CONTACT (3D) Table 3-8, "User Subroutines for Contact Problems" USER	Definition of normal stress for user elements in contact.
USDATA	USDATA	Definition of user-definer constants.
USINC	INITIAL DISP INITIAL VEL INITIAL TEMP	Definition of initial displacement, initial velocity, or temperature.
USSD	DYNAMIC CHANGE RESPONSE SPECTRUM	Definition of spectrum displacement density function.
UVELOC	HEAT	Definition of convective velocities.

■ FORCEM

Input of Nonuniform Distributed Loads

Description

This subroutine allows input of nonuniform distributed loads. This routine can be used to specify the load magnitude as a function of coordinate position and/or time.

Subroutine FORCEM is called during the calculation of the equivalent nodal loads, at each integration point needed to calculate the loads specified in the DIST LOADS option regardless of the use of the ALL POINTS or CENTROID parameters. The use of this subroutine is flagged by the appropriate load type in the DIST LOADS input option where the type chosen depends on the element type (see *Volume B, Element Library*).

Format

The definitions in FORCEM depend on the element dimensionality as follows:

For two-dimensional elements:

```
SUBROUTINE FORCEM (P,X1,X2,NN,N)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION N(7)
   user coding
RETURN
END
```

where P is the magnitude of the distributed load to be defined by you in this subroutine at the integration point being evaluated. The following values are given:

X1	is the first coordinate of the integration point.
X2	is the second coordinate of the integration point.
NN	is the integration point number.
N(1)	is the element number.
N(2)	is the parameter identifying the type of load.
N(3)	is the integration point number.
N(4)	is not used.
N(5)	is the distributed load index.
N(6)	is not used.
N(7)	is the internal element number.

For three-dimensional elements and shell element types 22, 49, 72, 75, 138, 139, and 140, the required headers are:

```
SUBROUTINE FORCEM (P,X1,X2,NN,N)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION X1(3), X2(3), N(7)
```

where P is the magnitude of the distributed load at this point, to be defined by you. In cases where a direction is also needed, X2 must also be given as the (x,y,z) components of a vector in the load direction. The following values are provided:

X1(3)	is the (x,y,z) position of the integration point.
X2(3)	is the vector describing direction of load.
NN	is the integration point number.
N(1)	is the element number.
N(2)	is the parameter identifying the type of load.
N(3)	is not used.
N(4)	is not used.
N(5)	is the distributed load index.

Load Control	User Supplies
AUTO LOAD AUTO LOAD with FOLLOW FOR	INCREMENTAL PRESSURE PRESSURE END OF INCREMENT
AUTO TIME AUTO TIME with FOLLOW FOR	INCREMENTAL PRESSURE PRESSURE END OF PERIOD
AUTO INCREMENT AUTO INCREMENT with FOLLOW FOR	PRESSURE END OF PERIOD PRESSURE END OF PERIOD (KPPASS = 1) PRESSURE BEGINING IN INCREMENT (KPPASS = 2

Subroutine FORCEM is called twice per increment when AUTO INCREMENT and FOLLOW FOR are used together in the analysis.

The reading of data is not recommended in FORCEM since this routine is in the recycling loop for nonlinear analysis, and you cannot know how many times per increment it is called.

Examples

It is often useful to have the distributed load vary with time in a dynamic analysis. To obtain the current time and increment of time add:

```
include 'xxx/common/creeps'
```

where:

CPTIM	is the time at the beginning of the increment
TIMINC	is the increment of time.

are variables in this common block.

To obtain transient time corresponding to heat transfer analysis where temperatures are read in using the CHANGE STATE/AUTO THERM option, add:

```
include 'xxx/common/heattm'
```

where:

CUTIME	is transient time at the beginning of the current increment from the heat transfer analysis.
DUTIME	is the time increment during the current increment from the heat transfer analysis.

To obtain the increment number add:

```
include 'xxx/common/concom'
```

where:

INC	is the current increment number.

In the example shown below, a beam is given a linearly varying distributed load.

$$P(X) = \left(\frac{MAX}{LEN}\right) X \qquad 0 \leq X \leq LEN$$

where LEN is the length of the beam and MAX is the load intensity at X=LEN.

The resulting user subroutine is as follows:

```
SUBROUTINE FORCEM  (P,X1,X2,NN,N)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION N(7)
REAL LEN,MAX
MAX =
LEN =
P = X1* MAX/LEN
RETURN
END
```

■ FLUX

Input of Nonuniform Fluxes

Description

For heat transfer analysis, this subroutine allows surface or body fluxes to be specified as functions of time, temperature, or position. The use of this subroutine is flagged by the appropriate flux type in the DIST FLUXES input option where the type chosen depends on element type (see *Volume B, Element Library*).

Format

Subroutine FLUX is written with the following headers:

```
SUBROUTINE FLUX(F,TS,N,TIME)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION TS(6), N(7)
   user coding
RETURN
END
```

where

`F`	is the surface or volumetric flux, to be defined at this integration point in this subroutine.
`TS(1)`	is the estimated temperature at the end of the increment.
`TS(2)`	is the current values of the area under the volumetric flux versus time curve, that is, $\int_0^t Qdt$. This total includes all uniform and nonuniform volumetric fluxes.
`TS(3)`	is the temperature at the beginning of the increment.
`TS(4), TS(5), TS(6)`	are the integration point coordinates.
`N(1)`	is the element number.
`N(2)`	is the parameter identifying the type of flux.
`N(3)`	is the integration point number.
`N(4)`	is the flux index.
`N(5)`	is not used.

N(6)	1 - heat transfer. 2 - joule. 3 - bearing. 4 - electrostatic. 5 - magnetostatic. 6 - acoustic.
N(7)	is the internal element number.
TIME	is the current time.

This subroutine is called at each time step for each integration point and element listed with an appropriate FLUX type in the DIST FLUXES input option.

The reading of data is not recommended in FLUX since this routine is in the recycling loop, and you cannot know how many times per increment it is called.

■ CUPFLX

Coupling of Inelastic Energy and Internal Heat Generation

Description

This subroutine allows you to modify the default routine for the calculation of the internal heat generated due to inelastic energy dissipation. This routine is only used if a coupled thermal-mechanical analysis is being performed and a DIST FLUXES type 101 is chosen.

Format

Subroutine CUPFLX is written with the following headers:

```
SUBROUTINE CUPFLX (F,TS,N,TIME,TIMINC,TOTPLE,DIFPLE,DEN,FCMECH)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION TS(1), N(1)
        user coding
RETURN
END
```

where

F	is the volumetric flux to be defined by you.
TS(1)	is the estimated temperature at the end of the increment.
TS(3)	is the temperature at the beginning of the increment.
TS(4), TS(5), TS(6)	are the integration point coordinates.
N(1)	is the element number.
N(2)	is 101.
N(3)	is the integration point number.
N(7)	is the internal element number.
TIME	is the time at the beginning of increment.
TIMINC	is the time increment.
TOTPLE	is the total plastic strain energy.
DIFPLE	is the incremental plastic strain energy.
DEN	is the mass density.
FCMECH	is the factor entered through the CONVERT model definition option.

■ UINSTR

Input of Initial State of Stresses

Definition

This subroutine is called in a loop over all the elements in the mesh when the ISTRESS parameter is used. Note that this routine is called twice for each point. During the first call, the user-defined stress vector S is used to define the net nodal force. During the second call, the user-defined stress vector S is used to define the initial stress at each point. In a rigid-plastic analysis, this routine is called at every increment; otherwise, only in increment zero.

Format

Subroutine UINSTR is written with the following headers:

```
      SUBROUTINE UINSTR (S,NDI,NSHEAR,N,NN,KC,XINTP,NCRD,
     +INC,TIME,TIMEINC)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION S(1), XINTP(NCRD), N(2)
          user coding
      RETURN
      END
```

where:

S	is the stress vector defined by you.
NDI	is the number of direct stress components.
NSHEAR	is the number of shear stress components.
N(1)	is the user element number.
N(2)	is the internal element number.
NN	is the integration point number.
KC	is the layer number (shells or beams).
XINTP	is the array of integration point coordinates.
NCRD	is the number of coordinates.
INC	is the increment number.
TIME	is the total time at beginning of increment.
TIMEINC	is the incremental time.

■ UFOUR

Input of a User-defined Function F(Θ) for Fourier Analysis

Description

This subroutine allows input of a function F(Θ) where it can be expressed analytically. The values of F(Θ) are then passed into a MARC routine that calculates the Fourier expansion coefficients.

Format

Subroutine UFOUR is written with the following headers:

```
SUBROUTINE UFOUR (F,N,NS)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION F(1)
   user coding
RETURN
END
```

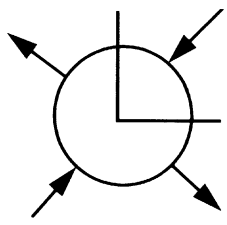
where

F	The F-array should contain the N values of F (Θ) in sequential order starting at Θ = 0° and ending with Θ = 360°. You specify the N values of F(Θ) in degrees sequentially from 0 to 360° in positions N + 1 through 2N of the F-array.
N	The number of stations around the circumference for which the function value F is specified. N is to be defined by you.
NS	The number of the Fourier series.

Example

For example, suppose the following function is to be expanded in a Fourier series:

$$F(\Theta) = \begin{cases} 1 & \Theta = 135°,\ 315° \\ -1 & \Theta = 45°,\ 225° \\ 0 & \text{elsewhere.} \end{cases}$$



This might be accomplished through the following code for subroutine UFOUR which calculates F(Θ) for 25 values of Θ from 0° to 360° by 15°.

```
      SUBROUTINE UFOUR (F,N,NS)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION F(1)
C
      DO 10 I=1,N
      F(I)=0
      F(I+N) = (I-1)*15
10    CONTINUE
      F(4)  = -1.0
      F(10) = +1.0
      F(16) = -1.0
      F(22) = +1.0
C
      RETURN
      END
```

The subroutine UFOUR is called by using the following model definition option:

```
FOURIER
0,0,25,
```

■ FORCDT

Input of Time Dependent Nodal Loads, Displacements or Time Dependent Nodal Fluxes, Temperatures for Heat Transfer

Stress Analysis

Description

Simple time dependent load or displacement histories can be input on data lines. However, in more general cases, when the load history is complex, it is often more convenient to input the history through a user subroutine. For distributed loads, this is achieved with subroutine FORCEM; for point loads, it is achieved via subroutine FORCDT.

This subroutine is flagged by introducing a model definition set, FORCDT, listing the node numbers for which this routine is called. Then, at each increment of the analysis, for each of the nodes on the list, the subroutine is called. In static analyses, displacement and load arrays are available and, for dynamics, velocity and acceleration analyses are also given. For nodes without kinematic boundary conditions, you can define increments of point loads (thus overwriting any point load input at the same nodes in the POINT LOAD block). For nodes with kinematic boundary conditions (that is, listed in the FIXED DISP or DISP CHANGE blocks), you can define increments of displacement.

Note: FORCDT cannot be used to modify Fourier type boundary conditions.

Format

Subroutine FORCDT is written with the following headers:

```
      SUBROUTINE FORCDT (U,V,A,DP,DU,TIME,DTIME,NDEG,NODE,
     1 UG,XORD,NCRD,IACFLG,INC, IPASS)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION U(NDEG),V(NDEG),A(NDEG),DP(NDEG),DU(NDEG),UG(1),XORD(1)
       user coding
      RETURN
      END
```

where

U	is the array of total displacements at this node.
V	is the array of total velocities at this node (dynamics only).
A	is the array of total accelerations at this node (dynamics only).
DP	is the array of incremental point loads at this node – can be set by you at degrees of freedom without kinematic boundary conditions.
DU	is the array of incremental displacements at this node or is the array of total accelerations at this node, see IACFLG – can be set by you for degrees of freedom listed as having kinematic boundary conditions.
TIME	is the total time (only relevant for dynamics or creep) at the beginning of the increment.
DTIME	is the increment of time (only relevant for dynamics or creep).
NDEG	is the number of degrees of freedom per node.
NODE	is the global node number.
UG	is the array of total displacements in the global system.
XORD	is the array of original nodal coordinates.
NCRD	is the number of coordinates per node.
IACFLG	is set to 1 if accelerations are prescribed in dynamic analysis.
INC	is the increment number.
IPASS	= 0 conventional analysis. = 1 stress portion of coupled analysis. = 2 heat portion of coupled analysis.

To obtain transient time corresponding to heat transfer analysis, where temperatures are read in using the CHANGE STATE/AUTO THERM option, add

```
include 'xxx/common/heattm'
```

where

CUTIME	is the time at the beginning of the current increment from heat transfer analysis.
DUTIME	is the change in time during current increment from heat transfer analysis.

are variables in the common block heattm.

As an example, suppose a sinusoidal forcing is required at the third degree of freedom at a node.

The forcing function is

$$P = B \sin \omega t$$

so

$$dp = B(\sin \omega (t + dt) - \sin \omega t)$$

Hence, we write the routine as follows:

```
      SUBROUTINE FORCDT (U,V,A,DP,DU,TIME,DTIME,NDEG,NODE,
     1 UG,XORD,NCRD,IACFLG,INC, IPASS)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION U(1),V(1),A(1),DP(1),DU(1),UG(1),XORD(1)
      B =
      OMEGA =
      DP(3) = B*(SIN(OMEGA*(TIME+DTIME)) - SIN(OMEGA*TIME))
      RETURN
      END
```

Heat Transfer Analysis

Description

Time dependent nodal fluxes or temperature boundary conditions can be input most conveniently through the use of subroutine FORCDT. For distributed fluxes, subroutine FLUX should be used to input the value of the distributed flux as a function of time and position.

Subroutine FORCDT is flagged by a model definition set, FORCDT, listing the node numbers. Then at each step in the analysis, for each of the nodes in the list, the subroutine is called. The current, calculated temperature is provided at the nodes. For nodes not specified as having temperature boundary conditions, you can give the point flux. For those nodes specified with temperature, boundary conditions (in FIXED TEMPERATURE or TEMP CHANGE) sets the temperature.

Format

Subroutine FORCDT is written with the following headers:

```
      SUBROUTINE FORCDT (X1,X2,X3,F,T,TIME,DTIME,NDEG,NODE,X4,
     1 XORD,NCRD,IACFLG,INC,IPASS)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION F(NDEG),T(NDEG),XORD(NCRD)
            user coding
      RETURN
      END
```

where

X1,X2,X3	are not used.
F	is the array of fluxes at the node – can be re-defined for nodes free of temperature boundary conditions.
T	is the array of temperatures at the node – can be redefined for nodes having temperature boundary conditions.
TIME	is the total time at the end of the current step.
DTIME	is the current time increment.
NDEG	is 1 unless heat transfer shell elements are used.
NODE	is the global node number.
X4	is not used.
XORD	is the array of nodal coordinates.
NCRD	is the number of coordinates per node.
IACFLG	is not used.
INC	is the increment number.
IPASS	= 0 conventional analysis. = 2 heat transfer portion of coupled analysis.

■ FORCDF

Input of Frequency Dependent Loads or Displacements in Harmonic Analysis

Description

Simple nodal load or displacement excitations can be input on data lines. However, in more general cases, when the load is nonhomogeneous, it is often more convenient to input the excitation through a user subroutine. For distributed loads, this is achieved with subroutine FORCEM; for point loads or displacements, it is achieved via subroutine FORCDF.

This subroutine is flagged by introducing a model definition set, FORCDT, listing the node numbers for which this routine is called. Then, at each harmonic sub-increment of the analysis, for each of the nodes on the list, the subroutine is called. For nodes without kinematic boundary conditions, you can define increments of point loads (thus, overwriting any point load input at the same nodes in the POINT LOAD block). For nodes with kinematic boundary conditions (that is, listed in the FIXED DISP or DISP CHANGE blocks), you can define increments of harmonic displacement.

Format

Subroutine FORCDF is written with the following headers:

```
      SUBROUTINE FORCDF (U,FR,FI,DUR,DUI,FREQ,DTIME,NDEG,NODE,
     1 UG,XORD,NCRD,ICOMPL,INC,INCSUB)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION (NDEG),FR(NDEG),FI(NDEG),DUR(NDEG),DUI(NDEG),UG(1),
     +XORD(1)
       user coding
      RETURN
      END
```

where

U	is the array of total displacements at this node.
FR	is the array of the real components of the harmonic point loads.
FI	is the array of the imaginary components of the harmonic point loads.
DUR	is the array of the real components of the harmonic displacements.

DUI	is the array of the imaginary components of the harmonic displacements.
FREQ	is the excitation frequency.
DTIME	is not used.
NDEG	is the number of degrees of freedom per node.
NODE	is the global node number.
UG	is the array of total displacements in the global system.
XORD	is the array of original nodal coordinates.
NCRD	is the number of coordinates per node.
ICOMPL	is 0 if real analysis; 1 if complex analysis.
INC	is the increment number.

■ FILM

Input of Nonuniform Film Coefficients

Description

In heat transfer analysis, it is often necessary to include nonuniform film coefficients and sink temperatures for the calculation of convection or radiation boundary conditions. Subroutine FILM facilitates this. It is called at each time step for each integration point on each element surface given in the FILMS model definition set, and allows you to modify the film coefficient and sink temperature that is input through the data lines. In coupled contact analyses, user subroutines UHTCOE and UHTCON are preferred

Format

Subroutine FILM is written with the following headers:

```
SUBROUTINE FILM (H,TINF,TS,N,TIME)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION N(7),TS(6)
   user coding
RETURN
END
```

where:

H	is the ratio of the desired film coefficient to that given on the FILMS data set for this element to be defined by you (preset to 1).
TINF	is the ratio of the desired sink temperature to that given on the FILMS data set for this element to be defined by you (preset to 1).
TS(1)	is the estimated surface temperature at the end of the increment.
TS(2)	is the surface temperature at the beginning of the increment.
TS(3)	is not used.
TS(4)	is the integration point 1st coordinate.
TS(5)	is the integration point 2nd coordinate.
TS(6)	is the integration point 3rd coordinate.
N(1)	is the element number.
N(2)	is the IBODY code.

N(3)	is the integration point number.
N(4)	is the film index.
N(5)	is the sink temperature index.
N(6)	is not used.
N(7)	is the internal element number.
TIME	is the current time.

Note that since H and TINF are defined as ratios, if you do not re-define them in this routine, the data set values are used. If you wish to give absolute values here, the corresponding values on the FILMS data set can be conveniently set to 1.

■ FLOW

Input of Mass Flow Rate and Inlet Temperature

Description

In a heat transfer analysis involving fluid channel elements, user subroutine FLOW is available to you for the modification of mass flow rate, inlet temperature, and film coefficient. Both the inlet temperature and mass flow rate can be dependent on time; the film coefficient can also be a function of streamline distance.

Format

Subroutine FLOW is written with the following header:

```
 SUBROUTINE FLOW (II,IFACE,N1,NBSURF,STOT,RATE,TINLET,SURFJ,TSURJ,
+HJ,TFLUID,TIMINC,CPTIME)
 IMPLICIT REAL *8 (A-H, O-Z)
 DIMENSION SURFJ(4),TSURJ(4),HJ(4)
   user coding
 RETURN
 END
```

where:

II	is the channel number.
IFACE	is the channel face identification, defining the flow direction.
N1	is the fluid channel element number.
NBSURF	is the number of channel surfaces.
STOT	is the total stream line distance.
RATE	is the mass flow rate (redefined by you in this subroutine).
TINLET	is the inlet temperature (redefined by you in this subroutine).
SURFJ(I)	is the channel surface area array.
TSURJ(I)	is the channel surface temperature array.
HJ(I)	is the film coefficient of the ith surface (redefined by you in this subroutine).

TFLUID	is the fluid element temperature.
TIMINC	is the time increment.
CPTIME	is the current total time.

In two-dimensional analyses, SURFJ(1), SURFJ(2) are the lengths of the edges bordering the channel element. In three-dimensional analyses, SURFJ(1) through SURFJ(4) are the areas on adjacent faces. In a similar manner, TSURJ is the average temperature on adjacent edges (for 2D) or adjacent faces (for 3D).

■ GAPT

Input of Thermal Contact (Conrad) Gap Temperature

Description

In a heat transfer analysis involving thermal contact (CONRAD) gap elements, the gap temperature is compared with a given gap closure temperature for the determination of gap open/closed condition. In MARC, the gap temperature is estimated from the average of gap nodal temperatures 0.5*(T1 + T2) and the gap closure temperature is entered through the model definition CONRAD GAP option.

The user subroutine GAPT allows for the redefinition of gap temperature (TGAP) based on the nodal temperatures T1 and T2. If the gap temperature (TGAP) is greater than or equal to the gap closure temperature (TCLOSE), the gap is closed. Otherwise, the gap is open.

Format

Subroutine GAPT is written with the following header:

```
SUBROUTINE GAPT(N,I1,I2,T1,T2,TCLOSE,TGAP,INC,TIME,TIMINC)
IMPLICIT REAL *8 (A-H, O-Z)
   user coding
RETURN
END
```

where:

N	is the gap (tie) number.
I1,I2	are the nodal numbers.
T1,T2	are the nodal temperatures.
TCLOSE	is the gap closure temperature.
TGAP	is the gap temperature (to be defined you).
INC	is the increment number.
TIME	is the total transient time.
TIMINC	is the time increment.

■ UFORMS

Definition of Constraint Conditions

Description

Subroutine UFORMS allows the definition of a constraint condition. The program's capability for applying arbitrary homogeneous constraints between nodal displacements is used through this routine. To distinguish user constraints from the program's built-in constraints, those constraints formed by you in UFORMS must be of type less than zero (ISTYP in the subroutine: first field of data block 3 of the TYING model definition option). The constraint conditions can be supplied by using the subroutine UFORMS. The conventions adopted for these constraints are:

1. A constraint is defined by:

$$\{u^a\} = [S]\begin{Bmatrix} u^b \\ u^c \\ \text{etc} \end{Bmatrix}$$

where:

$\{u^a\}$ The vector of displacement at node *a*, referred to as the tied mode.

$\begin{Bmatrix} u^b \\ u^c \\ \text{etc} \end{Bmatrix}$ Vector of displacements at *b*, *c*, etc.; these nodes are referred to as the retained nodes.

2. In the matrix [S], a row of zeros indicates that particular degree of freedom at node a is not constrained.
3. To apply a constraint between degrees of freedom at the same node, the node must appear on both sides of the equation, with rows of zeros in [S] corresponding to the degrees of freedom on the left-hand side, which are retained on the right-hand side, and columns of zeros in [S] corresponding to the tied nodes appearing on the left-hand side.

Note: When the retained nodes have transformations applied to them, the constraint matrix, S, is written with respect to the transformed displacements.

Format

The subroutine supplying the [S] matrix must have the following headers:

```
      SUBROUTINE UFORMS(S,NRETN,LONG,NDEG,ISTYP,ITI,ISTART,ITIE,
     1 LONGSM,ITIEM,LEVEL,NUMNP,DICOS,TRANSM,XORD,NPBT,NBCTRA,
     2 NCRD,TDICOS,LEVELM,II,LONGTM,DISP,ITYFL)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION S(NDEG,LONGSM),ITI(LONGTM,ITIEM)
      DIMENSION DICOS(NDEG,NDEG),TRANSM(6,1),
     1 XORD(NCRD,LONGTM),NPBT(1),TDICOS(NDEG,NDEG),
     2 DISP(NDEG,LONGTM)
            user coding
      RETURN
      END
```

where:

S	is the constraint matrix to be defined by you (dimension (NDEG, LONGSM), LONGSM = NDEG* (number of retained nodes)).
NRETN	is the number of retained nodes for this type of tying.
LONG	is NDEG*NRETN.
NDEG	is the number of degrees of freedom at a node.
ISTYP	is the type of this constraint equation (given in TYING set of blocks). The program adds 1000 to all ISTYP that are less than -1000 before calling UFORMS; any use of ISTYP in UFORM should account for this.
ITI(1, II)	is the node on left-hand side of this (the IIth) constraint equation.
ITI(2,II), ITI(3,II), etc.	are the nodes on the right-hand side of this constraint equation.
ISTART	is not used.
ITIE	is the number of constraint equations.
LONGSM	is the size of constraint matrix = ITIEM*NDEG.
ITIEM	is the maximum number of constraint equations.
LEVEL	is not used.
NUMNP	is the number of nodal points in mesh.
DICOS	is the work space.
TRANSM(6, 1)	are supplied as data (given in TRANSFORMATION set of blocks).
XORD(N1, N)	are the N1th coordinate of node ITI(N,II).
NPBT(L)	is the information about Lth boundary condition transformation.

NBCTRA	is the number of nodes with transformations.
NCRD	is the number of coordinate directions.
TDICOS	is the work space.
LEVELM	is not used.
II	is the tying number.
LONGTM	is the maximum number of retained nodes plus one.
DISP(N1,N)	is the N1th total displacement of node ITI(N,II) if ISTYP is less than 1000.
ITYFL	is set to zero to remove this tie constraint; note that this only works if re-assembly is forced.

Example

Suppose a change from a coarse to a fine mesh of two-dimensional isoparametric elements is required. For any node in the fine mesh which does not correspond to a node in the coarse mesh, a constraint is necessary. The displacement at these nodes can be expressed as a linear combination of the displacements of the two corner nodes of the coarse mesh since the displacement is linear between these nodes due to the element formulation.

In the coarse mesh:

$$\begin{Bmatrix} u_j \\ v_j \end{Bmatrix} = \begin{bmatrix} 1-\lambda & 0 & \lambda & 0 \\ 0 & 1-\lambda & 0 & \lambda \end{bmatrix} \begin{Bmatrix} \begin{bmatrix} u_i \\ v_i \end{bmatrix} \\ \begin{bmatrix} u_l \\ v_l \end{bmatrix} \end{Bmatrix}$$

where:

$$\lambda = x_{ij}/x_{il}$$

Thus, we supply the following subroutine:

```
      SUBROUTINE UFORMS(S,NRETN,LONG,NDEG,ISTYP,ITI,ISTART,ITIE
     1 LONGSM,ITIEM,LEVEL,NUMNP,DICOS,TRANSM,XORD,NPBT,NBCTRA,
     2 NCRD,TDICOS,LEVELM,II,LONGTM,DISP,ITYFL)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION S(NDEG,LONGSM),ITI(LONGTM,ITIEM)
      DIMENSION DICOS(NDEG,NDEG),TRANSM(6,1),
     1 XORD(NCRD,LONGTM),NPBT(1),TDICOS(NDEG,NDEG),
     2 DISP(NDEG,LONGTM)
```

```
J=1
I=2
L=3
XIJ = SQRT((XORD(1,I)-XORD(1,J))**2+(XORD(2,I)- XORD(2,J))**2)
XIL = SQRT((XORD(1,I)-XORD(1,L))**2+(XORD(2,I)- XORD(2,L))**2)
XLAMBD = XIJ/XIL
S(1,1) = 1. -XLAMBD
S(2,2) = 1. -XLAMBD
S(1,3) = XLAMBD
S(2,4) = XLAMBD
      user coding
RETURN
END
```

Assuming that nodes `j` and `k` are located between nodes `I` and `L` and nodes `m`, `n` are located between nodes `L` and `P`, the constraint is then imposed by specifying `j`, `k`, `m`, `n`, etc., on data lines as the tied nodes, and `I, L; I, L; L, P; L, P; etc.`, as the corresponding pairs of retained nodes. The TYING option would then become:

```
TYING
4,
-1,j,2
i,l,
-1,k,2
i,l,
-1,m,2
l,p,
-1,n,2
l,p,
```

Note that this coarse to fine mesh tying constraint is in the program as default tying types 31 and 32 for planar elements and as tying type 33 and 34 for three-dimensional brick elements. See *Volume A: User Information* for further details.

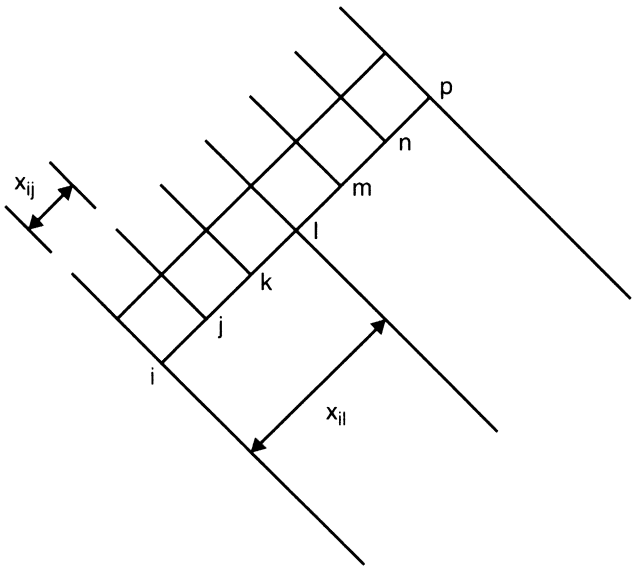


Figure 2-1 Coarse to Fine Mesh Example

■ CREDE

Input of Pre-specified State Variables

Description

Subroutine CREDE is available to you for the input of pre-specified state variables. The simplest option allows the specification of temperature increments throughout the mesh. Through the use of the STATE VARS parameter, the number of state variables per point in the structure can be increased. For example, radiation fluxes (in reactor core problems) can be included. The program always assumes temperature is the first state variable given at a particular point, since the first state variable is used in conjunction with the tables of temperature dependence input specified in the TEMPERATURE EFFECTS option, and the first state variable is used to compute thermal strains. All state variables are available to all constitutive routines.

Subroutine CREDE is called once per element in a loop over the elements when the THERMAL LOADS option is used. Any data blocks required should appear immediately after data block 2 of the THERMAL LOADS option in the input data. If the first field of data block 2 in the THERMAL LOADS option is a 3, total state variable values must be provided at all points of all elements at which constitutive calculations are made. If the first field is a 2, the incremental values are defined. Depending on the inclusion of the CENTROID or ALL POINTS parameters, centroidal values or values at all numerical integration points of an element are expected. For shell elements, the values of state variable increments must be given for each layer through the thickness at every integration point. For beam elements, the values of state variable increments must be given at all points used to define the beam section (16 for default element type 14, 25, 76 or 78; user-defined for element type 13, 77, or 79).

Format

Subroutine CREDE is written with the following headers:

```
SUBROUTINE CREDE (DTDL,M,NSTRES,NEQST,NSTATS)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION DTDL (NSTATS,NEQST,NSTRES)
       user coding
RETURN
END
```

where:

DTDL	is the array of state variable increments or total values (to be defined here by you).
M	is the element number and must remain unchanged in CREDE.
NSTRES	is the maximum number of integration points per element, if ALL POINTS was included in the parameters, and is 1 if the CENTROID parameter is used.
NEQST	is the maximum number of layers per element.
NSTATS	is the number of state variables requested by you in the STATE VARS parameter. (This number equals 1 if only temperature is required).

For meshes with several element types, NEQST and NSTRES take on maximum values, but the DTDL array need only be filled as far as necessary for a particular element type.

Example

As an example, suppose a linear gradient through the thickness is to be imposed on a shell with NSTATS = 1. The same gradient is imposed throughout the structure.

The following coding will suffice:

```
      SUBROUTINE CREDE (DTDL,M,NSTRES,NEQST,NSTATS)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION DTDL (NSTATS,NEQST,NSTRES)
      TOUT=500.0
      TIN=300.0
      T=TIN
      DT=(TOUT-TIN)/FLOAT(NEQST-1)
      DO 2 I=1, NEQST
      DO 1 J=1, NSTRES
    1 DTDL (1,I,J)=T
    2 T=T+DT
      RETURN
      END
```

■ INITSV

Initialize State Variable Values

Description

This subroutine is called in a loop over all the elements in the mesh when the INITIAL STATE option appears in the model definition blocks with a 2 in the second field of the second data block of that option. It allows you to define initial values of state variables.

Format

Subroutine INITSV is written with the following headers:

```
SUBROUTINE INITSV(SV,LAYERS,INTPTS,M,ID)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION SV(LAYERS,INTPTS)
   user coding
RETURN
END
```

where:

SV	is the array of values of this state variable; to be defined here for this element by you.
LAYERS	is the number of layers through the thickness if this is a shell element, or the number of points in the cross-section if this is a beam element. It is 1 for a continuum element.
INTPTS	is the number of integration points in this element if the ALL POINTS parameter is used. If the CENTROID parameter is used, INTPTS = 1.
M	is the element number.
ID	is the state variable number (from columns 1-5 of the second card series of the INITIAL STATE set).

■ NEWSV

Input New State Variable Values

Description

This subroutine is called in a loop over all the elements in the mesh when the CHANGE STATE option appears in the model definition or the history definition set with a 2 in the second field of the second data block of that option. It allows the new values of any state variable to be defined for the end of the current step.

Format

Subroutine NEWSV is written with the following headers:

```
SUBROUTINE NEWSV(SV,LAYERS,INTPTS,M,ID)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION SV(LAYERS,INTPTS)
   user coding
RETURN
END
```

where:

SV	is the array of new values of this state variable; to be defined here for this element by you.
LAYERS	is the number of layers through the thickness if this is a shell element, of the number of points in the cross-section if this is a beam element. It is 1 for a continuum element.
INTPTS	is the number of integration points in this element if the ALL POINTS parameter is used. If the CENTROID parameter is used, INTPTS=1.
M	is the element number.
ID	is the state variable number (from columns 1-5 of the second data block of the CHANGE STATE set [model definition or history definition]).

■ USSD

Input of Spectral Response Density

Description

Subroutine USSD allows you to input the spectral density function for the frequencies required in the spectrum response calculation. These frequencies are obtained by performing a modal analysis.

Format

Subroutine USSD is written with the following headers:

```
SUBROUTINE USSD(SD,OMEG,I)
IMPLICIT REAL *8 (A-H, O-Z)
   user coding
RETURN
END
```

where:

SD	is the spectral response density for the Ith degree of freedom to be defined by you.
OMEG	is the frequency in cycles per time unit.
I	is the degree of freedom.

■ USINC

Input of Initial Conditions

Description

This routine allows you to input initial displacements and velocities for dynamic stress analysis, initial temperatures for heat transfer analysis, or a spatially varying interference fit for contact analysis. You give the values for all degrees of freedom in vector F. This routine is used with either the INITIAL DISP, INITIAL VEL, or the INITIAL TEMP options. This routine is called for every node in the structure if it is used.

Format

Subroutine USINC is written with the following headers:

```
SUBROUTINE USINC(F,N,NDEG,IFLAG)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION F(NDEG)
   user coding
RETURN
END
```

where:

F	is the vector of initial conditions to be given by you.
N	is the node number.
NDEG	is the number of degrees of freedom per node.
IFLAG	is the type flag. = 1 initial displacement. = 2 initial velocities. = 3 initial temperatures.

■ USDATA

Input of Initial Data

Description

This subroutine is a mechanism to allow you to read data into a user-defined common block. This common block is stored on the restart file, and available in subsequent increments. The common block USDACM must be given the correct length in this routine. This common block can also be used in any other user subroutine.

Format

Subroutine USDATA is written with the following headers:

```
SUBROUTINE USDATA(KIN,KOU,IC)
COMMON/USDACM/MYDATA
IMPLICIT REAL *8 (A-H, O-Z)
   user coding
RETURN
END
```

where:

KIN	is the unit number for input, usually 5.
KOU	is unit number for output, usually 6.
IC	is the reader flag. = 1 pre-reader. = 2 real reader.

Note that the maximum length of USDACM should be defined here. It should agree in length in real *4 words as with that given on the USDATA model definition option.

UVELOC

Generation or Modification of Nodal Velocity Vectors

Description

In heat diffusion-convection, it is sometimes necessary to include a position dependent velocity field. Subroutine UVELOC, which is called for each node, allows you the specification or re-definition of previously specified nodal velocity vectors. The inclusion of convection is activated on the HEAT parameter. This routine should not be used in a coupled fluid-thermal analysis, as the velocities are calculated by the program

Format

Subroutine UVELOC is written with the following headers:

```
SUBROUTINE UVELOC (VELOC,COORD,NCRD,NODE)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION VELOC (NCRD),COORD(NCRD)
   user coding
RETURN
END
```

where:

VELOC	is the array of nodal velocity components to be defined.
COORD	is the array of coordinates at this node.
NCRD	is the number of coordinates.
NODE	is the node number.

■ MOTION

Definition of Rigid Surface Motion for 2D Contact

Description

This subroutine allows the definition of nonuniform rigid surface motions, in conjunction with the CONTACT option. Its call is triggered by the model definition option UMOTION. This routine should only be used with velocity controlled rigid surfaces.

Subroutine MOTION is called during the calculations at the beginning of each time increment and you return the surface velocities for that increment. Imposed displacement increments at nodal points in contact with rigid surfaces are obtained from the velocity multiplied by the time increment. The surface path becomes an explicit forward integration of velocities. Therefore, caution should be taken when there are abrupt changes in surface path direction or abrupt changes in velocity by making time increments as small as necessary.

If, at the start of the analysis, a surface is placed apart from the body to be deformed, subroutine MOTION is also used in the approaching phase.

If two-dimensional elements are being used, the surfaces have rigid body motions in two dimensions. It is assumed that such motions can be defined by a translation of a point (the center of rotation), plus a rotation around that point.

Format

Subroutine MOTION is written with the following headers:

```
SUBROUTINE MOTION (X,F,V,TIME,DTIME,NSURF,INC)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION X(1),V(1),F(1)
    user coding
RETURN
END
```

where:

X(3)	is the current coordinates defining surface. X(1) = first coordinate of center of rotation. X(2) = second coordinate of center of rotation. X(3) = angle rotated around z-axis.

F(3)	is the current surface load. F(1) = first component of load. F(2) = second component of load. F(3) = moment.
V(3)	is the current surface velocities. V(1) = first component of center of rotation velocity. V(2) = second component of center of rotation velocity. V(3) = component of angular velocity:
TIME	is current time.
DTIME	is the increment of time applied.
NSURF	is the surface number to which data applied.
INC	is the increment number.
V(1),V(2),V(3)	are to be defined by you.

Example

Assume that a rigid surface is identified as surface number 1, and is moving in the negative x-direction with a velocity of 1.0. The user subroutine MOTION can be written as follows:

```
SUBROUTINE MOTION(X,F,V,TIME,DTIME,NSURF,INC)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION X(1),V(1),F(1)
IF(NSURF.GT.1) RETURN
V(1)=-1.
V(2)=0.
V(3)=0.
RETURN
END
```

■ MOTION

Definition of Rigid Surface Motion for 3D Contact

Description

This subroutine allows the definition of nonuniform rigid surface motions in conjunction with the CONTACT option. Its call is triggered by the model definition option UMOTION. This routine should only be used with velocity controlled rigid surfaces.

Subroutine MOTION is called during the calculations at the beginning of each time increment and your return surface velocities for that increment. Imposed displacement increments at nodal points in contact with rigid surfaces are obtained from the velocity multiplied by the time increment. The surface path becomes an explicit forward integration of velocities. Therefore, caution should be taken when there are abrupt changes in surface path direction or abrupt changes in velocity by making time increments as small as necessary.

If, at the start of the analysis, a rigid surface is placed apart from the deformable body, subroutine MOTION is also used in the approaching phase.

If three-dimensional elements are used, the surfaces have rigid body motions in three dimensions. It is assumed that such motions can be defined by a translation of a point (the center of rotation), plus a rotation about the axis of rotation through that point.

Format

Subroutine MOTION is written with the following headers:

```
SUBROUTINE MOTION (X,F,V,TIME,DTIME,NSURF,INC)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION X(1),V(1),F(1)
    user coding
RETURN
END
```

where:

`X(6)` is the current coordinates defining surface.
`X(1)` = first coordinate of center of rotation.
`X(2)` = second coordinate of center of rotation.
`X(3)` = third coordinate of center of rotation.
`X(4)` = first component of direction cosine of rotation axis.
`X(5)` = second component of direction cosine of rotation axis.
`X(6)` = third component of direction cosine of rotation axis.

F(6)	is the current surface load. F(1) = first component of load. F(2) = second component of load. F(3) = third component of load. F(4) = first component of moment W.R.T. center of rotation. F(5) = second component of moment W.R.T. center of rotation. F(6) = third component of moment W.R.T. center of rotation.
V(4)	is the current surface velocities. V(1) = first component of center of rotation velocity. V(2) = second component of center of rotation velocity. V(3) = third component of center of rotation velocity. V(4) = component of angular velocity.
TIME	is current time.
DTIME	is the increment of time applied.
NSURF	is the surface number to which data applied.
INC	is the increment number.
V(1),V(2),V(3),V(4)	are to be defined by you.

Example

Assume that a rigid surface is identified as surface number 2 and is moving in the negative x-direction with a velocity of 1.0. The user subroutine MOTION can be written as follows:

```
SUBROUTINE MOTION(X,F,V,TIME,DTIME,NSURF,INC)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION X(1),V(1),F(1)
IF(NSURF.NE.2) RETURN
V(1)=-1.
V(2)=0.
V(3)=0.
V(4)=0.0
RETURN
END
```

■ UFRIC

Definition of Friction Coefficients

Description

With this subroutine you can define the variable friction coefficients or friction factors in conjunction with the model definition option CONTACT. Its call is triggered by the option UFRICTION.

Subroutine UFRIC is called for every element containing nodes that are in contact with surfaces at the nodes. These calls are made every iteration both during the assembly phase and during the stress recovery phase.

In case of the variable `IFRIC = 1 or 3`, a constant shear friction model is enacted and you return a friction factor `m` defined in the equation:

$$\mathbf{f}_t = -m\ k_y\ \mathbf{t}$$

where:

f_t	is the shear friction force being applied.
m	is the friction factor.
k_y	is the shear flow stress of the material being deformed.
t	is the tangent unit vector in the direction of relative sliding velocity.

In case the variable `IFRIC = 2, 4, or 5`, a Coulomb friction model is enacted, and you return a friction coefficient μ defined in the equation:

$$\mathbf{f}_t = -\mu\ f_n\ \mathbf{t}$$

where:

μ	is the friction coefficient.
f_n	is the normal stress/force at the point of contact.

Format

Subroutine UFRIC is written with the following headers:

```
SUBROUTINE UFRIC (MIBODY,X,FN,VREL,TEMP,YIEL,FRIC,TIME,INC,NSURF)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION X(2),MIBODY(4),VREL(1)
   user coding
RETURN
END
```

where

For friction based on nodal stresses:

MIBODY(1)	is your element number.
MIBODY(2)	is the side number.
MIBODY(3)	is the surface integration point number.
MIBODY(4)	is the internal element number.

For friction based on nodal forces:

MIBODY(1)	is your node number.
MIBODY(2)	is not used; enter 0.
MIBODY(3)	is not used; enter 0.
MIBODY(4)	is the internal node number.
X	is the updated coordinates of contact point where friction is being calculated.
FN	is the normal stress/force being applied at that point.
VREL	is the relative sliding velocity at contact point.
TEMP	is the temperature of contact point.
YIEL	is the flow stress of workpiece material at contact point.
FRIC	is the friction coefficient or friction factor to be provided by you.
TIME	is the current time.
INC	is the increment number.
NSURF	is the surface being contacted by the side for which friction calculations are being made.

■ DIGEOM

Definition of 3D Rigid Surface Patch

Description

In three-dimensional problems in which complicated rigid surfaces need to be entered, it might be easier to define them with other software aids, such as a CAD system or an FEA pre-processor. In such cases, this subroutine lets you enter the geometry directly. This subroutine is used in conjunction with the CONTACT option for three-dimensional problems only.

Rigid surfaces are normally entered by means of several geometrical entities. If the discrete representation is used these are internally subdivided into 4-point patches. This user subroutine allows you to directly enter the coordinates associated with each patch.

Subroutine DIGEOM is called for every geometrical entity of type 7 (patch) for which the Fortran logical unit from where data is read is declared as -1.

Format

Subroutine DIGEOM is written with the following headers:

```
SUBROUTINE DIGEOM (IPATCH,NDIE,XYZ,NPATCH)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION XYZ(3,4)
   user coding
RETURN
END
```

where:

IPATCH	is the current patch number of this entity.
NDIE	is the surface (body) number.
XYZ	are the three (x, y, z) coordinates of the four points of the patch to be entered by you.
NPATCH	is the total number of patches defining this entity.

SEPFOR

Definition of Separation Force

Description

This subroutine allows the definition of the separation force in conjunction with the model definition option CONTACT. The separation forces, FNORM and FTANG, are either calculated by the program or entered through the CONTACT option, and then passed into this subroutine. You decide whether these values at the current increment are appropriate to determine whether separation occurs.

FNORM is the normal reaction force above which a node in contact separates from a surface. Any compressive or negative value indicates real contact while a positive reaction force indicates a tendency to separate. The default is taken as the maximum value of the residual force in the structure for the current increment. This value can be reset by you through the input format. Defining a too small value can result in an increased number of iterations. Defining a very large value eliminates the possibility of separation. FTANG is the tangential force used to determine whether a nodal point positioned at a convex corner of surface should be sliding from patch to patch or remaining on its current patch. The default value is half of FNORM. These two default reaction forces vary from increment to increment.

Format

Subroutine SEPFOR is written with the following headers:

```
SUBROUTINE SEPFOR (FNORM,FTANG,IBODY,NNODE,INC)
IMPLICIT REAL *8 (A-H, O-Z)
   user coding
RETURN
END
```

where:

FNORM	is the normal separation force to be supplied by you.
FTANG	is the tangential separation force to be supplied by you.
IBODY	is the current body number the node touched.
NNODE	is the current touched external node number.
INC	is the current increment number.

Example

If you desire the default separation force for surface 2, and do not want any separation of nodes from surface 3, the subroutine is written as follows:

```
      SUBROUTINE SEPFOR (FNORM,FTANG,IBODY,NNODE,INC)
      IMPLICIT REAL *8 (A-H, O-Z)
      IF(IBODY.NE.3)GO TO 999
C     RESET FNORM TO A VERY LARGE VALUE TO ELIMINATE POSSIBILITY OF
C     SEPARATION
      FNORM=2.E7
C
      WRITE(6,101) IBODY,FNORM,NNODE,INC
  101 FORMAT('THE SEPARATION FORCE OF BODY ',15,
     *HAS BEEN RESET TO BE ',E15.5,
     *FOR NODE ',15,' AT INCREMENT ',15)
  999 CONTINUE
      RETURN
      END
```

SEPSTR

Definition of Separation Stress

Description

This subroutine allows the definition of the separation stress in conjunction with the model definition option CONTACT. The separation stresses, SNORM and STANG, are either calculated by the program or entered through the CONTACT option, and then passed into this subroutine. You decide whether these values at the current increment are appropriate to determine whether separation occurs.

SNORM is the stress normal to the surface above which a node in contact separates from another body. Any compressive or negative value indicates real contact while a positive stress indicates a tendency to separate. The default is taken as the maximum value of the residual force in the structure for the current increment divided by an effective area. This value can be reset by you through the input format. Defining a too small value can result in an increased number of iterations. Defining a very large value eliminates the possibility of separation. STANG is the tangential stress used to determine whether a nodal point positioned at a convex corner of surface should be sliding from patch to patch or remaining on its current patch. The default value is half of SNORM. These two default values vary from increment to increment.

Format

Subroutine SEPSTR is written with the following headers:

```
SUBROUTINE SEPSTR (SNORM,STANG,IBODY,NNODE,INC)
IMPLICIT REAL *8 (A-H, O-Z)
   user coding
RETURN
END
```

where:

SNORM	is the normal separation stress to be supplied by you.
STANG	is the tangential separation stress to be supplied by you.
IBODY	is the current body number the node touched.
NNODE	is the current touched external node number.
INC	is the current increment number.

■ UHTCOE

Definition of Environment Film Coefficient

Description

This subroutine allows the definition of variable film coefficients and sink temperatures of free surfaces, in conjunction with the CONTACT option and COUPLE parameter. Its call is triggered by the UHTCOEF option.

Subroutine UHTCOE is called at every element surface containing nodes that are on a free body boundary and for each surface at the trapezoidal rule integration points (that is, the nodes). These calls are made every iteration both during the assembly phase and the recovery phase of the heat transfer pass of a coupled analysis.

A distributed heat flux is being calculated according to the equation:

$$q = H(T - TS)$$

where:

`Q`	is the heat flux entering the surface.
`T`	is the surface temperature.
`TS`	is the sink temperature.
`H`	is the film coefficient.

By modifying `H` and `TS`, you can model varying heat transfer conditions along the boundary. Special attention has been given to provide you the capability of simulating radiation heat transfer, by making available the location and temperatures of all the surfaces in the environment.

You can either specify `H` and `TS` or specify the flux `Q` directly which is treated strictly as such.

Format

Subroutine UHTCOE is written with the following headers:

```
 SUBROUTINE UHTCOE(MIBODY,XP,TEMP,IBODY,NF,XORD,NVXORD,XT,NVXT,DXT,
+NVDXT,TMPALL,NVTMP,TMPALO,NVTMPO,TOTINC,TIMINC,
+INC,NCRD,NDEG,NBCD,NBCN,TSINK,HTCOEF,IFLAG)
 IMPLICIT REAL *8 (A-H, O-Z)
 DIMENSION MIBODY(4),XP(1),NF(NBCN,1),XORD(1),XT(1),DXT(1),
+TMPALL(1),NBCD(1), TMPALO(1)
    user coding
 RETURN
 END
```

where:

MIBODY(1)	is the element where the surface flux is being calculated.
MIBODY(2)	is the side of said element.
MIBODY(3)	is the integration point of said side.
MIBODY(4)	is the internal element number.
XP(NCRD)	are the coordinates of point where calculation is being made, updated to end of increment.
TEMP	is the temperature of said point.
IBODY	is the flexible surface to which point belongs.
NF(NBCN,1)	are the nodal points that make the boundary of deformable surfaces declared in option contact.
NBCN	is the upper bound to the number of nodes on a flexible surface boundary.
NBCD(1)	is the array of actual number of boundary nodes on flexible surfaces.
XORD(1)	is the array of original nodal point coordinates.
NVXORD	is the MARC coordinate vector number.
XT(1)	is the array of nodal point displacements.
NVXT	is the MARC displacement vector number.
DXT(1)	is the array of nodal displacement increments.
NVDXT	is the MARC displacement increment vector number.
TMPALL(1)	is the array of nodal temperatures (current estimate at end of increment).
NVTMP	is the MARC temperature vector number.
TMPALO(1)	is the array of nodal temperatures (at beginning of increment).
NVTMPO	is the MARC temperature vector number.

TOTINC	is the current accumulated time.
TIMINC	is the time increment.
INC	is the increment number.
NCRD	is the number of coordinates per node.
NDEG	is the number of degrees-of-freedom per node.
TSINK	is the sink temperature declared in contact option for this flexible surface.
IFLAG	=0 HTCOEF is a heat transfer coefficient =1 HTCOEF is a flux.
HTCOEF	is the heat transfer coefficient between surface and environment, such that the heat flux per unit area that leaves the surface is: Q = HTCOEF (TEMP - TSINK) or the heat flux per unit area that leaves the surface. HTCOEF is to be defined here.

■ UHTCON

Definition of Contact Film Coefficient

Description

This subroutine allows the definition of variable film coefficients of surfaces that are in contact with other surfaces in conjunction with the CONTACT option and COUPLE parameter. Its call is triggered by the UHTCON option.

Subroutine UHTCON is called at every element surface containing nodes that are on a body boundary that is in contact, and for each surface at the trapezoidal rule integration points (that is, the nodes). These calls are made every iteration during both the assembly phase and the stress recovery phase of the heat transfer pass of a coupled analysis.

A distributed heat flux is being calculated according to the equation

$$q = HD(T - TD)$$

where:

q	is the heat flux entering the surface.
T	is the surface temperature.
TD	is an interpolated temperature of the body being contacted.
HD	is the film coefficient.

By modifying HD, you can model varying heat transfer conditions along the contact regions.

Format

Subroutine UHTCON is written with the following headers:

```
      SUBROUTINE UHTCON(MIBODY,XP,T,IBODY,IOBODY,FN,TOTINC,TIMINC,
     +INC,NCRD,NDEG,TD,HD)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION MIBODY(4),XP(1)
         user coding
      RETURN
      END
```

where:

MIBODY(1)	is the element where the surface flux is being calculated.
MIBODY(2)	is the side of said element.
MIBODY(3)	is the integration point of said side.
MIBODY(4)	is the internal element number.
XP(NCRD)	is the coordinates of point where calculation is being made; it is updated to end of increment.
T	is the temperature of said pointer.
IBODY	is the flexible surface to which point belongs.
IOBODY	is the surface being contacted.
FN	is the contact pressure between contacting surfaces.
TOTINC	is the current accumulated time.
TIMINC	is the time increment.
INC	is the current increment.
NCRD	is the number of coordinates per node.
NDEG	is the number of degrees of freedom per node.
TD	is the temperature of surface being contacted.
HD	is the heat transfer coefficient between surfaces in contact, such that the heat flux per unit area that leaves the surface is: `Q = HD (T - TD).`

UNORST

Definition of Normal Stress, Flow Stress and Temperature at Contact Node

Description

With this subroutine, you can define the normal stress at each node in contact instead of using the extrapolated value from the integration points. It is only called for user-defined elements and is used in the calculation of Coulomb friction for contact analysis. The magnitude of the user-defined normal stress must be in the local system of the patch to which the nodal point is in contact with.

Format

Subroutine UNORST is written with the following headers:

```
      SUBROUTINE UNORST(STRNOD,STRINT,USTR,TRANS,NODE,IBODY,KC,
     +NDIE,NODCLS,LMM,NOD,M,N,TIMINC,NDIM,NDEG,NSTRMX,NNODE,INTEL)
      IMPLICIT REAL *8 (A-H, O-Z)
C
      DIMENSION STRNOD(NDIM,NNODE),STRINT(8,INTEL),USTR(NDIM,1)
      DIMENSION LMM(1),NODCLS(1),TRANS(3,3)
          user coding
C
C
      RETURN
      END
```

where:

STRNOD (1-NSTRMX,NODE)	are the stresses at node. If your element is not consistent with the MARC element, these values can be incorrect.
STRNOD (NSTRMX+1,NODE)	is the temperature at the node.
STRNOD (NSTRMX+2,NODE)	is the flow stress at the node.
STRNOD (NSTRMX+3,NODE)	is the previous sliding velocity in first local direction.
STRNOD (NSTRMX+4,NODE)	(in 3D contact) is the previous sliding velocity in the second local direction.
STRINT (1-NSTRMX,INTEL)	are the stresses at all integration points.

STRINT (NSTRMX+1,INTEL)	is the temperature at all integration points.
STRINT (NSTRMX+2,INTEL)	is the flow stress at all integration points.
USTR (2,NODE)	is the current sliding velocity in the first local direction.
USTR (3,NODE)	(in 3D contact) current sliding velocity in the second local direction.
TRANS	local transformation matrix at the node.
For 3D contact:	
TRANS(1-3,1)	three components of local x-direction.
TRANS(1-3,2)	three components of local y-direction.
TRANS(1-3,3)	three components of local z-direction.
For 2D contact:	
TRANS(1,1) and TRANS(2,2)	is the directional cosine.
TRANS(1,2)	-sine, TRANS(2,1) is the directional cosine.
TRANS(3,3)	= 1.
NODE	is the current local node number belonging to the element face (it is neither a MARC internal node number nor an external user node number).
IBODY	the element side or face number that the node belongs to.
KC	is the current layer number.
NDIE	is the die number that the current node touches.
NODCLS	is the node array to indicate if the nodes on the IBODY are currently in contact. Zero value indicates no contact and nonzero value is the die number it currently touches.
LMM	is the connectivity array for current element side or face (local node number). for 2D contact it contains IBODY and IBODY+1 for 3D contact it stores 1,2,3,4, for 3D shell element: IBODY=1 it stores 1,2,3,4, and 9,10,11,12 if 20-node element. IBODY=2 it stores 6,5,8,7, and 13,14,15,16 if 20-node element. IBODY=3 it stores 2,1,5,6, and 9,17,13,18 if 20-node element. IBODY=4 it stores 3,2,6,7, and 10,18,14,19 if 20-node element. IBODY=5 it stores 4,3,7,8, and 11,19,15,20 if 20-node element. IBODY=6 it stores 1,4,8,5, and 12,20,16,17 if 20-node element.
NOD	is the external user node number.
M	is the element number.

N	is the elsto buffer number.
TIMINC	is the time increment.
NDIM	is the NSTRMX+4 for 3-D contact.
NDIM	is the NSTRMX+3 for 2-D contact.
NDEG	is the number of degrees of freedom per node.
NSTRMX	is the maximum number of stress components.
NNODE	is the maximum number of nodes per element.
INTEL	is the number of integration points at which stresses are stored.

User entries (OUTPUT):

USTR (1,NODE)	is the normal stress at current node.
USTR (NSTRMX+1,NODE)	is the temperature at current node.
USTR (NSTRMX+2,NODE)	is the flow stress at current node.

Example

```
      SUBROUTINE UNORST(STRNOD,STRINT,USTR,TRANS,NODE,IBODY,KC,
     +NDIE,NODCLS,LMM,NOD,M,N,TIMINC,NDIM,NDEG,NSTRMX,NNODE,INTEL)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION STRNOD(NDIM,NNODE),STRINT(8,INTEL),USTR(NDIM,1)
      DIMENSION LMM(1),NODCLS(1),TRANS(3,3)
C
C  THE STRESS AT ALL CONTACT NODE IS 84 PER UNIT AREA.
C  NEGATIVE NUMBER MEANS THE MAGNITUDE ALONG NEGATIVE
C  LOCAL NORMAL DIRECTION OF THE TOUCHED FACE.
C
      USTR(1,NODE)=-84.0
      RETURN
      END
```

■ UCONTACT

User-defined Contact Condition

Description

This subroutine allows you to calculate your own contact conditions with rigid surfaces based upon the MARC contact algorithm. This subroutine requires the use of the UCONTACT model definition option and is triggered by this option. In general, the four stages in MARC contact procedures are: checking contact, enforce contact constraint, check separation, and check penetration. You can either specify contact conditions at one stage and let the MARC program do the work in the remainder of stages or you can substitute your calculations for all stages. The calls and your requirements are defined as follows:

`IFLAG=1` at first contact stage; check contact

`IFLAG=2` at second contact stage; enforce contact constraint

`IFLAG=3` at third contact stage; check separation

`IFLAG=4` at fourth contact stage; check penetration

You must set `IUSED=1` if your code is to be used.

Format

Subroutine `UCONTACT` is written with the following headers:

```
      SUBROUTINE UCONTACT(IPATCH,NDIE,XYZ,NPATCH,XP,DU,XNORM,
     +FNORM,FACTOR,DIST,DERROR,THICK,SNORM,TIMINC,CFORCE,
     +INC,NCYCLE,ICLOSE,ISEPAR,I2OR3,NODE,IUSED,IFLAG)
      IMPLICIT REAL *8 (A-H, O-Z)

      DIMENSION XYZ(I2OR3,1),XP(1),DU(1),XNORM(1),SNORM(1)
          user coding

      RETURN
      END
```

where:

`IPATCH`	is the current patch number.
`NDIE`	is the surface number.

XYZ(I2OR3,4)	are the coordinates of ipatch for 3D.
XYZ(I2OR3,2)	are the coordinates of ipatch for 2D.
NPATCH	is the total number of patches used to generate the surface.
XP	is the updated nodal coordinates.
DU	is the incremental displacement in global system.
XNORM	is the normal vector of current patch.
FNORM	is the contact forces at the node.
DIST	is the incremental displacement projected along the normal direction.
FACTOR	is the factor = (normal distance between node and patch)/dist.
DERROR	is the contact distance.
THICK	is the shell thickness at the node. For 3D shell element only.
SNORM	is the shell normal vector at the node. For 3D shell element only.
TIMINC	is the time step at current increment.
CFORCE	is the default contact separation force.
INC	is the current increment number.
NCYCLE	is the iteration number at current increment.
ICLOSE	indicates if the node touches the patch. 0 means no touch, 1 is touch. Only redefine if IFLAG=1.
ISEPAR	indicates if the node separates from the patch. 0 means no separation 1 is to separate. Only redefine if IFLAG=3.
I2OR3	2 for 2D contact 3 for 3D contact.
NODE	is your node number.
IUSED	0 means this routine is not used during current contact stage. 1 means this routine is used during current contact stage.

IFLAG

2D contact:
output: iused, iclose and xnorm(1), xnorm(2).
3D contact:
output: iused, iclose and xnorm(1), xnorm(2), and xnorm(3).
If IFLAG=2
2D contact:
output: iused, xnorm(1), xnorm(2). The xnorm are normal vector of ipatch.
3D contact:
output: iused, xnorm(1), xnorm(2), xnorm(3). xnorm(4), xnorm(5), and xnorm(6).
The xnorm(1-3) are the three components of normal vector of ipatch.
The xnorm(4-6) are the three components of tangent vector of ipatch to define the local-x direction of transformation system. Those two vectors must be unitized.
If IFLAG=3 output: iused, isepar.
If IFLAG=4 output: iused, dist, and factor.

■ INITPL

Initialize Equivalent Plastic Strain Values

Description

This subroutine is called in a loop over all the elements in the mesh when the INITIAL PLASTIC STRAIN option appears in the model definition options with a two in the second field of the second data block of that option. It allows you to define initial values of equivalent plastic strain. It is often necessary to enter the amount of previously accumulated plastic strain. This initial value is only used in the work (strain) hardening calculation.

Format

Subroutine INITPL is written with the following headers:

```
SUBROUTINE INITPL(SV,LAYERS,INTPTS,M)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION SV(LAYERS,INTPTS)
    user coding
RETURN
END
```

where:

SV	is the array of equivalent plastic strains, to be defined here for this element by you.
LAYERS	is the number of layers through the thickness if this is a shell element, or the number of points in the cross section if this is a beam element. It is 1 for a continuum element.
INTPTS	is the number of integration points in this element. It is 1 if the CENTROID parameter is used.
M	is the element number.

■ INITPO

Initialize Pore Pressure in an Uncoupled Fluid-Soil Analysis

Description

This subroutine allows you to prescribe the initial pore pressure in an uncoupled fluid-soil analysis. This user subroutine can only be used if an uncoupled analysis is chosen on the PORE parameter, and the routine is activated using the INITIAL PORE model definition option.

Format

Subroutine INITPO is written with the following headers:

```
SUBROUTINE INITPO(POREP,INTPTS,M)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION POREP(INTPTS)
    user coding
RETURN
END
```

where:

POREP	is the array of pore pressures to be defined for this element.
INTPTS	is the number of integration points associated with this element.
M	is your element number.

NEWPO

Modify Pore Pressure in an Uncoupled Fluid-Soil Analysis

Description

This subroutine allows you to modify the pore pressure in an uncoupled fluid-soil analysis. This user subroutine can only be used if a coupled analysis is chosen on the PORE parameter, and the routine is activated using the CHANGE PORE model definition option.

Format

Subroutine NEWPO is written with the following headers:

```
SUBROUTINE NEWPO(POREP,INTPTS,M)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION POREP(INTPTS)
    user coding
RETURN
END
```

where:

POREP	is the array of pore pressures to be defined for this element.
INTPTS	is the number of integration points associated with this element.
M	is your element number.

3 List of User-defined Anisotropy and Constitutive Relations Subroutines

Subroutines **Page**

ANELAS 3-4
ANEXP 3-19
ANKOND 3-21
ANPLAS 3-14

CRPLAW 3-28

GAPU 3-75
GENSTR 3-67

HOOKLW 3-12
HYPELA 3-56
HYPELA2 3-58

ORIENT 3-17

TENSOF 3-42

UBEAM 3-69
UCOMPL 3-73
UCRACK 3-40
UELDAM 3-54
UENERG 3-51
UEPS 3-23
UFAIL 3-16
UFINITE 3-62
UMOONY 3-50
UMU 3-24
UNEWTN 3-80
UOGDEN 3-52
UPERM 3-49

Subroutines

	Page
UPHI	3-71
UPOWDR	3-47
UPSTRECH	3-64
URPFLO	3-81
USELEM	3-77
USHRET	3-43
USIGMA	3-26
USPCHT	3-27
USPRNG	3-38
UVOID	3-44
UVOIDN	3-45
VSWELL	3-33
WKSLP	3-35

3 User-defined Anisotropy and Constitutive Relations Subroutines

This chapter describes the user subroutines available to allow you to provide material data to standard MARC constitutive relations, or for you to create your own model. The routines in this chapter cover the spectrum of anisotropic elasticity and plasticity, creep, plasticity, rate independent nonlinear elasticity, cracking and electrical and magnetic materials among others. These routines are, in general, called for each integration point for each element they have been invoked. This provides a powerful method to provide nonhomogeneous, nonlinear material behavior. Table 3-1 summarizes these routines and indicates what parameters or model definition options are required to invoke the user subroutine.

Table 3-1 User-defined Anisotrophy and Constitutive Relations Subroutine Requirements

User Subroutine	Required Parameters or Model Definition Options	Purpose
ANELAS	ORTHOTROPIC or ANISOTROPIC	Definition of factors to scale elastic stress strain law.
ANEXP	ORTHOTROPIC or ANISOTROPIC	Definition of thermal strain increment.
ANKOND	ORTHOTROPIC or ANISOTROPIC	Definition of thermal conductivity or electrical resistance in Joule heating.
ANPLAS	ORTHOTROPIC or ANISOTROPIC	Definition of parameters for Hill yield criteria
CRPLAW	CREEP	Definition of function to describe creep strain rate.
GAPU	GAP DATA	Definition of contact gap closure distance
GENSTR	SHELL SECT	Definition of generalized stress-strain law for shells.
HOOKLW	ORTHOTROPIC or ANISOTROPIC	Definition of elastic stress-strain or compliance relation.
HYPELA	HYPOELASTIC	Definition of nonlinear stress-strain relationship.
HYPELA2	HYPOELASTIC	Definition of nonlinear stress-strain relationship.

Table 3-1 User-defined Anisotrophy and Constitutive Relations Subroutine Requirements (Continued)

User Subroutine	Required Parameters or Model Definition Options	Purpose
ORIENT	ORIENTATION	Definition of preferred material orientation for orthotropic or anisotropic behavior.
TENSOF	ISOTROPIC CRACK DATA	Definition of tension softening modulus.
UBEAM	HYPOELASTIC	Definition of nonlinear generalized stress-strain law for element types 52 or 98.
UCOMPL	HARMONIC	Definition of stress-strain rate relationship for harmonic analysis.
UCRACK	ISOTROPIC CRACK DATA	Definition of ultimate stress for cracking analysis.
UELDAM	OGDEN DAMAGE	Definition of damage parameters for Ogden rubber model.
UENERG	MOONEY	Definition of strain energy function.
UEPS	ELECTRO or EL-MA ORTHOTROPIC	Definition of anisotropic electrical permittivity.
UFAIL	FAIL DATA	Definition of composite failure criteria.
UMOONEY	MOONEY	Definition of temperature dependent Mooney-Rivlin constants.
UMU	MAGNETO or EL-MA ORTHOTROPIC	Definition of anisotropic magnetic permeability.
UNEWTN	R-P FLOW or FLUID	Definition of material viscosity.
UOGDEN	OGDEN	Definition of Ogden material parameters.
UPERM	PORE	Definition of soil permeability.
UPHI	HARMONIC MOONEY PHI-COEFFICIENTS	Definition of phi coefficients for rubber-viscoelastic harmonic analysis.
UPOWDR	POWDER	Definition of powder material data.
UPSTRECH	ODGEN	Definition of generalized principal stretch based elasticity models.
URPFLO	R-P FLOW	Definition of yield surface for rigid plastic flow.

Table 3-1 User-defined Anisotrophy and Constitutive Relations Subroutine Requirements (Continued)

User Subroutine	Required Parameters or Model Definition Options	Purpose
USELEM	USER	Definition of consistent nodal loads, mass matrix, stiffness matrix, and residuals for user-defined element.
USHRET	ISOTROPIC CRACK DATA	Definition of shear retention factor for elements that have cracks.
USIGMA	EL-MA	Definition of anisotropic electrical conductivity.
USPCHT	HEAT or COUPLE or FLUID	Definition of specific heat.
USPRNG	SPRINGS or FOUNDATION	Definition of nonlinear spring or foundation stiffness.
UVOID	DAMAGE	Definition of initial void fraction for Gurson damage model.
UVOIDN	DAMAGE	Definition of void nucleation for Gurson damage model.
VSWELL	CREEP	Definition of volumetric swelling.
WKSLP	ISOTROPIC or ORTHOTROPIC or ANISOTROPIC WORK HARD	Definition of work hardening or strain hardening data.

■ ANELAS

Elastic Anisotropy

Description

This subroutine allows you to define the anisotropic elastic law. In the most generally allowed case, the isothermal stress-strain law in the preferred orientation is:

$$\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \tau_{12} \\ \tau_{23} \\ \tau_{31} \end{Bmatrix} = \begin{bmatrix} r_{11}D_{11} & r_{12}D_{12} & r_{13}D_{13} & 0 & 0 & 0 \\ & r_{22}D_{22} & r_{23}D_{23} & 0 & 0 & 0 \\ & & r_{33}D_{33} & 0 & 0 & 0 \\ \text{Symmetric} & & & r_{44}D_{44} & 0 & 0 \\ & & & & r_{55}D_{55} & 0 \\ & & & & & r_{66}D_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{23} \\ \gamma_{31} \end{Bmatrix}$$

The arrangement of the $\{\sigma\},\{\varepsilon\}$ vectors is defined for each element type in *Volume B: Element Library*. D_{ij} are the incremental elastic stress-strain relation calculated by MARC based on material data given through input data. The r_{ij} are supplied by you in subroutine ANELAS. It is often easier to directly specify the stress-strain for compliance relationship in subroutine HOOKLW.

Format

Subroutine ANELAS is written with the following headers:

```
 SUBROUTINE ANELAS (N,NN,KC,R,IRDIM,NDI,NSHEAR,MATS,DT,DTDL,D,
+RPROPS,IPROPS)
 IMPLICIT REAL *8 (A-H, O-Z)
 DIMENSION R (IRDIM,IRDIM),DT(1),DTDL(1),D(IRDIM,IRDIM),N(2),
 RPROPS(1),IPROPS(1)
       user coding
 RETURN
 END
```

where:

N(1)	is your element number.
N(2)	is the internal element number.
NN	is integration point number.
KC	is the layer number (always 1 for continuum elements).
R	is the r to be defined by you; the number of allowable r being given in Table 3-2.
IRDIM	is the dimension of the R array for the current element.
NDI	is the number of direct components.
NSHEAR	is the number of shear components.
MATS	is the material id.
DT	is the array of state variables.
DTDL	is the array of increments of state variables.
D	is the stress-strain law as calculated by MARC using input data. To modify this matrix directly, use user subroutine HOOKLW instead of ANELAS.
RPROPS	is the array of real properties, see introduction.
IPROPS	is the array of integer properties, see introduction.

Table 3-2 Allowable Anisotropy

Library Element Number	Allowable Transformations to Preferred Operation	Size of R. Matrix (IRDIM) for IRDIM=1 No Anisotropy Possible	Number of Direct Stresses (NDI)	Number of Shear Stresses (NSHEAR)
1	None	1	2	1
2	Orthogonal in z-r plane	4	3	1
3	Orthogonal in x-y plane	3	2	1
4	Any in θ^1 - θ^2 surface	3	2	1
5	None	1	1	0
6	Orthogonal in x-y plane	4	3	1
7	Orthogonal in (x,y,z) space	6	3	3
8	Any in θ^1 - θ^2 surface	6	2	1
9	None	1	1	0
10	Orthogonal in z-r plane	4	2	1

Table 3-2 Allowable Anisotropy (Continued)

Library Element Number	Allowable Transformations to Preferred Operation	Size of R. Matrix (IRDIM) for IRDIM=1 No Anisotropy Possible	Number of Direct Stresses (NDI)	Number of Shear Stresses (NSHEAR)
11	Orthogonal in x-y plane	4	3	1
12	None	1	0	0
13	None	1	1	0
14	None	1	1	1
15	None	1	2	0
16	None	1	1	0
17	None	1	2	0
18	Any in surface	3	2	1
19	Orthogonal in (x,y,z) space	4	2	1
20	Orthogonal in (x,y,z) space	6	3	3
21	Orthogonal in (x,y,z) space	6	3	3
22	Orthogonal in (x,y,z) space	5	2	3
23	None	1	1	0
24	Any in θ^1 - θ^2 surface	3	2	1
25	None	1	1	1
26	Orthogonal in x-y plane	3	2	1
27	Orthogonal in x-y plane	4	3	1
28	Orthogonal in x-y plane	4	3	1
29	Orthogonal in x-y plane	4	3	1
30	Any in surface	3	2	1
31	Not Available	-	-	-
32	Orthogonal in x-y plane	4	3	1
33	Orthogonal in z-r plane	4	3	2
34	Orthogonal in x-y plane	4	3	1
35	Orthogonal in (x,y,z) space	6	3	3
36-44	Use subroutine ANKOND to supply anisotropic conductivity	-	-	-
45	None	1	1	1
46-48	None	-	-	-

Table 3-2 Allowable Anisotropy (Continued)

Library Element Number	Allowable Transformations to Preferred Operation	Size of R. Matrix (IRDIM) for IRDIM=1 No Anisotropy Possible	Number of Direct Stresses (NDI)	Number of Shear Stresses (NSHEAR)
49	Any in $\underline{V}^1$ - $\underline{V}^2$	3	2	1
50	Use subroutine ANKOND to supply anisotropic conductivity	-	-	-
51	None	1	1	0
52	None	1	1	0
53	Orthogonal in x-y plane	3	2	1
54	Orthogonal in x-y plane	4	3	1
55	Orthogonal in z-r plane	4	3	1
56	Orthogonal in x-y plane	4	3	1
57	Orthogonal in (x,y,z) space	6	3	3
58	Orthogonal in x-y plane	4	3	1
59	Orthogonal in z-r plane	4	3	1
60	Orthogonal in x-y plane	4	3	1
61	Orthogonal in (x,y,z) space	6	3	3
62	Orthogonal in z-r plane	6	3	3
63	Orthogonal in z-r plane	6	3	3
64	None	1	1	0
65	None	1	0	0
66	Orthogonal in z-r plane	6	3	3
67	Orthogonal in z-r plane	6	3	3
68	None	1	0	1
69	Use subroutine ANKOND to supply anisotropic conductivity	-	-	-
70	Use subroutine ANKOND to supply anisotropic conductivity	-	-	-
71	Use subroutine ANKOND to supply anisotropic conductivity	-	-	-
72	Orthogonal in $\underline{V}^1$ - $\underline{V}^2$	3	2	1
73	Orthogonal in z-r plane	6	3	3

Table 3-2 Allowable Anisotropy (Continued)

Library Element Number	Allowable Transformations to Preferred Operation	Size of R. Matrix (IRDIM) for IRDIM=1 No Anisotropy Possible	Number of Direct Stresses (NDI)	Number of Shear Stresses (NSHEAR)
74	Orthogonal in z-r plane	6	3	3
75	Orthogonal in $\underline{V}^1$ - $\underline{V}^2$	5	2	3
76	None	1	1	1
77	None	1	1	0
78	None	1	1	1
79	None	1	1	0
80	Orthogonal in x-y plane	4	3	1
81	Orthogonal in x-y plane	4	3	1
82	Orthogonal in z-r plane	4	3	1
83	Orthogonal in z-r plane	4	3	1
84	Orthogonal in (x,y,z) space	6	3	3
85	Use subroutine ANKOND to supply anisotropic conductivity	-	-	-
86	Use subroutine ANKOND to supply anisotropic conductivity	-	-	-
87	Use subroutine ANKOND to supply anisotropic conductivity	-	-	-
88	Use subroutine ANKOND to supply anisotropic conductivity	-	-	-
89	None	1	2	1
90	None	1	2	3
91	Orthogonal in x-y plane	4	3	2
92	Orthogonal in z-r plane	4	3	1
93	Orthogonal in x-y plane	4	3	1
94	Orthogonal in z-r plane	4	3	1
95	Orthogonal in z-r plane	6	3	3
96	Orthogonal in z-r plane	6	3	3
97	None	1	0	0
98	None	1	1	2

Table 3-2 Allowable Anisotropy (Continued)

Library Element Number	Allowable Transformations to Preferred Operation	Size of R. Matrix (IRDIM) for IRDIM=1 No Anisotropy Possible	Number of Direct Stresses (NDI)	Number of Shear Stresses (NSHEAR)
99	None	-	-	-
100	None	-	-	-
101	Use subroutine ANKOND to supply anisotropic conductivity	-	-	-
102	Use subroutine ANKOND to supply anisotropic conductivity	-	-	-
103	Use subroutine ANKOND to supply anisotropic conductivity	-	-	-
104	Use subroutine ANKOND to supply anisotropic conductivity	-	-	-
105	Use subroutine ANKOND to supply anisotropic conductivity	-	-	-
106	Use subroutine ANKOND to supply anisotropic conductivity	-	-	-
107	Orthogonal in (x,y,z) space	6	3	3
108	Orthogonal in (x, y, z) space	6	3	3
109	Use subroutine UMU	-	-	-
109	Use subroutine UMU	-	-	-
110	Use subroutine UMU	-	-	-
111	Use subroutine UEPS, UMU, USIGMA	-	-	-
112	Use subroutine UEPS, UMU, USIGMA	-	-	-
113	Use subroutine UEPS, UMU, USIGMA	-	-	-
114	Orthogonal in x-y plane	3	2	1
115	Orthogonal in x-y plane	4	3	1
116	Orthogonal in z-r plane	4	3	1
117	Orthogonal in x,y,z space	6	3	3
118	Orthogonal in x-y plane	4	3	1
119	Orthogonal in z-r plane	4	3	1

Table 3-2 Allowable Anisotropy (Continued)

Library Element Number	Allowable Transformations to Preferred Operation	Size of R. Matrix (IRDIM) for IRDIM=1 No Anisotropy Possible	Number of Direct Stresses (NDI)	Number of Shear Stresses (NSHEAR)
120	Orthogonal in x,y,z space	6	3	3
121	Use subroutine ANKOND	-	-	-
122	Use subroutine ANKOND	-	-	-
123	Use subroutine ANKOND	-	-	-
124	Orthogonal in x-y plane	3	2	1
125	Orthogonal in x-y plane	4	3	1
126	Orthogonal in z-r plane	4	3	1
127	Orthogonal in x,y,z space	6	3	3
128	Orthogonal in x-y plane	4	3	1
129	Orthogonal in z-r plane	4	3	1
130	Orthogonal in x,y,z, space	6	3	3
131	Use subroutine ANKOND	-	-	-
132	Use subroutine ANKOND	-	-	-
133	Use subroutine ANKOND	-	-	-
134	Orthogonal in x,y,z space	6	3	3
135	Use subroutine ANKOND	-	-	-
138	Orthogonal in $\underline{V}^1$-$\underline{V}^2$	3	2	1
139	Orthogonal in $\underline{V}^1$-$\underline{V}^2$	3	2	1
140	Orthogonal in $\underline{V}^1$-$\underline{V}^2$	5	2	3
141	None	-	1	0
142	None	-	1	0
143	None	-	1	0
144	None	-	1	0
145	None	-	1	0
146	None	-	1	0
147	None	-	1	0
148	None	-	1	0

All parameters except the R array are defined by the program. R must be defined by you in this routine.

Note that the R and D matrices have the dimension appropriate for the number of stress components associated with the particular element (see Table 3-2). Thus, for example, in elements 3 or 18, the R matrix would be of size 3 by 3, and the stress strain law would take the form:

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_{12} \end{Bmatrix} = \begin{bmatrix} r_{11}\ D_{11} & r_{12}\ D_{12} & 0 \\ & r_{22}\ D_{22} & 0 \\ \text{Symmetric} & & r_{33}\ D_{33} \end{bmatrix} \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \gamma_{12} \end{Bmatrix}$$

To define an anisotropic stress-strain relation for the Herrmann incompressible elements in MARC, user subroutine ANELAS is used in a slightly different manner. The compliance strain-stress relation is given directly in the fourth argument R and is not used in the last argument D. For example, in the most generally allowed case, the compliance relation in the preferred orientation is:

$$\begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{23} \\ \gamma_{31} \end{Bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & 0 & 0 & 0 \\ & R_{22} & R_{23} & 0 & 0 & 0 \\ & & R_{33} & 0 & 0 & 0 \\ & & & R_{44} & 0 & 0 \\ & & & & R_{55} & 0 \\ & & & & & R_{66} \end{bmatrix} \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \tau_{12} \\ \tau_{23} \\ \tau_{31} \end{Bmatrix}$$

Note: This routine should not be used if you desire that the material constants should be design variables. Use the ORTHOTROPIC option instead.

■ HOOKLW

Anisotropic Elastic Law

Description

User subroutine HOOKLW is an alternative mechanism to user subroutine ANELAS. In this routine, the elastic stress-strain law is supplied by you. A maximum of 21 terms are necessary for a three-dimensional body. This law is given in terms of the coordinate system defined in the ORIENTATION option. You should insure that the stress-strain law is symmetric. Note that this routine is called for each integration point of those elements that have anisotropic properties. You can define either the stress-strain relation or the compliance strain-stress relation. The returned value of argument `IMOD` must be set accordingly. For example, if `IMOD=1`, the stress-strain law is given and you return to the array `B` such that:

$$\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \tau_{12} \\ \tau_{23} \\ \tau_{31} \end{Bmatrix} = \begin{bmatrix} B11 & B12 & B13 & B14 & B15 & B16 \\ B21 & B22 & B23 & B24 & B25 & B26 \\ B31 & B32 & B33 & B34 & B35 & B36 \\ B41 & B42 & B43 & B44 & B45 & B46 \\ B51 & B52 & B53 & B54 & B55 & B56 \\ B61 & B62 & B63 & B64 & B65 & B66 \end{bmatrix} \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \gamma_{12} \\ \gamma_{23} \\ \gamma_{31} \end{Bmatrix}$$

The arrangement of {s}, {ε} vectors are defined for each element type in *Volume B: Element Library.*

Format

Subroutine HOOKLW is written with the following headers:

```
      SUBROUTINE HOOKLW(M,NN,KC,B,NGENS,DT,DTDL,E,PR,NDI,NSHEAR,IMOD,
     +RPROPS,IPROPS)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION B(NGENS,NGENS),DT(1),DTDL(1),RPROPS(1),IPROPS(1),M(2)
        user coding
      RETURN
      END
```

where:

M(1)	is your element number.
M(2)	is the internal element number.
NN	is the integration point number.
KC	is the layer number.
B	is the user-defined stress-strain law if IMOD=1; or the user-defined compliance relation if IMOD=2 to be defined here.
NGENS	is the number of stresses and strain components.
DT	is the state variables at the beginning of the increment (temperature first).
DTDL	is the increment of state variables.
E	is the Young's modulus including temperature effects.
PR	is the Poisson's ratio including temperature effects.
NDI	number of direct components of stress.
NSHEAR	number of shear components of stress.
IMOD	Set to 0 if user subroutine ANELAS is used. Set to 1 to indicate that the stress-strain law has been given. Set to 2 to indicate that the compliance strain-stress, relation has been given.
RPROPS	array of real properties, see introduction.
IPROPS	array of integer properties, see introduction.

Note that for temperature dependent properties, this routine is called twice for each integration point. The first time to evaluate the stress-strain law at the beginning of the increment; the second time at the end of the increment.

Note: This routine should not be used if you desire that the material constants should be design variables. Use the ORTHOTROPIC option instead.

■ ANPLAS

Anisotropic Yield Surface and Creep Potential

Description

The anisotropic yield function and stress potential are assumed as:

$$a_1(\sigma_y - \sigma_z)^2 + a_2(\sigma_z - \sigma_x)^2 + a_3(\sigma_x - \sigma_y)^2 + 3a_4\tau_{yz}^2 + 3a_6\tau_{xy}^2 = 2\bar{\sigma}^2$$

(R. Hill - *Mathematical Theory of Plasticity*, Oxford, 1950)

where: $\bar{\sigma}$ is the equivalent tensile yield stress for isotropic behavior: $\bar{\sigma} = \bar{\sigma}(\varepsilon^P, T)$

and, for Mohr-Coulomb behavior: $\bar{\sigma} = \bar{\sigma}(J_1)$; $J_1 = \dfrac{\sigma_x + \sigma_y + \sigma_z}{3}$

You define ratios of actual to isotropic yield (in the preferred orientation) in the array `YRDIR` for direct tension yielding, and `YRSHR` for yield in shear (ratio of actual shear yield to $\sigma/\sqrt{3}$ = isotropic shear yield). Then the a_1 above are derived as (Hill):

$$a_1 = \frac{1}{YRDIR(2)^2} + \frac{1}{YRDIR(3)^2} - \frac{1}{YRDIR(2)^2}$$

$$a_2 = \frac{1}{YRDIR(3)^2} + \frac{1}{YRDIR(1)^2} - \frac{1}{YRDIR(2)^2}$$

$$a_3 = \frac{1}{YRDIR(1)^2} + \frac{1}{YRDIR(2)^2} - \frac{}{YRDIR(3)^2}$$

$$a_4 = \frac{2}{YRSHR(3)^2}$$

$$a_5 = \frac{2}{YRSHR(2)^2}$$

$$a_6 = \frac{2}{YRSHR(1)^2}$$

Note that YRDIR and YRSHR should be given in the order appropriate for the element (see Library Element Description).

On the output, the Mises intensity is not affected by these material parameters.

Format

Subroutine ANPLAS is written with the following headers:

```
SUBROUTINE ANPLAS(N,NN,LAYER,NDI,NSHEAR,MATS,YRDIR,YRSHR)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION YRDIR (1),YRSHR(1),N(2)
    user coding
RETURN
END
```

where:

N(1)	is your element number.
N(2)	is the internal element number.
NN	is the integration point number.
LAYER	is the layer number (always 1 for continuum elements).
NDI	is the number of direct stresses.
NSHEAR	is the number of shear stresses.
MATS	is the material id.
YRDIR	is the array of tensile yield ratios to be defined here.
YRSHR	is the array of shear yield ratios to be defined here.

All parameters except YRDIR and YRSHR are defined by the program. YRDIR and YRSHR are defined by you in this routine.

■ UFAIL

User-defined Failure Criterion

Description

User subroutine UFAIL is provided to allow you to calculate your own scalar failure criterion. To call subroutine UFAIL, you must specify failure criterion type UFAIL in the FAIL DATA model definition option. UFAIL will then be called for every integration point associated with the material id specified in the FAIL DATA option.

Format

Subroutine UFAIL is written with the following headers:

```
      SUBROUTINE UFAIL (N,NN,KC,MATS,
     1 STRESS,STRAIN,NDI,NSHEAR,FAILCR)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION STRESS(1),STRAIN(1),N(2)
       user coding
      RETURN
      END
```

where:

N(1)	is your element number.
N(2)	is the internal element number.
NN	is the integration point number.
KC	is the layer number.
MATS	is the material identification number.
STRESS	is the current total stress state.
STRAIN	is the current total strain.
NDI	is the number of direct stresses.
NSHEAR	is the number of shear stresses.
FAILCR	is your calculated failure criterion.

■ ORIENT

Specification of Preferred Orientation

Description

User subroutine ORIENT is used to supply a preferred orientation so that ANELAS, HOOKLW, ANKOND, and ANPLAS can supply anisotropic material constants in this orientation.

Format

Subroutine ORIENT is written with the following headers:

```
SUBROUTINE ORIENT (N,NN,KC,G)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION G(3,3),N(2)
    user coding
RETURN
END
```

where:

N(1)	is your element number.
N(2)	is the internal element number.
NN	is the integration point number.
KC	is the layer number (always 1 for continuum elements).
G	is the transformation matrix to be defined here.

All parameters except G are passed in by the program – you must supply the G matrix. G is the transformation to the preferred orientation from the usual MARC orientation:

$$\begin{Bmatrix} v'_1 \\ v'_2 \\ v'_3 \end{Bmatrix} = \begin{bmatrix} G_{11} & G_{12} & G_{13} \\ G_{21} & G_{22} & G_{23} \\ G_{31} & G_{32} & G_{33} \end{bmatrix} \begin{Bmatrix} v_1 \\ v_2 \\ v_3 \end{Bmatrix}$$

where:

V	is the vector in the MARC system.
V'	is the vector in the preferred system.

For curvilinear systems (for example, element types 4, 8, and 24), G is defined by $G(I,J) = g^i_j$. For planar transformations, G(3,I) = G(I,3) = 0; G(3,3) = 1.; I = 1,2 must be given.

Note: This routine should not be used if you desire that the material orientation be a design variable. Use the COMPOSITE option instead.

■ ANEXP

Anisotropic Thermal Expansion

Description

User subroutine ANEXP is used to specify anisotropic thermal strain increments in the orientation defined by the ORIENTATION option. You are given the temperature at the beginning of the increment, the temperature increment, and the base value of the thermal expansion coefficients given on the ISOTROPIC or ORTHOTROPIC options. You must supply the incremental thermal strain vector $\Delta\varepsilon^{th}_{ij}$ ($\Delta\varepsilon^{th}{}_{i}{}^{j}$ for doubly curved shell elements 4, 8, and 24) in the subroutine. Any components of the incremental thermal strain vector not defined in the routine assume their default program calculated values.

Format

Subroutine ANEXP is written with the following headers:

```
SUBROUTINE ANEXP (N,NN,KC,T,TINC,COED,NDI,NSHEAR,EQEXP)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION EQEXP(1),TINC(1),T(1),COED(NDI),N(2)
   user coding
RETURN
END
```

where:

N(1)	is your element number.
N(2)	is the internal element number.
NN	is the integration point number.
KC	is the layer number for shells or beams (always 1 for continuum elements).
T(1)	is the total temperature at the beginning of the increment.
T(2)	is the total values of other state variables at the beginning of the increment.
TINC(1)	is the temperature increment.
TINC(2) etc.	are the increments of other state variables.
COED(I)	is the base value of the Ith coefficient of thermal expansion as given through the input data. There are NDI coefficients for each element.

NDI	is the number of direct components of strain at this point.
NSHEAR	is the number of shear components of strain at this point.
EQEXP	is the thermal strain increment vector, to be defined by you in this subroutine. All parameters except EQEXP are supplied by the program.

Note: For the curvilinear coordinate elements (doubly curved shell elements 4,8,24) the mixed strain tensor shear components, $\varepsilon_1{}^2$, $\varepsilon_2{}^1$, are stored. Otherwise, shear components are engineering shear strain.

■ ANKOND

Input of Anisotropic Thermal Conductivity Matrix

Description

For anisotropic heat transfer analysis, this subroutine allows you to define an anisotropic conductivity matrix at each integration point in each element. The anisotropic conductivity matrix is defined with respect to the preferred orientation specified in ORIENTATION option. This routine is also used for anisotropic electrical resistance in a Joule heating analysis.

Format

Subroutine ANKOND is written with the following headers:

```
      SUBROUTINE ANKOND (COND,CANISO,N,NN,KC,MATNO,ID,T,DT,TIME,DELTME,
     * JOULHT)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION COND(ID,ID),CANISO(3)
          user coding
      RETURN
      END
```

where:

COND	is the conductivity matrix, k_{ij}: $$q_i = k_{ij}\frac{\partial T}{\partial x_j}$$ This is to be re-defined as necessary by you. This matrix is passed in as set-up for anisotropic conductivity. If you do not re-define it, it remains anisotropic according to k_{ij} (T) given on the ISOTROPIC, ORTHOTROPIC, and TEMPERATURE EFFECTS options.
CANISO	are the anisotropic conductivities k_{ij} (T) established by you via data blocks.
N	is the element number.
NN	is the integration point number.
KC	is the layer number.
MATNO	is the material identifier.

ID	is the size of the COND matrix; that is, the number of derivatives. $\frac{\partial T}{\partial x_j}$
T	is the temperature at the beginning of the time increment.
DT	is the estimated temperature increment.
TIME	is the transient time at the beginning of the increment.
DELTME	is the increment of time.
JOULHT	= 0 return thermal conductivity. = 1 return electrical conductivity.

■ UEPS

Input of Anisotropic Permittivity Matrix

Description

For anisotropic electrostatic or electromagnetic analysis, this subroutine allows you to define an anisotropic permittivity matrix at each integration point in each element. The anisotropic permittivity matrix is defined with respect to the preferred orientation specified in the ORIENTATION option.

Format

Subroutine UEPS is written with the following headers:

```
SUBROUTINE UEPS (EPS,M,NN,MATNO,ID)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION EPS(ID,ID),M(2)
    user coding
RETURN
END
```

where:

EPS	is the permittivity matrix, [ε] (D = [ε]E). This is to be re-defined as necessary by you. This matrix is passed in as set-up for anisotropic permittivity. If you do not re-define it, it remains as given through the ISOTROPIC or ORTHOTROPIC options.
M(1)	is your element number.
M(2)	is the internal element number.
NN	is the integration point number.
MATNO	is the material identifier.
ID	is the size of the matrix.

■ UMU

Input of Anisotropic Permeability Matrix

Description

For anisotropic magnetostatic or electromagnetic analysis, this subroutine allows you to define an anisotropic permeability matrix at each integration point in each element. The anisotropic permeability matrix is defined with respect to the preferred orientation specified in the ORIENTATION option.

The permeability μ is used in the relation:

$B=\mu H + B_r$

where:

B	is the magnetic induction.
H	is the magnetic field intensity.
m	is the permeability.
B_r	is the remanence.

Note: B is complex in a harmonic analysis.

Format

Subroutine UMU is written with the following headers:

```
SUBROUTINE UMU (XMU,M,NN,MATNO,ID,CPTIM,DTIME,B)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION XMU(ID,ID),B(3),M(2)
    user coding
RETURN
END
```

where:

XMU	is the reluctivity matrix $\left[\frac{1}{\mu}\right]\left(H = \left[\frac{1}{\mu}\right](B - B_r)\right)$. This is to be re-defined as necessary by you. This matrix is passed in as set-up for anisotropic permeability. If you do not re-define it, it remains anisotropic according to μ (T) given on the ISOTROPIC or ORTHOTROPIC options.
M(1)	is your element number.
M(2)	is the internal element number.
NN	is the integration point number.
MATNO	is the material identifier.
ID	is the size of the matrix.
CPTIM	is the frequency in a harmonic analysis.
DTIME	is the increment of time.
B	is the magnetic flux density in a transient analysis, or is zero in a harmonic analysis.

■ USIGMA

Input of Anisotropic Electric Conductivity Matrix

Description

For anisotropic electromagnetic analysis, this subroutine allows you to define an anisotropic conductivity matrix at each integration point in each element. The anisotropic permittivity matrix is defined with respect to the preferred orientation specified in the ORIENTATION option.

Format

Subroutine USIGMA is written with the following headers:

```
SUBROUTINE USIGMA (SIGMA,M,NN,MATNO,ID,CPTIM,DTIME)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION SIGMA(3,3),M(2)
    user coding
RETURN
END
```

where:

SIGMA	is the electric conductivity matrix, [σ] (J = [σ]E). This is to be re-defined as necessary by you. This matrix is passed in as set-up for anisotropic conductivity. If you do not re-define it, it remains as given through the ISOTROPIC or ORTHOTROPIC options.
M(1)	is your element number.
M(2)	is the internal element number.
NN	is the integration point number.
MATNO	is the material identifier.
CPTIM	is the transient time at the beginning of the increment; in a harmonic analysis, it is the frequency.
DTIME	is the increment of time.
ID	is the size of the matrix.

■ USPCHT

Definition of Specific Heat

Description

This subroutine allows you to define the specific heat in a heat transfer or coupled analysis. This is an alternative to the use of the ISOTROPIC or ORTHOTROPIC and TEMPERATURE EFFECTS options. This routine is called at each increment for every element in the mesh; hence, allowing you to specify a nonlinear relationship. This is often useful in welding or casting analyses.

Format

Subroutine USPCHT is written with the following headers:

```
 SUBROUTINE USPCHT (SPHEAT,M,NN,KC,INC,NCYCLE,MATS,NSTATS,DT,DTDL,
+CPTIM,TIMINC)
 IMPLICIT REAL *8 (A-H, O-Z)
 DIMENSION SIGMA(3,3),M(2),DT(NSTATS),DTDL(NSTATS)
    user coding
 RETURN
 END
```

where:

SPHEAT	is the specific heat per unit mass. This is to be defined by you.
M(1)	is your element number.
M(2)	is the internal element number.
NN	is the integration point number.
KC	is the layer number.
INC	is the increment number.
NCYCLE	is the cycle number.
MATS	is the material id.
NSTATS	is the number of state variables.
DT	is the temperature at the start of the increment.
DTDL	is the estimated increment of temperature.
CPTIM	is the time at the beginning of the increment.
TIMINC	is the increment of time.

■ CRPLAW

Input of Special Creep Law

Description

User subroutine CRPLAW allows you to specify the increment of creep strain.

The use of such a routine is flagged by setting the fifth field of the second block in the CREEP model definition option to zero. This subroutine is called as required during the analysis because of possible re-cycling due to nonconvergence. The number of times the routine is called in each increment is not fixed.

The program allows you to input your own creep law through subroutine CRPLAW.

The assumed form of the law is:

$$\dot{\varepsilon}^c = f(\bar{\sigma}, T, t, \bar{\varepsilon}^c, p, \alpha_1, \alpha_2, \text{etc})$$

where:

$\dot{\varepsilon}^c$	is the equivalent creep strain rate, in uniaxial tension.
$\bar{\sigma}$	is the current equivalent (J) stress, normalized for uniaxial tension.
T	is the current total temperature.
t	is the current total time.
$\bar{\varepsilon}^c$	is the current total equivalent creep strain, normalized for uniaxial tension.
p	is the hydrostatic stress.
α_1, α_2, etc.	are the state variables. The program requires you to program your creep law so that an equivalent creep strain increment is defined.

Format

Subroutine CRPLAW is written with the following headers:

```
 SUBROUTINE CRPLAW(EQCP,EQCPNC,STR,CRPE,T,DT,TIMINC,CPTIM,M,NN,KC,
+MATS,NDI,NSHEAR)
 IMPLICIT REAL *8 (A-H, O-Z)
 DIMENSION T(3),DT(1),STR(1),CRPE(1)
    user coding
 RETURN
 END
```

where:

EQCP — Passed in as total equivalent creep strain.

$$\bar{\varepsilon}^c = \Sigma\left(\frac{2}{3}\Delta\varepsilon^c{}_{ij}\Delta\varepsilon^c{}_{ij}\right)^{1/2}$$

or, for ORNL Constitutive Theory, equivalent total creep strain,

$$\bar{\varepsilon}^c = \left(\frac{2}{3}\Sigma\Delta\varepsilon^c{}_{ij}\Sigma\Delta\varepsilon^c{}_{ij}\right)^{1/2}$$

to be re-defined as equivalent primary creep strain increment.

EQCPNC — is the equivalent creep strain increment; to be defined by you in this routine. For ORNL Constitutive Theory, passed in as equivalent total primary creep strain. Otherwise undefined when passed in.
Must be redefined by you as equivalent creep strain increment.

STR — is the stress array.

CRPE — is the incremental creep strain array. If you want to define a creep strain law not following the normality condition, the creep strain increment can be defined here.

T(1) — is the current total equivalent (J_2) stress.

T(2) — is the current total hydrostatic stress.

T(3) — is the current total swelling strain (from subroutine VSWELL).

DT(1) — is the current total temperature.

DT(2),DT(3) — are the additional state variables read in user subroutine CREDE.

TIMINC — is the current time increment.

CPTIM — is the current total time.

M	is the current element number.
NN	is the integration point number.
KC	the layer number.
MATS	is the material id.
NDI	is the number of direct components of strain.
NSHEAR	is the number of shear components of strain.

The simplest way to define a creep strain increment from a given rate law $\dot{\varepsilon}^c = f(\sigma, \text{etc})$ is to multiply by Δt, the time increment:

$$\Delta\bar{\varepsilon}^c = \Delta t \cdot f(\bar{\sigma}, \text{etc})$$

As an example, suppose we wish to use the creep law (where A and B are constants):

$$\dot{\varepsilon}^c = A \sinh\left(\frac{\sigma}{B}\right)$$

This would be programmed as follows:

```
      SUBROUTINE CRPLAW(EQCP,EQCPNC,STR,CRPE,T,DT,TIMINC,CPTIM,M,
     +NN,KC,MATS,NDI,NSHEAR)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION T(1),DT(1),STR(1),CRPE(1)
C   DEFINE A AND B
      A = CONSTANT1
      B = CONSTANT2
C   OBTAIN SINH (T/B)
      S = T(1)
      IF (S.EQ.0.) GO TO 1
      SINHT = .5*(EXP(S/B)-EXP(-S/B))
      GO TO 2
 1    SINHT = 0.
 2    CONTINUE
C   NON DEFINE EQCPNC
      EQCPNC = TIMINC*A*SINHT
      RETURN
      END
```

The ORNL recommendations include the use of a strain hardening creep formulation. The following example of subroutine CRPLAW shows a simple technique of numerical solution for a strain hardening formulation based on equivalent total creep strain. The example is based on

a Blackburn formulation with a single primary term, but the technique is general and can be used for more complex formulations. The numerical inversion of the total creep equation for equivalent time is achieved by Newton's method:

$$t_{n+1} = t_n + \Delta t$$

$$t = \frac{-\bar{\varepsilon}^c + f(T, \bar{\sigma}, t_n)}{-f'(T, \bar{\sigma}, t_n)}$$

t_n is the solution for equivalent time at the nth iteration.

Δt is the correction to t at the nth iteration, and the total creep equation is

$$\bar{\varepsilon}^c = f(T, \bar{\sigma}, t)$$

with

$$f' = \frac{\partial f}{\partial t}$$

A tolerance of 10^{-6} has been placed on $\left|\frac{\Delta t}{t_n}\right|$.

Practical experience shows this needs about four or five iterations for the creep law in the example. The listing of CRPLAW follows:

```
      SUBROUTINE CRPLAW(EQCP,EQCPNC,STR,CRPE,T,DT,TIMINC,CPTIM,M,
     +NN,DC,MATS,NDI,NSHEAR)
      IMPILCIT *8 (A-H, O-Z)
      DIMENSION T(1),DT(1),STR(1),CRPE(1)
C  THIS ROUTINE FORMULATES THE STRAIN HARDENING FORMULATION OF
C  THE BLACKBURN CREEP LAW.
C  EPSILON
C  DOT=1/TIME SUB CAP T * (EPSILON SUR T - EPSILON SUSUPER T)
C  + EPSILON DOT SUB M
C  THE FOLLOWING DEFINITION APPLIES TO THE FUNCTION CODED BELOW
C  A IS LN(A)
C  B IS ALPHA*SIGMA
C  E IS N
C  C IS Q
C  C IS T
C  EPSILON SUB T, T, SUB T AND EPSILON SUB M DOT ARE GIVEN BY A CURVE
C  FUN + A*SINH TO N OF ALPHA SIGMA TIMES E TO Q/T EXP
```

```
C  IHARD=0 USES STRAIN HARDENING
C  STRAIN NOW DIMENSIONAL
C  TEMPERATURE IN FAHRENHEIT
C  STRESS IN PSI
C  TIME IN HOURS
      FTN(A,B,C,D,E)=EXP(A)*(.5*(EXP(B)-EXP(-B))**)E*EXP(C/D))
      IHARD=0
      IHARD=1
      EQCPNC=0
      IF(T(1).LT.25.)GO TO 1
      TRANK=DT(1)+459.67
      ET=FTN(2.76,1.976E-3*T(1),-1,03E4,TRANK,.08778)
      TT=FTN(-21.38,.09546E-3*T(1),4.54E4,TRANK,-2.31)
      EDOT=FTN(57.2,.02345E-3*T(1),-9.98E4,TRANK,6.933)
C  THE FOLLOWING IS A NEWTON METHOD TO EXPRESS T IN TERMS OF KNOWN
C  QUANTITIES. INITIAL GUESS IS T= (F SUB C- ET) / E DOT M
      IF(IHARD.EQ.1) GO TO 10
      TIME=CPTIM GO TO 2
 10   CONTINUE
      TIME=(100.*EQCP-ET)/EDOT
      FT=ET/TT
      IF(EQCP.EQ.0.) GO TO 4
  2   EFT=EXP (-TIME/TT)
      FT=FT*EFT/TT
  4   ST=EDOT
      EQCPNC=(FT+ST)*TININC*0.01
  1   RETURN
      END
```

■ VSWELL

Input of Special Swelling Law

Description

The user subroutine VSWELL allows you to include pure swelling (dilatational) creep in MARC.

Format

Subroutine VSWELL is written with the following headers:

```
SUBROUTINE VSWELL(SWELL,SIG,TEMP,N,NN,KC,CPTIM,TIMINC,MAT,DTEMP)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION SIG(3),TEMP(1),DTEMP(1)
    user coding
RETURN
END
```

where:

SWELL	is the increment of volumetric swelling $\left(= \frac{DV}{V}\right)$ defined by you.
SIG(1)	is the uniaxial equivalent of J2 stress.
SIG(2)	is the hydrostatic stress.
SIG(3)	is the current total swelling strain (accumulated from this routine).
	Note: This is a uniaxial component; that is, $\frac{1}{3}\frac{DV}{V}$
TEMP(1)	is the temperature.
TEMP(2),TEMP(3), etc.	are the additional state variables read in through user subroutine CREDE.
N	is the element number.
NN	is the integration point number.
KC	is the layer number.
CPTIM	is the total creep time.
TIMINC	is the current time increment.
MAT	is the material identification number.
DTEMP(1)	is the temperature increment.
DTEMP(2), etc.	are the increments of additional state variables.

You define the increment of dilatational creep by this routine, which is called at each integration point where constitutive calculations are being performed by the program. It is called automatically when any CREEP incremental option is used (AUTO CREEP, CREEP INCREMENT, etc.) and can be used alone or in combination with a Mises type creep law (subroutine CRPLAW). This subroutine is called as required during the analysis, so that, because of possible re-cycling due to nonconvergence, the number of times the routine is called in each increment is not fixed.

Example

The following is a typical irradiation swelling formulation:

$$\left(\frac{DV}{V}\right) = c(q \cdot t)^a \exp\left(b_0 + \frac{b_1}{T^1} + \frac{b_2}{T^2}\right)$$

where:

a, b_0, b_1, b_2, c are numerical constants, q is flux, t is time and T is temperature.

Differentiating with respect to time,

$$\frac{d}{dt}\left(\frac{DV}{V}\right) = acq^a t^{a-1} \exp\left(b_0 + \frac{b_1}{T^1} + \frac{b_2}{T^2}\right)$$

so that subroutine VSWELL becomes:

```
SUBROUTINE VSWELL(SWELL,SIG,TEMP,N,NN,KC,CPTIM,TIMINC,MAT,DTEMP)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION SIG(3),TEMP(1),DTEMP(1)
C=
Q=TEMP(2)
A=
B0=
B1=
B2=
SWELL=A*C*Q**A*CPTIM**(A-1.)
SWELL=SWELL*EXP(B0+B1/TEMP+B2/TEMP**2)
SWELL=SWELL*TIMINC
RETURN
END
```

This assumes flux increments q are entered into the second state variable using CREDE.

WKSLP

User Subroutine for Work-hardening Slope Definition

Description

This subroutine makes it possible for you to program the yield stress and the corresponding work-hardening slope directly as a function of equivalent plastic strain and temperature. See WORK HARD model definition option. You need to define the value of the slope of the equivalent stress vs. equivalent plastic strain. The current yield stress can be defined also. The specification of the latter is optional. If the value of the current yield is not given here, the program calculates it from the initial yield value and the work-hardening slopes defined in this routine.

In order to use this subroutine instead of the slope-break point data, you should set the number of work-hardening slopes equal to -1. No work-hardening slope break point data blocks should be included. The routine is called as required by the program during the elastic-plastic calculations. The number of times it is called per increment depends on the number of points going plastic, on the nonlinearity of the work-hardening curve, and on temperature dependence.

Format

Subroutine WKSLP is written with the following headers:

```
SUBROUTINE WKSLP(M,NN,KC,MATS,SLOPE,EBARP,ERAT,STRYT,DT,IFIRST)
IMPLICIT REAL *8 (A-H, O-Z)
    user coding
*RETURN
END
```

where:

M	is the current element number.
NN	is the integration point number.
KC	is the layer number.
MATS	is the material id.

SLOPE is the work-hardening slope to be defined by you as:

$$\frac{d\bar{\sigma}}{d\bar{\varepsilon}^p} + \frac{\bar{\sigma}(\bar{\varepsilon}^p, \dot{\bar{\varepsilon}}^p) - (\bar{\varepsilon}^p, 0)}{\dot{\bar{\varepsilon}}^p \Delta t}$$

$\bar{\sigma}$ = equivalent tensile stress = $\sqrt{\frac{3}{2} S_{ij} S_{ij}}$

$$S_{ij} = \sigma_{ij} - \frac{1}{3}\delta_{ij}\sigma_{kk}$$

EBARP is the current total equivalent plastic strain, $\bar{\varepsilon}^p$

$$\bar{\varepsilon}^p = \Sigma d\bar{\varepsilon}^p$$

$$d\bar{\varepsilon}^p = \sqrt{\frac{2}{3} d\varepsilon_{ij}^p d\varepsilon_{ij}^p}$$

ERAT is the equivalent plastic strain rate, $\dot{\bar{\varepsilon}}^p$

DT is the current total temperature.

IFIRST is passed in as 1 for initial yield curve.is passed in as 2 for the tenth cycle yield curve when ORNL constitutive theory is flagged.

STYRT is the current yield stress σ.

Note: $\frac{d\bar{\sigma}}{d\bar{\varepsilon}^p}$ is not the slope of the tensile stress-strain curve, which is $\frac{d\bar{\sigma}}{d\bar{\varepsilon}}$ with:

$$d\bar{\varepsilon} = d\bar{\varepsilon}^e + d\bar{\varepsilon}^p$$

The time increment, Δt, is given by variable TIMINC in common block CREEPS. You must take care to provide rate of change of stress with respect to plastic strain, not total strain. The second term in the SLOPE expression allows you to include strain-rate effect if desired. You must define SLOPE and STRYT in this routine. EBARP, DT, and IFIRST should not be changed.

Note: If UPDATE is used, the stresses are Cauchy (true) stress and the strains are logarithmic strains.

Example

Let us assume that yield surface can be expressed as:

$$\sigma_y = A(1 + \bar{\varepsilon}^p)^n$$

then,

$$\frac{\partial \sigma_y}{\partial \bar{\varepsilon}^p} = nA(1 + \bar{\varepsilon}^p)^{n-1}$$

the user subroutine would look like:

```
SUBROUTINE WKSLP(M,NN,KC,MATS,SLOPE,EBARP,ERAT,STRYT,DT,IFIRST)
IMPLICIT REAL *8 (A-H, O-Z)
A=
N=
N1=n-1
SLOPE=n*A*(1.+EBARP)**N1
STRYT=A*(1.+EBARP)**N
RETURN
END
```

■ USPRNG

Input of Nonlinear Spring, Dashpot and Foundation Stiffness

Description

User subroutine USPRNG permits the introduction of nonlinear spring constants for use with the SPRINGS and/or FOUNDATION options and input of nonlinear damping if the DASHPOT option is to be used. Your coding must supply both the ratio of the current value of spring stiffness to the data input value and the total spring force. For dynamic analysis, the ratio of damping coefficient can also be provided. The data value of the spring/dashpot constant, total time, and the element or spring number are made available to the subroutine. For harmonic analysis, the spring/dashpot constants can be a function of the frequency. Subroutine USPRNG is accessible whenever either the SPRINGS or the FOUNDATION option is used.

Format

Subroutine USPRNG is written with the following headers:

```
SUBROUTINE USPRNG(RATK,F,DATAK,U,TIME,N,NN,NSPRNG)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION RATK(2),DATAK(2),U(2),TIME(2),N(2),F(2)
    user coding
RETURN
END
```

where:

RATK(1)	is the ratio of the present value of spring stiffness to the data value given in the option input; to be defined by you.
RATK(2)	is the ratio of the present value of the damping coefficient to the data value given in the input; to be defined by you. This applies to SPRINGS in dynamic analysis only.
F(1)	is the force to be defined by you. (a) For springs: F = spring force. (b) For elastic foundation: F = pressure per unit area.
F(2)	is the damping force.
DATAK(1)	is the data value of spring constant (or foundation stiffness) as defined by you in SPRINGS/FOUNDATION options data input.
DATAK(2)	is the data value of the damping constant as defined by you in the SPRINGS option data input.

U(1)	(a) For springs: U(1) = U_2- U_1. (b) For elastic foundation: U(1) = U_n. (positive in the direction specified by face identification given in the FOUNDATION option).
U(2)	For dynamic spring/dashpot U(2) = $U_2 - U_1$.
TIME(1)	is the total time (for dynamic or creep analysis).
TIME(2)	is the frequency (for harmonic analysis with spring/dashpot).
N(1)	is the element number (only for elastic foundation).
N(2)	is the face number (only for elastic foundation).
NN	is the integration point number (only for elastic foundation).
NSPRNG	is the spring number, the position of the spring in the input data list (only for springs).

If the routine is called for an elastic foundation point, NSPRNG is zero.

If the routine is called for a spring, N and NN are zero.

Note that if you prefer to give the absolute value of the spring constant, rather than a ratio, the corresponding value in the SPRINGS or FOUNDATION option should be set to 1. The same applies for a damping constant.

■ UCRACK

Input of Ultimate Stress for Cracking Analysis

Description

This subroutine allows you to input a constant or a temperature dependent ultimate stress at each integration point of an element for cracking analysis. In addition, you can define the strain softening modulus and the crushing strain.

Format

Subroutine UCRACK is written with the following headers:

```
      SUBROUTINE UCRACK (SCRACK,ESOFT,ECRUSH,ECP,DT,DTDL,N,NN,KC,
     1 INC, NDI, NSHEAR, SHRFAC)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION ECP(1), DT(1), DTDL(1)
        user coding
      RETURN
      END
```

where:

SCRACK	is the user-defined ultimate cracking stress.
ESOFT	is the user-defined strain softening moduli.
ECRUSH	is the user-defined strain at which crushing occurs.
ECP	is the array of crack strains.
DT	is the array of state variables, temperature first.
DTDL	is the array of incremental state variables, temperature first.
N	is the element number.
NN	is the integration point number.
KC	is the layer number.
INC	is the increment number.
NDI	is the number of direct components.
NSHEAR	is the number of shear components.
SHRFAC	is the user-defined shear retention factor.

Let us assume that the ultimate stress looks like

$$\sigma^{CR} = A(1 - e^{-RT})$$

The user subroutine would look like

```
      SUBROUTINE UCRACK(SCRACK,ESOFT,ECRUSH,ECP,DT,DTDL,N,NN,KC,INC,
     1 NDI,NSHEAR, SHRFAC)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION ECP(1),DT(1),DTDL(1)
      A=
      R=
      TT=DT(1)+DTDL(1)+473.0
      SCRACK=A*(1.0D0-EXP(-R*TT))
      RETURN
      END
```

■ TENSOF

Input of Tension Softening Modulus for Cracking Analysis

Description

The tension softening modulus defines the post-failure behavior at an integration point. By default in the program, the reduction of the cracking stress to zero is a linear function of the crack strain. This routine allows you to define for instance a nonlinear behavior. The routine is automatically called for every crack in the analysis.

Format

Subroutine TENSOF is written with the following headers:

```
SUBROUTINE TENSOF (D,SP,GFP,DEP,ECP,SCRACK,SOSTR,ETSNEW,
                   ETSOFT,XH,SPECLN,JSOFT)
```

where:

D	is the stiffness in the crack direction term to be defined by you.
SP	is the stress at end of increment as function of crack strain to be defined by you.
GFP	is the change in stress due to incremental crack growth.
DEP	is the current strain increment.
ECP	is the crack strain at end of increment.
SCRACK	is the critical cracking stress given in input.
SOSTR	is the current cracking stress based on previous softening.
ETSNEW	is the current value of temperature dependent Young's modulus.
ESOFT	is the tension softening modulus given in input.
XH	is the characteristic element length.
SPECLN	is the test specimen length.
JSOFT	is the status indicator for softening. Used for plotting only.
= 1	inside softening range.
= 2	outside softening range.

Note that the definition of the stiffness D does not need to be exact. The correct definition of the stiffness only determines the speed of the convergence. In fact, in the above routine, a large negative value of the stiffness term should never be used as this would result in convergence problems. The stress definition, however, must be exact; otherwise, the wrong solution is obtained.

■ USHRET

Input of Shear Retention Factor for Cracking Analysis

Description

The shear retention factor is used to define the residual shear stiffness for a cracked integration point in a cracking analysis. The shear retention factor is defined as the factor with which the initial shear stiffness is multiplied. With this routine, you can define the shear retention factor to be, for instance, a function of the crack strain. The routine is automatically called for each existing crack.

Format

Subroutine USHRET is written with the following headers:

```
SUBROUTINE USHRET (FACTOR,ECRA1,ECRA2,ECRA12)
IMPLICIT REAL *8 (A-H, O-Z)

   user coding

RETURN
END
```

where:

FACTOR	is a user-defined shear retention factor to be defined here.
ECRA1	is the crack strain in the first crack direction.
ECRA2	is the crack strain in the second crack direction.
ECRA12	is the shear strain over the crack.

■ UVOID

Definition of the Initial Void Volume Fraction

Description

This subroutine allows the definition of the initial void fraction in an elastic plastic material when the damage model is being used. This subroutine is automatically called if the Gurson damage model is specified for a specific material.

Format

Subroutine UVOID is written with the following header:

```
SUBROUTINE UVOID(VOIDFI,M,NN,KC,MATS,X)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION M(2),X(1)

   user coding

RETURN
END
```

where:

VOIDFI	is the initial void fraction to be defined here.
M(1)	is your element number.
M(2)	is the internal element number.
NN	is the integration point number.
KC	is the layer number.
MATS	is the material id.
X	is the coordinate position of integration point.

■ UVOIDN

Definition of the Void Nucleation Rate

Description

This subroutine allows the definition of the void nucleation rate in a material using the Gurson model. This subroutine is called if the void nucleation method under the DAMAGE model definition option is set to 3.

In this model, the yield surface is given as:

$$F = \frac{\sigma_e^2}{\sigma_m^2} + 2q_1 f \cosh\left(\frac{q_2\sigma_{KK}}{2\sigma_m}\right) - (1-q_1 f)^2 = 0$$

where:

σ_e	is the effective stress.
σ_m	is the equivalent tensile stress.
f	is the void ratio.

Format

Subroutine UVOIDN is written with the following headers:

```
      SUBROUTINE UVOIDN(A,B,M,NN,KC,MATS,EPL,EPLAS,S,NDI,NSHEAR,DT,
     +DTDL)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION M(2),DT(1),DTDL(1),EPL(1)

        user coding

      RETURN
      END
```

where:

A	is the multiplier as shown below.
B	is the multiplier as shown below.
M(1)	is your element number.

M(2)	is the internal element number.
NN	is the integration point number.
KC	is the layer number.
MATS	is the material id.
EPL	is the plastic strain components.
EPLAS	is the equivalent plastic strain.
S	is the stress array.
NDI	is the number of direct components.
NSHEAR	is the number of shear components.
DT	is the array of state variables, temperature first.
DTDL	is the array of increment of state variables.

In this subroutine, the following type of stress controlled nucleation rate can be specified:

$$\dot{f} = A\dot{\bar{\sigma}} + B\dot{\sigma}_{kk}$$

where $\dot{\bar{\sigma}}$ is the von Mises equivalent stress rate, and $\dot{\sigma}_{kk}$ is the hydrostatic stress rate.

■ UPOWDR

Definition of Material Data for Powder Metallurgy Model

Description

Material properties of powder metals which are used in Hot Isostatic Pressing (HIP) are typically dependent upon both the temperature and the relative density of the material. This user subroutine provides an alternative mechanism to enter this data. This subroutine is called for all elements for which the POWDER option is used. The elastic, plastic, and thermal properties can be defined in this routine. In this model, the yield function, F, is defined as:

$$F = \frac{1}{\gamma}\left(S_{ij}S_{ij} + \frac{P^2}{\beta^2}\right)^{1/2} - \sigma_y$$

where:

γ and β	are material parameters to be entered here.
S	is the deviatoric stress.
P	is the hydrostatic stress.
σ_y	is the equivalent tensile stress.

The equivalent inelastic strain rate, $\dot{\varepsilon}$, is defined as:

$$\dot{\varepsilon} = \frac{1}{\eta}\left(\frac{F}{\sigma_y}\right)$$

where:

η	is the viscosity.

Format

Subroutine UPOWDR is written with the following headers:

```
      SUBROUTINE UPOWDR(E,G,POISS,GAMMA,BETA,VISC,SIGY,AMB,COMPF,REDENS,
     +DT,DTDL,DET,IHEAT,IHCPS)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION POWDAT(32)

        user coding

      RETURN
      END
```

where:

E	is the Young's moduli.
G	is the shear moduli.
POISS	is the is the Poisson's ratio.
GAMMA	is the parameter γ in the yield function.
BETA	is the parameter β in the yield function.
VISC	is the is the viscosity μ.
SIGY	is the temperature-dependent equivalent tensile stress σ_y
AMB	is the conductivity in a coupled analysis.
COMPF	is the specific heat in a coupled analysis.
REDENS	is the relative density.
DT	is the array of state variables, temperature first.
DTDT	is the array of increment of state variables.
DET	is the determinant which gives the change in volume.
IHEAT	is the indicates if this is the heat transfer calculation in a coupled analysis. = 0 stress pass. = 1 heat pass.

In the stress pass, you should define E, G, POISS, GAMMA, VISC, and SIGY.

In the heat transfer pass, you should define AMB and COMPF.

The values of E, G, POISS, GAMMA, BETA, VISC, AMB, COMPF upon entrance are the values calculated by the program based upon your input.

■ UPERM

Definition of Soil Permeability

Description

In a soil analysis, it might be necessary to define the permeability as a function of the porosity or other variables. This user subroutine allows you to enter a general nonlinear relationship. It is called during any coupled fluid-soil analysis.

Format

Subroutine UPERM is written with the following headers:

```
SUBROUTINE UPERM(PERMEA,M,NN,POROP,POROS,X,K,STRESS,NGENS)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION PERMEA(K,K),X(1),STRESS(NGENS)

   user coding

RETURN
END
```

where:

PERMEA	is the permeability matrix.
M	is the element number.
NN	is the integration point number.
POREP	is the pore pressure.
POROS	is the soil porosity.
X	is the array of integration point coordinates.
K	is the dimension of the permeability matrix.
STRESS	is the effective stress matrix.
NGENS	is the number of stress components.

■ UMOONY

Mooney-Rivlin Material

Description

This subroutine allows you to redefine the constants used in the strain energy function. This data is normally entered through the MOONEY model definition option.

The form of the strain energy function is:

$$W = C_{10}(I_1 - 3) + C_{01}(I_2 - 3) + C_{11}(I_1 - 3)(I_2 - 3) + C_{20}(I_1 - 3)^2 + C_{30}(I_1 - 3)^3$$

Format

Subroutine UMOONY is written with the following headers:

```
SUBROUTINE UMOONY(C10,C01,C11,C20,C30,T,N,NN,MATS)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION N(2)
   user coding
RETURN
END
```

where:

C10,C01,C11,C20,C30	are the values used in the strain energy function to be defined you.
T	is the temperature.
N(1)	is the user element number.
N(2)	is the internal element number.
NN	is the integration point number.
MATS	is the material identifier.

■ UENERG

Strain Energy Function

Description

This subroutine allows you to define your own elastic strain energy function for incompressible materials. Normally the five constant second order model is entered using the MOONEY model definition option. This option must still be used to invoke this user subroutine. This routine can be used when either the total Lagrange or updated Lagrange procedure is used.

Format

Subroutine UENERG is written with the following headers:

```
 SUBROUTINE UENERG (W,W1,W2,W11,W12,W22,WI1,WI2,C10,C01,C11,C20,
+C30,N,NN)
 IMPLICIT REAL *8 (A-H, O-Z)
 DIMENSION N(2)
   user coding
 RETURN
 END
```

where:

W	is the strain energy density.
W1	is $\partial W/\partial I_1$.
W2	is $\partial W/\partial I_2$.
W11	is $\partial^2 W/\partial I_1^2$.
W12	is $\partial^2 W/\partial I_1 \partial I_2$.
W22	is $\partial^2 W/\partial I_2^2$.
WI1	is the first strain invariant of the left Cauchy-Green tensor - 3.0.
WI2	is the second strain invariant of the left Cauchy-Green tensor - 3.0.
C10, C01, C11, C20, C30	are the five material parameters of the Mooney formulation.
N(1)	is the user element number.
N(2)	is the internal element number.
NN	is the integration point number.

■ UOGDEN

Definition of Ogden Material Parameters

Description

This subroutine allows the definition of the Ogden material parameters. Additionally, any temperature dependence of these properties can be entered here. The OGDEN option must be used to indicate that the element uses this material law, and the number of terms in the series must be entered through the model definition option. When the Ogden model is used in the updated Lagrange formulation, this routine is called twice per integration point. The first time for the bulk modulus; the second time for the μ and λ coefficients.

The strain energy function for this material is written as:

$$W = \sum_{i=1}^{n} \frac{\mu_i}{\alpha_i}(\lambda_1^{\alpha_i} + \lambda_2^{\alpha_i} + \lambda_3^{\alpha_i} - 3) + 4.5K(J^{1/3} - 1)^2$$

Format

Subroutine UOGDEN is written with the following headers:

```
 SUBROUTINE UOGDEN(MATS,NSER,M,NN,KC,INC,CPTIM,TIMINC,XMTDAT,BULK,
+DT,DTDT)
 IMPLICIT REAL *8 (A-H, O-Z)
 DIMENSION XMTDAT(2,NSER),M(2),DT(1),DTDL(1)

   user coding

 RETURN
 END
```

where:

MATS	is the material id.
NSER	is the number of terms in the series.
M(1)	is your element number.
M(2)	is the internal element number.
NN	is the integration point number.

KC	is the layer number.
INC	is the increment number.
CPTIM	is the time at the beginning of the increment.
TIMINC	is the time step.
XMTDAT(1,i)	is the value of μ_i.
XMTDAT(2,i)	is the value of α_i.
BULK	is the bulk modulus.
DT	is the array of state variables, temperature first.
DTDT	is the array of increments of state variables.

■ UELDAM

Definition of Damage Parameters in Ogden Model

Description

This subroutine allows you to define the damage parameters for the Ogden model. There are two types of damage: one is associated with the deviatoric (shear) behavior, and one is associated with the dilatational (volumetric) behavior (additional details can be found in *Volume A: User Information*). This routine is, therefore, called twice per integration point, once for deviatoric behavior and once for volumetric behavior. This routine is called only if the damage type is set to 6 through the DAMAGE model definition option.

Format

Subroutine UELDAM is written with the following headers:

```
      SUBROUTINE UELDAM(M,N,NN,KC,INC,LOVL,MATS,TIMINC,CPTIM,TOTEN,
     2                  DEVEN,TOTEND,TOTENV,SURFC,SURFD,DT,DTDL,
     3                  DAMD,DDAMD)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION DT(1),DTDL(1)
        user coding
      RETURN
      END
```

where:

M	is your element number.
N	is the internal element/elsto number.
NN	is the integration point number.
KC	is the layer number.
INC	is the increment number.
LOVL	is 4 for assembly phase. is 6 for stress recovery phase.
MATS	is the material set number.
TIMINC	is the time increment.
CPTIM	is the time at the beginning of the increment.

TOTEN is the total instantaneous strain energy at the end of the current step excluding damage.

DEVEN is the deviatoric part of the instantaneous strain energy at the end of the current step excluding damage.

TOTEND is the stored deviatoric energy at previous step (including damage).

TOTENV is the stored volumetric energy at previous step (including damage).

SURFC is the current radius of continuous damage surface.

SURFD is the current radius of discontinuous damage surface.

DT is the temperature.

DTDL is the incremental temperature.

Required Output

DAMD is the value of Kachanov deviatoric damage parameter.

DDAMD is the derivative of the damage parameter with respect to the maximum total strain energy.

■ HYPELA

Hypoelastic Material

Description

This subroutine is used to define elastic constants as functions of elastic strain and state variables. The hypoelastic option is described in *Volume A: User Information*.

Format

Subroutine HYPELA is written with the following headers

```
      SUBROUTINE HYPELA(D,G,E,DE,S,TEMP,
     1 DTEMP,NGENS,N,NN,KC,MATS,NDI,NSHEAR)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION D(NGENS,NGENS),G(NGENS),E(NGENS),
     1 DE(NGENS),S(NGENS),TEMP(1),DTEMP(1),N(2)

        user coding

      RETURN
      END
```

where:

D(NGENS,NGENS)	is the strain increment – stress increment transformation matrix (to be defined in this routine).
G(NGENS)	is the stress increment caused by change in state variables (to be defined in this routine).
E(NGENS)	are the total elastic strains at the beginning of the increment.
DE(NGENS)	are the increments of elastic strains.
S(NGENS)	are passed in as the total stresses at the beginning of the increment, $\sigma_{ij}(\varepsilon, T)$. This should be updated by you to be $\sigma_{ij}(\varepsilon_{ij} + \Delta\varepsilon_{ij}, T + DT)$.
TEMP(1)	are the total state variables at the beginning of the increment (temperature first).
DTEMP(1)	are the increments of state variables (temperature first).
NGENS	is the size of the D matrix.
N(1)	is the user element number.
N(2)	is the internal element number.

NN	is the integration point number.
KC	is the layer number (shells or beams).
MATS	is the material identifier.
NDI	is the number of direct stress components.
NSHEAR	is the number of shear stress components.

This routine is called at each point where hypoelastic constitutive calculations are required as defined in the HYPOELASTIC option. For Herrmann formulation of elements, the last strain component is the volumetric strain and the last stress component the mean pressure constraint. For curvilinear based elements (doubly curved shell elements 4, 8, and 24) mixed tensor components are used:

$$\Delta\sigma^{i}_{j} = D^{i}_{jkl}\Delta\varepsilon^{lk} + G^{i}_{j}$$

Note that the total stress at the end of the increment should be defined exactly in S – not as the stress at the beginning of the increment plus the linearized stress increment. By this means, the residual load correction (Newton-Raphson procedure) can operate effectively. This point is critical for accurate solutions. The change in stress consists of the changes due to both mechanical deformation $\Delta\varepsilon$ and the changes due to temperature dependent properties. The effects of thermal strains are taken care of by including the coefficient of thermal expansion in the HYPOELASTIC option.

The strain measure passed into this routine and the stress measure that you need to calculate and pass back depend on the parameter chosen in the PARAMETER option. These measures are summarized below:

Parameter	Strain	Stress
None	Engineering	Engineering
LARGE DISP	Green Lagrange	Second Piola Kirchhoff
LARGE DISP FINITE UPDATE	Logarithmic	Cauchy

■ HYPELA2

User-defined Material Behavior

Description

This subroutine is used to define user-defined material behavior in conjunction with the HYPOELASTIC model definition option of MARC. You must define the stress and the tangent stress-strain relationship and other state variables for the current strain increment. The HYPOELASTIC option is described in *Volume C: Program Input.*

Format

Subroutine HYPELA2 is written with the following headers

```
      SUBROUTINE HYPELA2(D,G,E,DE,S,T,DT,NGENS,N,NN,KC,MATS,NDI,NSHEAR,
     2                   DISP,DISPT,COORD,FFN,FROTN,STRECHN,EIGVN,FFN1,
     3                   FROTN1,STRECHN1,EIGVN1,NCRD,ITEL,NDEG,NDM,NNODE,
     4                   JTYPE,LCLASS,IFR,IFU)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION E(1),DE(1),T(1),DT(1),G(1),D(NGENS,NGENS),S(1)
      DIMENSION N(2),COORD(NCRD,NNODE),DISP(NDEG,NNODE),
     2          DISPT(NDEG,NNODE),FFN(ITEL,ITEL),FROTN(ITEL,ITEL)
     3          STRECHN(ITEL),EIGVN(ITEL,ITEL),FFN1(ITEL,ITEL)
     4          FROTN1(ITEL,ITEL),STRECHN1(ITEL),EIGVN1(ITEL,ITEL)

        user coding

      RETURN
      END
```

where:

D	is the stress strain law to be formed.
G	is the change in stress due to temperature effects.
E	is the total strain.
DE	is the increment of strain.
S	is the stress to be updated by you.
T	is the state variables (comes in at t = n; must be updated to have state variables at t = n +1).
DT	is the increment of state variables.

NGENS	is the size of the stress-strain law.
N	is the element number.
NN	is the integration point number.
KC	is the layer number.
MATS	is the material identification number.
NDI	is the number of direct components.
NSHEAR	is the number of shear components.
DISP	is the incremental displacements.
DISPT	is the displacements at t = n (at assembly lovl = 4) and the displacements at t = n +1 (at stress recovery lovl = 6).
COORD	is the coordinates.
NCRD	Is the number of coordinates.
NDEG	is the number of degrees of freedom.
ITEL	is the dimension of F and R; 2 for plane-stress and 3 for the rest of the cases.
NNODE	is the number of nodes per element.
JTYPE	is the element type.
LCLASS	is the element class.
IFR	is set to 1 if R has been calculated.
IFU	is set to 1 if STRECH has been calculated.

at t = n (or the beginning of the increment):

FFN	is the deformation gradient.
FROTN	is the rotation tensor.
STRECHN	is the square of principal stretch ratios, lambda (i).
EIGVN (I,J)	I principal direction components for J eigenvalues.

at t = n +1 (or the current time step):

FFN1	is the deformation gradient.
FROTN1	is the rotation tensor.
STRECHN1	is the square of principal stretch ratios, lambda (i).
EIGVN1(I,J)	is the I principal direction components for J eigenvalues.

The following operation obtains UN1 (stretch tensor) at t = n +1:

```
CALL SCLA(UN1,0.D0,ITEL,ITEL,1)
DO 3 K+1,3
 DO 2 i=1,3
  DO 1 j=1,3
   UN1 (I,J)=UN1(I,J)+DSQRT(STRECHN1(K)*EIGVN1(I,K)*EIGVN1(J.K)
  CONTINUE
 CONTINUE
CONTINUE
```

This subroutine gives you the ability to implement arbitrary material models. MARC supplies you the total displacement, incremental displacement, total strain, the strain increment, a vector of state variables, and other information. You are expected to calculate the stress that corresponds to the current strain and state variables.

For large strain behavior (elastic, viscoelastic, etc.), only the LARGE DISP (for total Lagrangian procedure) or LARGE DISP and UPDATE flags (for updated Lagrangian procedure) are used.

If the flags UPDATE, LARGE DISP, and FINITE are used, large strain calculations are done using a rate based stress-strain law (refer to "Finite Strain Plasticity with additive deocmposition of strain rates" in Volume A: Theory, Chapter 5, under "Nonlinear Analyses", "Elastic-Plastic Analysis Computational Procedures for Elastic-Plastic Analysis". This set of parameters is more suited to calculations involving large strain inelastic behavior (plasticity).

The use of stress-strain measures (also reflected in the output) for the material model is summarized below:

Parameter	Strain	Stress
None	Engineering	Engineering
LARGE DISP	Green Lagrange	Second Piola Kirchhoff
LARGE DISP UPDATE	Logarithmic	Cauchy
LARGE DISP FINITE UPDATE	Logarithmic	Cauchy

The total and incremental strains are passed through the argument list. The deformation gradient, principal stretches, rotation tensor, and eigenvectors are also passed at the beginning and end of the increment. STRECHN1 stores the value of the squares of the trial stretches, and EIGVN1(I,J) stores the i^{th} eigenvector component corresponding to the j^{th} eigenvalue of C where C is the right Cauchy-Green tensor at t = n + 1.

For thermal stress problems, you need to calculate and return the change in stress G, due to temperature dependent material properties.

$$G_i = (D^{\theta_{n+1}} - D^{\theta_n})_{ij} (\varepsilon_n^e)_j$$

where θ_n and θ_{n+1} are the temperatures at time t = n and t = n + 1, respectively.

If there are any state variables in the problem, you can use the array T() to update and return these state variables. The increments of the state variables should be calculated and returned as the array DT(). T() and DT() are the size NSTATS if NSTATS is the number of state variables and is set in the PARAMETER option in the input file. It must be remembered that T(1) and DT(1) are reserved for the temperature and the temperature increment respectively and is supplied to you by MARC. You must not change the values of T(1) and DT(1) even in isothermal problems with state variables. All variables T(2) to T(NSTATS) and DT(2) to DT(NSTATS) are accessible to you.

You also need to provide the tangent stiffness D for the current increment.

$$D_{ij} = \frac{\partial(\Delta\sigma_i)}{\partial(\Delta\varepsilon_j)}$$

The rate of convergence for a nonlinear problem depends critically on your supplied tangent stiffness, D. Before using the subroutine for large problems, you should check the subroutine for one element problems under displacement and load control boundary conditions. The displacement controlled boundary condition problem checks the accuracy of the stress update procedure while the load controlled problem checks the accuracy of the tangent stiffness. A fully consistent exact tangent stiffness provides quadratic convergence of the displacement or residual norm.

■ UFINITE

Finite Deformation Isotropic Material Models

Description

This subroutine is used for finite deformation isotropic material models based on principal stretches. Both nonlinear elasticity and large strain plasticity models can be implemented using this subroutine. This routine requires the use of the PLASTICITY, 5 parameter. UFINITE is available for plane strain, generalized plane strain, axisymmetric, axisymmetric with twist and 3D elements.

Format

Subroutine UFINITE is written with the following headers:

```
      SUBROUTINE UFINITE(STRECH,EIGV,DETFE,DETFT,DEFGR,DEFINV,DT,DTDL,
     2                   STRESS,TANGENT,M,NN,GF,D)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION STRECH(3),STRESS(3),TANGENT(3,3),EIGV(3,3),DEFGR(3,3),
      DIMENSION N(2),COORD(NCRD,NNODE),DISP(NDEG,NNODE),
     2          DEFINV(3,3),DT(1),DTDL(1),GF(1),D(1)

        user coding

      RETURN
      END
```

where

Input:

STRECH	is the squares of deviatoric trial elastic principal stretch ratios.
EIGV(I,J)	is the I principal direction components for J eigenvalues of the trial elastic left Cauchy-Green tensor (Finger tensor).
DETFE	is the elastic part of the Jacobian.
DETFT	is the total Jacobian.
DEFGR	is the total deformation gradient for continuum elements.
DEFINV	is the inverse of the total deformation gradient.
DT	is the array of the total state variables (temperature is first).
DTDL	is the array of the incremental state variables.
M	is your element number.

NN	is the integration point number.
D	is the array for material properties defined as D(1) = bulk modulus at (DT + DTDL) D(2) = shear modulus at (DT +DTDL) D(3) = initial yield stress (at zero effective plastic strain) D(4) = kinematic hardening modulus at (DT + DTDL) D(5) = portion of isotropic hardening D(6) = portion of kinematic hardening D(7) = coefficient of linear thermal expansion D(8) = bulk modulus at DT D(9) = shear modulus at DT.

Required Output:

STRESS	is the principal deviatoric Kirchhoff stress at the end of the increment.
TANGENT	is the elasto-plastic material tangent in the principal space; relating the total deviatoric Kirchhoff stress in principal space to the total principal deviatoric logarithmic strains.
GF	is the stress change due to temperature dependent properties.

This subroutine allows you to implement arbitrary finite elasticity and large strain plasticity models. You do not need to be concerned with preserving objectivity under large rotations in large strain problems, but must only deal with the small strain problem. You need to update principal deviatoric Kirchhoff stresses and provide a consistent deviatoric part of tangent in principal space and calculate any change in stresses due to temperature dependent thermal properties. MARC calculates the kinematic large strain contributions to the tangent automatically. You do not need to calculate the pressure or the volumetic part of the tangent. Also, transformation from the principal to global space for both stresses and the tangent is done automatically by MARC. You can refer to user subroutine UPSTRECH for the analogus formulation for elasticity

If there are any state variables in the problem, you can use the array DT() to update and return these state variables. The increments of the state variables must be calculated and returned as the array DTDL(). DT() and DTDL() are the size NSTATS where NSTATS is the number of state variables and is set in the PARAMETER option in the input file. It must be remembered that DT(1) and DTDL(1) are reserved for the temperature and the temperature increment, respectively and are supplied to you by MARC. You must not change the values of DT(1) and DTDL(1) even in isothermal problems. All variables DT(2) to DT(NSTATS) and DTDL(2) to DTDL(NSTATS) are accessible to you.

The framework used in this subroutine is based on principal stretches of the trial left Cauchy-Green tensor. For more details, please refer to the work of Simo and coworkers.

The rate of convergence of the global residual in this approach is critically dependent on the accuracy of the consistent tangent and the accuracy of the stress update procedure.

■ UPSTRECH

Definition of Generalized Principal Stretch-based Elasticity Models

Description

This subroutine allows you to define a nonlinear elasticity law based upon the principal stretches. This model is activated using the OGDEN option. The analysis is done in an updated Lagrange framework. The ELASTICITY parameter must be used. This routine allows you to specify the nonlinear deviatoric behavior. A nonlinear volumetric behavior can be controlled using the UOGDEN subroutine to modify the bulk modulus.

Format

Subroutine UPSTRECH is written with the following headers:

```
 SUBROUTINE UPSTRECH(STRECH,EIGV,DETFE,DETFT,DEFGR,DEFINV,DT,DTDL,
2                    STRESS,TANGENT,N,NN,LOVL,MATS,ENERGY)

 IMPLICIT REAL*8 (A-H,O-Z)

 DIMENSION STRECH(3),STRESS(3),TANGENT(3,3)EIGV(3,3)DEFGR(3,3)
2          DEFINV(3,3),DT(1),DTDL(1)

    user coding

  RETURN
  END
```

where:

Input:

STRECH	is the squares of elastic deviatoric principal stretch ratios.
EIGV(I,J)	is the I principal direction components for J eigenvalues.
DETFE	is the elastic part of the Jacobian.
DETFT	is the total Jacobian.
DEFGR	is the total deformation gradient for continuum elements.
DEFINV	is the inverse of total deformation gradient.

DT	is the array of total state variables (temperature is first).
DTDL	is the array of incremental state variables.
M(1)	is your element number.
M(2)	is the internal element number.
LOVL	is 4 for assembly. is 6 for stress recovery phase.
MATS	is the material identifier.

Required Output:

STRESS	is the deviatoric principal Kirchhoff stress, first derivative of the deviatoric strain energy with respect to deviatoric stretches.
TANGENT	is the deviatoric material tangent in principal space, second derivative of the deviatoric strain energy.
ENERGY	is the total deviatoric energy if rubber damage model is used.

To obtain U (stretch tensor), you must do the following operation:

```
      DIMENSION U(3,3)
      CALL SCLA(U,0.D0,3,3,1)
      DO 3 K=1,3
       DO 2 I=1,3
        DO 1 J=1,3
         U(I,J)=U(I,J)+DSQRT(STRECH(K))*EIGV(I,K)*EIGV(J,K)
1      CONTINUE
2     CONTINUE
3    CONTINUE
```

As an example, if the deviatoric part of the strain energy function is given as:

$$W = W(\lambda_i) \qquad (i = 1, 3)$$

where λ_i is the elastic deviatoric principle stretches passes through the argument list.

Hence, the deviatoric Kirchhoff stress, β_j, stored in STRESS is:

$$\beta_j = \lambda_j \frac{\partial W}{\partial \lambda_i} \delta_{ij} - \frac{1}{3} \sum_{i=1}^{3} \lambda_i \frac{\partial W}{\partial \lambda_i} \qquad (j = 1, 3)$$

Also the deviatoric material tangent in principal space, γ_{ij}, stored in tangent is:

$$\gamma_{ij} = \lambda_j \frac{\partial}{\partial \lambda_j}\left(\lambda_i \frac{\partial W}{\partial \lambda_i}\right) + \frac{1}{9}\sum_{p=1}^{3}\sum_{q=1}^{3}\lambda_q \frac{\partial}{\partial \lambda_q}\left(\lambda_p \frac{\partial W}{\partial \lambda_p}\right) - \frac{1}{3}\left[\sum_{p=1}^{3}\lambda_j \frac{\partial}{\partial \lambda_j}\left(\lambda_p \frac{\partial W}{\partial \lambda_p}\right)\right] + \sum_{p=1}^{3}\lambda_p \frac{\partial}{\partial \lambda_p}\left(\lambda_i \frac{\partial W}{\partial \lambda_i}\right) \qquad (i = 1, 3 \text{ and } j = 1, 3)$$

You do not have to calculate the pressure or the volumetric part of the tangent in this model. Also, transformation from the principal to global space for both stresses and the tangent is done automatically by MARC.

■ GENSTR

Generalized Stress Strain Law (Shells & Beams)

Description

This subroutine allows you to enter the generalized stress-strain law for shells and beams which are conventionally integrated through their thickness. This is often convenient in composite analysis where the experimental information is for the total material, not individual plies. This option is activated using the SHELL SECT parameter. As no layer integration is performed, the number of layers can be set to one.

You need to provide the generalized stress-strain law D and the total generalized stress at the end of the increment.

Format

Subroutine GENSTR is written with the following headers:

```
      SUBROUTINE GENSTR(D,DC,FCRP,ETOTA,DE,HT,S,T,DT,ER,EC,
     * SR,SC,NGENS,N,NN,MATS,IHRESP,ICRESP)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION D(NGENS,NGENS),DC(NGENS,NGENS),FCRP(1),ETOTA(1),
     * DE(1),S(1),T(1),DT(1),ER(1),EC(1),SR(1),SC(1),N(2)
        user coding
      RETURN
      END
```

where:

D	is the generalized real stress-strain law to be defined here.
DC	is the generalized imaginary stress-strain law.
FCRP	is the change in stress due to 'temperature effects' to be defined here.
ETOTA	is the total strain array.
DE	is the increment of strain array.
HT	is the shell thickness.
S	is the stress array.
T	are the state variables (temperature).
DT	are the increments of state variables.

ER	is the real strain array during harmonic sub-increment.
EC	is the imaginary strain array during harmonic sub-increment.
SR	is the real harmonic stress.
SC	is the imaginary harmonic stress.
NGENS	is the number of generalized stress.
N(1)	is the user element number.
N(2)	is the internal element number.
NN	is the integration point number.
MAT	is the material number.
IHRESP	is the flag to indicate harmonic sub-increment.
IHRESP=0	during a transient analysis.
IHRESP=1	during a harmonic sub-increment.
ICRESP	Indicates complex harmonic sub-increment.

During transient increments, you define D, S, and FCRP.

During harmonic subincrements the user defines D, DC, SR, and SC.

■ UBEAM

Input for Nonlinear Beam

Description

User subroutine UBEAM allows you to define nonlinear elastic cross-section properties as a function of generalized elastic strains and state variables for beam element 52 or beam element 98:

This is used in conjunction with the hypoelastic option. You must use the HYPOELASTIC model definition option.

Format

Subroutine UBEAM is written with the following headers.

```
 SUBROUTINE UBEAM (D,FCRP,DF,DFI,ETOT,DE,DEI,S,SI,GS,GSI,TEMP,
+DTEMP,NGENS,N,NN,MATS)
 IMPLICIT REAL *8 (A-H, O-Z)
 DIMENSION D(NGENS,NGENS),DF(1),S(1),GS(NGENS),DE(NGENS),TEMP(1),
+DTEMP(1),FCRP(1),ETOT(1),DFI(1),DEI(NGENS),SI(1),GSI(1)
   user coding
 RETURN
 END
```

where:

D	is the matrix of cross-section stiffness properties (to be defined in this routine).
FCRP	is the generalized stress increment caused by change in state variables (to be defined in this routine).
DF	are the increments of generalized stress (to be defined in this routine).
DFI	are the increments of imaginary generalized stress, if complex harmonic analysis.
ETOT	are the total generalized strains.
DE	are the increments of generalized strain.
DEI	are the increments of imaginary generalized strain, if complex harmonic analysis.
S	is passed in as the total stress at the beginning of the increment and must be redefined as the total stress at the end of increment.
SI	are the increments of imaginary stress, if complex harmonic analysis.

GS	is passed in as the total generalized stress at the beginning of the increment, and must be redefined as the total stress generalized at the end of the increment.
GSI	are the increments of generalized harmonic stress, if complex harmonic analysis.
TEMP	are the total state variables at the beginning of the increment.
DTEMP	are the increments of state variables.
NGENS	is the number of generalized stress.
N	is the element number.
NN	is the integration point number.
MATS	is the material identifier.

The components of generalized strain and stress for element 52 are:

ETOT(1)	Axial strain
ETOT(2)	Curvature change in first bending direction
ETOT(3)	Curvature change in second bending direction
ETOT(4)	Twist of the beam
S(1)	Axial force
S(2)	Bending moment in first bending direction
S(3)	Bending moment in second bending direction
S(4)	Twisting moment

The components of generalized strain and stress for element 98 are:

ETOT(1)	Axial strain
ETOT(2)	Local γ_{xy} shear
ETOT(3)	Local γ_{yz} shear
ETOT(4)	Curvature change in first bending direction
ETOT(5)	Curvature change in second bending direction
ETOT(6)	Twist of the beam
S(1)	Axial force
S(2)	Local τ_{xy} shear
S(3)	Local τ_{yz} shear
S(4)	Bending moment in the first bending direction
S(5)	Bending moment in the second bending direction
S(6)	Twisting moment

Note: This routine should not be used if you desire that the material properties or the beam cross-section data are design variables. Use the ISOTROPIC and GEOMETRY option instead.

UPHI

Input of PHI Function in Harmonic Analysis

Description

This subroutine allows the input of PHI functions to be expressed analytically. The values of PHI are then passed into a MARC routine where they are used in calculation of the Laplace transform for harmonic analysis.

Format

Subroutine UPHI is written with the following headers:

```
      SUBROUTINE UPHI(ELCG,FREQ,WI1,WI2,C10,C01,C11,C20,C30,NDI,NSHEAR,
     *FI0,FI1,FI2,FI11,FI12,FI21,FI22,IFLAG,DERIVS)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION ELCG(1),DERIVS(1)

        user coding
      RETURN
      END
```

where:

ELCG	is the left Cauchy-Green strain vector.
FREQ	is the excitation frequency in radians/ time unit.
WI1,WI2	are the first and second invariants of ELCG.
C10,C01,C11,C20,C30	are the five material parameters of the Mooney formulation.
NDI	is the number of direct strain components.
NSHEAR	is the number of shear strain components.
FI0,FI1,FI2,FI11,FI12,FI21,FI22	are the seven PHI functions which should be defined in this routine by you.
IFLAG = 1:	The sine PHI functions should be defined.
IFLAG = 2:	The cosine PHI functions should be defined.
DERIVS	is the array which contains the variables W, W1, W2, W11, W12, W21, and W22.

where:

`W`	is the strain energy density.
`W1`	is $\partial W / \partial I_1$.
`W2`	is $\partial W / \partial I_2$.
`W11`	is $\partial^2 W / \partial I_1^2$.
`W12`	is $\partial^2 W / \partial I_1 \partial I_2$.
`W21`	is $\partial^2 W / \partial I_2 \partial I_1$.
`W22`	is $\partial^2 W / \partial I_2^2$.

UCOMPL

Input of Viscous Stress Strain Relationship

Description

Subroutine UCOMPL allows you to input a real (elastic) and imaginary (damping) stress-strain relation for complex harmonic analysis. If not used, only the real portion is formed in the conventional manner. This routine is called for all elements, integration points, and layers in a harmonic subincrement. You specify the C matrix and can alter the existing B matrix if necessary. The stress is then calculated from $\sigma = B\varepsilon + C\dot{\varepsilon}$ where ε, $\dot{\varepsilon}$ are the harmonic strain and strain rate, respectively.

Format

Subroutine UCOMPL is written with the following headers:

```
      SUBROUTINE UCOMPL(C,B,ETOT,EELAS,EPLAS,S,T,XINTP,COORD,
     2 DISPT,FREQ,N,NN,KC,NGENS,INC,INCSUB,NDEG,NCRD,NDI,NSHEAR)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION C(NGENS,NGENS),B(NGENS,NGENS),ETOT(1),
     2 EELAS(1),EPLAS(1),T(1),XINTP(NCRD),COORD(NCRD,1),
     2 DISPT(NDEG,1),N(2)
C     USER SUBROUTINE TO INPUT A COMPLEX STRESS STRAIN LAW
C     FOR HARMONIC ANALYSIS C IS IMAGINARY PART
C     B IS REAL PART

        user coding
      RETURN
      END
```

where:

C	is the imaginary damping part of the stress-strain law.
B	is the real elastic part of the stress-strain law.
ETOT	are the total strains.
EELAS	are the total elastic strains.
EPLAS	are the plastic strains.
S	are the stresses.

`T`	are the total state variables (temperature first).
`XINTP`	are the coordinates of this integration point.
`COORD`	are the coordinates of the nodes of this element.
`DISPT`	are the total displacements of the nodes of this element.
`FREQ`	is the harmonic frequency in radians/time unit.
`N(1)`	is your element number.
`N(2)`	is the internal element number.
`NN`	is the integration point number.
`KC`	is the layer number.
`NGENS`	is the number of stress-strain components.
`INC`	is the increment number.
`INCSUB`	is the subincrement number.
`NDEG`	is the number of degrees of freedom per node.
`NCRD`	is the number of coordinate directions per node.
`NDI`	is the number of direct component of stress.
`NSHEAR`	is the number of shear components of stress.

GAPU

Input of Gap Direction And Closure Distance

Description

This subroutine allows input or modification of the direction and closure distance of gap element type 12 and 97 based on the current position of the end nodes of the element. This makes it possible to model contact sliding along curved surfaces which may occur in the analysis of metal forming problems. Although the gap direction and closing distance can be changed, this subroutine does not allow for finite sliding of two meshes with respect to each other, since the load transfer path is unchanged. In addition, it allows for specification of a nonlinear relationship between the normal force and the maximum friction force instead of the regular linear Coulomb relation.

Note: If this subroutine is used to change the direction of the gap, friction should not be included.

The subroutine also allows you to specify certain tolerances to control gap closure and friction iterations. This last feature is not generally used.

Format

Subroutine GAPU calls for the following headers:

```
      SUBROUTINE GAPU(DIR,DIST,X1,X4,TOL1,TOL2,TOL3,M,MSUB,INC,
     +NCR,FN,FF)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION DIR(3),X1(3),X4(3),M(2)
        user coding
      RETURN
      END
```

where:

DIR	is the array of direction cosines of the current gap direction. This can be modified by you.
DIST	is the current closure distance (distance that the nodes must travel to obtain closure), which is to be defined by you.
X1	is the current array of coordinates of the first node of the element.
X4	is the current array of coordinates of the fourth node of the element.
TOL1	is the tolerance on gap overclosure. Default is 0.

TOL2	is the tolerance on gap force which allows the gap to remain closed even if small negative force.
TOL3	is the tolerance on frictional force. Default is 0.
M(1)	is your element number.
M(2)	is the internal element number.
MSUB	is the subelement number (only for MARC element type 97).
INC	is the current increment number.
NCR	is the dimension of the gap. 2 for 2D problems, 3 for 3D problems.
FN	is the current gap force.
FF	is the frictional force limit, to be specified by you.

In two dimensional problems, DIR, X1, and X4 have two components; otherwise, DIR, X1, and X4 have three components.

■ USELEM

User-Defined Element

Description

This subroutine allows you to calculate your own finite element stiffness or mass matrix. This can also be used as interface with other numerical techniques. In general, in the finite element calculation, several matrices are required; hence, for a particular element, this routine is called a multiple number of times. The calls and your requirements are defined as follows:

IFLAG=1	Return the equivalent nodal loads (F) given distributed surface or body loads. If the ELASTIC, FOLLOW FOR parameters or the AUTO STEP, AUTO TIME, AUTO INCREMENT options are used, these are total loads or else incremental loads. In a heat transfer analysis, this is the total flux vector.
IFLAG=2	Return the element tangent stiffness matrix (K). For an elastic analysis, this is the usual stiffness. For a heat transfer matrix analysis, this is the conductivity matrix. Also calculate the total internal forces (R). This is not necessary in a linear elastic analysis if the LOAD COR parameter has been turned off.
IFLAG=3	Return the mass matrix (M) for a dynamic analysis or specific heat matrix for a heat transfer problem.
IFLAG=4	Calculate the incremental strains (DE), generalized stresses (GSIGS) and the internal force (R). For a linear elastic solution, if only displacements are required, the user does not need to return any values.
IFLAG=5	Output element results if so desired.

To use this option, the USER parameter must be included to define the size of the element stiffness matrix and other critical dimensions and the element type given on the connectivity must be a negative number.

Format

Subroutine USELEM calls for the following headers:

```
      SUBROUTINE USELEM(M,XK,XM,NNODE,NDEG,F,R,
     * JTYPE,DISPT,DISP,NDI,NSHEAR,IPASS,NSTATS,NGENS,
     * NSTRES,COORD,NCRD,IFLAG,IDSS,T,DT,ETOTA,GSIGS,DE,
     * GEOM1,GEOM2,GEOM3,GEOM4,GEOM5,GEOM6, SIGXX, NSTRMX)
      IMPLICIT REAL *8 (A-H, O-Z)
```

```
      DIMENSION XK(IDSS,IDSS),XM(IDSS,IDSS),DISPT(NDEG,1),DISP(NDEG,1)
      DIMENSION T(NSTATS,1),DT(NSTATS,1),COORD(NCRD,1)
      DIMENSION ETOTA(NGENS,1),GSIGS(NGENS,1),DE(NGENS,1)
      DIMENSION F(NDEG,1),R(NDEG,1),SIGXX(NSTRMX,1)
       user coding
      RETURN
      END
```

where:

M	is your element number.
XK	is the stiffness matrix.
XM	is the mass matrix.
NNODE	is the number of nodes per element.
NDEG	is the number of degrees of freedom per node.
F	is the externally applied equivalent nodal loads array.
R	is the internal forces array.
JTYPE	is the element type.
DISPT	is the total nodal displacements array of this element.
DISP	is the incremental nodal displacements of this element.
NDI	is the number of direct components of stress.
NSHEAR	is the number of shear components of stress.
IPASS	Flag to indicate which pass for coupled analysis. =0 during an uncoupled analysis. =1 during a stress analysis pass. =2 during a heat transfer pass.
NSTATS	is the number of state variables.
NGENS	is the number of generalized strains.
NSTRESS	is the number of integration points.
COORD	is the original nodal coordinates array.
NCRD	is the number of coordinates per node.

IFLAG	indicates what is to be returned by you. =1 Called by OPRESS during formation of load vector. You return F. =2 Called by OASEMB during formation of stiffness matrix. You return XK, R. =3 Called by OASMAS during formation of mass matrix. You return XM. =4 Called by OGETST during stress recovery. You return R, GSIGS, DE, ETOTA, SIGXX. =5 Called by SCIMP during output phase. You print results.
IDSS	is the size of element stiffness matrix.
T	is the state variables.
DT	is the increment of state variables.
ETOTA	is the total strain array.
GSIGS	is the generalized stress array.
DE	is the increment of strain array.
GEOM1	is the first geometric parameter.
GEOM2	is the second geometric parameter.
GEOM3	is the third geometric parameter.
GEOM4	is the fourth geometric parameter.
GEOM5	is the fifth geometric parameter.
GEOM6	is the sixth geometric parameter.
SIGXX	is layer stresses for shell elements and is equal to GSIGS for continuum element.
NSTRMX	maximum number of stresses per integration points and is equal to NGENS for continuum element.

Note that the stiffness matrix is normally symmetric. If a nonsymmetric formulation is used, the SOLVER option should be used to indicate this.

■ UNEWTN

Input of Viscosity in Flow Analysis

Description

It is possible to solve Newtonian and non-Newtonian laminar incompressible steady state fluid analyses using the R-P FLOW parameter in MARC. User subroutine UNEWTN is used to define the viscosity at a particular spatial location. An Eulerian approach is then used to solve for the nodal velocities. This routine can also be used to define the nonlinear viscosity in Navier Stokes fluid analysis when the FLUID parameter is used.

Format

Subroutine UNEWTN is written with the following headers:

```
SUBROUTINE UNEWTN (N,NN,V,E,NGENS,DT,DTDL)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION E(NGENS),N(2)
   user coding
RETURN
END
```

where:

N(1)	is your element number.
N(2)	is the internal element number.
NN	is the integration point number.
V	is the viscosity to be defined here.
E	are the components of the strain rate.
NGENS	is the number of components.
DT	is the temperature at the beginning of the increment.
DTDL	is the increment of temperature.

Note: If Herrmann elements are used, the last component of E represents a rate of change of volumetric strain.

■ URPFLO

User Routine for Rigid-Plastic Flow

Description

This subroutine allows you to define the current yield stress as a function of the equivalent strain rate, equivalent strain, temperature, and user-defined state variables. This subroutine is used in conjunction with the transient R-P FLOW parameter.

Format

Subroutine URPFLO is written with the following headers:

```
      SUBROUTINE URPFLO(MDUM,NN,LAYERS,MATS,INC,NDI,NGENS,NCRD,NSTAT,
     +CPTIM,TIMINC,EBAR,ERATE,DT,DTDL,STATS,DSTATS,COORD,YD)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION MDUM(2), STATS(NSTAT), DSTATS(NSTAT), COORD(NCRD)

         user coding

      RETURN
      END
```

where:

passed into routine:

MDUM(1)	element number.
MDUM(2)	internal element/elsto number.
NN	integration point number.
LAYERS	layer number.
MATS	material id.
INC	increment number.
NDI	number of direct components.
NGENS	total number of components.
NCRD	number of coordinates.
NSTAT	number of state variables excluding temperature.
CPTIM	time at beginning of increment.
TIMINC	incremental time.
DT	temperature at beginning of increment.
DTDL	incremental temperature.

EBAR	total equivalent strain at beginning of increment.
STATS	values of state variables excluding temperature at beginning of increment.
ERATE	equivalent strain rate.
COORD	integration point coordinates.

to be passed back:

YD	equivalent stress; if not calculated here, program will find the value of yd from the input data.
DSTATS	incremental state variables (excluding temperature).

References

1. Simo, J. C. and Taylor, R. L., "Quasi incompressible finite elasticity in principal stretches. Continuum basis and numerical algorithms", *Comp. Meth. App. Mech. Engrg.*, 85, pp. 273-310, 1991.
2. Simo, J. C., "Algorithms for static and dynamic multiplicative plasticity that preserve the classical return mapping schemes of the infinitesimal theory", *Comp. Meth. App. Mech. Engrg.*, 99, pp. 61-112, 1992.

4 List of Viscoplasticity and Generalized Plasticity Subroutines

Subroutines	Page
ASSOC	4-10
CRPLAW	4-4
NASSOC	4-6
SINCER	4-11
UVSCPL	4-2
YIEL	4-9
ZERO	4-8

4 Viscoplasticity and Generalized Plasticity Subroutines

The subroutines in this chapter are used to describe viscoplastic materials or a user-defined general plasticity model. There are two numeric procedures for viscoplasticity, implicit and explicit. This is activated on the CREEP parameter. The implicit method is preferred. The generalized plasticity allows you to develop a yield surface, equivalent stress, and flow rate that is different from one provided by MARC. Table 4-1 summarizes these routines and indicates what parameters or model definition options are required to invoke the user subroutine.

Table 4-1 Viscoplasticity and Generalized Plasticity User Subroutine Requirements

User Subroutine	Required Parameters or Model Definition Options	Purpose
ASSOC	ISOTROPIC – GEN-PLAST	Definition of the direction of incremental plastic strain in generalized plasticity model.
CRPLAW	CREEP	Definition of inelastic strain rate for explicit viscoplasticity model.
NASSOC	CREEP	Definition of direction of incremental viscoplastic strain for explicit viscoplasticity model.
SINCER	ISOTROPIC – GEN-PLAST	Definition of fraction of increment which is elastic for generalized plasticity model.
UVSCPL	CREEP ISOTROPIC – VISCO-PLASTIC	Definition of inelastic strain rate for implicit viscoplastic model.
YIEL	ISOTROPIC, ORTHOTROPIC or ANISOTROPIC	Definition of yield stress.
ZERO	ISOTROPIC, ORTHOTROPIC or ANISOTROPIC	Definition of equivalent stress.

■ UVSCPL

Definition of the Inelastic Strain Rate

Description

This user subroutine is used for computing the inelastic strain increment for an elastic-viscoplastic material. This routine allows very general material laws to be entered. You must define the inelastic strain and the stress increment.

Format

Subroutine **UVSCPL** is written with the following headers:

```
      SUBROUTINE UVSCPL(YOUNG,POISS,SHEAR,B,USTRRT,ETOT,E,THMSTI,EELAS,
     1 S,SINC,GF,EPL,AVGINE,EQCRP,EQCPNC,YD,YD1,VSCPAR,DT,DTDL,CPTIM,
     2 TIMINC,XINTP,NGENS,M,NN,KC,MAT,NDI,NSHEAR,NCRD,IANISO,NSTATS,
     3 INC,NCYCLE,LOVL,NVSPLM)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION POISS(3,2),YOUNG(3,2),B(NGENS,NGENS),USTRRT(NGENS),
     1 ETOT(NGENS),E(NGENS),THMSTI(NGENS),EELAS(NGENS),S(NGENS),
     2 SINC(NGENS),GF(NGENS),EPL(NGENS),AVGINE(NGENS),DT(NSTATS),
     3 DTDL(NSTATS),XINTP(NCRD),SHEAR(3,2),VSCPAR(NVSPLM)
       user coding
      RETURN
      END
```

where:

YOUNG	is the Young's modulus.
POISS	is the Poisson's modulus.
SHEAR	is the shear modulus.
B	is the tangent elastic matrix.
USTRRT	is the inelastic strain rate.
ETOT	is the accumulated total strain at beginning of increment.
E	is the current strain increment.
THMSTI	is the thermal strain increment.
EELAS	is the accumulated elastic strain at beginning of increment.
S	is the accumulated stress at beginning of increment.
SINC	is the stress increment.

GF	is the change in stress due to change in elastic material properties associated with DT.
EPL	is the accumulated inelastic strain at beginning of increment.
AVGINE	is the inelastic strain increment.
EQRCP	is the equivalent inelastic strain at beginning of increment.
EQCPNC	is the increment equivalent inelastic strain.
YD	is the flow stress at temperature t.
YDL	is the flow stress at temperature t + dt.
VSCPAR	is the viscoplastic data read off isotropic or orthotropic op.
DT	is the state variables at beginning of increment.
DTDL	is the incremental state variables.
CPTIM	is the elapsed time at beginning of increment.
TIMINC	is the time increment.
XINTP	is the integration point coordinates.
NGENS	is the number of strain components.
M	is the element number.
NN	is the integration point number.
KC	is the layer number.
MAT	is the material id.
NDI	is the number of direct components.
NSHEAR	is the number of shear components.
NCRD	is the number of coordinate directions.
IANISO	is the flag to indicate nonisotropic elasticity.
NSTATS	is the number of state variables.
INC	is the increment number.
NCYCLE	is the cycle number.
LOVL	= 4 during stiffness formation. = 6 during residual calculation.
NVSPLM	is the number of viscoplastic data read from input.

Note: To ensure convergence, it should be noted that the returned values of these quantities must be mutually compatible; that is, they simultaneously must satisfy within tolerance:

1. `SINC = B*(E - AVGINE - THMSTI) + GF`
2. the creep law employed.

The tolerance should be at least one order of magnitude smaller than the global Newton-Raphson tolerance. The values of USTRRT, AVGINE, and SINC are expected to be returned from the routine for both LOVL=4 and LOVL=6.

■ CRPLAW

Input of Special Viscoplastic Strain Rate Law

Description

Subroutine CRPLAW can also be used for calculating the viscoplastic behavior. All the creep quantities are treated as viscoplastic strain quantities when the appropriate flag is set on the CREEP parameter.

The basic information on the use of this subroutine can be found in Chapter 3, on page 28 of this manual. Additionally, you can use common block VISCPL.

The variables in common block VISCPL are:

YD	is the equivalent stress at first yield.
YD1	is the equivalent yield stress including current work hardening and temperature effects.
YD2	is the equivalent stress for ORNL tenth cycle yield.
YD21	is the equivalent stress including current work hardening and temperature effects for ORNL tenth cycle yield.
YDZER	is the equivalent yield stress including Mohr-Coulomb terms (defaults to YD1).

Example

The following is a simple viscoplastic strain rate law that depends on the differences between the current stress state and the static yield stress, raised to the nth power. (Note that T(1), the current equivalent stress also includes Mohr-Coulomb terms when the option is flagged.)

$$\dot{\bar{\varepsilon}} = c(\bar{\sigma} - \bar{\sigma}_y)^n$$

where:

$\bar{\sigma}$	is the current total equivalent stress.
$\bar{\sigma}_y$	is the current equivalent yield stress including workhardening, temperature effects and Mohr-Coulomb terms.
n	is the index of the power law.
c	is the constant that depends on the index *n*. Here the strain rate equation is made dimensionless in stress by setting $c = 0.01/\bar{\sigma}^n{}_{yo}$ where σ_{yo} is the equivalent stress at first yield.

This is programmed as follows; for n=2

```
 SUBROUTINE CRPLAW(EQCP,EQCPNC,STR,CRPE,T,DT,TIMINC,CPTIM,M,NN,KC,
+MAT,NDI,NSHEAR)
 IMPLICIT REAL *8 (A-H, O-Z)
 DIMENSION T(1),DT(1),STR(1),CRPE(1)
 C = 0.01/(YD*YD)
 S = T(1) - YDZER
 EQCPNC = 0.0
 IF(S.LT.0.0)RETURN
 S=S*S
 EQCPNC=C*S*TIMINC
 RETURN
 END
```

■ NASSOC

Input of a Nonassociated Flow Law

Description

Subroutine NASSOC allows you to calculate a strain increment with a flow rule differing from the normality rule of plasticity, which is the default used by the program.

Format

Subroutine **NASSOC** is written with the following headers:

```
      SUBROUTINE NASSOC(EQCPNC,STOT,SINC,E,
     1 AMOHR,NGENS,NDI,T,TZERO)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION STOT(NGENS),SINC(NGENS),E(NGENS),T(2)
        user coding
      RETURN
      END
```

where:

EQCPNC	is the increment of the equivalent viscoplastic strain.
STOT	is the current stress array.
SINC	is the dimensionless flow directions $\frac{\partial\bar{\sigma}}{\partial\sigma}$. The current values in this subroutine are associated with the yield criterion used. You are free to vary the flow rule in NASSOC by changing SINC.
E(I)	is the Ith viscoplastic strain increment. It is later set equal to EQCPNC*SINC(I) in the program; thus, it is not set in this subroutine.
AMOHR	is the Mohr-Coulomb parameter entered in the ISOTROPIC option (third field).
NGENS	is the number of stresses or strains.
NDI	is the number of direct stresses.
T(1)	is the current equivalent stress.
T(2)	is the current mean hydrostatic stress.
TZERO	is the equivalent stress including Mohr-Coulomb terms, temperature and work hardening effects.

It is often useful to have the information regarding the yield surface. This can be obtained from common block VISCPL

```
yd, yd1, yd2, yd21, ydzer
```

where:

YD	is the equivalent stress at first yield.
YD1	is the equivalent yield stress including current work hardening and temperature effects.
YD2	is the equivalent stress for ORNL tenth cycle yield.
YD21	is the equivalent stress including current work hardening and temperature effects for ORNL tenth cycle yield.
YDZER	is the equivalent yield stress including Mohr-Coulomb terms (defaults to YD1).

Example

The following example calculates a nonassociated flow rule for a Mohr-Coulomb problem. The default flow rule is the one associated with the von Mises yield criterion.

```
      SUBROUTINE NASSOC(EQCPNC,STOT,SINC,E,
     + AMOHR,NGENS,NDI,T,TZERO)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION STOT(1),SINC(1),E(1),T(1)
      NSHEAR=NGENS-NDI
      DO 1 I=1,NDI
    1 SINC(I)=0.5*3.0*(STOT(1)-T(2))
      TR=1./TZERO
      DO 2 I=1,NDI
    2 SINC(I)=SINC(I)*TR
      RETURN
      END
```

■ ZERO

Calculation of Equivalent Stress

Description

Subroutine ZERO is used to calculate the equivalent yield stress based on the current total stresses. The subroutine ZERO in the program applies the von Mises yield criterion as a default. You can substitute another yield criterion by writing a new subroutine ZERO. Mohr Coulomb models specified in the ISOTROPIC option should not be used when user subroutine ZERO is used because of the danger of taking into account the effects of hydrostatic pressure twice.

Format

Subroutine ZERO is written with the following headers:

```
REAL*8 FUNCTION ZERO(NDI,NSHEAR,T,IORT,IANISO,YRDIR,YRSHR,AMM,AO)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION T(1),YRDIR(3),YRSHR(3),AMM(3)
    user coding
RETURN
END
```

where:

NDI	is the number of direct components of stress.
NSHEAR	is the number of shear components of stress.
T(I)	is the Ith component of stress.
IORT	is the flag indicating if curvilinear coordinates are used. This is 1 for element types 4, 8, and 24.
IANISO	is the flag indicating if anisotropy is used.
YRDIR	are the components for Hill's anisotropic plasticity.
YRSHR	are the shear components for Hill's anisotropic plasticity.
AMM	is the metric if curvilinear coordinates are used.
AO	is the metric scale factor if curvilinear coordinates are used.

YIEL

Calculation of Current Yield

Description

Subroutine YIEL is used to define the yield stress based on the current work hardening and other state variables.

Format

Subroutine YIEL is written with the following headers:

```
REAL*8 FUNCTION YIEL(M,NN,KC,YIELD,IFIRST,DT,EPLAS,ERATE,MATS,JPROPS)
IMPLICIT REAL *8 (A-H, O-Z)
   user coding
RETURN
END
```

where:

M	is the element numbers.
NN	is the integration point number.
KC	is the layer number.
YIELD	is the yield stress entered as data in the ISOTROPIC block.
IFIRST	=1 Calculate yield stress. =2 Calculate 10th cycle yield stress (ORNL only). =3 Calculate 100th cycle yield stress (ORNL only).
DT	is the current temperature
EPLAS	is the total equivalent plastic strain. Note that this is implied by the yield criterion used in subroutine ZERO (or the Mohr-Coulomb yield criterion, if that is used).
ERATE	is the equivalent plastic strain rate. Not available for viscoplasticity.
MATS	is the material id.
JPROPS	is the table id associated with the yield.

■ ASSOC

Input of Associated Flow Law

Description

When used in conjunction with the generalized plasticity option (defined in the ISOTROPIC option), subroutine ASSOC can be used to define the flow direction for plasticity. The default is the associated flow law with the von Mises (J2) yield surface.

Format

Subroutine ASSOC is written with the following headers:

```
SUBROUTINE ASSOC(STOT,SINC,SC,T,NGENS,NDI,NSHEAR,N,NN,KC)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION STOT(NGENS),SINC(NGENS)
   user coding
RETURN
END
```

where:

STOT	is the current stress array.
SINC	is the dimensionless flow direction $\frac{\partial\bar{\sigma}}{\partial\sigma}$ to be defined by you.
SC	is the hydrostatic stress.
T	is the equivalent stress.
NGENS	is the number of stress components.
NDI	is the number of shear stress components.
N	is the element number.
NN	is the integration point number.
KC	is the layer number for shells or beams.

■ SINCER

User Subroutine for Improving Accuracy

Description

Subroutine SINCER can be used to define how much an "elastic" stress increment exceeds the yield stress. This allows the program to accurately take large increments such that the material goes from elastic to elastic-plastic. You return the value of FPLAS, which is the fraction of the stress increment beyond the yield surface. This routine should only be used if a yield surface other than the von Mises (J2) is used in conjunction with the generalized plasticity option (defined in the ISOTROPIC option).

Format

Subroutine SINCER is written with the following headers:

```
SUBROUTINE SINCER(FPLAS,SINC,STOT,NGENS)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION SINC(NGENS), STOT(NGENS)
    user coding
RETURN
END
```

where:

FPLAS	is the fraction of stress increment beyond the yield stress to be defined you.
SINC	is the estimated elastic increment of stress.
STOT	is the stress at the beginning of the increment.
NGENS	is the number of stress components.

5 List of Viscoelasticity Subroutines

Subroutines	Page
CRPVIS	5-2
HOOKVI	5-7
TRSFAC	5-5

5 Viscoelasticity Subroutines

This chapter describes routines used for viscoelastic analysis. There are two procedures available. The explicit procedure uses the CRPVIS subroutine to describe a generalized Kelvin model. The implicit procedure uses a hereditary integral approach and is the preferred choice. Table 5-1 summarizes these routines and indicates what parameters or model definition options are required to invoke the user subroutine.

Table 5-1 Viscoplasticity Subroutines Requirements

User Subroutine	Required Parameters or Model Definition Options	Purpose
CRPVIS	VISCO ELAS	Definition of generalized Kelvin model using explicit procedure.
HOOKVI	VISCELORTH	Definition of anisotropic viscoelastic material law for a particular relaxation time.
TRSFAC	VISCELPROP or VISCELORTH or VISCELMOON or VISCELOGDEN and SHIFT FUNCTION	Definition of shift function for thermo-rheologically simple material.

■ CRPVIS

Viscoelasticity – Generalized Kelvin Material Behavior

Description

In addition to the nonlinear Maxwell type model allowed in the CREEP option, a general Kelvin model can be included by requesting it on the CREEP parameter. In this case, the program assumes an additional creep strain ε_{ij}^{K} , governed by $\frac{d}{dt}\varepsilon_{ij}^{K} = A_{ijkl}S_{kl} - B_{ijkl}\varepsilon_{kl}^{K}$

where:

[A] and [B]	are defined by the user in the subroutine described below,
s_{ij}	are the deviatoric stress components $s_{ij} = \sigma_{ij} - \delta_{ij}\frac{\sigma_{kk}}{3}$

and the total strain is: $\varepsilon_{ij} = \varepsilon_{ij}^{e} + \varepsilon_{ij}^{p} + \varepsilon_{ij}^{c} + \varepsilon_{ij}^{K} + \varepsilon_{ij}^{th}$

where:

ε_{ij}^{th}	are the thermal strain components.
ε_{ij}^{e}	are the elastic strain components (instantaneous response).
ε_{ij}^{p}	are the plastic strain components.
ε_{ij}^{c}	are the creep strains defined via CRPLAW and VSWELL subroutines and using the CREEP option.
ε_{ij}^{K}	are the Kelvin model strain components as defined above.

Format

Subroutine CRPVIS is written with the following headers:

```
      SUBROUTINE CRPVIS(CRPR,TSIG,SINC,AE,BE,NGENS,
     1 DT,DTDL,N,NN,KC,MATS,NDI,NSHEAR,TIME,TIMINC)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION CRPR(1),TSIG(1),SINC(1),AE(NGENS,NGENS),
     1 BE(NGENS,NGENS),DT(1),DTDL(1),N(2)
       user coding
      RETURN
      END
```

where:

CRPR	are the Kelvin creep strain components.
TSIG(1)	is the second invariant of the deviatoric stress = $\left(\frac{3}{2}s_{ij}s_{ij}\right)^{1/2}$.
TSIG(2)	is the hydrostatic stress = $\frac{1}{3}\sigma_{kk}$.
SINC	are the deviatoric stress components (s_{ij}).
AE	is the matrix A_{ijkl} above, to be defined here by you.
BE	is the matrix B_{ijkl} above, to be defined here by you.
NGENS	is the number of stress (strain) components.
DT	are the total state variables at this point (temperature first).
DTDL	are the increments of state variables at this point during this step of the solution.
N(1)	is your element number.
N(2)	is the internal element number.
NN	is the integration point number.
KC	is the layer number.
MATS	is the material identification number.
NDI	is the number of direct components.
NSHEAR	is the number of shear components.
TIME	is the total time.
TIMINC	is the time increment.

Only AE and BE are to be defined by you – the other variables are provided to assist in calculations, for example when a nonlinear Kelvin model is used.

This routine is called at each integration point of each element when necessary, when the VISCO ELAS parameter is present. Note that the use of the VISCO ELAS parameter also requires the use of the CREEP option in the model definition data as well. The CREEP option is required to set the tolerance control for the maximum strain in any increment. In viscoelastic two-dimensional analysis, the stress does not change appreciably so that all time steps are controlled by the maximum increment in strain. The recommended and default value of this strain increment is 0.005 of the total maximum strain. Note that this value is ten times smaller than the default

value for normal creep problems. Because of the use of the CREEP option, Maxwell models can be included in series with the Kelvin model. The ordering of stress and strain components is given in *Volume B: Element Library* for each element type.

When used with doubly curved shell elements (shell elements 4, 8, and 24), the above relation is written in a mixed formulation:

$$\frac{d}{dt}(\varepsilon^{\alpha}{}_{\beta}{}^{K}) = A^{\alpha}{}_{\beta}{}^{\gamma}{}_{\delta}\, S_{\gamma}{}^{\delta} - B^{\alpha}{}_{\beta}{}^{\gamma}{}_{\delta}\, \varepsilon^{d}_{\gamma}{}^{K} \quad \alpha,\beta \text{ etc.} = 1,2$$

with two shear components stored, $\varepsilon^{1}{}_{2}$, then $\varepsilon^{2}{}_{1}$.

TRSFAC

Define a Shift Function for Thermo-Rheologically Simple (T.R.S.) Material Behavior

Description

This subroutine allows you to define the shift function for the relaxation function.

A description of T.R.S. material behavior is given in *Volume A: User Information*. You are reminded that this option is only available in conjunction with the hereditary integral form of viscoelastic constitutive representation. The use of this subroutine to define a shift function for a particular viscoelastic material group is indicated by inserting a negative value in the first field of block 2 in the SHIFT FUNCTION model definition block.

The program will proceed to compute the increment of pseudo- or reduced time $\Delta\varepsilon\ (x,t)$ according to the relationship:

$$\Delta\varepsilon(x, t) = \int_{t}^{(t+\Delta t)} 10^{B[T(x,\, t')]} dt'$$

where the shift factor, `B`, is a function of the spatially and time dependent temperature `T(x,t)`. A five-point Simpson's rule is used to numerically integrate this expression.

In this subroutine, you are expected to define the shift function, ϕ, which is the logarithm of the shift factor: that is,

$$[\phi(x, t)] = -Log_{10}\{B[T(x, t)]\}$$

The subroutine is called five times at each point. These points can be the centroids of the elements or each integrating point if the ALL POINTS parameter has been invoked.

Format

User subroutine TRSFAC is written with the following headers:

```
      SUBROUTINE TRSFAC(SHFTLG,MATV,NSHFT,N,NN,KC,DT,DTDL,TGLASS,
     *CPTIM,HXITOT,TIMINC,TINT)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION DT(1),DTDL(1),N(2)
         User Coding
      RETURN
      END
```

where:

SHFTLG	is the logarithm of the shift function, ϕ, which must be defined by you.

The following parameters are passed into this subroutine and **must not** be redefined:

MATV	is the viscoelastic material group identifier or number associated with the point, x, currently being considered.
NSHFT	is the negative number associated with the particular user defined shift function for the viscoelastic material group, MATV. This number was specified in the first field of the second data line in the SHIFT FUNCTION model definition option.
N(1)	is your element number.
N(2)	is the internal element number.
NN	is the current integrating point number (or centroidal point if the ALL POINTS parameter is not used).
KC	is the current layer or beam section number if a shell or beam element is being considered.
DT	is the total temperature at this point corresponding to the beginning of the current increment.
DTDL	is the current incremental change in temperature for this point.
TGLASS	is the reference or glassy transition temperature used in defining the shift function.
CPTIM	is the total creep or viscoelastic time up to the beginning of this increment.
HXITOT	is the total pseudo- or reduced-time at this point, corresponding to the beginning of the increment.
TIMINC	is the increment of real time.
TINT	is a linearly interpolated value of the total temperature at one of the five integrating stations between the beginning and end of the increment. This is the variable which should be used in computing the value of the shift function.

■ HOOKVI

User Defined Anisotropic Viscoelasticity

Description

You can specify the time dependent properties of an orthotropic material through the VISCELORTH model definition option. You can then modify this data by use of subroutine HOOKVI which is automatically called for every material defined in that option.

Format

Subroutine HOOKVI is called with the following header codes:

```
SUBROUTINE HOOKVI (M,NN,KC,ITERM,B,DT,DTDL,E,PR,G)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION B(6,6),DT(1),DTDL(1),E(3),PR(3),G(3),M(2)
   user coding
RETURN
END
```

where:

M(1)	is your element number.
M(2)	is the internal element number.
NN	is the integration point number.
KC	is the layer number.
ITERM	is the viscoelastic series number.
B	is the user-defined 6 x 6 matrix of viscoelastic time dependent constants for this element and series number.
DT	is the current temperature.
DTDL	is the current increment in temperature.
E	is vector of time dependent Young's moduli input in the VISCELORTH option.
PR	is the vector of time dependent Poisson's ratios input in the VISCELORTH option.
G	is the vector of time dependent shear moduli given in the VISCELORTH option.

5 *Viscoelasticity Subroutines*

6 List of Geometry Modifications Subroutines

Subroutines	Page
MAP2D	6-5
REBAR	6-8
SSTRAN	6-15
UACTIVE	6-7
UACTUAT	6-17
UADAP	6-12
UCOORD	6-11
UFCONN	6-4
UFRORD	6-9
UFXORD	6-3
UPNOD	6-6
URCONN	6-10
USHELL	6-14
UTHICK	6-16
UTRANS	6-13

6 Geometry Modifications Subroutines

The user subroutines described in this section are provided to allow you to define the initial geometry of the finite element mesh, or to change the mesh due to rezoning or rigid plastic analyses. Often these user subroutines are used to customize already existing meshes. User subroutine UTRANS is a powerful way to provide transformations to the degrees of freedom of a node. Table 6-1 summarizes these routines and indicates what parameters or model definition options are required to invoke the user subroutine.

Table 6-1 Geometry Modifications Subroutines Requirements

User Subroutine	Required Parameters or Model Definition Options	Purpose
MAP2D	MESH2D MAPPER	Define the coordinates of key boundary nodes for mesh generation.
REBAR	ELEMENTS (rebar element types)	Define the orientation and effective thickness of the elements.
SSTRAN	SUBSTRUCTURE	Define the transformation matrix applied to a superelement.
UACTIVE		Activate or deactivate elements.
UACTUAT	GEOMETRY	Define the length of the actuator element.
UADAP	ADAPTIVE (parameter) ADAPTIVE (model definition option)	Define a user-defined error criteria for adaptive meshing.
UCOORD	ADAPTIVE (parameter) ADAPTIVE (model definition option)	Describe of the location of newly created nodes.
UFCONN	UFCONN	Modify the connectivity of an element.
UFRORD	REZONING REZONE UFRORD	Modify the coordinates of a node during rezoning.
UFXORD	UFXORD	Modify the initial nodal coordinates.
UPNOD	R-P FLOW	Update the nodal coordinates in a rigid plastic analysis using the Eularian procedure.

Table 6-1 Geometry Modifications Subroutines Requirements (Continued)

User Subroutine	Required Parameters or Model Definition Options	Purpose
URCONN	UFCON	Modify the connectivity of an element during rezoning.
USHELL		Define the integration point thickness for shell elements.
UTHICK	NODAL THICKNESS	Define the initial thickness at the nodes for shell elements.
UTRANS	UTRANFORM	Define a transformation to be applied to the degrees of freedom at a node.

■ UFXORD

Coordinate Generation or Modification

Description

The user subroutine UFXORD can be used to modify (or expand) coordinates input through use of the COORDINATES option, or as an internal coordinate generator. You must input the model definition option UFXORD, followed by a block giving the nodes for which UFXORD is used. The program calls UFXORD for each node in the list, so that the coordinates of that node can be modified or generated. The UFXORD option can be repeated as many times as necessary.

Format

Subroutine UFXORD is written with the following headers:

```
SUBROUTINE UFXORD (XORD, NCRD, N)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION XORD (NCRD)
    user coding
RETURN
END
```

where:

XORD	is the array of coordinates in the Nth node and will be passed in containing coordinates previously generated at the Nth node by COORDINATES, FXORD or UFXORD blocks.
NCRD	is the number of coordinates per node.
N	is the node number.

The user subroutine is most commonly used with shell or beam elements (for example, Elements 4, 8, 13, 15) where the full coordinate set is usually generated on the basis of reduced set of coordinates. See, for example, the description of the FXORD option in *Volume A: User Information.* You can also use this routine to generate special coordinate systems (for example, cylindrical or spherical) or to convert from special coordinate systems to a rectangular system.

■ UFCONN

Connectivity Generation or Modification

Description

The user subroutine UFCONN can be used to modify (or expand) input given through use of the CONNECTIVITY option, or as an internal connectivity generator. You must input the model definition option UFCONN, followed by a block giving the elements for which UFCONN is used. The program calls UFCONN for each element in the series, so that the connectivity of that element can be modified or generated. The UFCONN option can be repeated as many times as necessary.

Format

Subroutine UFCONN is written with the following headers:

```
SUBROUTINE UFCONN(J,ITYPE,LM,NNODMX)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION LM(1)
    user coding
RETURN
END
```

where:

J	is the element number.
ITYPE	is the element type.
LM	is the array of nodes making up the element.
NNODMX	is the maximum number of nodes in an element.

LM is passed in containing the connectivity of the Jth element already generated by previous CONNECTIVITY, UFCONN, or other generators. Similarly, ITYPE is the element type if previously defined. You can modify or define ITYPE or LM in this routine. Note there is no checking to determine if ITYPE has been defined on SIZING or ELEMENTS parameter, or if node numbers are in the range $1 \le N \le NUMNP$.

■ MAP2D

Boundary Node Coordinates Modification in Mesh2D

Description

The user subroutine MAP2D can be used to modify coordinates input for the boundary nodes in MESH2D by the BOUNDARY option. You must input the MAPPER option as part of the two-dimensional mesh generation. The program calls MAP2D once, so that the coordinates of all the boundary nodes can be modified or generated.

Format

Subroutine MAP2D is written with the following headers:

```
SUBROUTINE MAP2D(NNO,X,Y)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION X(NNO),Y(NNO)
    user coding
RETURN
END
```

where:

NNO	is the number of boundary nodes.
X and Y	are the user-defined coordinates of the boundary nodes.

■ UPNOD

Update Nodal Positions in Flow Solutions

Description

This subroutine is used in conjunction with Eulerian flow solutions (for example, R-P FLOW parameter) to update the mesh after a velocity field has been found. You can access the velocity field and hence re-define the nodal coordinates. The subroutine is called in a loop over all the nodes in the mesh at the end of convergent step of the flow calculation. This routine should not be used in conjunction with the CONTACT option.

Format

Subroutine UPNOD is written with the following headers:

```
SUBROUTINE UPNOD (XORD,VEL,NCRD,NDEG,NODE)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSIONAL XORD (NCRD), VEL(NDEG)
    user coding
RETURN
END
```

where:

XORD	is the array of coordinates at this node, to be redefined in this routine as required.
VEL	is the array of current velocities at this node.
NCRD	is the size of the XORD array (number of coordinates per node).
NDEG	is the size of the VEL array (number of velocity components per node).
NODE	is the node number.

Example

A typical subroutine UPNOD for use with higher order elements would be

```
      SUBROUTINE UPNOD(XORD,VEL,NCRD,NODE)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION XORD(NCRD),VEL(NDEG)
      TIME=
      DO 5 I=1,NCRD
      XORD(I)=XORD(I)+VEL(I)*TIME
5     CONTINUE
      RETURN
      END
```

UACTIVE

Activate or Deactivate Elements

Description

The user subroutine UACTIVE can be used to either activate or deactivate elements in the model. The routine is called at the beginning of the analysis and at the end of each increment. A deactivated element does not contribute to the load, mass, stiffness, or internal force calculation. If an element is activated after previously being deactivated, you can specify if the material is to come back in its previous state or in a modified state.

Format

Subroutine UACTIVE is written with the following headers:

```
SUBROUTINE UACTIVE(M,N,MODE,IRSTSTR,IRSTSTN,INC,TIME,TIMINC)
IMPLICIT REAL*8 (A-H,O-Z)
    user coding
RETURN
END
```

where:

M	is the element number.
NN	is the internal element number.
MODE	-1 deactivate element. 2 leave in current status. 1 activate element.
IRSTSTR	set to 1 to reset stresses to zero.
IRSTSTN	set to 1 to reset strains to zero.
INC	is the increment number.
TIME	is the time at the beginning of the increment.
TIMINC	is the incremental time.

■ REBAR

Input of Rebar Positions, Areas and Orientations

Description

This subroutine is used in conjunction with the single strain rebar elements (23, 46, 47, 48, 142-148). See the description of these elements for details of the use of this subroutine.

Format

Subroutine REBAR is written with the following headers:

```
SUBROUTINE REBAR (N,NN,T,PR,TR,A)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION A(3),NN(3)
    user coding
RETURN
END
```

where:

N	is the element number.
NN(1)	is the integration point number.
NN(2)	is the layer number.
NN(3)	is the integration point number in this layer.
T	is the nominal size in thickness direction.
PR	is the relative position of rebar layer with respect to T.
	The program uses the ratio PR/T to position the rebar layer in the thickness direction.
TR	is the equivalent thickness of rebar.
A	is the direction cosines of the rebar.
T,PR,TR,A	are to be defined by you.

Note that three entries are to be defined in A in all cases.

■ UFRORD

Rezoning Coordinate Generation or Modification

Description

The user subroutine UFRORD can be used to modify (or expand) coordinate change input in a rezoning analysis. You must input the rezoning option UFRORD, followed by a block giving a list of nodes for which UFRORD is used. The program calls UFRORD for each node in the list, so that the coordinates for that node can be modified or generated. The UFRORD option can be repeated as many times as necessary.

Format

Subroutine UFRORD is written with the following headers:

```
SUBROUTINE UFRORD(XORD,NCRD,DISPT,NDEG,N)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION XORD(NCRD),DISPT(NDEG)
   user coding
RETURN
END
```

where:

XORD	are the coordinates of node N which should be generated or modified in this subroutine.
NCRD	is the number of coordinates per node.
DISPT	are the total displacements of node N.
NDEG	is the number of degrees of freedom per node.
N	is the node number.

■ URCONN

Rezoning Connectivity Generation or Modification

Description

The user subroutine URCONN can be used to modify (or expand) input given through use of the CONNECTIVITY CHANGE option, or as an internal connectivity generator. You must input the model definition option URCONN, followed by a block giving the elements for which URCONN is used. The program calls URCONN for each element in the series, so that the connectivity of that element can be modified or generated.

Format

Subroutine URCONN is written with the following headers:

```
SUBROUTINE URCONN(J,ITYPE,LM,NNODMX)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION LM(1)
    user coding
RETURN
END
```

where:

J	is the element number.
ITYPE	is the element type.
LM	is the array of nodes making up the element.
NNODMX	is the maximum number of nodes in an element.

LM is passed in containing the connectivity of the Jth element already generated by previous CONNECTIVITY, UFCONN, CONNECTIVITY CHANGE, URCONN, or other generators. Similarly, ITYPE is the element type if previously defined. You can modify or define ITYPE or LM in this routine. Note there is no checking to determine if ITYPE has been defined on SIZING or ELEMENTS parameter, or if node numbers are in the range $1 \leq N \leq NUMNP$.

■ UCOORD

Relocate Nodes Created During Adaptive Meshing

Description

The user subroutine UCOORD can be used to define the location of a new node created due to adaptive meshing. The default if this routine is not used is to put the newly created node geometrically half way between the old nodes. This routine is called for each new node created.

Format

Subroutine UCOORD is written with the following headers:

```
SUBROUTINE UCOORD(XORD, NCRD, INOD, LM, NNODE)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION XORD(NCRD,1),LM(1)
   user coding
RETURN
END
```

where:

`XORD(J,I)`	current coordinate `j` of node i.
`NCRD`	number of coordinates per node.
`INOD`	node number of new node.
`LM(I)`	nodes on which `INOD` is depending.
`NNODE`	number of nodes on which `INOD` is depending.
`NNODE`	=2 middle of edge between `LM(1)` and `LM(2)`. =3 center of triangle `LM(1)`, `LM(2)`, `LM(3)`. =4 center of plane `LM(1)`, `LM(2)`, `LM(3)`, `LM(4)`. =4 center of `tet4` `LM(1)`, `LM(2)`, `LM(3)`, `LM(4)`. =8 center of brick `LM(1)`, `LM(2)`, `LM(3)`, `LM(4)`, `LM(5)`, `LM(6)`, `LM(7)`, `LM(8)`.

Update `XORD(J,INOD)` if desired.

■ UADAP

User Defined Error Criteria

Description

The user subroutine UADAP can be used to define an error criteria for adaptive meshing. The value of USERCR must be returned. It is a measure of the quality of this element. If the value of USERCR is greater than `f1 * user_max` or greater than `f2`, the element refines.

Note that the `f1` and `f2` must be specified on the ADAPTIVE model definition option. `User_max` is the largest value of USERCR over all of the elements.

Format

Subroutine UADAP is written with the following headers:

```
SUBROUTINE UADAP(MM,XORD,DSXT,NCRDMX,NDEGMX,LM,NNODE,USERCR)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION XORD(NCRDMX,*),DSXT(NDEGMX,*),LM(*)
   user coding
RETURN
END
```

where:

`MM`	user element number.
`XORD`	original coordinates.
`DSXT`	total displacements.
`NCRDMX`	maximum number of coordinates per node.
`NDEGMX`	maximum number of degrees of freedom per node.
`LM`	node numbers of this element.
`NNODE`	number of nodes per element.
`USERCR`	user error criteria is to be defined here.

UTRANS

Implement Local Coordinate System

Description

This subroutine allows you to specify a local coordinate system for user specified nodes. The node numbers are given in the UTRANFORM model definition option. This routine is called a multiple number of times for each increment of analysis. The local coordinate system can be modified (updated) at each increment to facilitate the input of complex boundary conditions. Incremental nodal displacements and reaction forces are output in both the local and global coordinate system. All total nodal quantities are output in the global system.

Format

Subroutine UTRANS is written with the following headers:

```
SUBROUTINE UTRANS (DICOS, NDEG, XORD, NCRD, I, N)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION DICOS (NDEG, NDEG), XORD(NCRD)
    user coding
RETURN
END
```

where:

DICOS	is the user defined rotation matrix from the local to global coordinate system. Note that this matrix must be proper orthogonal.
NDEG	is the number of degrees of freedom.
XORD	are the coordinates of the node updated if either UPDATE or FOLLOW FOR parameter is used.
NCRD	is the number of coordinates per node.
I	is your node number.
N	is the transformation number.

■ USHELL

Modify Thickness of Shell Elements

Description

This subroutine allows you to specify the thickness of shell elements for each integration point. This routine is called twice for each increment of analysis. It is not advisable to change the thickness during an analysis.

Note: This routine should not be used if the thickness is to be considered a design variable. Use the GEOMETRY option instead.

Format

Subroutine USHELL is written with the following headers:

```
SUBROUTINE USHELL (THICK,XINTP,NCRD,M,NN)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION XINTP(NCRD),M(2)
   user coding
RETURN
END
```

where:

THICK	is the thickness of shell, to be modified by you.
XINTP	is the integration point coordinates.
NCRD	is the number of coordinates per point.
M(1)	is the user element number.
M(2)	is the internal element number.
NN	is the integration point number.

■ SSTRAN

Transformation of Substructures

Description

This subroutine allows you to transform a substructure. The substructure can be either rotated, mirrored or both. This routine is called every time a previously generated substructure is used.

Format

Subroutine SSTRAN is written with the following headers:

```
SUBROUTINE SSTRAN (NLEV,NSS,ICODE,NDEG,TDICTM)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION TDICTM (NDEG,NDEG)
    user code
RETURN
END
```

where:

NLEV	is the substructure level number.
NSS	is the substructure number.
ICODE	set to 1 if transformation given.
NDEG	is the number of degrees of freedom per node.
TDICTM	is the transformation matrix, defined by you.

Before a substructure is used, it is transformed such that the direction cosines of a triad in the new system with respect to the old system is TDICTM.

■ UTHICK

User-Specified Nodal Thicknesses

Description

Subroutine UTHICK is called automatically by the NODAL THICKNESS model definition block. The value of the THICK argument upon input is the value for nodal thickness entered by you. If this subroutine is not used, the nodal thickness data entered through the NODAL THICKNESS block are used.

Note: This routine should not be used if the thickness is to be considered a design variable. Use the GEOMETRY option instead.

Format

Subroutine UTHICK is called with the following headers:

```
SUBROUTINE UTHICK (THICK,COORD,NCRD,NOD,BEARC,NBEARF,INC,INCSUB)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION COORD (NCRD), BEARC(6, NBEARF)
 user coding
RETURN
END
```

where:

THICK	is the user-defined nodal thickness of node NOD. Upon input, THICK takes the value input through the NODAL THICKNESS block.
COORD	is the array of coordinates for this node. This array is only available if the COORDINATES option (and UFXORD option, if used) precedes the NODAL THICKNESS option.
NCRD	is the maximum number of coordinates per node.
NOD	is the node number.
BEARC	is not used.
NBEARF	is not used.
INC	is not used.
INCSUB	is not used.

■ UACTUAT

Prescribe the Length of an Actuator

Description

Subroutine UACTUAT allows you to control the length of an actuator in an incremental analysis. This is often useful in mechanism analyses, where the kinematics are prescribed. This is used with the truss element, type 9, when an initial length is given in the fourth field of the geometry option.

Format

Subroutine UACTUAT is called with the following headers:

```
SUBROUTINE UATUAT (M,INC,CPTIM,TIMINC,XLNGTH,OLNGTH)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION M(2)
 user coding
RETURN
END
```

where:

M(1)	your element number.
M(2)	internal element number.
INC	increment number.
CPTIM	time.
TIMINC	time increment.
XLNGTH	length of actuator to be set by you.
OLNGTH	current length of actuator.

7 List of Output Quantities Subroutines

Subroutines **Page**

ELEVAR 7-16
ELEVEC 7-18

IMPD 7-7
INTCRD 7-20

PLOTV 7-2

UBGINC 7-21
UBGITR 7-23
UEDINC 7-22
UELOOP 7-24
UPOSTV 7-4

7 Output Quantities Subroutines

This chapter describes user subroutines which can be used to obtain results from the analysis and manipulate it for postprocessing. There are also four dummy routines that can be used to set parameters for the advanced user. Table 7-1 summarizes these routines and indicates what parameters or model definition options are required to invoke the user subroutine.

Table 7-1 Output Quantities Subroutines Requirements

User Subroutine	Required Parameters or Model Definition Options	Purpose
ELEVAR	UDUMP	Allows postprocessing of element results.
ELEVEC	UDUMP	Allows postprocessing of element results in harmonic analysis.
INTCRD		Makes available integration point coordinates.
IMPD	UDUMP	Allows postprocessing of nodal vector results.
PLOTV	POST	Defines element quantity to be written to post file.
UBGINC		Dummy routine available at the beginning of each increment.
UBGITR		Dummy routine available at the beginning of each iteration.
UEDINC		Dummy routine available at the end of each increment.
UELOOP		Dummy routine available during major element loops.
UPOSTV	POST	Defines nodal vecors that are to be written to a post file.

■ PLOTV

User-selected Postprocessing of Element Variables

Description

Subroutine PLOTV is used in conjunction with either element code 19 or a negative code entered in the POST option. This allows you to define an element variable to be written to the post file.

Format

Subroutine PLOTV is written with the following headers:

```
 SUBROUTINE PLOTV(V,S,SP,ETOT,EPLAS,ECREEP,T,M,NN,LAYER,NDI,
+NSHEAR,JPLTCD)
 IMPLICIT REAL *8 (A-H, O-Z)
 DIMENSION S(1),SP(1),ETOT(1),EPLAS(1),ECREEP(1),M(2)
  user coding
 RETURN
 END
```

where:

V	is the variable to be plotted or put onto the post file, to be defined in this routine.
S	is the array of stresses at this integration point. For heat transfer analysis, S contains $\partial T/\partial X_i$ and $K_i \partial T/\partial X_i$. For a magnetostatic analysis, S contains the magnetic induction (B) and the magnetic field intensity (H).
SP	is the array stresses in the preferred direction if ORIENTATION is used.
ETOT	is the total strain (generalized) at this integration point.
EPLAS	is the total plastic strain at this integration point.
ECREEP	is the total creep strain at this integration point.
T	is the temperature at this integration point.
M(1)	is your element number.
M(2)	is the internal element number.
NN	is the integration point number.
LAYER	is the layer number (for beams or shells).
NDI	is the number of direct stresses.
NSHEAR	is the number of shear stresses.
JPLTCD	is the absolute value of the user entered code.

Example

For example, suppose you wish to output the sum of the squares of the two shear stresses in the friction theory. These are `S(2)` and `S(3)`, so the routine would appear as:

```
 SUBROUTINE PLOTV(V,S,SP,ETOT,EPLAS,ECREEP,T,M,NN,LAYER,NDI,NSHEAR,
+JPLTCD)
 IMPLICIT REAL *8 (A-H, O-Z)
 DIMENSION S(1),SP(1),ETOT(1),EPLAS(1),ECREEP(1)
 V=SQRT(S(2)**2 + S(3)**2)
 RETURN
 END
```

This quantity could then be postprocessed using Mentat.

■ UPOSTV

User-selected Postprocessing of Nodal Variables

Description

Subroutine UPOSTV is used in conjunction with the POST option to define a vector quantity that is to be written to the post file.

Format

Subroutine UPOSTV is written with the following headers:

```
      SUBROUTINE UPOSTV(N,NDEG,NCRD,NUMNP,IANTYP,JNODE,IUID,UPOST,
     * XORD,VECTORS,INC,CPTIM)
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION UPOST(NDEG),XORD(NCRD),VECTOR(NDEG,JNODE)
         user coding
      RETURN
      END
```

where:

N	is your node number.
NDEG	is the number of degrees of freedom per node.
NUMNP	is the number of nodes in the mesh.
IANTYP	is the analysis type – see PLDUMP in Chapter 9.
JNODE	is the number of vector quantities already defined – see PLDUMP in Chapter 9.
IUID	is your vector number.
UPOST	is user-defined components of vector for this node.
XORD	is the coordinates of this node.
VECTORS	is the displacement, etc. of this node. See IANTYP/JNODE Table on page 6 in PLDUMP section in Chapter 9.
INC	is the increment number.
CPTIM	is the total time.

Example

For example, you would like to post process the relative displacement of all nodes with respect to your node 5 for all time. You would need to obtain the displacement of node 5 and subtract this from the other displacements. This is done as follows:

```
      SUBROUTINE UPOSTV(N,NDEG,NCRD,NUMNP,IANTYP,JNODE,IUID,UPOST,
     * XORD,VECTOR,INC,CPTIM)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C     USER SUBROUTINE TO DEFINE NODAL POST VARIABLES
C
C     N          USER NODE NUMBER
C     NDEG       NUMBER OF DEGREES OF FREEDOM PER NODE
C     NCRD       NUMBER OF COORDINATES PER NODE
C     NUMNP      NUMBER OF NODES IN MESH
C     IANTYP     ANALYSIS TYPE - SEE PLDUMP IN VOLUME D
C     JNODE      NUMBER OF VECTOR QUANTITIES ALREADY DEFINED
C                - SEE PLDUMP IN VOLUME D
C     IUID       USER VECTOR NUMBER
C     UPOST      USER DEFINED COMPONENTS OF VECTOR FOR THIS NODE
C     XORD       COORDINATES OF THIS NODE
C     VECTORS    DISPLACEMENT, ETC OF THIS NODE.
C                SEE IANTYP/JNODE TABLE IN PLDUMP SECTION IN VOLUME D
C     INC        INCREMENT NUMBER
C     CPTIM      TOTAL TIME
C
      DIMENSION UPOST(NDEG),XORD(NCRD),VECTOR(NDEG,JNODE)
      INCLUDE '../COMMON/SPACE'
      INCLUDE '../COMMON/ARRAYS'
      INCLUDE '../COMMON/ARRAY2'
      INCLUDE '../COMMON/DEVELP'
      DIMENSION DISP5(12)
C     SET REFERENCE NODE LEXT=5
      LEXT=5
C     GET INTERNAL NODE NUMBER
```

```
      LINT=IBSRCH(LEXT,INTS(INOIDS),NUMNP,1)
      LA2=INPNUM+LINT-1
      IF(JOPTIT.NE.0) LINT=IGETSH(INTS(LA2),0)
C     GET REFERENCE DISPLACEMENT AND STORE INTO DISP5
      LA3=IDSXT+(LINT-1)*NDEG
      CALL MCPY(VARS(LA3),DISP5,NDEG,1,0)
C
C     GET DISPLACEMENT OF CURRENT NODE FROM VECTOR AND
C     SUBTRACT OFF REFERENCE DISPLACEMENT AND STORE BACK INTO UPOST
C
      DO I=1,NDEG
        UPOST(I)=VECTOR(I,1)-DISP5(I)
      ENDDO
C
      RETURN
      END
```

■ IMPD

Output of Nodal Quantities

Description

IMPD makes the displacements, coordinates, reaction forces, velocities and accelerations available at the end of each increment so that you can save them in any form convenient for postprocessing. During harmonic subincrements, IMPD allows you to obtain the complex displacements and reactions. In heat transfer (or Joule heating) analysis, this subroutine allows you to obtain nodal temperatures, fluxes, and voltages for your postprocessing. This routine is used in conjunction with the UDUMP option.

Stress Analysis

Format

Subroutine IMPD is written with the following headers:

```
SUBROUTINE IMPD (N,DD,TD,XORD,F,V,A,NDEG,NCRD)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION DD (NDEG), TD (NDEG), XORD (NCRD),F(NDEG),V(NDEG),A(NDEG)
    user coding
RETURN
END
```

where:

N	is the node number (the subroutine is called once per node per increment).
DD	is the array of displacement increments at this node.
TD	is the array of total displacements at this node.
XORD	are the coordinates of this node.
F	are the reaction forces at prescribed boundary conditions; residual load correction elsewhere at this node.
V	is the total velocity at this node.
A	is the total acceleration at this node.
NDEG	is the number of degrees of freedom per node (that is, the size of the DD, TD, V, and A arrays).
NCRD	is the number of coordinate directions per node (equals the size of the XORD array).

During harmonic subincrements:

DD	is the array of real displacements.
TD	is the array of imaginary displacements.
F	is the array of real reaction forces.
V	is the array of imaginary reaction forces.

Example

For example, suppose you wish to write on a file the displaced position of a three-dimensional solid structure for subsequent plotting. A scale factor of 5 is used on the displacements.

```
      SUBROUTINE IMPD (N,DD,TD,XORD,F,V,A,NDEG,NCRD)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION DD(NDEG), TD(NDEG),XORD(NCRD),F(NDEG),
     1 V(NDEG), A(NDEG)
      DIMENSION TXORD(3)
C     TXORD WILL BE THE COORDS + 5X TOTAL DISPLACEMENTS
      DO 1 I = 1,3 1 TXORD(I) = XORD(I) + 5.0*TD(I)
C     WRITE OUT DISPLACED POSITIONS ON TAPE 20.
      WRITE (20) N, TXORD
      RETURN
      END
```

After each increment, there are NUMNP records (number of nodal points) on logical unit 20, each containing a node number and three adjusted coordinates. Note that any additional file unit must be taken care of with the appropriate machine dependent JCL.

Heat Transfer (or Joule Heating) Analysis

Format

Subroutine IMPD is written with the following headers:

```
SUBROUTINE IMPD (N,DD,TD,XORD,F,V,A,NDEG,NCRD)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION DD (NDEG),TD(NDEG),XORD(NCRD),F(NDEG),V(NDEG),A(NDEG)
    user coding
RETURN
END
```

where:

N	is the node number (the subroutine is called once per node per increment).
DD	is the array of temperatures at this node.
TD	is the array of fluxes at this node.
XORD	is the coordinates of this node.
F	is the voltage at this node in a Joule heating analysis.
V	is not used.
A	is not used.
NDEG	is the number of degrees of freedom per node (that is, the size of the DD, TD, V, and A arrays).
NCRD	is the number of coordinate directions per node (equals the size of the XORD array).

Electrostatic Analysis

Format

Subroutine IMPD is written with the following headers:

```
SUBROUTINE IMPD (N,DD,TD,XORD,F,V,A,NDEG,NCRD)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION XORD(NCRD)
    user coding
RETURN
END
```

where:

N	is the node number (the subroutine is called once per node per increment).
DD	is the potential at this node.
TD	is the charge at this node.
XORD	is the coordinates of this node.
F	is not used.
V	is not used.
A	is not used.
NDEG	is the number of degrees of freedom per node = 1.
NCRD	is the number of coordinate directions per node (equals the size of the XORD array).

Magnetostatic Analysis

Format

Subroutine IMPD is written with the following headers:

```
SUBROUTINE IMPD (N,DD,TD,XORD,F,V,A,NDEG,NCRD)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION XORD(NCRD)
    user coding
RETURN
END
```

where:

N	is the node number (the subroutine is called once per node per increment).
DD	is the potential at this node.
TD	is the current at this node.
XORD	is the coordinates of this node.
F	is not used.
V	is not used.
A	is not used.
NDEG	is the number of degrees of freedom per node = 1.
NCRD	is the number of coordinate directions per node (equals the size of the XORD array).

Harmonic Electromagnetic Analysis

Format

Subroutine IMPD is written with the following headers:

```
SUBROUTINE IMPD (N,DD,TD,XORD,F,V,A,NDEG,NCRD)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION DD (NDEG),TD(NDEG),XORD(NCRD),F(NDEG),V(NDEG),A(NDEG)
    user coding
RETURN
END
```

where:

N	is the node number (the subroutine is called once per node per increment).
DD	is the array of real component of potential at this node.
TD	is the array of imaginary component of potential at this node.
XORD	is the coordinates of this node.
F	is the real component of the reaction.
V	is the imaginary component of the reaction.
A	is not used.
NDEG	is the number of degrees of freedom per node = 1.
NCRD	is the number of coordinate directions per node (equals the size of the XORD array).

Transient Electromagnetic Analysis

Format

Subroutine IMPD is written with the following headers:

```
SUBROUTINE IMPD (N,DD,TD,XORD,F,V,A,NDEG,NCRD)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION DD (NDEG),TD(NDEG),XORD(NCRD),F(NDEG)
    user coding
RETURN
END
```

where:

N	is the node number (the subroutine is called once per node per increment).
DD	is the array of incremental potential at this node.
TD	is the array of total potential at this node.
XORD	is the coordinates of this node.
F	is the reaction forces at applied boundary conditions.
V	is not used.
A	is not used.
NDEG	is the number of degrees of freedom per node (that is, the size of the DD, TD, and F arrays).
NCRD	is the number of coordinate directions per node (equals the size of the XORD array).

Accoustic Analysis

Format

Subroutine IMPD is written with the following headers:

```
SUBROUTINE IMPD (N,DD,TD,XORD,F,V,A,NDEG,NCRD)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION XORD(NCRD)
    user coding
RETURN
END
```

where:

N	is the node number (the subroutine is called once per node per increment).
DD	is the incremental pressure at this node.
TD	is the total pressure at this node.
XORD	is the coordinates of this node.
F	is the reaction force.
V	is the pressure velocity.
A	is the pressure acceleration.
NDEG	is the number of degrees of freedom per node = 1.
NCRD	is the number of coordinate directions per node (equals the size of the XORD array).

Fluid or Fluid-Thermal Analysis

Format

Subroutine IMPD is written with the following headers:

```
SUBROUTINE IMPD (N,DD,TD,XORD,F,V,A,NDEG,NCRD)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION DD (NDEG),TD(NDEG),XORD(NCRD),F(NDEG),V(1),A(1)
    user coding
RETURN
END
```

where:

N	is the node number (the subroutine is called once per node per increment).
DD	is not used.
TD	is the array of velocities at this node.
XORD	is the coordinates of this node.
F	is the array of forces.
V	is the temperature at this node in a fluid-thermal analysis.
A	is the flux at this node in a fluid-thermal analysis.
NDEG	is the number of degrees of freedom per node (that is, the size of the DD, TD, and F arrays).
NCRD	is the number of coordinate directions per node (equals the size of the XORD array).

■ ELEVAR

Output of Element Quantities

Description

ELEVAR makes element (integration point) quantities available at the end of each increment so that you can save them in any form convenient for postprocessing. This routine is used in conjunction with the UDUMP option.

Format

Subroutine ELEVAR is written with the following headers:

```
      SUBROUTINE ELEVAR(N,NN,KC,GSTRAN,GSTRES,STRESS,PSTRAN,
     1 CSTRAN,VSTRAN,CAUCHY,EPLAS,EQUIVC,SWELL,KRTYP,PRANG,DT,
     2 GSV,NGENS,NGEN1,NSTATS,NSTASS,THERM)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION GSTRAN(NGENS),GSTRES(NGENS),
     1 STRESS(NGEN1),PSTRAN(NGEN1),CSTRAN(NGEN1),VSTRAN(NGEN1),
     2 CAUCHY(NGEN1),DT(NSTATS),GSV(1),THERM(NGEN1),KRTYP(4),
     3 PRANG(3,2)
        User Coding
      RETURN
      END
```

where:

N	is the element number.
NN	is the integration point number.
KC	is the layer number for beam or shell elements.
GSTRAN	is the total strain array.
GSTRES	is the generalized force array.
STRESS	is the total stresses array.
PSTRAN	is the plastic strain array.
CSTRAN	is the creep strain array.
VSTRAN	is the viscoelastic strain array.
CAUCHY	is the Cauchy stress array.

EPLAS	is the equivalent plastic strain.
EQUIVC	is the equivalent creep strain.
SWELL	is the swelling strain.
KRTYP(1)	is the crack indicator for the first crack direction: 0 = no crack in this direction. 1 = open crack, developed in this increment. 2 = open crack, developed in previous increment. 3 = closed crack.
KRTYP(2)	is the crack indicator for the second crack direction.
KRTYP(3)	is the crack indicator for the third crack direction.
KRTYP(4)	is the crushing indicator: 0 = no crushing. 1 = crushing occurring in this increment. 2 = crushing occurred in previous increment.
PRANG	(i,1) = components of normal to the first crack plane. (i,2) = components of normal to the second crack plane (3D only).
DT	is the state variables array, temperature first.
GSV	is the global state variable array.
NGENS	is the number of generalized strains.
NGEN1	is the number of physical components.
NSTATS	is the number of state variables.
NSTASS	is the number of global state variables.
THERM	is the total thermal strain array.

■ ELEVEC

Output of Element Quantities in Harmonic Analysis

Description

ELEVEC makes element (integration point) quantities available at the end of each harmonic subincrement so that you can save them in any form convenient for your postprocessing. This routine is used in conjunction with the UDUMP option.

Format

Subroutine ELEVEC is written with the following headers:

```
      SUBROUTINE ELEVEC(N,NN,KC,GSTRAN,GSTRES,STRESS,PSTRAN,
     1 CSTRAN,VSTRAN,CAUCHY,EPLAS,EQUIVC,SWELL,KRTYP,PRANG,DT,
     2 GSV,NGENS,NGEN1,NSTATS,NSTASS,STSRE,STSIM,STNRE,STNIM)
      IMPLICIT REAL *8 (A-H, O-Z)
      DIMENSION GSTRAN(NGENS),GSTRES(NGENS),
     1 STRESS(NGEN1),PSTRAN(NGEN1),CSTRAN(NGEN1),VSTRAN(NGEN1),
     2 CAUCHY(NGEN1),DT(NSTATS),GSV(NSTASS),
     3 STSRE(NGEN1),STSIM(NGEN1),STNRE(NGEN1),STNIM(NGEN1)
        user coding
      RETURN
      END
```

where:

N	is the element number.
NN	is the integration point number.
KC	is the layer number for beam or shell elements.
GSTRAN	is the total strain array.
GSTRES	is the generalized force array.
STRESS	is the total stresses array.
PSTRAN	is the plastic strain array.
CSTRAN	is the creep strain array.
VSTRAN	is the viscoelastic strain array.
CAUCHY	is the Cauchy stress array.

EPLAS	is the equivalent plastic strain.
EQUIVC	is the equivalent stress.
SWELL	is the swelling strain.
KRTYP	is the cracking type.
PRANG	is the crack angle.
DT	is the state variables array, temperature first.
GSV	is the global state variable array.
NGENS	is the number of generalized strains.
NGEN1	is the number of physical components.
NSTATS	is the number of state variables.
NSTASS	is the number of global state variables.
STSRE	is the real harmonic stress.
STSIM	is the imaginary harmonic stress.
STNRE	is the real harmonic strain.
STNIN	is the imaginary harmonic strain.

■ INTCRD

Output of Integration Point Coordinates

Description

Subroutine INTCRD makes the integration point coordinates for the stiffness matrix available at each increment. You can save them in any form convenient for postprocessing.

Format

Subroutine INTCRD is written with the following headers:

```
SUBROUTINE INTCRD(M,NN,XORD,NCRD)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION XORD(NCRD)
   user coding
RETURN
END
```

where:

M	is the element number.
NN	is the integration point number.
XORD	are the coordinates of this integration point.
NCRD	is the number of coordinate directions.

■ UBGINC

Beginning of Increment

Description

This subroutine is called at the beginning of each new increment. It can be used to define or modify data variables stored in common blocks.

Note: No special flag is required in the input file.

Format

Subroutine UBGINC is written with the following headers:

```
SUBROUTINE UBGINC(INC,INCSUB)
IMPLICIT REAL *8 (A-H, O-Z)
   user coding
RETURN
END
```

where:

INC	is the increment number.
INCSUB	is the subincrement number.

■ UEDINC

End of Increment

Description

This subroutine is called at the end of each increment. It can be used to define or modify data variables stored in common blocks.

Note: No special flag is required in the input file.

Format

Subroutine UEDINC is written with the following headers:

```
SUBROUTINE UEDINC(INC,INCSUB)
IMPLICIT REAL *8 (A-H, O-Z)
    user coding
RETURN
END
```

where:

INC	is the increment number.
INCSUB	is the subincrement number.

■ UBGITR

Beginning of Iteration

Description

This subroutine is called at the beginning of each iteration in the solution of the nonlinear problem. It can be used to define or modify data variables stored in common blocks.

Note: No special flag is required in the input file.

Format

Subroutine UBGITR is written with the following headers:

```
SUBROUTINE UBGITR(INC,INCSUB,NCYCLE)
IMPLICIT REAL *8 (A-H, O-Z)
    user coding
RETURN
END
```

where:

INC	is the increment number.
INCSUB	is the subincrement number.
NCYCLE	is the iteration number (the first is labeled zero).

■ UELOOP

Beginning of Element Loop

Description

This subroutine is called in a loop over the elements. It can be used to define or modify data variables stored in common blocks.

Note: No special flag is required in the input file.

Format

Subroutine UELOOP is written with the following header cards:

```
SUBROUTINE UELOOP(M,N,IL)
IMPLICIT REAL *8 (A-H, O-Z)
    user coding
RETURN
END
```

where:

M	is your element number.
N	is the internal element number.
IL	is the loop flag. = 1 form consistent nodal loads from distributed loads. = 2 stiffness matrix formation. = 3 mass matrix formation. = 4 stress recovery.

8 List of Hydrodynamic Lubrication Subroutines

Subroutines	Page
UBEAR	8-2
UGROOV	8-3
URESTR	8-4
UTHICK	8-5
UVELOC	8-6

8 Hydrodynamic Lubrication Subroutines

This chapter describes user subroutines that can be used to customize a hydrodynamic bearing analysis. In such problems, the geometry can be complicated by grooves in the bearing surface or nonuniform lubricant. The user routines provided here facilitate the input of this data. Table 8-1 summarizes these routines and indicates what parameters or model definition options are required to invoke the user subroutine.

Table 8-1 Hydrodynamic Lubrication Subroutines Requirements

User Subroutine	Required Parameters or Model Definition Options	Purpose
UBEAR	BEARING	Define the orientation of the film surface.
UGROOV	BEARING	Define the groove depth.
URESTR	BEARING RESTRICTOR	Define the nonuniform restrictor coefficient and pump pressures.
UTHICK	BEARING NODAL THICKNESS THICKNS CHANGE	Define the lubricant thickness.
UVELOC	BEARING VELOCITY	Define the nodal velocity of bearing surface.

■ UBEAR

Input of Spatial Orientation of Lubricant Thickness

Description

In bearing analysis, the lubricant is modeled by a planar mesh due to the absence of pressure gradients across the film height. The program integrates the obtained pressure distribution over the entire mesh. This yields a set of equivalent consistent nodal forces perpendicular to the lubricant. In order to calculate the load capacity of a particular bearing system, these forces must be transformed to the global coordinate system. For this purpose, information is required about the direction cosines of the lubricant normal. This can be done in using subroutine UBEAR which is called for each node.

Format

Subroutine UBEAR is written with the following headers:

```
SUBROUTINE UBEAR (COORD,COS,NODE,NCRD)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION COORD (NCRD), COS(3)
   user coding
RETURN
END
```

where:

COORD	is the array of coordinates at this node.
COS	is the array of direction cosines of the vector perpendicular to the lubricant; to be defined in this routine. A default vector (0,0,1) is assumed if not specified.
NODE	is the node number.
NCRD	is the number of coordinates per node.

■ UGROOV

Input of Groove Depths

Description

In bearing analysis, discontinuous film thicknesses are often applied to increase the load carrying capacity. This is usually done by grooves, which can be defined in the GEOMETRY block. However, this is not possible if position dependent groove depths have to be included. In such situations, subroutine UGROOV must be used. It is called at each integration point and allows you to specify the groove depth at these points. In addition, this routine can be used for selecting elements which are located at grooves in case complex groove patterns have to be modeled.

Format

Subroutine UGROOV is written with the following headers:

```
SUBROUTINE UGROOV (THICK,COORD,M,NN,NCRD)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION COORD (NCRD)
   user coding
RETURN
END
```

where:

THICK	is the groove depth magnitude to be specified.
COORD	is the array of coordinates at this integration point.
M	is the element number.
NN	is the integration point number.
NCRD	is the number of coordinates per node.

■ URESTR

Input of Nonuniform Restrictor Coefficients

Description

In bearing analysis, it is often necessary to include nonuniform restrictor coefficients and pump pressures. Subroutine URESTR allows this. It is called at each increment for each integration point on each element-surface, given in the RESTRICTOR model definition set, and allows you to modify the restrictor coefficient and pump pressure input on the data blocks.

Format

Subroutine URESTR is written with the following headers:

```
SUBROUTINE URESTR (CR,PP,PS,N,INC)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION N(3)
   user coding
RETURN
END
```

where:

CR	is the ratio of the desired restrictor coefficient to that given on the RESTRICTOR data set for this element to be defined by you (preset to 1).
PP	is the ratio of the desired pump pressure to that given on the RESTRICTOR data set for this element to be defined by you (preset to 1).
PS	is the surface pressure at the beginning of the increment.
N(1)	is the element number.
N(2)	is the face number.
N(3)	is the integration point number.
INC	is the current increment number.

Note that since CR and PP are defined as ratios, if you do not re-define them in this routine, the data block values are used. If you wish to give absolute values here, the corresponding values on the RESTRICTOR data set can be conveniently set to 1.

■ UTHICK

Generation or Modification of Nodal Thickness or Thickness Change Field

Description

In bearing analysis, the film height usually varies over the entire lubricant region. Subroutine UTHICK allows you to define, or to redefine previously specified, nodal thicknesses. It is called for each node in the mesh.

In addition, this routine can be used to define thickness increments in incremental analysis or within subincrements when evaluating damping and/or stiffness coefficients. In order to enable the specification of thickness increments as function of previously calculated bearing properties, you have access to the latter quantities in this routine.

Format

Subroutine UTHICK is written with the following headers:

```
SUBROUTINE UTHICK (THICK,COORD,NCRD,NOD,BEARC,NBEARF,INC,INCSUB)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION COORD (NCRD), BEARC (6,NBEARF)
   user coding
RETURN
END
```

where:

THICK	is the lubricant thickness or incremental lubricant thickness magnitude to be specified for this node.
COORD	is the array of coordinates for this node.
NCRD	is the number of coordinates per node.
NOD	is the node number.
BEARC	is the matrix of previously calculated bearing properties. Each column contains three bearing force and three bearing moment components. The quantities calculated in the previous increment are stored in the first column. Each subsequent column contains the properties pertaining to the previous set of subincrements.
NBEARF	is the maximum number of subincrement as given on the BEARING parameter.
INC	is the increment number.
INCSUB	is the subincrement number.

■ UVELOC

Generation or Modification of Nodal Velocity Vectors

Description

In bearing analysis, it is sometimes necessary to include a position dependent velocity field. Subroutine UVELOC, which is called for each node, allows you the specification or re-definition of previously specified nodal velocity vectors.

Note: No special flag is required in the input file.

Format

Subroutine UVELOC is written with the following headers:

```
SUBROUTINE UVELOC (VELOC,COORD,NCRD,NODE)
IMPLICIT REAL *8 (A-H, O-Z)
DIMENSION VELOC (NCRD),COORD(NCRD)
   user coding
RETURN
END
```

where:

VELOC	is the array of nodal velocity components to be defined.
COORD	is the array of coordinates at this node.
NCRD	is the number of coordinates.
NODE	is the node number.

9 List of Special Routines

Special Routines	Page
PLDUMP	9-2

9

9 Special Routines – MARC Post File Processor PLDUMP

This chapter discusses a stand-alone program that provides examination of the post-processing file created by the POST option. This allows you to perform additional calculations based upon results calculated in MARC, and to create a post file. These results can then be viewed with Mentat.

■ PLDUMP

MARC Post File Processor

PLDUMP is a small utility program which is used to access, analyze, convert, and process MARC binary and formatted post files. The source code to PLDUMP is supplied at no additional charge to MARC customers and is available on the MARC installation media. You can modify this source as necessary to suit your requirements. A description of the structure of the post file follows.

When PLDUMP is executed, you are asked several questions, as follows:

1. Output file name:
2. Write post data to output option:

   Valid responses are:

   `NONE` – do not write post file to output

   `PARTIAL` – write analysis control data to output

   `FULL` – write entire post file to output
3. Type of post file to read:

   Valid responses are:

   `BINARY` – to read a binary file

   `FORMATTED` – to read a formatted file
4. Old post file file name:
5. Type of post file to write:

   Valid responses are:

   `NONE` – do not write a new post file

   `BINARY` – write a new formatted file

   `FORMATTED` – write a new binary file
6. New post file name:
7. Highest increment number to process:

   Processing stops after this increment number is processed. To process only a nonincremental header, enter a `-1`.

Marc Post File Layout

The following describes the contents of the nonincremental data which is found on a MARC post file. This information is the same for either the binary or the formatted post file.

These subroutines are written with the following blocks:

Subroutine	Blocks
POSTHD	1-6
CHKPST	7-13
DUMPNURB	14
OSCINC	15-18, 20-22
NEWPST	19
SCIMP	23
INCRE	24 in stress
IMPHT	24 in heat transfer.
IMPBER	24 in bearing
INCRACO	24 in bearing
INSPRP	24 in spectrum response
INCREFLD	24 in fluid analysis
PSTSEN	25, 26
PSTOPT	27

MARC subroutine SETSV reads a heat transfer file into stress.

MARC subroutine INITC reads initial conditions for dynamics.

MARC subroutine POSTHD copies a previous post file onto the beginning of new file for restart.

Notes: All character data stored on the post file is stored in one character per word (A1) format regardless of machine (except for the Beginning of Increment Indicator, Block 10, which is still 4H****).

Column labeled "Record" is the record number in the current set of records in a binary post file only.

Column labeled "Line" is the line number in the current block for a formatted post file only. If "Line" is blank, then the entire set of lines is repeated throughout the block.

Column labeled "Format" is the format specification for a formatted post file. For character data, the format also indicates how the character data was stored in the words read from a binary post file record.

This first column indicates whether a particular part of the post file is new in K7 (by means of a "7"), is corrected from K6 or identical to K6 (by means of a " ").

The post file for K7 has several significant structural changes relative to previous releases. These changes are to support domain decomposition (see block 5); you can add your own nodal post variables through user subroutine UPOSTV (see 10th field, POST model definition option), new analysis types (see IANTYP, block 2), support of design sensitivity and optimization, and changes due to contact. In K6 and previous post revisions, the rigid contact bodies were represented as piecewise linear segments (lines or patches) and were treated as elements at each increment. In the K7 and future releases, they are placed on the post file only once and, if necessary, their position is recalculated based upon the informaion given in block 21. This results in a substantially smaller post file.

```
-----------------------------------------------------------------------
 Block 1 - Analysis title
           Block 1 is REQUIRED.
           Number of records for binary : 1  Words/record: 70
           Number of lines for formatted: 1 line of 70 characters

 Record Line Format Variable  Description
    1     1  70A1   TITLE(J)  J=1,70  70 character analysis title

-----------------------------------------------------------------------
 Block 2 - Analysis verification data
           Block 2 is REQUIRED.
           Number of records for binary : 1  Words/record: 18
           Number of lines for formatted: 3
           Items per line for formatted : 6 I13 integers

 Record Line Format Variable  Description
    1     1  6I13   INUM      Number of element postvariables
                    LNUM      Total number of nodes in the mesh
                    MNUM      Total number of elements in the mesh
                    NDEG      Maximum number of dof's for any node in
                              the mesh
                    NSTRES    Maximum number of integration points for
                              any element in the mesh
                    INOD      Number of nodal postvariables
                              NOTE: This number also appears
                                    in the incremental header.
                                    See KNOD, Block 11.
                              INOD  = JNODE*NDEG
K7                            JNODE = number of nodal vectors
                                    appearing on the posttape plus number
                                      of user defined vectors.
                                      (See IANTYP, below)
          2  6I13   IPSTCC    Topological data available on the tape
                              IPSTCC = 0  not available
                              IPSTCC = 1      available
                              NOTE: In K2 and later, IPSTCC = 1 always
                    NADTIE    Number of ties from adaptive meshing
                    NCRD      Maximum number of coordinates for any
                              node in the mesh
                    NNODMX    Maximum number of nodes per element for
                              any element in the mesh
```

IANTYP Analysis type indicator (See Table 1)
NOTE: This number also appears in the incremental header. See JANTYP, Block 17.

Table 1 - Analysis type indicator

IANTYP	Description	JNODE
1	displacements	2
2	displacements w/r	3
3	dynamic	4
4	dynamic w/r	5
5	rigid plastic	3
6	rigid plastic w/r	4
7	heat transfer	1
8	heat transfer w/f	2
9	joule heat	2
10	joule heat w/f	3
11	r harmonic	1
12	r harmonic w/r	2
13	r harm & dyn	4
14	r harm & dyn w/r	5
15	c harmonic	2
16	c harmonic w/r	4
17	c harm & dyn	4
18	c harm & dyn w/r	6
19	cpld static	3
20	cpld static w/r	5
21	cpld dynamic	5
22	cpld dynamic w/r	7
23	cpld + joule static	4
24	cpld + joule static w/r	6
25	cpld + joule dynamic	6
26	cpld + joule dynamic w/r	8
27	eigenvector modal	2
28	eigenvector buckle	1
29	bearing	5
30	modal w/r	2
31	modal + mass	2
32	modal + mass w/r	3
33	random response	3
34	random response w/r	4
35	auto correlation	3

```
                36        auto correlation w/r                                4
                37        load create                                         1
                38        -----------------                                   0
                39        buckle                                              1
                40        buckle w/r                                          2
                41        rigid plastic coupled                               4
                42        rigid plastic coupled w/r                           6
                43        electrostatics                                      2
                45        magnetostatics                                      2
                47        acoustics                                           2
                48        transient electromagnetic                           2
                49        harmonic electromagnetic                            4
                50        pore pressure calculation                           1
                51        coupled soil analysis                               3
K7              52        auto forge                                          5
K7              53        coupled auto forge                                  7
 .              54        static w/contact-fric                               4
 .              55        dynamic w/contact-frict                             6
 .              56        r-p flow w/contact-frict                            5
 .              57        coupled-static w/contact-frict                      6
 .              58        coupled-dynamic 2/contact-frict                     8
 .              59        coupled rigid plastic w/contact-frict               7
 .              60        sensitivity                                         3
 .              61        design optimization                                 3
 .              62        shape optimization                                  3
 .              63        fluid analysis - mixed method                       4
 .              64        fluid analysis - penalty method                     3
 .              65        fluid-thermal mixed method w/flux                   7
 .              66        fluid-thermal penalty mithod w/flux                 6
 .              67        fluid-thermal mixed method                          6
 .              68        fluid-thermal penalty method                        5
 .              69        fluid-solid mixed method                            6
 .              70        fluid-solid penalty method                          5
 .              71        fluid-thermal-solid mixed method w/flux             9
 .              72        fluid-thermal-solid pnealty method w/flux           8
 .              73        fluid-thermal-solid mixed method                    8
K7              74        fluid-thermal-solid penalty method                  7
                        ICOMPL    Solution space indicator
                                  ICOMPL = 0  real    space
                                  ICOMPL = 1  complex space
          3  6I13       NBCTRA    Number of nodes with transformations
                        POSTRV    Posttape revision number
```

```
                                    POSTRV = 0  prior to K1
                                    POSTRV = 1  K1
                                    POSTRV = 2  K2
                                    POSTRV = 3  K3
                                    POSTRV = 4  K4
                                    POSTRV = 5  K5
                                    POSTRV = 6  K6
K7                                  POSTRV = 7  K7
                        NDISTL      Number of distributed loads
                                    NOTE: NDISTL always >= 3 in MARC.
                        NSET        Total number of sets
                        NSPRNG      Number of springs
                        NDIE        Number of dies
```

```
 Block 3  - Set information
            Block 3 is REQUIRED.
            Number of records for binary : 1  Words/record: 12
            Number of lines for formatted: 2
            Items per line for formatted : 6 I13 integers

 Record Line Format Variable  Description
    1     1  6I13   NESETS    The number of element sets
                    NNSETS    The number of node sets
                    NISETS    The number of int. points sets
                    NLSETS    The number of layer sets
                    NDSETS    The number of dof sets
                    NINSET    The number of increments sets
           2  6I13   KELEM     The total # of items in element sets
                    KNODE     The total # of items in node sets
                    KINT      The total # of items in int. point sets
                    KLAYR     The total # of items in layer sets
                    KDOF      The total # of items in dof sets
                    KINC      The total # of items in increment sets

K7
K7 Block 4  - Dummy
            Block 4 is only included if K7 or greater post revision.
.           Number of records for binary : 1  Words/record: 1
.           Number of lines for formatted: 1
.           Items per line for formatted : 1 I13 integers
.
. Record Line Format Variable  Description
K7    1     1   I13   IDUN      Not used
```

```
K7   Block 5  - Domain Decomposition identifier.
K7              Block 5 is only included if K7 or greater post revision
 .              Number of records for binary : 1  Words/record: 2
 .              Number of lines for formatted: 1
 .              Items per line for formatted : 2 2I13 integers
 .
 .    Record Line Format Variable  Description
 .       1     1  6I13   NPROCD    Number of domains used. If zero, then no
 .                                  domain composition
 .                       IDOMID    Domain identifier
 .
 .
 .             Note:  If IDOMID=0 and NPROCD is greater than 1, this post file
 .                  is for the highest level and contains no analysis results.
 .                    This post file may be closed and one should open the
K7                   associated post files for each domain

     ------------------------------------------------------------------------
     Block 6 - Element variable postcodes
               Block 6 is OPTIONAL.  Omitted if INUM = 0 (Block 2).
               Number of records for binary : INUM   Words/record: 25
               Number of lines for formatted: INUM
               Items per line for formatted : 1 I13 integer, 24 chars

               Note:  See also MARC documentation Vol. C, Section C3.3 for
                      more information on the MARC POST option and
                      post codes

     Record Line Format Variable  Description
        J     J  I13,   JPOST(J)  Jth element variable identifier

                                  JPOST(J) = ICODE(J)+1000*LCODE(J)

                                  where ICODE(J) = post code J
                                        LCODE(J) = layer number for post
                                                   code J
                                  Note: LCODE = 0 indicates a
                                        layer number is not relevant, or
                                        generalized quantity is given for
                                        shells, plates and beams

                  24A1   LABL(I,J) I=1,24  24 char label for post code J
       J=1,INUM
```

```
-----------------------------------------------------------------------
 Block 7 - Element connectivities
           Block is OPTIONAL.  Omitted if MNUM   = 0 (Block 2)
                                       of IPSTCC = 0 (Block 2).
           Number of records for binary : MNUM Words/record: NNODMX+3
           Number of lines for formatted: ((NNODMX+2)/6+1)*MNUM
           Number of lines per element  :  (NNODMX+2)/6+1
           Items per line for formatted : 6 I13 integers

 Record Line Format Variable  Description
    J        I13,   IDELEM    Element id
             I13,   JTYPE     Element type
             I13,   NNODE     Number of nodes in this element
          3I13/6I13/...
                    LM(I)     I=1,NNODMX  Element connectivity
   J=1,MNUM

-----------------------------------------------------------------------
 Block 8 - Nodal coordinates
           Block is OPTIONAL.  Omitted if IPSTCC = 0 (Block 2)
                                       or NCRD   = 0 (Block 2).
           Number of records for binary : LNUM  Words/record: NCRD+1
           Number of lines for formatted: (NCRD/6+1)*LNUM
           Number of lines per node     :  NCRD/6+1
           Items per line for formatted : 1 I13 integer plus
                                          5 E13.6 reals
           Continuation lines per node  : 6 E13.6 reals

 Record Line Format Variable  Description
    J        I13,   IDNODE    Node id
        5E13.6/6E13.6/...
                    XORD(I)   I=1,NCRD  Nodal coordinates
   J=1,NCRD

-----------------------------------------------------------------------
 Block 9 - Spring data
           Block is OPTIONAL.  Omitted if NSPRNG = 0 (Block 2)
           Number of records for binary : NSPRNG  Words/record: 5
           Number of lines for formatted: NSPRNG
           Items per line for formatted : 5 I13 integers
```

```
 Record Line Format Variable  Description
    J     J  5I13   ISPRNG(I) I = 1  Spring id
  J=1,NSPRNG                   = 2  Node 1
                               = 3  Degree of freedom 1
                               = 4  Node 2
                               = 5  Degree of freedom 2

-----------------------------------------------------------------------
 Block 10 - Nodal codes and nodal transformations
            Block is REQUIRED.
            Number of records for binary : 1   Words/record: LNUM
            Number of lines for formatted: (LNUM-1)/6+1
            Items per line for formatted : 6 I13 integers

 Record Line Format Variable  Description
    1        6I13   NCODE(I)  I=1,LNUM  Nodal codes

                               NCODE(I) = IDOF + 1000*ITRN

                               where

                               IDOF = node code for node I
                                      (See Table 4 in incremental data)
                               ITRN = transformation number
                                      for node I

-----------------------------------------------------------------------
 Block 11 - Ties due to Adaptive Meshing
            Block is OPTIONAL.  Omitted if NADTIE = 0 (Block 2)
                                        or POSTRV ≤ 5
          Number of records for binary : 2*NADTIE Words/record:
                                          variable
            Number of lines for formatted: 2*NADTIE
            Items per line for formatted : 6 I13

            Records come in NADTIE pairs

 Record Line Format Variable  Description
    J        2I13             1: Tie id either 91 or 92
                              2: Number of nodes associated with tie;
                                 either 3 or 5
   J+1       6I13             Actual Node numbers associated with tie
```

```
-----------------------------------------------------------------------
 Block 12 - Transformations
            Block is OPTIONAL.  Omitted if NBCTRA = 0 (Block 2)
                                        or IPSTCC = 0 (Block 2).
            Number of records for binary : NBCTRA  Words/record: 9
            Number of lines for formatted: 2*NBCTRA
            Items per line for formatted : 6 E13.6 reals
            Continuation lines           : 3 E13.6 reals

 Record Line Format Variable  Description
    J    6E13.3/3E13.3
                   ((D(I,K), I=1,3),K=1,3)  Direction cosines
                             for transformation from local
   J=1,NBCTRA                coordinates to global

-----------------------------------------------------------------------
 Block 13 - Set definition
            Block is OPTIONAL.  Omitted if NSET   = 0 (Block 2)
                                      or IPSTCC = 0 (Block 2).
            Number of records for binary : 3 records/set
            Number of lines for formatted: (NITEMS+5)/6+2

 Record Line Format Variable  Description
    1     1  12A1   NAME(I)   I=1,12  12 character set name
    2     2  2I13   NITEMS    Number of items in set
                    SETTYP    Set type = 0  element set
                                       = 1  nodal set
                                       = 2  integ. pt. set
                                       = 3  layer set
                                       = 4  dof set
                                       = 5  increment set
    3      6I13/... ITEMS(I)  I=1,NITEMS  Set items
                              (Omit record 3 if NITEMS=0)
                        Marc posttape layout
                             version K3

    The following describes the contents of the header of
    the incremental data part on a MARC posttape.

-----------------------------------------------------------------------
```

The contact geometry data described in block 14 contains multi-record data depending upon the type of body. Symbolically, it appears as:

```
do   n = 1 , ndie
     subblock 14a
     if (ibodtyp.ne.0) then
         subblock 14b
         do    ℓc = 1, numcrv
               subblock 14c
              if (ibodtyp.eq.1) then
                 do ℓp = 1, npatch
                   subblock 14.d1
                 enddo
              endif
              if (ibodtyp.eq.2) then
                  do ℓp = 1, npatch
                      subblock 14.d2
                  enddo
              endif
              if (ibodtyp.eq.3) then
                  do ℓc = 1, nptu
                     subblock 14.d3
                  enddo
              endif
              if (ibodtyp.eq.4) then
                  do ℓc = 1, nptu*nptv
                     subblock 14.d4
                  enddo
              endif
              if (ibodtyp.eq.1, or ibodtyp.eq.2) then
                  do ℓp = 1, npoint
                     subblock 14e.1
                  enddo
              endif
              if (ibodtyp.eq.3) then
                  do ℓc = 1, nptu
                     subblock Ae.3
                  enddo
              endif
              if (ibodtyp.eq.4) then
                  do ℓc = 1, nptu*nptv
                     subblock 14e.3
```

```
                                    enddo
                                endif
                                if (ibodtyp.eq.3) then
                                    do &k = 1, nptu + noru
                                       subblock 14f.3
                                    enddo
                                endif
                                if (ibodtyp.eq.4) then
                                    do &k = 1, nptu + nptu + nptv + norv
                                       subblock 14f.3
                                    enddo
                                endif
                                if (ibodtyp.eq.4, and ntrim.gt.0)
                                    do nt = 1, ntrip
                                       subblock 14g.4
                                       do    ntp = 1, npttr
                                            subblock 14.h.4
                                       enddo
                                    enddo
                                endif
                            enddo
                        endif
                    enddo
    ------------------------------------------------------------------------
K7  Block 14 - Contact Geometric Data
K7             This block is optional. If it is skipped, POSTRV is less than
 .             7 or NDIE=0
 .             Data for each surface is given in groups,
 .             There are NDIE groups
 .
 .   Subblock 14a - Body header information
 .   Record Line Format Variable   Description
 .      1     1  3I13   IBOD       Body number
 .      2     2  2I13   IBODTYP    Body type
 .                                  0 - deformable
 .                                  1 - 2d links type 9
 .                                  2 - 3d links type 18
 .                                  3 - 2d NURB curve
 .                                  4 - 3d NURB surface
 .                      NUMCRV     Number of curves or surfaces if NURB
 . ************************************
K7 If IBODTYP = 0, skip all remaining subblocks
```

```
K7   Subblock 14b - Reference point
K7   Record Line Format Variable   Description
        1     1  6E13    RPX    ┐
                         RPY    ├    Reference Point Coordinates
                         RPZ    ┘
                         DCX    ┐
                         DCY    ├    Direction cosine of Axis of Rotation
                         DCZ    ┘

     ----------------------------------
     Subblock 14c - Surface information
     Record Line Format Variable   Description
        1     1  6I13              Depends on IBODTYP
        If IBODTYP = 0  NUMELM     Number of elements in body
        If IBODTYP = 1  NPATCH     Number of links
                        NPOINT     Number of points
        If IBODTYP = 2  NPATCH     Number of patches
                        NPOINT     Number of points
        If IBODTYP = 3
                        NPTV       Number of control points
                        NPTV ≡ 1
                        NORU       Order of NURB
                        NORU ≡ 1
                        NTRIM ≡ 0
        If IBODTYP = 4
                       NPTU       Number of control points along U direction
                       NPTV       Number of control points along V direction
                        NDRU       Order along U direction
                        NDRV       Order along V direction
                        NTRIM      Number of trimming curves

     ----------------------------------
     Subblock 14d - (Depends upon IBODTYP)

     Subblock 14d.1 - Link descriptor
     If IBODTYP = 1     NPATCH records
     Record Line Format Variable          Description
        1     1  6E13    IPATCH           Patch Id number
                         IPATCHTYP = 9    Patch type
                         NPAT1            Patch point number
                         NPAT2            Patch point number

K7   ----------------------------------
```

```
K7   Subblock 14d.2 - Patch descriptor
K7   If IBODTYP = 2     NPATCH records
     Record Line Format Variable           Description
        1     1  6E13   IPATCH             Patch Id number
                        IPATCHTYP = 18     Patch type
                        NPAT1              Patch point number
                        NPAT2              Patch point number
                        NPAT3              Patch point number
                        NPAT4              Patch point number

       ----------------------------------
     Subblock 14d.3 - Control point coordinates
     If IBODTYP = 3     NPTU records
     Record Line Format Variable           Description
        Q     Q  3E13   XPT       \
                        YPT        }       Coordinates of control point
                        ZPT = 0   /

       ----------------------------------
     Subblock 14d.4 - Control point coordinates
     If IBODTYP = 4     NPTU * NPTV records
     Record Line Format Variable           Description
        1     1  3E13   XPT       \
                        YPT        }       Coordinates of control point
                        ZPT       /

       ----------------------------------
     Record 14e - (Depends upon IBODTYP)

     Subblock 14e.1 - Link/patch coordinates
     If IBODTYP = 1 or IBODTYP = 2     NPOINT records
     Record Line Format Variable           Description
        1     1   I13   IPPN              Patch point number
                 3E13   XPT(IPPN)  \
                        YPT(IPPN)   }      Coordinates of control point
                        ZPT(IPPN)  /

       ----------------------------------
     Subblock 14e.3 - Homogeneous coordinates
     If IBODTYP = 3 or IBODTYP = 4     NPTU * NPTV records
     Record Line Format Variable           Description
K7      1     1   E13   HCD(ℓ)            Homogeneous coordinates
```

```
K7    ----------------------------------
K7  Subblock 14f.3 - Knot vector
 .  Record 14F - (Only if IBODTYP = 3 or IBODTYP = 4)
 .               (NPTU + NORU) + (NPTV + NORV) records
 .  Record Line Format Variable          Description
 .     1     1   E13    KB(6)            Knot vector
 .
 .    ----------------------------------
 .  Subblock 14g.4 - Trimming curve information
 .  Record 14G - (Only if JBODTYP = 4 and NTRIM >0
 .                  (Trimming curve information got each trimming curve)
 .  Record Line Format Variable          Description
 .              5I13   NPTR              Number of points
 .                     NPTV = 0
 .                     NORV              Order of curve MARC used
 .                     NORV = 0
 .
 .    ----------------------------------
 .  Subblock 14h.4 - Points on trimming curve
 .                   (Only if IBODTYP = 4 and NTRIM > 0
 .  Record Line Format Variable          Description
 .   NPTR       2E13.6   XP              X coordinate in parametric space
K7                       YP              Y coordinate in parametric space

     ------------------------------------------------------------------------
    Block 15 - Begin Increment Indicator
               Block 10 is REQUIRED if incremental data is to appear.
               Number of records for binary : 1  Words/record: 1
               Number of lines for formatted: 1

    Record Line Format Variable  Description
       1     1  1A4    ISEE      1 character = 4H****

                                 Beginning of increment indicator.

   ------------------------------------------------------------------------
K7  Block 16 - Loadcase title
K7             Block 16 is only included if K7 or greater post revision.
 .             Number of records for binary : 1  Words/record: 70
 .             Number of lines for formatted: 1 line of 70 characters
 .
 .  Record Line Format Variable  Description
K7     1     1  170A1  TITLE(J)  J=1, 70, 70 character analysis title
```

```
 ------------------------------------------------------------------------
  Block 17 - Integer Increment verification data
           This block used for all analysis type except design sensitivity
             or design optimization.
             Number of records for binary : 1   Words/record: 6
             Number of lines for formatted: 1   6 I13 integers

  Record Line Format Variable  Description
     1     1  6I13   NEWCC     Switch to flag the existance of data for
                              blocks 2 to 13 following block 17 on the
                              incremental posttape.
                              NEWCC = 0  info not given
                              NEWCC = 1  rezone
                              NEWCC = 2  substructure
                    INC       Increment number
                    SUBINC    Subincrement number
                    JANTYP    = IANTYP + 100 if element quantities
                                        appear in this increment.
                              IANTYP = See Table 1, Block 2.
                              Note: This IANTYP is not necessarily
                                    equal to IANTYP in Block 2!
                    KNOD      Number of nodal postvalues in this incr.
                              = JNODE*NDEG  (See Table 1, Block 1)
                    IDMY1     reserved for future expansion

 ------------------------------------------------------------------------
K7  Block 17.ds - Integer Increment verification data
K7           This block is used for design sensitivity (JANTYP = 60)
 .           Number of records for binary : 1   Words/record: 6
 .           Number of lines for formatted: 1   6 I13 integers
 .
 .  Record Line Format Variable  Description
 .     1     1  6I13   NEWCC     Switch to flag the existance of data for
 .                              blocks 2 to 13 following block 17 on the
 .                              incremental posttape.
 .                              NEWCC = 0  info not given
 .                              NEWCC = 1  rezone
 .                              NEWCC = 2  substructure
 .                    INC       Increment number
 .                    SUBINC    Constraint reference number
 .                    JANTYP    = 60 (indicates sensitivity results)
 .                    KNOD      Not used
K7                   NDSVAR     Number of design variables
```

```
K7  Block 17.do - Integer Increment verification data
K7             Block used for design optimization.
               Number of records for binary : 1   Words/record: 6
               Number of lines for formatted: 1   6 I13 integers

    Record Line Format Variable  Description
       1     1  6I13   NEWCC     Switch to flag the existance of data for
                                  blocks 2 to 13 following block 17 on the
                                  incremental posttape.
                                  NEWCC = 0  info not given
                                  NEWCC = 1  rezone
                                  NEWCC = 2  substructure
                        INC       Increment number
                        SUBINC    Optimixation cycle number
                        JANTYP    61 (indicates optimization results)
                        IFLAG     2, 1, -1, -2
                                   2: best feasible design so far
                                   1: feasible desigm
                                  -1: infeasible design
                                  -2: best feasible design so far,
                                      infeasible design
K7                      NDSVAR    Number of design variables

    Block 18 - Real increment verification data
               Block 18 is used for all analysis except design sensitivity
               and design optimization.
               Number of records for binary : 1  Words/record: 6
               Number of lines for formatted: 1  6 E13.6 reals

    Record Line Format Variable  Description
       1     1  6E13.6 TIME      Transient time
                       FREQ      Frequency
                       GMAS      Generalized mass
                       DMY2      reserved for future expansion
                       DMY3      reserved for future expansion
                       DMY4      reserved for future expansion
```

```
K7  Block 18.ds - Real increment verification data
K7             This block is used for design sensitivity only.
               Number of records for binary : 1  Words/record: 6
               Number of lines for formatted: 1  6 E13.6 reals

    Record Line Format Variable  Description
       1     1  6E13.6 TIME      Transient time
                       FREQ      Frequency
                       GMAS      Generalized mass
                       RESPONSE  Sensitivity generated check value
                                 of the response
                       RSPLIM    Liniting value on the response
                       ENTRY     reserved for future expansion

    ------------------------------------------------------------------
    Block 18.do - Real increment verification data
               This block is used for design optimization.
               Number of records for binary : 1  Words/record: 6
               Number of lines for formatted: 1  6 E13.6 reals

    Record Line Format Variable  Description
       1     1  6E13.6 TIME      Transient time
                       FREQ      Frequency
                       GMAS      Generalized mass
                       OBJEC     Objective function value for this cycle
                       CONVAL    Normalized value of most
                                 critical constraint
K7                     ENTRY     reserved for future expansion

    ------------------------------------------------------------------
    Block 19 - New non-incremental data
               Block 19 is OPTIONAL.  Omitted if NEWCC = 0 (Block 17)

               If NEWCC is non-zero, repeat Blocks 2-13 here.
               (Read in new-non-incremental data.)

    ------------------------------------------------------------------
    Block 20 - Magnitude of distributed loads
               Block 20 is OPTIONAL.  Omitted if NDISTL = 0 (Block 2), or
               JANTYP = 60 or JANTYP = 61
               Number of records for binary : 1   Words/record: NDISTL
               Number of lines for formatted: (NDISTL-1)/6+1
```

```
 Record Line Format Variable  Description
    1       6E13.6  DLOAD(I)  I=1,NDISTL  Distributed load magnitude

-----------------------------------------------------------------------
 Block 21 - Magnitudes of spring forces
            Block 21 is OPTIONAL.  Omitted if NSPRNG = 0 (Block 2), or
            JANTYP = 60 or JANTYP = 61
            Number of records for binary : NSPRNG Words/record: 2
            Number of lines for formatted: NSPRNG

 Record Line Format Variable  Description
    J     J  2E13.6 REALF     Real spring force
                    CMPLXF    Complex spring force
   J=1,NSPRNG                 = 0.0 if ICOMPL = 0 (Block 2)

-----------------------------------------------------------------------
 Block 22 - Magnitudes of die forces
            Block 22 is OPTIONAL.  Omitted if NDIE = 0 (Block 2), or
            JANTYP = 60 or JANTYP = 61
            Number of records for binary : NDIE  Words/record: 18
            Number of lines for formatted: 3*NDIE
            Items pre line for formatted : 3 lines per die,
                                           6 E13.6 reals per line
            Number of lines per die      : 3

 Record Line Format Variable  Description
    J   6E13.6,6E13.6,6E13.6
                    XDIE(I)   I=1,18  Die variables for die J

                         I =  1  X position
                           =  2  Y position
                           =  3  Z position
                           =  4  ----------------------
                           =  5  ----------------------
                           =  6  Angular position about axis of rotation
                           =  7  X velocity
                           =  8  Y velocity
                           =  9  Z velocity
                           = 10  ----------------------
                           = 11  ----------------------
                           = 12  Angular velocity about axis of rotation
```

```
                              = 13  X force
                              = 14  Y force
                              = 15  Z force
                              = 16  Moment about x
                              = 17  Moment about y
                              = 18  Moment about z

  J=1,NDIE

-----------------------------------------------------------------------
 Block 23 - Values of element integration point variables
            Block is OPTIONAL.  Omitted if INUM = 0 (Block 2)
                                        or MNUM = 0 (Block 2)
                                        or JANTYP < 100 (Block 17)
            Number of records for binary : NSTRES*MNUM
            Words per record for binary  : INUM
            Number of lines for formatted: ((INUM-1)/6+1)*NSTRES*MNUM

 Record Line Format Variable  Description
    J     6E13.6/6E13.6/...
                    EVAR(I)   I=1,INUM  Element integration point
 J=1,NSTRES*MNUM                        variables

   NOTE: J points to the (MOD(J-1,NSTRES)+1)-th integration point
                 for the ((J-1)/NSTRES + 1)-th element.

  ALTERNATIVELY:  Let J be calculated by a double DO loop -
                  do K=1,MNUM
                   do L=1,NSTRES
            C       read Jth record where J=(K-1)*NSTRES+L
                    read (EVAR(I),I=1,INUM)
                   enddo
                  enddo

-----------------------------------------------------------------------
 Block 24 - Values of nodal variables
            Block 24 is OPTIONAL.  Omitted if KNOD = 0 (Block 17), or
            JANTYP = 60 or JANTYP = 61
            Number of records for binary : LNUM   Words/record: KNOD
            Number of lines for formatted: ((KNOD-1)/6+1)*LNUM

 Record Line Format Variable  Description
    J     6E13.6/...
  J=1,LNUM          DOFVAR(I) I=1,KNOD Values of nodal variables
```

```
      NOTE: There are JNODE=KNOD/NDEG nodal vectors given in this block.
            The vectors appearing here depend on IANTYP (Block 17).
            The vectors appear in the order given below.  Within each
             vector, the degrees of freedom appear in order from 1 to NDEG.
            For a particular node,the meaning of the degrees of freedom
            are given by the value of IDOF (Block 7).  Each node code
            indicates a particular number and sequence of degrees
            of freedom as given in Table 4 below.

   ------------------------------------------------------------------------
K7  Block 25 - Response gradients
K7              Block 25 is OPTIONAL.  Used only if JANTYP = 60.
 .              Number of records for binary : 1   Words/record: NDSVAR
 .              Number of lines for formatted: (NDSVAR)/6+1
 .
 .
 .   Record Line Format Variable  Description
 .      J     6E13.6/...
 .                      RESGRD(I) I=1,NDSVAR Values of response gradient
 .
 .
 .  ------------------------------------------------------------------------
 .   Block 26 - Element contribution to the response
 .              Block 26 is OPTIONAL.  Used only if JANTYP = 60.
 .              Number of records for binary : 1   Words/record: NUMEL
 .              Number of lines for formatted: (NUMEL-1)/6+1
 .
 .
 .   Record Line Format Variable  Description
 .      J     6E13.6/...
 .                      ELMCRES(I) I=1,NUMEL Values of element contribution
 .
 .
 .  ------------------------------------------------------------------------
 .   Block 27 - Design variables for this cycle
 .              Block 27 is OPTIONAL.  Used only if JANTYP = 61.
 .              Number of records for binary : 1   Words/record: NDSVAR
 .              Number of lines for formatted: (NDSVAR-1)/6+1
 .
 .
 .   Record Line Format Variable  Description
 .      J     6E13.6/...
K7                      DESVAR(I) I=1,NDSVAR Values of design variables
   ------------------------------------------------------------------------
```

Table 1 - IANTYP Index
Index to a description of the JNODE quantities for each IANTYP

Key to index

```
                      d : displacements
                     xf : external applied forces
                      r : reaction forces
                      v : velocities
                      a : accelerations
                      f : fluxes
                     nv : nodal voltages
                     id : imaginary part of complex displacements
                     rd : real part of complex displacements
                    ixf : imaginary part of complex xtrnl forces
                    rxf : real part of complex xtrnl forces
                     ir : imaginary part of complex reactions
                     rr : real part of complex reactions
                     mf : mass fluxes
                      t : temperatures
                     fx : force in x direction
                     fy : force in y direction
                     fz : force in z direction
                      p : pressure
                      m : modal mass
                    psi : magnetic potential
                      J : current
                    phi : electric potential
                    rho : charge
                      s : acoustic source
                     pp : pore pressure
                     fr : friction force
                    xfl : externaly applied flux
                   rfld : reaction force in fluid
```

IANTYP/JNODE	1	2	3	4	5	6	7	8	9	10
1	d	xf								
2	d	xf	r							
3	d	v	a	xf						
4	d	v	a	xf	r					
5	d	xf	v							
6	d	xf	v	r						
7	t									
8	t	f								

```
 9          t    nv
10          t    nv   f
11          d
12          d    r
13
14
15          rd   id
16          rd   id   rr   ir
17
18
19          d    xf   t
20          d    xf   t    r    f
21          d    v    a    xf   t
22          d    v    a    xf   t    r    f
23          d    xf   t    nv
24          d    xf   t    nv   r    f
25          d    v    a    xf   t    nv
26          d    v    a    xf   t    nv   r    f
27          d    m
28          d
29          p    fx   fy   fz   mf
30          d    r
31          d    m
32          d    m    r
33
34
35
36
37          xf
38
39          d
40          d    r
41          d    xf   t    v
42          d    v    t    v    r    f
43          phi  rho
45          psi  J
47          p    s
48          psi  J
49          id   rd   ixf  rxf
50          pp
51          d    xf   r
51
```

```
  52
  53
  54          d    xf   r    fr
  55          d    v    a    xf   f    fr
  56          d    xf   v    r    fr
  57          d    xf   t    r    f    fr
  58          d    v    a    xf   t    r    f    fr
  59          d    xf   t    v    r    f    fr
  60
  61
  62
  63          v    p    xf   f
  64          v    xf   r
  65          v    p    t    xf   xfl  r    f
  66          v    t    xf   xfl  r    f
  67          v    p    t    xf   xfl  r
  68          v    t    xf   xfl  r
  69          d    v    p    xf   r    rfld
  70          d    v    xf   r    rfld
  71          d    v    p    t    xf   xfl  r    rfld f
  72          d    v    t    xf   xfl  r    rfld f
  73          d    v    p    t    xf   xfl  f    rfld
  74          d    v    t    xf   xfl  r    rfld
-------------------------------------------------------------------
```

Table 4 - Degrees of freedom as a function of node code

```
IDOF  Degrees of freedom
      1    2    3    4    5    6    7    8    9   10   11   12
-------------------------------------------------------------------

 0    NOT CONNECTED
 1    T
 2    U    V
 3    U    V    W
 4    U    V    PH
 5    U    U,1  U,2  V    V,1  V,2  W    W,1  W,2  U,12 V,12 W,12
 6    U    U,1  U,2  V    V,1  V,2  W    W,1  W,2
 7    U    V
 8    L1   L2
 9    U    U,1  V    V,1  W    W,1  PH   PH,1
10    U    V    W    PHX  PHY  PHZ
```

```
11    U    V    U,1  V,1
12    U    V    DW
13    PHX  PHY
14    U    V    PHX
15    U    V    W    PH1  PH2
16    U,N  V,N  W,N
17    U    V    L1
18    U    V    W    L1
19    U    V    PH1  PH2
20
21    U    V    PH1  L1
22    PH1
23    PH2
24    U    V    W    PH1  PH2  PH3  PH4
25    U    V    W    PH1  PH2  PH3  V,1
26    U    V    W    PH1  PH2  PH12
27    U    V    PH
28    T1   T2
29    T1   T2   T3
30    U1   V1   U2   V2   PH
31    U1   V1   U2   V2
32    U    V     W   PH1  PH2
33    U    V     W
```

```
INDEX TO ELEMENT TYPE

IDOF    Element types associated with IDOF

 0      No element type associated
 1      All heat transfer continuum elements
 2      All structural planar or axisymmetric continuum elements
 3      All structural solid elements
 4        1   5  45  89
 5        4
 6        8  24
 7       12  80  81  82  83  84  97 118 119 120
 8       12  97
 9       13
 10      14  22  25  31  52  75  76  78  98 138 139 140
 11      15  16  17
 12      19  29  34  47  56  60  81
```

```
13      19  29  34  47  56  60  81
14      20  67  83
15
16      24
17      32  33  34  58  59  60 128 129
18      35  61 130
19
20      62  63  66  67  73  74
21      63  66  74
22      49  72  76  77
23
24      77  79
25      25
26
27
28      85  86  87  88
29      85  87  86  88
30      95  96  86  88
31      97
32      90
33     111 112 113
```

A

User Subroutines and Special Routines

Subroutines	Page
ANELAS	3-4
ANEXP	3-19
ANKOND	3-21
ANPLAS	3-14
ASSOC	4-10
CREDE	2-30
CRPLAW	3-28
CRPLAW	4-4
CRPVIS.	5-2
CUPFLX	2-10
DIGEOM	2-44
ELEVAR	7-16
ELEVEC	7-18
FILM	2-20
FLOW	2-22
FLUX	2-8
FORCDF	2-18
FORCDT	2-14
FORCEM	2-4
GAPT.	2-24
GAPU	3-75
GENSTR	3-67
HOOKLW	3-12
HOOKVI	5-7
HYPELA	3-56
HYPELA2	3-58

Subroutines	Page
IMPD	7-7
INITPL	2-59
INITPO	2-60
INITSV	2-32
INTCRD	7-20
MAP2D	6-5
MOTION (2D)	2-38
MOTION (3D)	2-40
NASSOC	4-6
NEWPO	2-61
NEWSV	2-33
ORIENT	3-17
PLOTV	7-2
REBAR	6-8
SEPFOR	2-45
SEPSTR	2-47
SINCER	4-11
SSTRAN	6-15
TENSOF	3-42
TRSFAC	5-5
UACTIVE	6-7
UACTUAT	6-17
UADAP	6-12
UBEAM	3-69
UBEAR	8-2
UBGINC	7-21
UBGITR	7-23
UCOMPL	3-73
UCONTACT	2-56

Subroutines	Page
UCOORD	6-11
UCRACK	3-40
UEDINC	7-22
UELDAM	3-54
UELOOP	7-24
UENERG	3-51
UEPS.	3-23
UFAIL	3-16
UFCONN	6-4
UFINITE	3-62
UFORMS	2-25
UFOUR.	2-12
UFRIC	2-42
UFRORD	6-9
UFXORD	6-3
UGROOV	8-3
UHTCOE	2-48
UHTCON	2-51
UINSTR.	2-11
UMOONY	3-50
UMU	3-24
UNEWTN	3-80
UNORST	2-53
UOGDEN	3-52
UPERM.	3-49
UPHI	3-71
UPNOD.	6-6
UPOSTV	7-4
UPOWDR	3-47
UPSTRECH	3-64
URCONN	6-10
URESTR	8-4
URPFLO	3-81
USDATA	2-36
USELEM	3-77
USHELL	6-14
USHRET	3-43

Subroutines	Page
USIGMA	3-26
USINC	2-35
USPCHT	3-27
USPRNG	3-38
USSD	2-34
UTHICK.	6-16
UTHICK.	8-5
UTRANS	6-13
UVELOC	2-37
UVELOC	8-6
UVOID	3-44
UVOIDN	3-45
UVSCPL	4-2
VSWELL	3-33
WKSLP	3-35
YIEL	4-9
ZERO	4-8

Special Routines	Page
PLDUMP	9-2

MARC
Analysis
Research
Corporation

Document Title: **MARC Volume D: User Subroutines and Special Routines K7**
Part Number: RF-3004-07
Revision Date: August, 1997

CALIFORNIA - Palo Alto
MARC Corporate Headquarters
260 Sheridan Avenue, Suite 309
Palo Alto, CA 94306, USA
Tel: 1 650 329 6800
Fax: 1 650 323 5892
Email: support@marc.com

CALIFORNIA - San Diego
MARC Analysis Research Corp.
4330 LaJolla Village Drive, Suite 320
San Diego, CA 92122
Tel: 1 619 658 9588
Fax: 1 619 587 8710

CONNECTICUT - E. Berlin
MARC Analysis Research Corp.
1224 Mill Street
E. Berlin, CT 06023
Tel: 1 860 828-2055
Fax: 1 860 828 2031

MICHIGAN - Ann Arbor
MARC Analysis Research Corp.
24 Frank Lloyd Wright Drive
Ann Arbor, MI 48106-0523
Tel: 1 313 998 0540
Fax: 1 313 998 0542

CZECH REPUBLIC
MARC Overseas, Inc.
Podolska 50
147 00 Praha 4
Czech Republic
Tel: 011 420 (2) 61 21 41 23
Mobil: 011 420 (602) 21 23 26
Fax: 011 420 (2) 61 21 41 23

FRANCE
MARC France S.A.
30, Rue de Rocroy
94100 Saint-Maur Des Fossés (near Paris)
France
Tel: 011 33 (1) 41 81 1 0 94
Fax: 011 33 (1) 41 81 1 0 95

GERMANY - Munich
MARC Software Deutschland GmbH
Ismaninger Str. 9
85609 Aschheim (bie Munich)
Germany
Tel: 011 49 (89) 904 50 33
Fax: 011 49 (89) 903 06 76
Email: system@marc.co.de

GERMANY - Hannover
MARC Software Deutschland Gmbh
Alte Dohrener Str. 66
30173 Hannover, Germany
Tel: 011 49 (511) 980 5182
Fax: 011 49 (511) 980 5187

ITALY
Espri-MARC s.r.l.
16129 Viale Brigata Bisagno 2/10
16129 Genova
Italy
Tel: 011 39 (10) 585 949
Fax: 011 39 (10) 585 949

THE NETHERLANDS
MARC Analysis Research Corporation
Dublinstraat 32
2713 HS Zoetermeer
The Netherlands
Tel: 011 31 (79) 3510 411
Fax: 011 31 (79) 3517 560
Email: support@marc.nl

UNITED KINGDOM
MARC UK Ltd.
35, Shenley Pavilions
Chalkdell Drive
Shenley Wood
Milton Keynes, MK56LB
Tel: 011 44 (1908) 506 505
Fax: 011 44 (1908) 506 522

CHINA
MARC Overseas, Inc.
Bright Chang An Building, Suite 321
7# Jian Guo Men Nei Street
Beijing, China 100 005
Tel: 011 86 (10) 6510 2056
011 86 (10) 6510 2057
011 86 (10) 6510 2058
Fax: 011 86 (10) 6510 2053

JAPAN - Nagoya
Nippon MARC Co., Ltd.
Nagoya Nishiki Daiichi Seimei Bldg.
2F, 6-5 Nishiki 1-chome
Naka-ku, Nagoya, Aichi 460
Tel: 011 81 (52) 202 5661
Fax: 011 81 (52) 202-5665

JAPAN - Osaka
Nippon MARC Co., Ltd.
4F 2nd Kimi Building
2-11 Toyotucho Suita-shi
Osaka 564
Tel: 011 81 (6) 385 1101
Fax: 011 81 (6) 385-4343

JAPAN - Tokyo
(Asian Operations Headquarters)
Nippon MARC Co., Ltd.
Shinjuku Daiichi Seimei Building
P.O. Box 5056
2-7-1 Nishi-Shinjuku, Shinjuku-ku
Tokyo 163
Tel: 011 81 (3) 3345 0181
Fax: 011 81 (3) 3345 1529
Email: system@marc.co.jp

KOREA
Dong Kyung Bldg. 7FL
824-19 Yuksam-Dong
Kangnam-Ku
Seoul, Korea
Tel: 011 82 (2) 561 7543
Fax: 011 82 (2) 561 7767