SIEMENS

Multi-Step Nonlinear User's Guide

Contents

Figures

Proprietary & Restricted Rights Notice

TAUCS Copyright and License

TAUCS Version 2.0, November 29, 2001. Copyright (c) 2001, 2002, 2003 by Sivan Toledo, Tel-Aviv University, stoledo@tau.ac.il. All Rights Reserved.

TAUCS License:

Your use or distribution of TAUCS or any derivative code implies that you agree to this License.

THIS MATERIAL IS PROVIDED AS IS, WITH ABSOLUTELY NO WARRANTY EXPRESSED OR IMPLIED. ANY USE IS AT YOUR OWN RISK.

Permission is hereby granted to use or copy this program, provided that the Copyright, this License, and the Availability of the original version is retained on all copies. User documentation of any code that uses this code or any derivative code must cite the Copyright, this License, the Availability note, and "Used by permission." If this code or any derivative code is accessible from within MATLAB, then typing "help taucs" must cite the Copyright, and "type taucs" must also cite this License and the Availability note. Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. This software is provided to you free of charge.

Availability (TAUCS)

As of version 2.1, we distribute the code in 4 formats: zip and tarred-gzipped (tgz), with or without binaries for external libraries. The bundled external libraries should allow you to build the test programs on Linux, Windows, and MacOS X without installing additional software. We recommend that you download the full distributions, and then perhaps replace the bundled libraries by higher performance ones (e.g., with a BLAS library that is specifically optimized for your machine). If you want to conserve bandwidth and you want to install the required libraries yourself, download the lean distributions. The zip and tgz files are identical, except that on Linux, Unix, and MacOS, unpacking the tgz file ensures that the configure script is marked as executable (unpack with tar zxvpf), otherwise you will have to change its permissions manually.

Chapter 1: Introduction

1.1 Overview of nonlinear capabilities

This book covers nonlinear structural analysis with the solution sequence, SOL 401 - NLSTEP. SOL 401 is a multistep, structural solution which supports a combination of static (linear or nonlinear) subcases and modal (real eigenvalue) subcases.

SOL 401 is the structural solution used by the NX Multiphysics environment within the NX Advanced Simulation product. The NX Multiphysics environment supports all combinations of structural-to-thermal and thermal-to-structural coupling with the NX Thermal solution. SOL 401 is also supported as a stand-alone NX Nastran solution.

Primary operations for nonlinear elements are updating element coordinates and applied loads for large displacements. The geometric nonlinearity becomes discernible when the structure is subjected to large displacement and rotation. Geometric nonlinear effects are prominent in two different aspects: geometric stiffening due to initial displacements and stresses, and follower forces due to a change in loads as a function of displacements. The large deformation effect resulting in large strains has not been implemented.

Material nonlinearity is an inherent property of any engineering material. Material nonlinear effects may be classified into many categories. Included are plasticity, nonlinear elasticity, creep, and viscoelasticity. SOL 401 supports plasticity and creep.

The primary solution operations are time increments, iterations with convergence tests for acceptable equilibrium error, and stiffness matrix updates. The iterative process is based on variations of Newton's method. The stiffness matrix updates are performed to improve the computational efficiency, but may be overridden at your discretion.

1.2 Program architecture

The software has a modular structure to separate functional capabilities which are organized under an efficient executive system. The program is divided into a series of independent subprograms, called functional modules. A functional module is capable of performing a pre-defined subset of operations. It is the Executive System that identifies every module to execute by MPL (Module Properties List).

The Executive System processes the input data by IFP (Input File Processor) and the general initialization, which are known as Preface,operations. It then establishes and controls the sequence of module executions in the OSCAR (Operation Sequence Control Array) based on the user-specified DMAP (Direct Matrix Abstraction Program) or solution sequence. The Executive System allocates system files to the data blocks in the FIAT (File Allocation Table) and maintains a parameter table for module interface. The Executive System is also responsible for the database management and all the input and output operations by GINO (General Input/Output Routines).

The functional module consists of a number of subroutines. Modules communicate with each other only through secondary storage files, called data blocks (matrix or table). Each module performs a certain function with input data blocks and produces output data blocks. A module may communicate

with the Executive System and with other modules through parameters, which may be input and/or output variables of the module. Modules utilize main memory dynamically. If the size of the main memory is insufficient to complete an operation, the module uses scratch files, which reside in the secondary storage as an extension of the main memory. This is known as a spill operation.

DMAP is a kind of macro program using a data block oriented language. The solution sequence is a collection of module statements written in the DMAP language tailored to process a sequential series of operations, resulting in a specific type of structural analysis. A typical solution sequence consists of three phases of functional operations: formation, assembly, and reduction of matrices; solution of equations; and data recovery. Solution sequences that process superelements have built-in superelement loops in the first and the last phases.

The nonlinear solution sequences have built-in loops in the second phase for subcase changes, load increments, and stiffness matrix updates. Nested in this DMAP loop, nonlinear solution processes comprise a number of internal iteration loops. Confining the discussion to SOL 401, the hierarchy of the nonlinear looping is shown in the table below. Central to the nonlinear processes is module NLTRD3. The module is self-contained to perform iterations for converged solutions.

Table 1-1. Hierarchy of Nonlinear Looping

	Name or Loop	Type
1	Subcases (boundaries, temperatures, loads, outputs)	DMAP Control
2	Time Steps (NLTRD3)	Module Control
3	Stiffness Matrix Updates	The actual stiffness update is under DMAP control, but the request for a stiffness update in the middle of a solution is under Module control. Decomposition is under module control.
4	Iterations (Vector Arithmetic)	Module Control
5	Elements (NLEMG)	Subroutine Control
6	Volume Integration (Gauss Points)	Subroutine Control

1.3 Rudiments of the user interface

The input data structure, as was designed originally, still has the most popular format consisting of an optional header, executive control section, case control section, and the bulk data section. All the features and principles for the user interface are common in the software. Needless to say, all the features of the user interface for nonlinear analysis are compatible with those for the linear analysis. Any exceptions for nonlinear analysis will be explained in the relevant chapters that follow.

Mechanical design is dictated by the strength, dynamic, and stability characteristics of the structure. The software provides the analysis capabilities of these characteristics with solution sequences, each of which is designed for specific applications. The type of desired analysis is specified in the executive control section by using a solution sequence identification. SOL 401 is designed for static and quasi-static.

The basic input data required for a finite element analysis may be classified as follows:

- Geometric data

- Element data
- Material data
- Boundary conditions and constraints
- Loads and enforced motions
- Solution methods

The first three classes of data may not be changed during the course of an analysis whereas the last three classes of data may be changed in midcourse via subcases under the case control section.

1.3.1 Case control section

The primary purpose of the case control is to define subcases. The subcase structure provides a unique means of changing loads, boundary conditions, and solution methods by making selections from the bulk data. Confining the discussion to SOL 401, loads and solution methods may change from subcase to subcase on an incremental basis. Constraints can be changed from subcase to subcase. As a result, the subcase structure determines a sequence of loading and constraint paths in a nonlinear analysis. The subcase structure also allows the user to select and change output requests for printout, etc., by specifying set numbers with keywords. Any selections made above the subcase specifications are applicable to all the subcases. Selections made in an individual subcase supersede the selections made above the subcases. The table below summarizes the case control data for nonlinear analysis.

Table 1-2. Summary of Case Control

ADAPTERR	GPFORCE	OPRESS
ANALYSIS	GPKE	OTEMP
BCRESULTS	GPLSTRN	PARAM
BCTSET	GROUNDCHECK	PLSTRN
BEGIN BULK	GSTRAIN	SEQDEP
BGRESULTS	GSTRESS	SET
BGSET	GTHSTRN	SMETHOD
BOLTLD	INCLUDE	SPC
CRSTRN	JINTEG	SPCFORCES
DISPLACEMENT	LABEL	STRAIN
DLOAD	LINE	STRESS
DTEMP	MAXLINES	SUBCASE
ECHO	MEFFMASS	SUBTITLE
EKE	METHOD	TEMPERATURE
ELSTRN	MPC	THSTRN
ELSUM	MPCFORCES	TITLE
ESE	NLCNTL	TSTEP
FORCE	NSM	WEIGHTCHECK
GCRSTRN	OLOAD	
GELSTRN	OMODES	

1.3.2 Bulk data section

Most of the input data is specified in the bulk data section. Some of the bulk data may not be used during a program execution if they are not selected with the case control data. The SOL 401 bulk data is summarized in the following table. The input data specially designed for nonlinear analysis is described in the relevant chapters that follow.

ACCEL	CORD3G	FORCE1	PSOLID
ACCEL1	CPENTA	FORCE2	RBAR
BCRPARA	CPLSTN3	GRAV	RBE2
BCTPARM	CPLSTN4	GRDSET	RBE3
BCTSET	CPLSTN6	GRID	RFORCE
BEDGE	CPLSTN8	GROUP	RFORCE1
BGADD	CPLSTS3	INCLUDE	SLOAD
BGPARM	CPLSTS4	MAT1	SPC
BGSET	CPLSTS6	MAT11	SPC1
BOLT	CPLSTS8	MAT9	SPCADD
BOLTFOR	CPYRAM	MATCID	SPCD
BOLTLD	CQUADX4	MATCRP	SPOINT
BSURFS	CQUADX8	MATFT	TABLED1
CHEXA	CRAKTP	MATS1	TABLED2
CMASS1	CTETRA	MATT1	TABLED3
CMASS2	CTRAX3	MATT11	TABLED4
CMASS3	CTRAX6	MATT9	TABLEM1
CMASS4	DAREA	MPC	TABLEM2
CONM1	DLOAD	MPCADD	TABLEM3
CONM2	DTEMP	NLCNTL	TABLEM4
CORD1C	DTEMPEX	PARAM	TEMP
CORD1R	ECHOOFF	PCOMPS	TEMPD
CORD1S	ECHOON	PLOAD	TEMPEX
CORD2C	EIGRL	PLOAD4	TLOAD1
CORD2R	ENDDATA	PLOTEL	TSTEP1
CORD2S	FORCE	PMASS	VCEV

1.3.3 Parameters

Parameters are used for requesting special features or specifying miscellaneous data. Parameters are initialized in the MPL, which can be overridden by a DMAP initialization. Modules may change the parameter values while the program is running.

There are two different types of parameters: user parameters (V,Y,name in the DMAP) and DMAP (non-user) parameters. For the user parameters, users are authorized to change the initial values by specifying PARAM data in the bulk data section or occasionally in the case control section. The following table lists the parameters supported in SOL 401.

COLPHEXA	OMAXR	RGBEAMA
COUPMASS	OMPT	RGBEAME

F56	OPG	RGLCRIT
GRDPNT	OUGCORD	RGSPRGK
LGDISP	POST	UNITSYS
MATNL	POSTEXT	TINY
MAXRATIO	POSTOPT	WTMASS
NOFISR	PRGPST	
OGEOM	PROUT	

1.4 Nonlinear characteristics and general recommendations

The modeling guidelines for nonlinear analysis and linear analysis are summarized as follows:

- The analyst should have some insight into the behavior of the structure to be modeled; otherwise, a simple model should be the starting point.
- The size of the model should be determined based on the purpose of the analysis, the trade-offs between accuracy and efficiency, and the scheduled deadline.
- Prior contemplation of the geometric modeling will increase efficiency in the long run. Factors to be considered include selection of coordinate systems, symmetric considerations for simplification, and systematic numbering of nodal points and elements for easy classification of locality.
- Discretization should be based on the anticipated stress gradient, i.e., a finer mesh in the area of stress concentrations.
- Element types and the mesh size should be judiciously chosen. For example, avoid highly distorted and/or stretched elements (with high aspect ratio).
- The model should be verified prior to the analysis by some visual means, such as plots and graphic displays.

Nonlinear analysis requires better insight into structural behavior. First of all, the type of nonlinearities involved must be determined. The geometric nonlinearity is characterized by large rotations which usually cause large displacements. Intuitively, geometric nonlinear effects should be significant if the deformed shape of the structure appears distinctive from the original geometry without amplifying the displacements. There is no distinct limit for large displacements because geometric nonlinear effects are related to the dimensions of the structure and the boundary conditions. The key to this issue is to know where the loading point is in the load-deflection curve of the critical area.

Additional recommendations are important for nonlinear analysis:

- PARAM,LGDISP,1 must be defined to turn on geometry nonlinearity.
- Material nonlinear effects can also be included. See Support for plasticity analysis and Support for creep analysis.
- The nonlinear region usually requires a finer mesh. Use a finer mesh if severe element distortions or stress concentrations are anticipated.

- The subcase structure should be utilized properly to divide the load or time history for conveniences in data recovery, and database storage control, not to mention changing constraints and loading paths.

- Many options are available in solution methods to be specified on the NLCNTL and the TSTEP1 bulk entries. The defaults should be used on all options before gaining experience.

Chapter 2: User Interface

2.1 Element and material summary

The following is a summary of the elements and materials supported in SOL 401. See Finite elements in nonlinear analysis.

- The 3D solids elements CTETRA, CHEXA, CPENTA and CPYRAM elements are supported for linear, geometric nonlinear, and material nonlinear analysis.
- The axisymmetric elements CQUADX4, CQUADX8, CTRAX3, CTRAX6, the plane strain elements CPLSTN3, CPLSTN4, CPLSTN6, CPLSTN8, and the plane stress elements CPLSTS3, CPLSTS4, CPLSTS6, CPLSTS8 are supported for linear, geometric nonlinear, and material nonlinear analysis.

  The grid points on these elements must all lie in either the XZ plane, or all in the XY plane of the basic coordinate system. The software automatically determines the orientation.

  When axisymmetric elements are defined on the XZ plane, X is the radial direction, and Z is the axial direction. The grid points defining these elements must have $X \geq 0$.

  When axisymmetric elements are defined on the XY plane, Y is the radial direction, and X is the axial direction. The grid points defining these elements must have $Y \geq 0$.
- A special, generalized plane strain formulation is available using the CPLSTN3, CPLSTN4, CPLSTN6, and CPLSTN8 element types. See Generalized plane strain analysis.
- The RBE2 and RBAR rigid elements are supported with optional large displacement effects and thermal expansion. The RBE3 rigid element is also supported, but it does not support the large displacement effects or thermal expansion. See Rigid element support.
- The mass elements CMASSi and CONMi are supported.
- The PSOLID or the PCOMPS bulk entries define the element properties. The PCOMPS is optionally used to define a layered solid composite property.

  The PGPLSN bulk entry is defines the special generalized plane strain element properties.
- The supported material types include the following.

  The MAT1 and MATT1 (temperature dependent) bulk entries define isotropic materials.

  The MAT3 and MATT3 (temperature dependent) bulk entries define isotropic materials.

  The MAT9 and MATT9 (temperature dependent) bulk entries define anisotropic materials.

  The MAT11 and MATT11 (temperature dependent) bulk entries define orthotropic materials.

  Plastic and creep materials can optionally be assigned to the 3D solid elements, axisymmetric elements, the plane stress elements, and the plane strain elements. You can enable one or both plasticity/creep in all subcases, or in specific subcases.

See Support for plasticity analysis and Support for creep analysis.

- You can request stress norm, stress error norm, strain energy norm, and strain energy error norm output. The output is computed and stored on an individual element basis. The NX Advanced Simulation product uses the output for adaptive meshing. See Error estimator for mesh refinement.

- You can compute and output the j-integral in a crack simulation. The j-integral output can be requested and used by third-party software like Zencrack to perform a fracture mechanics analysis. The CHEXA bulk entry allows for a collapsed element definition. See Crack simulation.

2.2 Subcase analysis type

SOL 401 allows any combination of static subcases and modal (real eigenvalue) subcases. The ANALYSIS case control command defines the subcase analysis type.

- Modal subcase: You include ANALYSIS=MODAL in a subcase.

- Static subcase: You include ANALYSIS=STATIC in a subcase.

The ANALYSIS case control command does not have a default in SOL 401. You must define it in every subcase, and it cannot be defined above the subcases (globally). The first subcase can be either a static or a modal subcase.

The modal subcase should include the METHOD case control command which selects the EIGRL bulk entry. The EIGRL entry defines the data needed to perform the real eigenvalue analysis with the Lanczos method. The modal subcase automatically includes the stress stiffening from the previous static subcase, and can potentially include follower stiffness and spin softening depending on the type of loading in the previous static subcase. The NLCNTL bulk entry has parameter inputs which allow you to control the stiffness contributions for the modal subcase.

2.3 Nonlinear Effects

The parameter LGDISP turns the nonlinear large displacement capability on/off for the static subcases. If you define the parameter LGDISP for SOL 401, you must include it in the bulk data portion of your input file. The single PARAM,LGDISP setting applies to all static subcases.

- PARAM,LGDISP,-1 (default) – Large displacement effects are turned off. Subcases which include ANALYSIS=STATIC are linear static subcases.

- PARAM,LGDISP,1 – Large displacement effects are turned on. Subcases which include ANALYSIS=STATIC are nonlinear static subcases.

PARAM,LGDISP,1 turns on large displacement effects, but small strains are assumed.

Material nonlinear effects can also be included. See Support for plasticity analysis and Support for creep analysis.

2.4 Subcase sequencing

You can use the SEQDEP case control command to define a static or modal subcase as sequentially dependent (SD), or non-sequentially dependent (NSD).

- SEQDEP=YES (default) – the static or modal subcase is a SD subcase.

  SOL 401 uses time as the variable to increment temperatures and loads. An SD static subcase uses the final time from the previous static subcase for its start time. The start time is used to compute the solution time steps in a static subcase. See Defining Solution Time Steps. A SD subcase also receives the final state variables from the previous static subcase. For example, plastic strains, creep strains, and displacements.

- SEQDEP=NO – the static or modal subcase is a NSD subcase.

  A NSD subcase is independent. The start time for a static NSD subcase is 0.0. See Defining Solution Time Steps.

  A NSD static or modal subcase does not use any data from a previous static subcase, regardless of the parameter settings on the NLCNTL bulk entry.

- Time does not apply to modal subcases.

2.5 Defining solution time steps

Loads are defined in SOL 401 as a function of time. SOL 401 is a static solution, and time is only used as the mechanism to increment loads. The TSTEP1 bulk entry defines the time step intervals in which a solution will be generated and output in a static subcase. You include the TSTEP case control command in the static subcase to select a specific TSTEP1 definition in the bulk data.

The TSTEP1 entry includes the end times ($Tend_i$), the number of increments ($Ninc_i$), and the increment for computing output ($Nout_i$). The start time for a particular subcase depends if it is sequentially dependent (SD) or not sequentially dependent (NSD).

1	2	3	4	5	6	7	8	9	10
TSTEP1	SID	Tend1	Ninc1	Nout1					
		Tend2	Ninc2	Nout2					
		Tend3	Ninc3	Nout3					
		-etc-							

TSTEP1 Input Example:

TSTEP1	1	10.0	5	2					
		50.0	4	3					
		100	2	ALL					

In this example, assuming a start time=0.0 for the subcase, the resulting time steps are as follows. The time steps in which output occurs are highlighted. Output always occurs at the end time.

The 1st row has an end time of 10.0, 5 increments, and output at every 2nd time step.

Time Step 1	Time Step 2	Time Step 3	Time Step 4	Time Step 5
2.0	4.0	6.0	8.0	10.0

The 2nd row has an end time of 50.0, 4 increments, and output frequency at every 3rd time step.

Time Step 1	Time Step 2	Time Step 3	Time Step 4
20.0	30.0	40.0	50.0

The 3rd row has an end time of 100.0, 2 increments, and output at all time steps.

Time Step 1	Time Step 2
75.0	100.0

In the same example, assuming a start time=5.0 for the subcase, the resulting time steps for the first row are as follows.

Time Step 1	Time Step 2	Time Step 3	Time Step 4	Time Step 5
6.0	7.0	8.0	9.0	10.0

The 2nd and 3rd row are the same:

Time Step 1	Time Step 2	Time Step 3	Time Step 4
20.0	30.0	40.0	50.0

Time Step 1	Time Step 2
75.0	100.0

Additional Information about TSTEP1:

- Output always occurs at $Tend_i$.
- $Tend_i$ must be increasing ($Tend_i < Tend_{i+1}$).
- When Tend1=0.0,
  - No other times are allowed. This is the only time for the associated subcase.
  - $Ninc_i$ can be defined.
  - $Nout_i$ is ignored.
  - Results are output at time = 0.0.
- $Nout_i$ controls the frequency of results output. The table below summarizes the input options.

Nout	Output frequency
YES	Output occurs at all increments defined on TSTEP1.
END	Output occurs at the end time.

ALL	Output occurs at all increments on TSTEP1 and any software subincrements. Note: When Nouti=ALL in the context of a NX Multiphysics solution, the result output time steps will be a combination of the structural output steps as well as the coupled time steps.
Integer ≥ 0	Output is computed at every Nout increment specific with TSTEP1.
CPLD	Output occurs only at coupling times. This option can only be defined by the NX Multiphysics environment.

- The start time (T_{start}) for a static subcase is determined as follows:
  - If a static subcase definition in the case control includes SEQDEP=NO, that subcase is not sequentially dependent (NSD). The start time for an NSD subcase is 0.0.
  - For a sequentially dependent (SD) static subcase (default), the final Tendi from a previous SD or NSD static subcase is the start time (T_{start}) for the current SD subcase. If an SD subcase has no previous SD or NSD static subcases, the start time is 0.0 for that SD subcase, and Tend1=0.0 is permitted. Otherwise, Tend1 > T_{start} for all other SD subcases.
- If a creep material is included, the software uses adaptive time stepping by default. The adaptive time stepping can result in additional solution time steps which are not defined by the TSTEP1 entry. See Support for creep analysis.

  Similarily, when running SOL 401 in the context of the NX Multiphysics environment, additional solution time steps beyond what is defined on the TSTEP1 entries are possible.

  For both of these cases, the Nout field on the TSTEP1 bulk entry still determines the frequency of results output.

2.6 Loads overview

The solution strategy in nonlinear is to apply the loads in an incremental fashion until the desired load level is reached. The algorithms remember the loads from one subcase to the next.

The methods employed to define loads in SOL 401 are similar to those used in a time history solutions. A single degree-of-freedom or a set of GRID points may be loaded with force pattern that varies with time. Functions may be tabular such as an earthquake or a booster liftoff, or they may be simple analytic functions such as a sine wave. Simple static load sets may be used to create the dynamic loads. They may be scaled and combined with other loads to simulate complex loading problems.

Time history loads define the loadings as functions of time and the location. They can be a load applied at a particular degree-of-freedom, pressure over the surface area, or the body force simulating an acceleration. The time history is provided by TLOADi bulk entry.

2.6.1 Mechanical loads

Mechanical loads can be defined in SOL 401 as time-assigned or time-unassigned. SOL 401 is a static solution, and time is only used as the mechanism to increment loads. Time-assigned and time-unassigned loads can be combined in the same static subcase.

- Load selection in Case Control:

- Time-unassigned loads are selected with the LOAD case control command,

  ```
  LOAD=n
  ```

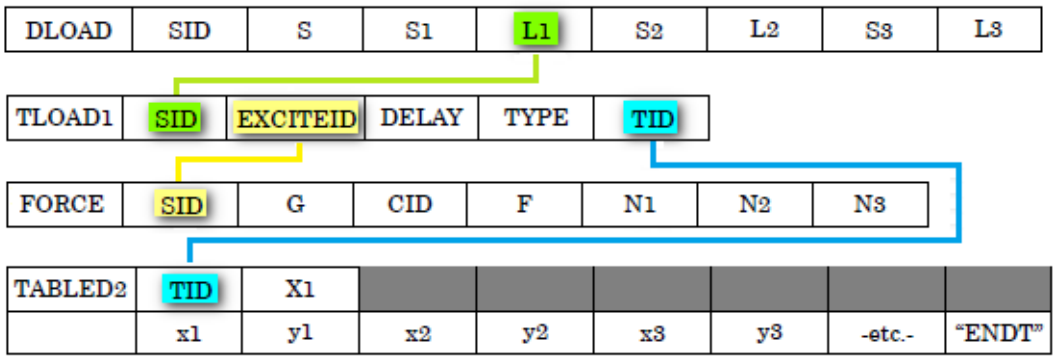
  where n points to a DAREA, FORCE, FORCE1, FORCE2, GRAV, PLOAD, PLOAD4, RFORCE, RFORCE1, SLOAD, SPCD, or LOADSET entry.

- Time-assigned loads are selected with the DLOAD case control command,

  ```
  DLOAD=n
  ```

  where n points to a load set defined by a TLOAD1 bulk entry, or a DLOAD bulk entry if you want to combine multiple TLOAD1 entries into a single load set.

- Time-assigned load definition in Bulk Data:
  - TLOAD1 - Defines a time-assigned load.
  - TABLEDi (i=1,2,3,4) - Table that defines the load variation with time.
  - DLOAD - Combines several TLOAD1 entries.
- Defining the TLOAD1 entry:
  - The EXCITEID field on the TLOAD1 entry selects the static load set IDs.
  - The supported static load inputs are the DAREA, FORCE, FORCE1, FORCE2, GRAV, PLOAD, PLOAD4, RFORCE, RFORCE1, SLOAD, and SPCD bulk entries.
  - The TYPE field on the TLOAD1 entry should be “0” for all load inputs selected by the EXCITEID field, except for the SPCD entry. The SPCD entry requires “1” in the TYPE field.
  - A real value is supported in the DELAY field on a TLOAD1 entry to optionally shift the time steps used to compute the associated loads.
  - A temperature load cannot be selected on the EXCITEID field. See Thermal Loads.
  - The TID field selects a TABLEDi, which defines a load scaling versus time function.
  - The figure below demonstrates how the DLOAD, TLOAD1, FORCE (for example), and TABLEDi bulk entries relate to one another.



  - Load Input Example 1:

When there is more than one time-assigned load set, the DLOAD bulk entry is required:

```
$2345678$2345678$2345678$2345678$2345678$2345678$2345678$2345678$2345678
$
$ DLOAD COMBINES MULTIPLE TLOAD1 (102 AND 105)
DLOAD         17      1.      1.     102      1.     105
$
$ TIME-ASSIGNED FORCE, EXCITEID=125, TYPE=0 (DEFAULT), TIME FUNCTION TID=13
TLOAD1       102     125                      13
$
FORCE        125      80       0      1.      3.      0.      0.
$
$ TIME FUNCTION 13 USED FOR FORCE LOAD
TABLED2       13      0.                                                    +
+              0.      0.      1.    100.      2.      0.    ENDT
$
$ TIME-ASSIGNED FORCE EXCITEID=3, TYPE=0 (DEFAULT), TIME FUNCTION TID=12
TLOAD1       105       3               1      12
$
FORCE          3      73       0      2.      8.      0.      0.
$
$ TIME FUNCTION 12 USED FOR FORCE LOAD
TABLED2       12      0.                                                    +
+              0.      0.      2.      1.    ENDT
```

- Load Input Example 2:

  When there is only one time-assigned load set, the DLOAD entry is not required:

```
$2345678$2345678$2345678$2345678$2345678$2345678$2345678$2345678$2345678
$
$ TIME-ASSIGNED FORCE, EXCITEID=125, TYPE=0 (DEFAULT), TIME FUNCTION TID=13
TLOAD1       102     125                      13
$
FORCE        125      80       0      1.      3.      0.      0.
PLOAD        125   100.0      21      30      18     10
PLOAD        125   100.0      10      18      22     25
$
$ TIME FUNCTION 13 USED FOR LOAD
TABLED2       13      0.                                                    +
+              0.      0.      1.    100.      2.      0.    ENDT
$
```

- Additional Information about mechanical loads:

  - The bolt preload capability allows you to model bolts with either 3D solid elements or 2D plane stress elements. A bolt preload can be combined with geometric nonlinear and plasticity conditions. See Bolt preload.

  - Loads in any subcase are total loads as opposed to incremental loads from the previous subcase. In other words, the ending load from a previous subcase does not become the initial loading for the consecutive subcase.

  - If no load is applied in a subcase, the total load is zero.

  - LOAD=n or DLOAD=n defined at the global level is used in all statics subcases unless a different LOAD=n or DLOAD=n is defined in a subcase.

- If a time-assigned and time-unassigned enforced displacement condition is defined with the SPCD entry, a constraint must also be defined with the SPC entry on the same DOF referenced by the SPCD entry.
- The TSTEP1 bulk entry defines the time step intervals in which a solution will be generated and output in a static subcase. If your time steps defined by the TSTEP1 entry exceed the time values defined in your TABLEDi entry, by default, the software will extrapolate the data defined in the TABLEDi entry. The software will issue a warning if extrapolation occurs. If you do not want the software to extrapolate the data, you can enter “1” in the EXTRAP field on the TABLEDi entry.
- In SOL 401, when RFORCE or RFORCE1 entries are referenced by the EXCITEID field on a TLOAD1 entry, the data on the associated TABLEDi, along with the scale factors S and Si on a DLOAD entry (if defined), scale the angular velocity (ω) and acceleration (α), which are used to compute an inertia force in the equation F = [m] [ω x (ω x r)) + α x r]. Since ω is squared in the force computation, the resulting scaling is not linearly related to the computed force (F). All other solutions scale the computed force (F).

2.6.2 Thermal loads

A thermal load requires a load temperature (T_{load}), an initial temperature (T_{init}), and a reference temperature (T_{ref}).

Thermal strain is calculated by

$\varepsilon = \alpha_{load}(T_{load} - T_{ref}) - \alpha_{init}(T_{init} - T_{ref})$

- T_{load} is the temperature load which induces a thermal strain.
- T_{init} is the strain free temperature used in the analysis.
- T_{ref} is the initial temperature used when computing the temperature dependent coefficient of thermal expansion, and is defined on the MATi entry. See Computing the coefficient of thermal expansion.
- If either T_{load} or T_{init} are defined, they both must be defined.
- If the coefficient of thermal expansion is defined as temperature dependent with the MATTi entries, α_{load} is evaluated at T_{load}, and α_{init} is evaluated at T_{init} .

  If the coefficient of thermal expansion is not defined as temperature dependent, α_{load} and α_{init} are assigned the single value defined on the MATi entry.
- T_{init} is defined using the TEMP(INIT) case control command, and must be the same for all subcases. Typically, the TEMP(INIT) command is defined globally, and selects one of the following.
  - The TEMP(INIT) can select the TEMP and TEMPD entries in the bulk data.

    For example,

```
...
TEMP(INIT) = 100
...
BEGIN BULK
```

```
...
$ Initial temperatures defined in the bulk data
TEMP,100,5,232.0,6,354.4,...etc
...
$ TEMPD defines a temperature for grid points not included on a TEMP entry
TEMPD,100,450.0
...
```

- The TEMP(INIT) command can select the TEMPEX and TEMPD bulk entries. The new TEMPEX entry references an external BUN file using the unit number defined with an ASSIGN statement. The unit number must be unique to other BUN files, and to other reserved unit numbers. The BUN file used to define T_{init} must only include a single set of temperature data.

  If the BUN file only defines temperatures for a portion of the model (subset), the TEMPD entry must be included in the bulk data to define a temperature for the grid points not included in the BUN file.

  For example,

```
...
ASSIGN BUN='temperature0.bun' UNIT=21
...
TEMP(INIT) = 100
...
BEGIN BULK
...
$ Initial temperatures defined in the BUN file
TEMPEX,100,21
$ Temperature for grid points not in the BUN file
TEMPD,100,630.2
...
```

- There are a variety of options to define T_{load}. These options can be defined globally and in a subcase. Any subcase definition will override any global definition. For example, if you define a time-unassigned T_{load} globally using the TEMP(LOAD) command, and you define a time-assigned T_{load} in a subcase using the DTEMP command, the time-assigned T_{load} is used for that subcase.
  - You can define a time-unassigned T_{load} with all temperatures defined in the bulk data. The TEMP(LOAD) case control command selects the TEMP and TEMPD entries in the bulk data.

    For example,

```
...
SUBCASE 5
     TEMP(LOAD) = 150
BEGIN BULK
...
$ time-unassigned grid point load temperatures for subcase 5
TEMP,150,74,232.0,23,354.4,...
$ TEMPD defines a temperature for grid points not included on a TEMP entry
TEMPD,150,450.0
...
```

  - You can define a time-unassigned T_{load} with temperatures defined in an external BUN file. The TEMP(LOAD) case control command selects the TEMPEX bulk entry and optionally the TEMPD entry. The new TEMPEX entry references the external file using the unit number defined with an ASSIGN statement. The unit number must be unique to other BUN files, and to other reserved unit numbers. The BUN file selected with the TEMPEX bulk entry must only include a single set of temperature data.

If the BUN file only defines temperatures for a portion of the model (subset), the TEMPD entry must be included in the bulk data to define a temperature for the grid points not included in the BUN file.

TEMPEX example:

```
...
ASSIGN BUN='temperature1.bun' UNIT=22
$...
SUBCASE 10
     TEMP(LOAD) = 200
BEGIN BULK
...
$ Time-unassigned load temperatures for subcase 10
TEMPEX, 200, 22
$ Temperature for grid points not in the BUN file
TEMPD,200,630.2
...
```

- You can define a time-assigned T_{load} with temperatures defined in the bulk data or in a BUN file. The new DTEMP case control command selects the new DTEMP bulk entry, which defines a list of time points versus set IDs. The set IDs are either the IDs of TEMP and TEMPD entries in the bulk data, or the IDs of TEMPEX and TEMPD entries in the bulk data. You cannot combine TEMP and TEMPEX entries with the same set ID.

  Example with TEMP and TEMPD entries in the bulk data:

```
Note: This example assumes the TEMP entries for temperature sets 500 and 501 define temperatu
...
SUBCASE 15
     DTEMP(LOAD) = 250
...
BEGIN BULK
...
$ DTEMP is a list of time points versus set IDs
DTEMP,250,,,,,,,,+
+,.2,500,.4,501,.6,502
...
$ Load temperatures at t=.2
TEMP,500,5,232.0,6,354.4,7,284.2
...
$ Load temperatures at t=.4
TEMP,501,5,234.1,6,356.3,7,287.8
...
$ Load temperatures at t=.6
TEMP,502,5,237.3,6,358.4,7,292.4
$ Temperature for grid points not defined with TEMP entry 502.
TEMPD,502,630.2...
```

  Example with TEMPEX and TEMPD entries in the bulk data:

```
Note: This example asssumes the BUN files for temperature sets 501 and 502 define temperatures
...
ASSIGN BUN='temperature1.bun' UNIT=22
ASSIGN BUN='temperature2.bun' UNIT=23
ASSIGN BUN='temperature3.bun' UNIT=24
...
SUBCASE 15
     DTEMP(LOAD) = 250
...
BEGIN BULK
...
$ DTEMP is a list of time points versus set IDs
```

```
DTEMP,250,,,,,,,,+
+,.2,500,.4,501,.6,502
...
$ Load temperatures at t=.2
TEMPEX,500,22
$ Temperature for grid points not defined in BUN file
TEMPD,500,345.4
...
$ Load temperatures at t=.4
TEMPEX,501,23
...
$ Load temperatures at t=.6
TEMPEX,502,24
...
$ If the BUN file for t=.2 and t=.4 includes data for all grid points, the TEMPD is not needed
```

- You can define a time-assigned T_{load} with temperatures defined in a single, external BUN file. The new DTEMP case control command selects the new DTEMPEX bulk entry, which references the external file using the unit number defined in the ASSIGN statement. The unit number must be unique to other BUN files, and to other reserved unit numbers. The single BUN file selected with the DTEMPEX bulk entry must include temperature data for all grid points, and for multiple time points. The BUN file can include temperatures for grids which are not in the model, but unlike the TEMPEX example above, the BUN file selected with the DTEMPEX cannot define temperatures for only a portion of the model (subset). The TEMPD entry cannot be combined with the DTEMPEX entry.

  DTEMPEX example:

```
...
ASSIGN BUN='temperature.bun' UNIT=23
$...
SUBCASE 20
   DTEMP = 300
BEGIN BULK
...
$ Time-assigned load temperatures for subcase 20
DTEMPEX, 300, 23
...
```

Additional information:

- The specification of TEMP(MATERIAL) or TEMP(BOTH) are unsupported and will cause a fatal error if defined.

- The TVAR parameter on the NLCNTL bulk entry controls if time-unassigned temperature loads selected with the TEMP(LOAD) case control command are ramped, or not ramped for each subcase.

  - When TVAR=RAMP, the software ramps the load temperatures from the final T_{load} defined for the previous static subcase to the T_{load} defined for the current subcase. The software determines the load temperature increments using the total number of time increments defined for that subcase. If T_{load} is not defined in the previous subcase, the software ramps from T_{init} to the current T_{load}.

  - When TVAR=STEP, the load temperatures are not ramped.

The default is “RAMP” except when Tend1 = 0.0 is defined on the TSTEP1 entry in the first static subcase. “STEP” occurs in this case.

- For the time-assigned temperature data, the software will interpolate the grid point temperatures when times are defined between the time points in the data. Although, if a solution time is outside the data range, the software will use the data at the closest time point, and a warning will be written to the f06 file.
- You can turn off the thermal strain computation by defining the parameter setting THRMST=NO (default=YES) on the NLCNTL bulk entry. This is useful for temperature dependent material evaluation without thermal loading.
- When temperature dependent material properties are defined with the MATTi entries for a static subcase, the properties are evaluated at T_{load} selected with either the TEMP(LOAD) or DTEMP case control. Both T_{load} and T_{init} must be defined when temperature dependent properties are defined.
- A modal subcase which is not sequentially dependent (SEQDEP=NO) can include temperature dependent material properties defined with the MATTi entries. The properties are evaluated at T_{load} selected with the TEMP(LOAD) case control. The DTEMP case control command is not supported in a modal subcase. Both T_{load} and T_{init} must be defined when temperature dependent properties are defined.
- The OTEMP case control command can be included to request solution temperature output.

Computing the coefficient of thermal expansion

You use temperature versus strain (length) test data to compute the temperature dependent coefficient of thermal expansion (α). This data begins with the test specimen of initial length L at a reference temperature (T_{ref}). The axial strain (L_i) is then measured at consecutive temperatures T_i. To calculate α_i:

$$\alpha_i = \frac{L_i - L}{L\left(T_i - T_{ref}\right)}$$

2.7 Constraints and enforced motion

SOL 401 allows you to change the boundary conditions, or enforce the displacements at specified grids.

2.7.1 Boundary conditions

In the Case Control section, an SPC entry is used to select a single point constraint set (s-set) which will be applied to the structural model. The specified set identification must be identical to the SID field of an SPC, SPC1 or SPCADD bulk entry. Notice that SPCADD entries take precedence over SPC or SPC1 entries. If both have the same SID, only the SPCADD entry will be used.

A significant application of SPC is the imposition of boundary conditions. The PS field in the GRID entry is also able to specify single-point constraints associated with a grid point. Although, these constraints are so-called permanent constraints which can not be changed during the analysis. An

advantage of using SPC to specify boundary conditions is that these boundary conditions can be changed from subcase to subcase by selecting a different SPC set inside each subcase.

SPC input summary

- The SPC=n case control command selects either the SPC, SPC1, or SPCADD bulk entry.

  The SPC condition can change between subcases.

  The SPC entry can optionally be used to define a time-unassigned enforced displacement. It cannot be defined as time-assigned. That is, it cannot be selected with the EXCITEID on the TLOAD1 entry. For a time-assigned enforced displacement, you can use the SPCD entry. See Mechanical Loads for information on the SPCD entry.

  The SPCFORCES case control command is supported to request the SPC force output.

2.7.2 Multipoint constraint

While a single point constraint (SPC) is used to constrain the motion of a degree-of-freedom, a multipoint constraint (MPC) is used to tie the motion of one degree-of-freedom to other degrees-of-freedom. The MPC command in the Case Control section is used to select a multipoint constraint set in the Bulk Data. The specified MPC set identification must appear at least in one MPC or MPCADD bulk entry. Each MPC bulk entry may be used to define a constraint equation involving a group of degrees-of-freedom in which the first degree-of-freedom is assumed to be the dependent degree-of-freedom and included in the m-set. All the degree-of-freedom in m-set will be condensed out prior to the matrix operations. Their response will be directly recovered from those of the independent degrees-of-freedom according to the specified constraint equation. An MPC cannot be changed from subcase to subcase.

MPC input summary

- The MPC=n case control command selects either the MPC or MPCADD bulk entry.

  MPCs can change between subcases, so the MPC=n case control command must be defined globally.

  MPCs do not update for large displacements (PARAM,LGDISP,1).

2.7.3 Enforced displacements

Enforced displacements may be specified in the Bulk Data section using SPC or SPCD entries. An SPC can be used to define time-unassigned enforced displacements. An SPCD can be used to define time-assigned or time-unassigned enforced displacements. See Boundary conditions and Mechanical Loads for input details.

Each SPC entry may define enforced displacements for up to two grid or scalar points. Several SPC entries which reference the same SID may be used if enforced displacements for more than two grid or scalar points are desired. The only disadvantage of this method is that the entire s-set must be redefined if the enforced displacement conditions vary among subcases.

If a time-assigned or time-unassigned enforced displacement condition is defined with the SPCD entry, a constraint must also be defined with the SPC entry on the same DOF referenced by the SPCD entry.

If multiple enforced displacement conditions are applied to the same DOF, the software uses the following precedence.

- A time-assigned enforced displacement defined with the SPCD entry, which is referenced by the EXCITEID on the TLOAD1 entry, will overwrite time-unassigned enforced displacements defined with the SPCD or SPC entries.

- A time-unassigned enforced displacement defined with the SPCD entry, which is referenced by the LOAD=n case control command, will overwrite a time-unassigned enforced displacement defined with the SPC entry.

2.8 Glue support

- Surface-to-surface glue

  You can define surface-to-surface glue between the faces of the CTETRA, CHEXA, CPENTA and CPYRAM elements. You create solid element face regions with the BSURFS, BCPROP, or BCPROPS bulk entries, then pair the regions using the source and target fields on the BGSET bulk entry.

- Edge-to-edge glue

  You can define edge-to-edge glue between the edges of the axisymmetric elements CTRAX3, CQUADX4, CTRAX6, CQUADX8, the plane stress elements CPLSTS3, CPLSTS4, CPLSTS6, CPLSTS8, and the plane strain elements CPLSTN3, CPLSTN4, CPLSTN6, CPLSTN8. You create edge regions with the BEDGE bulk entry, then pair the regions using the source and target fields on the BGSET bulk entry.

Additional information.

- The inputs to define contact conditions are consistent with other solutions (the BGSET case control and the BGSET, BGPARM, BGADD, BSURFS, BCPROPS, BEDGE bulk entries).

- The BGSET case control command must be above the subcases. As a result, the glue conditions apply to all static and modal subcases.

- The source side element characteristics are used to define the glue stiffness. Therefore it is possible for differences depending on which element faces or edges are selected as the source region.

- The axisymmetric, plane stress, and plane strain elements can be defined in either the XZ plane or in the XY plane. Edge-to-edge glue is supported in either orientation. Glue conditions are computed on the axisymmetric elements on a 2*PI section by default.

- Only GLUETYPE=2 is supported. If you request GLUETYPE=1, the software will continue the solution using GLUETYPE=2. See the BGPARM entry in the *Quick Reference Guide* for a definition of GLUETYPE.

- When large displacement occur with PARM,LGDISP,1, the glue stiffness orientation will update as a result of large displacement effects. The glue stiffness is only computed once, at the beginning of a solution.

- The BGRESULTS case control command is supported to request the glue forces and tractions in SORT1 format.

- The generalized plane strain element is not supported by glue or contact regions.

2.9 Contact support summary

Contact conditions prevent element faces or edges from penetrating, and allow finite sliding with optional friction effects. The inputs to define contact conditions are generally consistent with other solutions (the BCSET case control and the BCTSET, BCTPARM, BCRPARA, BGADD, BSURFS, BCPROPS, BEDGE bulk entries). See Contact conditions in SOL 401.

Contact input summary

- Surface-to-surface contact

  You can define surface-to-surface contact between the faces of the solid elements CTETRA, CHEXA, CPENTA and CPYRAM. You create solid element face regions with the BSURFS, or BCPROPS bulk entries, then pair the regions using the source and target fields on the BCTSET bulk entry.

- Edge-to-edge contact

  You can define edge-to-edge contact between the edges of the axisymmetric elements CTRAX3, CQUADX4, CTRAX6, CQUADX8, the plane stress elements CPLSTS3, CPLSTS4, CPLSTS6, CPLSTS8, and the plane strain elements CPLSTN3, CPLSTN4, CPLSTN6, CPLSTN8. You create edge regions with the BEDGE bulk entry, then pair the regions using the source and target fields on the BCTSET bulk entry.

- The BCSET case control command must be above the subcases. As a result, the contact definition applies to all static subcases.

  A modal subcase which is sequentially dependent (default) uses the final stiffness from the previous static subcase. When contact is defined, the final stiffness from the static subcase includes the contact stiffness.

- The contact condition can update when large displacement effects are turned on with PARAM,LGDISP,1. The active and inactive contact elements are updated along with any force adjustments. When sliding occurs, the orientation of the current contact stiffness are updated. If large enough sliding occurs within a user defined tolerance, contact elements are recreated in the current deformed configuration, contact stiffness is recomputed, and contact tractions from the previous iteration are applied to the newly created contact elements.

- The BCTPARM entry can be used to define SOL 401 contact parameters. The OFFSET region parameter on the BCRPARA entry is supported.

- Contact results can be requested with the BCRESULTS case control command.

- The generalized plane strain element is not supported by glue or contact regions.

2.10 Nonlinear Parameters: NLCNTL entry

The NLCNTL bulk entry can be used to define strategies for the incremental and iterative solution processes. It is difficult to choose the optimal combination of all the options for a specific problem. However, based on a considerable number of numerical experiments, the default option was intended to provide the best workable method for a general class of problems. You should start with the default settings.

The NLCNTL bulk entry defines the parameters for SOL 401 control. The NLCNTL=*n* case control command selects the NLCNTL bulk entry, and can be defined in a subcase or globally. You can define the parameters on the NLCNTL bulk entry using the following format.

1	2	3	4	5	6	7	8	9	10
NLCNTL	ID	Param1	Value1	Param2	Value2	Param3	Value3		
	Param4	Value4	Param5	Value5	-etc-				

For example,

```
NLCNTL         1    EPSU    1E-3    EPSP    1E-3    EPSW    1E-7        +
+           CONV      PW   KSTEP       5 MAXITER      25
```

The supported parameters are listed below.

Name	Description
CONV	Specifies the convergence criteria. (Character = “U”, “P”, “W”, or any combination; Default = “W”)
CREEP	Include creep effects. (Character = “YES” or “NO”; Default = “YES”)
CRCERAT	For the ratio of maximum creep increment to elastic strain method, the ratio of maximum creep increment to elastic strain that is used to calculate the next time step. Valid for creep analysis only. (Real ≥ 0.0; Default = 0.1)
CRCINC	For the maximum creep increment method, the maximum creep increment that is used to calculate the next time step. Valid for creep analysis only. (Real ≥ 0.0; Default = 1.0E-4)
CRICOFF	Creep strain increment below which the next time step is the product of the current time step and CRMFMX. Valid for creep analysis only. (0.0 < Real < 1.0; Default = 1.0E-6)
CRINFAC	Integration factor used to calculate incremental creep strain. Valid for creep analysis only. (0.0 ≤ Real ≤ 1.0; Default = 0.5)
CRINTS	Initial time step or constant time step. Valid for creep analysis only. (Real > 0.0; Default = 0.01)

Name	Description
CRMFMN	Minimum time step multiplying factor. If the next time step calculated by the adaptive time stepping algorithm is smaller than the product of the current time step and CRMFMN, the software halves the current time step, recalculates the current creep strain increment, and reenters the adaptive time stepping algorithm at the point the creep strain increment is compared to CRICOFF. Valid for creep analysis only. (0 ≤ Real ≤ 1.0; Default = 0.1)
CRMFMX	Maximum time step multiplying factor. See the CRICOFF parameter for additional information. Valid for creep analysis only. (Real ≥ 1.0; Default = 5.0)
CRSBCDT	Controls whether the first time step in a sequential subcase uses CRINTS or the time step calculated at the end of the previous subcase. Valid for creep analysis only. (Integer; Default = 1) 0: Use the time step calculated at the end of the previous subcase 1: Use CRINTS
CRTEABS	Maximum absolute truncation error. Valid for creep analysis only. (0.0 ≤ Real < 1.0; Default = 1.0E-4)
CRTECO	For the error truncation method, use CRTEABS to calculate the next time step if the creep strain is less than CRTECO, and use CRTEREL to calculate the next time step if the creep strain is greater than CRTECO. Valid for creep analysis only. (0.0 ≤ Real < 1.0; Default = 0.01)
CRTEREL	Maximum relative truncation error. Valid for creep analysis only. (0.0 ≤ Real < 1.0; Default = 0.01)
CRTSC	Specifies the time stepping method. Valid for creep analysis only. (Integer or blank; Default = 12) 0: Use constant time stepping 1: Use adaptive time stepping based on the error truncation method only 2: Use adaptive time stepping based on the ratio of maximum creep increment to elastic strain method only 3: Use adaptive time stepping based on the maximum creep increment method only 12: Use adaptive time stepping based on both the error truncation method and the ratio of maximum creep increment to elastic strain method 13: Use adaptive time stepping based on both the error truncation method and the ratio of maximum creep increment method 23: Use adaptive time stepping based on both the ratio of maximum creep increment to elastic strain method and the maximum creep increment method 123: Use adaptive time stepping based on the error truncation method, the ratio of maximum creep increment to elastic strain method, and the maximum creep increment method

Name	Description
CRTSMN	Minimum time step. If the next time step is larger than CRTSMN, the software accepts the next time step. If the next time step is smaller than CRTSMN, the software halves the current time step, recalculates the current creep strain increment, and reenters the adaptive time stepping algorithm at the point the creep strain increment is compared to CRICOFF. Valid for creep analysis only. (0.0 ≤ Real ≤ CRTSMX; Default = 0.001*CRINTS)
CRTSMX	Maximum time step. If CRTSMX is set to 0.0 (default), the software accepts the next time step. If CRTSMX is nonzero and the next time step is larger than CRTSMX, the software uses CRTSMX as the next time step. Otherwise, the next time step is compared to CRTSMN. Valid for creep analysis only. (Real ≥ 0.0; Default = 0.0)
EPSBOLT	Bolt preload convergence tolerance. (Real > 0.0; Default = 1.0E-2)
EPSP	Error tolerance on force. (Real > 0.0; Default = 1.0E-2)
EPSU	Error tolerance on displacement. (Real > 0.0; Default = 1.0E-2)
EPSW	Error tolerance on work. (Real > 0.0; Default = 1.0E-6)
FOLLOWK	Include follower stiffness. Follower stiffness is computed from the follower loads defined with the FORCE1, FORCE2, PLOAD, or PLOAD4 entries. (Character = “YES” or “NO”; Statics default = “NO”; Modal default = “YES”)
ITRBOLT	Maximum number of bolt iterations before the bolt preload calculation is considered non-converged. (Integer > 0; Default = 20)
KUPDATE	Stiffness update strategy. (Integer; Default = 0) -1: Initial stiffness approach 0: Auto stiffness update 1: Full Newton-Raphson > 1: Quasi Newton-Raphson, and KUPDATE is the number of iterations before a stiffness update
MAXBIS	Maximum number of bisections allowed. (0 < Integer < 10; Default = 5)
MAXDIV	Number of divergences before solution is assumed to diverge. (Integer > 0; Default = 3)
MAXITER	Maximum number of iterations per time step. (Integer > 0; Default = 25)
MAXQN	Maximum number of quasi-Newton correction vectors to be saved. (Integer ≥ 0; Default = 10)
MSGLVL	Diagnostic level. (Integer = 0 or 1; Default = 0) 0: No additional diagnostic output 1: Convergence information is output for each iteration
PLASTIC	Include plasticity effects. (Character = “YES” or “NO”; Default = “YES”)

Name	Description
SOLVER	Specifies the solver. (Character = ”SPARSE”, ”PARDISO”, or ”ELEMITER”; Default = “SPARSE”)
SPINK	Include spin softening. (Character = “YES” or “NO”; Statics default = “NO”; Modal default = “YES”)
STFOPTN	Material stiffness matrix option. (Integer ≥ 0; Default = 3) 1: The elastic stiffness matrix is used. 2: The tangent stiffness matrix is used. 3: The elastic stiffness matrix is used to start each subcase, the elastic stiffness matrix is used if a stiffness update is requested prior to the beginning of a new time step, and the tangent stiffness matrix is used at any intermediate stiffness update.
STRESSK	Include stress stiffening. (Character = “YES” or “NO”; Statics default = “YES”; Modal default = “YES”)
THRMST	Include thermal strain in a static analysis (Character = “YES” or “NO”; Default = “YES”)
TSTEPK	Stiffness is updated at the beginning of the time step. Applicable only if KUPDATE>1. (Character = “YES” or “NO”; Default = “NO”)
TVAR	Specifies whether temperature loads are ramped or stepped. (Character = ”RAMP” or ”STEP”; Default = “RAMP”)

2.11 Iteration related output data

At the end of every iteration, the relevant data from the iteration process are printed under the following heading:

TIME	Solution time
ITERATION NO	Iteration count for the current timestep
DISP	Relative error in terms of displacements. See Error functions and weighted normalization.
LOAD	Relative error in terms of loads. See Error functions and weighted normalization.
WORK	Relative error in terms of work. See Error functions and weighted normalization.
TOTAL STIFFNESS UPDATES	Number of stiffness updates in the current time step.
NO. OF BISECTIONS	Number of occurrences of bisection conditions during the iteration. See Divergence criteria.
NO. OF ITR DIV	Number of occurrences of probable divergence during the iteration. See Divergence criteria.

STIFFNESS PARAMETER CURRENT	Value for the current stiffness parameter for the current iteration.
STIFFNESS PARAMETER % CHANGE	% Change in the value for the current stiffness parameter between and prior iteration.

2.12 Supported output

Case Control	Description
ADAPTERR	Requests error estimates.
BCRESULTS	Requests contact forces, tractions, separation distance, and the total and incremental slide distances.
BGRESULTS	Requests glue forces and tractions.
CRSTRN	Requests grid point creep strains on elements.
DISPLACEMENT	Requests displacement output.
EKE	Requests element kinetic energy output.
ELSTRN	Requests elastic strain at grid points on elements.
ESE	Requests the output of the strain energy.
FORCE	Requests element force output.
GCRSTRN	Requests gauss point creep strains on elements.
GELSTRN	Requests elastic strain at gauss points.
GPFORCE	Requests grid point force balance output.
GPKE	Requests kinetic energy at grid points in a modal subcase.
GPLSTRN	Requests gauss point plastic strain output on elements.
GSTRAIN	Requests strain at gauss points.
GSTRESS	Requests stress at gauss points.
GTHSTRN	Requests thermal strain at gauss points.
JINTEG	Requests output of the j-integral for crack analysis.
MEFFMASS	Requests modal effective mass output in a modal subcase.
MPCFORCES	Requests multipoint constraint force output.
OLOAD	Requests the form and type of applied load vector output.
OMODES	Requests selects a set of modes for output.
OPRESS	Requests the solution pressures, which are from NX Thermal in the context of a coupled NX multi-physics analysis, be included in the SOL 401 output.
OTEMP	Requests solution temperatures output on grid points.
PLSTRN	Requests grid point plastic strain output on elements.
SPCFORCES	Requests single-point force of constraint vector output.
STRAIN	Requests element strain output.
STRESS	Requests element stress output.
THSTRN	Requests thermal strain at grid points on elements.

2.13 Solver Support

SOL 401 supports the sparse direct solver (default), the element iterative solver, or the PARDISO solver (NLTRD3 nonlinear solution module). To select the SOL 401 solver type, supply a pair of fields on the NLCNTL card of the form “SOLVER SPARSE”, “SOLVER ELEMITER”, or “SOLVER PARDISO”. The default is SPARSE.

- The sparse direct solver is a robust and reliable option, well-suited to sparse models where accuracy is desired.

- The element iterative solver, which is already supported in SOL 101, performs particularly well with solid element-dominated models. It may be a faster choice if lower accuracy is acceptable. As in SOL 101, convergence tolerances and other options may be set by supplying an SMETHOD card in case control and matching ITER card in bulk data.

- For problems involving contact and 3D solid elements, the element iterative solver is generally faster as compared to the sparse direct solver.

- The PARDISO solver is a hybrid direct-iterative solver, potentially faster with larger numbers of cores than the sparse solver but with slightly lower accuracy.

Chapter 3: Element support

3.1 Element Overview

In nonlinear finite element analysis, lower-order elements are often preferred over higher-order ones because of their robustness and reasonable accuracy at reduced costs. The software employs linear elements, rather than quadratic or cubic elements, to process nonlinearity. When using lower-order elements, quadrilateral and hexahedral elements are generally preferred over triangular, pentahedral or tetrahedral elements. Triangular and tetrahedral elements can exhibit excessively stiff behavior, and caution is needed when using these elements.

Caution is also needed when different element types are combined in a model, and if these elements are incompatible. In such cases, some provision (e.g., appropriate constraints) may be necessary at the interface boundary. Modeling the joints (such as bolted, riveted, or welded) is particularly difficult. For lack of better information, the joints are usually modeled as rigid or free in certain degrees-of-freedom. If improved accuracy is required at such joints, the characteristics of the joint (stiffness and/or damping) may have to be identified from experiments or the local analysis of a detailed model at the joint. Modeling of the boundary conditions at the supports poses similar difficulties. Ideal boundaries are represented as free, clamped, pinned, roller or ball joints. The reality tends to be in smeared condition.

Elements become actively nonlinear if the parameter LGDISP is tumed. As for geometric nonlinearity, the software does not currently support large strain capability. However, large displacement is treated effectively by computing element stresses and strains in the updated element coordinates.

The following is a summary of the elements and materials supported in SOL 401.

- The 3D solids elements CTETRA, CHEXA, CPENTA and CPYRAM elements are supported for linear, geometric nonlinear, and material nonlinear analysis.

- The axisymmetric elements CQUADX4, CQUADX8, CTRAX3, CTRAX6, the plane strain elements CPLSTN3, CPLSTN4, CPLSTN6, CPLSTN8, and the plane stress elements CPLSTS3, CPLSTS4, CPLSTS6, CPLSTS8 are supported for linear, geometric nonlinear, and material nonlinear analysis.

  The grid points on these elements must all lie in either the XZ plane, or all in the XY plane of the basic coordinate system. The software automatically determines the orientation.

  When axisymmetric elements are defined on the XZ plane, X is the radial direction, and Z is the axial direction. The grid points defining these elements must have $X \geq 0$.

  When axisymmetric elements are defined on the XY plane, Y is the radial direction, and X is the axial direction. The grid points defining these elements must have $Y \geq 0$.

- A special, generalized plane strain formulation is available using the CPLSTN3, CPLSTN4, CPLSTN6, and CPLSTN8 element types. See Generalized plane strain analysis.

- The RBE2 and RBAR rigid elements are supported with optional large displacement effects and thermal expansion. The RBE3 rigid element is also supported, but it does not support the large displacement effects or thermal expansion. See Rigid element support.
- The mass elements CMASSi and CONMi are supported.
- The PSOLID or the PCOMPS bulk entries define the element properties. The PCOMPS is optionally used to define a layered solid composite property.

  The PGPLSN bulk entry is defines the special generalized plane strain element properties.
- You can request stress norm, stress error norm, strain energy norm, and strain energy error norm output. The output is computed and stored on an individual element basis. The NX Advanced Simulation product uses the output for adaptive meshing. See Error estimator for mesh refinement.
- You can compute and output the j-integral in a crack simulation. The j-integral output can be requested and used by third-party software like Zencrack to perform a fracture mechanics analysis. The CHEXA bulk entry allows for a collapsed element definition. See Crack simulation.

3.2 Generalized plane strain

A special plane strain formulation called *generalized plane strain* is available as an optional extension to the standard plane strain formulation. Both formulations use the CPLSTN3, CPLSTN4, CPLSTN6, and CPLSTN8 plane strain element types. To invoke the generalized plane strain option, the plane strain elements need to reference a PGPLSN property bulk entry. The generalized plane strain option is only supported in SOL 401, and is only applicable to small strain, small deflection structural analyses. These structural analyses include linear static, creep, and plasticity analyses, and combination creep and plasticity analysis.

Analysis with the generalized plane strain formulation is highly specialized and typically used to evaluate the behavior of gas turbine compressor and turbine blades. For such an analysis, you mesh the cross section of the blade with CPLSTN3, CPLSTN4, CPLSTN6, or CPLSTN8 elements. All of the elements in the mesh should reference a single PGPLSN property bulk entry.

With the PGPLSN bulk entry, you can specify the following data:

- The material bulk entry that is referenced by the PGPLSN bulk entry. MAT1 and MAT3 material bulk entries can be referenced.
- The control grid point. The control grid point is the location where out-of-plane loads or enforced displacements are applied to the set of elements that reference the PGPLSN bulk entry.
- The element thickness in the undeformed state.
- Optional user-defined additive normal stiffness and rotational stiffness values.

For the generalized plane strain analysis, NX Nastran calculates the standard in-plane plane strain stiffness, but also calculates three net out-of-plane stiffness values relative to the displacement coordinate system of the control grid point. Consequently, how you specify the displacement coordinate system for the control grid point is very important. You should specify the displacement coordinate system of the control grid point such that one axis is normal to the cross section and the

other two axes are parallel to the principal axes of the cross section. By so doing, the three net out-of-plane stiffness values that NX Nastran calculates represent the normal stiffness of the cross section, and the two bending stiffness for symmetrical bending of the cross section.

Because the CPLSTN3, CPLSTN4, CPLSTN6, and CPLSTN8 plane strain elements can only be defined in the XY- or XZ-planes of the basic coordinate system, the direction normal to the cross section is always in the Z- or Y-direction, respectively of the basic coordinate system. NX Nastran checks that one of the axes of the displacement coordinate system of the control grid point coincides with the correct normal direction and issues an error if one does not.

NX Nastran does not check the other two coordinate directions of the displacement coordinate system for the control grid point. It is your responsibility to assure that these directions are parallel to the principal axes of the cross section.

If you specify additive stiffness, the normal stiffness is added to the normal stiffness that NX Nastran calculates for the cross section . The additive rotational stiffness values are added to the bending stiffness values as follows:

- If the model lies in the XY-plane of the basic coordinate system, the KR1 value on the PGPLSN bulk entry is added to the bending stiffness about the X-axis of the displacement coordinate system of the control grid point. the KR2 value on the PGPLSN bulk entry is added to the bending stiffness about the Y-axis of the displacement coordinate system of the control grid point.

- If the model lies in the XZ-plane of the basic coordinate system, the KR1 value on the PGPLSN bulk entry is added to the bending stiffness about the X-axis of the displacement coordinate system of the control grid point. the KR2 value on the PGPLSN bulk entry is added to the bending stiffness about the Z-axis of the displacement coordinate system of the control grid point.

You can apply loads to the control grid point and to the generalized plane strain element mesh. At the control grid point, you can account for the centrifugal force that is attributable to the portion of the blade from the cross section you are modeling to the blade tip. To allow you to define a mechanically equivalent loading at the control grid point, you can specify not only a force that acts normal to the cross section, but also the bending moments that act on the cross section about axes parallel to the principal axes of the cross section.

To the generalized plane strain mesh, apply surface tractions, body forces, and in-plane enforced displacements that you want to include in the analysis. For example, you can apply aerodynamic forces to the grid points that lie on the periphery of the mesh.

From the net out-of-plane stiffness values and the loads that are applied to the control grid point, NX Nastran calculates the thickness change over the cross section. Similar to planes remaining plane in pure bending of beams, NX Nastran enforces that the surface defined by the thickness change is planar. From the thickness change over the cross section, NX Nastran calculates the out-of-plane strain of the elements at the grid locations. During the solution of the finite element model, NX Nastran uses the out-of-plane strain and any surface tractions, body forces, and in-plane enforced displacements that you specified.

If an enforced displacement and enforced rotations are applied at the control grid point, the thickness change of the cross section is directly specified. From the thickness change, NX Nastran calculates the out-of-plane strain directly and the solution of the finite element model is as before.

Note the generalized plane strain element is not supported by glue or contact regions.

The following constitutive models are available with generalized plane strain elements:

- To model plasticity of an isotropic material, use the MAT1 and MATS1 bulk entries in combination.
- To model plasticity of an isotropic material with temperature-dependent properties, use some combination of the MAT1, MATS1, MATT1, TABLEST, and TABLES1 bulk entries.
- To model plasticity of an orthotropic material, use the MAT3 and MATS1 bulk entries in combination. The elastic portion of the response is orthotropic, and the plastic portion of the response is isotropic.
- To model plasticity of an orthotropic material with temperature-dependent properties, use some combination of the MAT3, MATS1, MATT3, TABLEST, and TABLES1 bulk entries. The elastic portion of the response is orthotropic, and the plastic portion of the response is isotropic.
- To model creep of an isotropic material, use the MAT1 and MATCRP bulk entries in combination.
- To model creep of an isotropic material with temperature-dependent properties, use the MAT1, MATT1, and MATCRP bulk entries in combination.
- To model creep of an orthotropic material, use the MAT3 and MATCRP bulk entries in combination.
- To model creep of an orthotropic material with temperature-dependent properties, use the MAT3, MATT3, MATCRP, and TABLEM1 bulk entries in combination. The elastic portion of the response is orthotropic, and the creep portion of the response is isotropic.

For additional information, see the PGPLSN bulk entry in the *NX Nastran Quick Reference Guide*.

3.3 Rigid element support

The RBE2 and RBAR rigid elements are supported in SOL 401 with optional large displacement effects and thermal expansion. The RBE3 rigid element is also supported, but it does not support the large displacement effects or thermal expansion.

The RIGID case control command includes the AUTO and STIFF options to select the RBE2 and RBAR rigid element behavior. When RIGID=AUTO, which is the default for SOL 401, the behavior depends if large displacement effects are turned off with PARAM,LGDISP,-1 (default), or on with PARAM,LGDISP,1. The RIGID case control command must be defined globally, and it applies to all subcases. The input combinations are as follows.

- When RIGID=AUTO and PARAM,LGDISP,-1, the software automatically applies the RIGID=LINEAR option. RBE2 and RBAR elements do not include large displacement effects or thermal expansion.
- When RIGID=AUTO and PARAM,LGDISP,1, the software automatically applies the RIGID=STIFF option. RBE2 and RBAR elements include large displacement effects and thermal expansion.
- When RIGID=STIFF and PARAM,LGDISP,-1, the large displacement effects are not included, in general. RBE2 and RBAR elements include thermal expansion.
- When RIGID=STIFF and PARAM,LGDISP,1, the RBE2 and RBAR element behavior is the same as RIGID=AUTO and PARAM,LGDISP,1. RBE2 and RBAR elements include large displacement effects and thermal expansion.

- When RIGID=LINEAR, the RBE2 and RBAR elements do not include large displacement effects or thermal expansion. This behavior is independent of the PARAM,LGDISP setting.

To compute large displacement effects and thermal expansion, the software internally replaces the RBE2 and RBAR elements with either a stiff beam element, or a stiff spring element. A coincident grid tolerance is used to determine if a beam or a spring is used. For the RBAR, if the distance between the connecting grids is less than the tolerance, the stiff spring formulation is used. For the RBE2, if the distance between the grid defined in the GN field on the RBE2 entry, and any of the grids defined in the GM fields on the RBE2 entry, is less than the tolerance, the stiff spring formulation is used. You can optionally define the coincident grid tolerance explicitly with the parameter RGLCRIT. By default, it is automatically computed by the software:

Coincident Grid Tolerance = 1E-6 * LMODEL (units=length)

where LMODEL is the largest dimension of the model determined by the software.

You can optionally define the beam stiffness and area explicitly using the parameters RGBEAME and RGBEAMA, respectively. By default, they are automatically computed by:

Beam Stiffness = 1e+2 * EMAX (units=force/length^2)

Beam Area = (LMODEL * 1e-2)^2 (units=length^2)

where EMAX is the largest Young's modulus in the model. If no material is specified in the model, EMAX is set to 1.0E12.

You can optionally define the spring stiffness explicitly using the parameter RGSPRGK. By default, it is automatically computed by:

Spring Stiffness = EMAX * LMODEL (units = force/length)

Additional information:

- MPCFORCE and GPFORCE output are supported with all of the rigid elements. Since the software internally replaces an RBAR or RBE2 with a stiff beam or spring element when RIGID=STIFF, these elements are no longer represented as MPC equations. As a result, MPCFORCE output is not applicable to these elements. GPFORCE and FORCE output is applicable.

- GPFORCE output will correctly account for large displacements, except for DOF which are included in MPC equations.

- In general, MPCFORCE output can be requested with large displacements (PARAM,LGDISP,1). Although, it is computed based on the initial, undeformed configuration. MPCFORCE output may not be accurate in regions where large displacements occur.

- The TEMP(LOAD) and TEMP(INIT) value used on RBAR elements is an average calculated from the grid point values. On RBE2 elements, an average TEMP(LOAD) and TEMP(INIT) is calculated for each leg of the element using the values on the independent/dependent grid pairs such that each leg can have a different thermal strain if the temperatures vary at the grids.

  The rigid element thermal strains are calculated from

  $\varepsilon_{thermal} = \alpha(AVGTEMP(LOAD) - AVGTEMP(INIT))$

  If TEMP(LOAD) or TEMP(INIT) are not defined, they are assumed to be zero.

See the RIGID case control command.

3.4 Error estimator for mesh refinement

You can request stress norm, stress error norm, strain energy norm, and strain energy error norm output when using SOL 401. The output is computed and stored on an individual element basis. The NX Advanced Simulation product uses the output for adaptive meshing.

The output is supported for the following element types:

Solid elements	CHEXA, CPENTA, CPYRAM, CTETRA (excludes CHEXA and CPENTA elements referencing PCOMPS bulk entries)
Axisymmetric elements	CQUADX4, CQUADX8, CTRAX3, CTRAX6
Plane strain elements	CPLSTN3, CPLSTN4, CPLSTN6, CPLSTN8 (includes elements referencing PGPLSN bulk entries)
Plane stress elements	CPLSTS3, CPLSTS4, CPLSTS6, CPLSTS8

- The stress norm is calculated from:

$$\|u\|_{L_2} = \sqrt{\frac{1}{\Omega}\int_{\Omega} \sigma_{unaveraged}^{T}\sigma_{unaveraged}d\Omega}$$

- The stress error norm is calculated from:

$$\|e_{\sigma}\| = \sqrt{\frac{1}{\Omega}\int_{\Omega}\left(\sigma_{unaveraged} - \sigma_{averaged}\right)^{T}\left(\sigma_{unaveraged} - \sigma_{averaged}\right)d\Omega}$$

- The strain energy norm is calculated from:

$$\|u\| = \sqrt{\int_{\Omega} \sigma_{unaveraged}^{T} D^{-1}\sigma_{unaveraged}d\Omega}$$

- The strain energy error norm is calculated from:

$$\|e_{U}\| = \sqrt{\int_{\Omega}\left(\sigma_{unaveraged} - \sigma_{averaged}\right)^{T} D^{-1}\left(\sigma_{unaveraged} - \sigma_{averaged}\right)d\Omega}$$

where

Ω is the element volume,

$\sigma_{unaveraged}$ is the unaveraged stress vector,

$\sigma_{averaged}$ is an averaged stress vector computed at a grid point using the stress vectors from elements connected to the grid point,

D matrix is the constitutive relation.

When computing $\sigma_{averaged}$, stress values are not averaged across different element families, material properties, material coordinate systems, orientation angles in 2D solid elements, and thicknesses in plane stress elements.

You use the STRESS, STNERGY and STEP describers on the new ADAPTERR case control command to request the output.

- The ADAPTERR case control command must be defined above the subcases (globally).
- The STRESS describer requests the stress norm and the stress error norm.
- The STNERGY describer requests the strain energy norm and strain energy error norm.
- You can specify both the STRESS and STNERGY describers to request stress norm, stress error normal, strain energy norm and strain energy error norm output.
- The software always outputs the maximum value on each element, for each output type requested, by comparing the values from all solution steps. In addition, if you specify the STEP describer, the software will output what you have requested at the output increment steps defined with the TSTEP1 entries.

For additional information, see the ADAPTERR case control command.

3.5 Crack simulation

You can compute and output the j-integral for a given crack geometry. This capability is only supported for SOL 401. The j-integral output can be used by third-party software like Zencrack to perform a fracture mechanics analysis.

- You can use the JINTEG case control command to control the computation and output of the j-integral. With the JINTEG case control command, you can direct the j-integral output to either .op2 or .f06 files.

  For additional information, see the JINTEG case control command.

- The creation of the CRAKTP bulk entry. You can use the CRAKTP bulk entry to specify information related to the crack tip.

  For additional information, see the CRAKTP bulk entry in the *NX Nastran Quick Reference Guide*.

- The creation of the VCEV bulk entry. You can use the VCEV bulk entry to define virtual crack tip extension vectors.

  For additional information, see the VCEV bulk entry in the *NX Nastran Quick Reference Guide*.

- The modification of the CHEXA bulk entry to allow for collapsed CHEXA element definition. Note that the collapsed CHEXA element is not supported in a glue or contact region.
- The creation of the COLPHEXA parameter. You can allow collapsed CHEXA elements to bypass input file checks with the new COLPHEXA parameter. To do so, specify PARAM,COLPHEXA,YES in the bulk section of the input file.

Collapsed CHEXA elements

Any face of a CHEXA element can be collapsed to an edge. The edge of the collapsed face represents the crack front.

Figure 3-1 shows the connectivity for a standard CHEXA element.

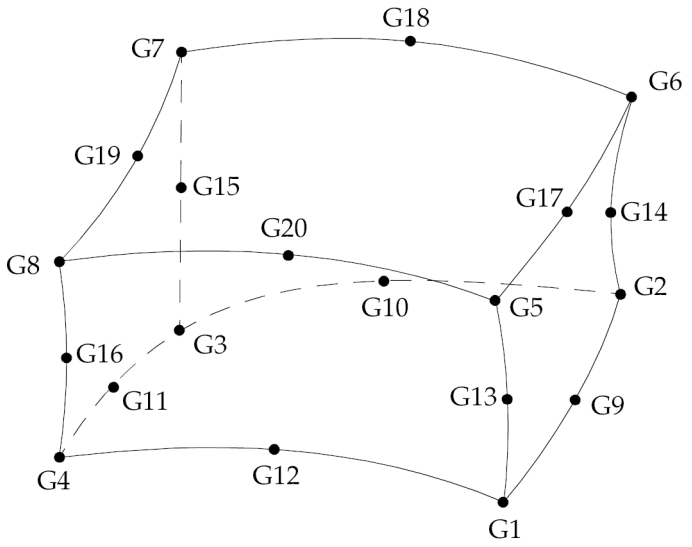


Figure 3-1. Standard CHEXA Element

Figure 3-2 shows the CHEXA element of Figure 3-1 with the G2–G14–G6–G18–G7–G15–G3–G10 face collapsed so that the G2–G14–G6 edge and the G3–G15–G7 edge become the crack front. Alternately, the G2–G14–G6–G18–G7–G15–G3–G10 face could be collapsed so that the G2–G10–G3 edge and the G6–G18–G7 edge would become the crack front.

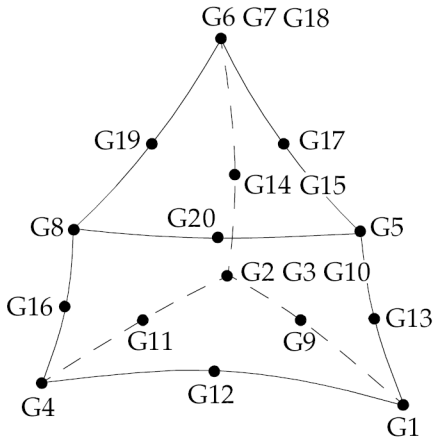


Figure 3-2. Collapsed CHEXA Element

Two options are available for specifying a CHEXA element with a collapsed face:

- In Format 1, 15 unique grid IDs are specified in the 20 grid ID fields of the CHEXA bulk entry. Format 1 is typically used for elastic material models. With Format 1, mid-side grids can move to the quarter-span locations closest to the crack front.

  For the collapsed CHEXA element shown in Figure 3-2, the Format 1 specification is as follows:

1	2	3	4	5	6	7	8	9	10
CHEXA	EID	PID	G1	G2	G2	G4	G5	G6	
	G6	G8	G9	G2	G11	G12	G13	G14	

1	2	3	4	5	6	7	8	9	10
	G14	G16	G17	G6	G19	G20			

where the same grid ID is entered in the G2, G3, and G10 fields, another grid ID is entered in both the G14 and G15 fields, and another grid ID is entered in the G6, G7, and G18 fields.

- In Format 2, 20 unique grid IDs are specified in the 20 grid ID fields of the CHEXA bulk entry. However, eight of the grid IDs do not have unique coordinates. Format 2 is typically used for elasto-plastic material models. With Format 2, mid-side grids remain at the mid-span locations.

  For the collapsed CHEXA element shown in Figure 3-2, the Format 2 specification is as follows:

1	2	3	4	5	6	7	8	9	10
CHEXA	EID	PID	G1	G2	G3	G4	G5	G6	
	G7	G8	G9	G10	G11	G12	G13	G14	
	G15	G16	G17	G18	G19	G20			

  where the grids entered in the G2, G3, and G10 fields would share the same coordinates, the grids entered in the G14 and G15 fields would share the same coordinates, and the grids entered in the G6, G7, and G18 fields would share the same coordinates. Unlike Format 1 where grids in the CHEXA element connectivity are merged, Format 2 does not merge coincident grids in the CHEXA element connectivity. Thus, these grids can move independently of one another.

3.6 Formulation of isoparametric elements

In the finite element method, parametric mapping is frequently used to map an irregular region into a regular one. The coordinate system used in the parametric mapping is a natural coordinate system suitable for the geometry. For instance, a natural coordinate system (ξ,η) is used for a quadrilateral surface in which each corner node has an extremum value of +1 or -1 in ξ and η. Then, the geometry of the internal points of the finite element can be described in terms of the nodal coordinates by the parametric mapping. The mathematical expression for this parametric mapping or interpolation is called a shape function. The displacement field inside the element should also be interpolated in terms of nodal displacements. The mapping is isoparametric if the same shape function is used to interpolate the displacement field as well as the geometry. The merit of isoparametric mapping is that the displacement field is invariant to the orientation of the Cartesian coordinate system x and y.

Most of the elements are isoparametric elements, in which the shape functions are expressed in terms of isoparametric coordinates. In what follows, derivations are shown for the isoparametric coordinates, shape functions, element matrix describing the strain-displacement relations, volume integration for stiffness and mass matrices by Gauss quadrature, and element loads are derived for a tetrahedron element to illustrate element related operations.

3.6.1 Isoparametric coordinates

Cartesian coordinates are not convenient to describe the geometry or the displacement field of a tetrahedron element. Let us introduce a set of volume coordinates ($L_1\ L_2\ L_3\ L_4$), such that

$$L_1 = \frac{Volume(P234)}{Volume(1234)}, etc.$$

as shown in Figure 3-3. The relation between volume and Cartesian coordinates can be established for a linear tetrahedron (without midside nodes) as follows:

$x = L_1x_1 + L_2x_2 + L_3x_3 + L_4x_4$

$y = L_1y_1 + L_2y_2 + L_3y_3 + L_4y_4$

$z = L_1z_1 + L_2z_2 + L_3z_3 + L_4z_4$

$1 = L_1 + L_2 + L_3 + L_4$

Equation 3-1.

It is obvious from Equation 3-1 that the shape functions are simply the volume coordinates, i.e.,

$N_1 = L_1$, $N_2 = L_2$, ..., etc.

Equation 3-2.

because $x = \Sigma\ N_i x_i$.

Shape functions for the quadratic tetrahedron can be derived using Lagrangian interpolation. These are

$N_1 = (2L_1 - 1)L_1$, etc. for corner nodes

and $N_5 = 4L_1, L_2$, etc. for midside nodes.

Equation 3-3.

Notice that these shape functions, Equations 3-2 and 3-3, satisfy element convergence criteria: integrability (C^{n-1} continuity for n-th derivative) and completeness (no straining by a rigid body mode, constant strain condition, and continuous displacement field).

With the isoparametric element, the same shape functions are used to describe the displacement field as well as the geometry, i.e.,

$$x = \sum_{1}^{10} N_i x_i\ ,\ etc. \quad and \quad u = \sum_{1}^{10} N_i u_i\ ,\ etc.$$

Equation 3-4.

The strains are calculated as

$$\{\varepsilon\} = [B]\{u\}$$

Equation 3-5.

where strain vector

$$\{\varepsilon\}^T = \langle \varepsilon_x \ \ \varepsilon_y \ \ \varepsilon_z \ \ \gamma_{xy} \ \ \gamma_{yz} \ \ \gamma_{zx} \rangle$$

Equation 3-6.

element matrix

$$\left[B\right]=\left[B_1 B_1 ... B_i .... B_{10}\right]$$

Equation 3-7.

nodal displacement vector

$$\left\{u\right\}^T=\left\langle a_1^T a_2^T ... a_i^T .... a_{10}^T\right\rangle$$

Equation 3-8.

with

$$\left\{a_i\right\}=\begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix}$$

Equation 3-9.

$$\left[B_i\right]=\begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_i}{\partial y} & 0 \\ 0 & \frac{\partial N_i}{\partial x} & \frac{\partial N_i}{\partial z} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} & 0 \\ 0 & \frac{\partial N_i}{\partial z} & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} & 0 & \frac{\partial N_i}{\partial x} \end{bmatrix}$$

Equation 3-10.

Because the shape functions are defined in terms of local coordinates, a coordinate transformation is required to obtain global derivatives. Volume coordinates involve four coordinates (one of which is dependent), and the Jacobian matrix will become rectangular. To avoid this difficulty, let us introduce local coordinates (ξ,η,ζ) as follows:

$$\begin{aligned} L_1 &= \xi \\ L_2 &= \eta \\ L_3 &= \varsigma \\ L_4 &= 1-\xi-\eta-\varsigma \end{aligned}$$

Equation 3-11.

Derivatives with respect to the local coordinates can be expressed in terms of global derivatives using the chain rule, i.e.,

$$\underset{3x10}{\begin{Bmatrix} \frac{\partial N_i}{\partial \varepsilon} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \varsigma} \end{Bmatrix}} = \underset{3x3}{\begin{bmatrix} \frac{\partial x}{\partial \varepsilon} & \frac{\partial y}{\partial \varepsilon} & \frac{\partial z}{\partial \varepsilon} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \varsigma} & \frac{\partial y}{\partial \varsigma} & \frac{\partial z}{\partial \varsigma} \end{bmatrix}} \underset{3x10}{\begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{Bmatrix}}$$

Equation 3-12.

where the Jacobian matrix is expressed in terms of shape functions as

$$[J] = \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \cdots & \frac{\partial N_{10}}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \cdots & \frac{\partial N_{10}}{\partial \eta} \\ \frac{\partial N_1}{\partial \varsigma} & \frac{\partial N_2}{\partial \varsigma} & \cdots & \frac{\partial N_{10}}{\partial \varsigma} \end{bmatrix} \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_{10} & y_{10} & z_{10} \end{bmatrix}$$

Equation 3-13.

This Jacobian matrix must be inverted to obtain global derivatives, from which the element matrix is formed. Notice that the determinant of the Jacobian matrix is called Jacobian which represents a volume change, i.e.,

$$dx\,dy\,dz = \det J \; d\xi\,d\eta\,d\varsigma$$

Equation 3-14.

3.6.2 Shape functions

To make tetrahedron element compatible with other solid elements (HEXA and PENTA), deletion of any or.all of.the midside nodes is permitted. The shape functions are modified with Kroneker deltas (δ_5 - δ_{10}) where

δ_i = 0 if the midside node i is deleted

= 1 if the midside node i is not deleted

The goal is to construct functions which are unity at the associated node and zero at all other nodes, regardless of any combination of deleted midside nodes, by the following scheme. At the corner nodes

$$N_1 = \xi - \frac{1}{2}N_5 - \frac{1}{2}N_6\frac{1}{2}N_7, etc.$$

At the midside nodes

$$N_5 = 4\xi\eta\delta_5, etc.$$

Shape functions are identified as follows:

$$N_1 = \xi\left[1-2\eta\delta_5-2\varsigma\delta_6-2\left(1-\xi-\eta-\varsigma\right)\delta_7\right]$$
$$N_2 = \eta\left[1-2\xi\delta_5-2\varsigma\delta_8-2\left(1-\xi-\eta-\varsigma\right)\delta_{10}\right]$$
$$N_3 = \varsigma\left[1-2\xi\delta_6-2\eta\delta_8-2\left(1-\xi-\eta-\varsigma\right)\delta_9\right]$$
$$N_4 = \left(1-\xi-\eta-\varsigma\right)\left(1-2\xi\delta_7-2\varsigma\delta_9-2\eta\delta_{10}\right)$$
$$N_5 = 4\xi\eta\delta_5$$
$$N_6 = 4\xi\varsigma\delta_6$$
$$N_7 = 4\xi\left(1-\xi-\eta-\varsigma\right)\delta_7$$
$$N_8 = 4\eta\varsigma\delta_8$$
$$N_9 = 4\varsigma\left(1-\xi-\eta-\varsigma\right)\delta_9$$
$$N_{10} = 4\eta\left(1-\xi-\eta-\varsigma\right)\delta_{10}$$

Derivatives of the shape functions with respect to the local coordinates are obtained as follows:

$$\frac{\partial N_1}{\partial\xi} = 1-2\eta\delta_5-2\varsigma\delta_6-2\left(-2\xi-\eta-\varsigma+1\right)\delta_7$$
$$\frac{\partial N_2}{\partial\xi} = -2\eta\left(\delta_5-\delta_{10}\right)$$
$$\vdots$$
$$\frac{\partial N_{10}}{\partial\xi} = -4\eta\delta_{10}$$
$$\frac{\partial N_1}{\partial\eta} = -2\xi\left(\delta_5-\delta_7\right)$$
$$\frac{\partial N_2}{\partial\eta} = 1-2\xi\delta_5-2\varsigma\delta_8-2\left(1-\xi-2\eta-\varsigma\right)\delta_{10}$$
$$\vdots$$
$$\frac{\partial N_{10}}{\partial\eta} = 4\left(1-\xi-2\eta-\varsigma\right)\delta_{10}$$

$$\frac{\partial N_1}{\partial \varsigma} = -2\xi\left(\delta_6 - \delta_7\right)$$
$$\frac{\partial N_2}{\partial \varsigma} = -2\eta\left(\delta_8 - \delta_{10}\right)$$
$$\vdots$$
$$\frac{\partial N_{10}}{\partial \varsigma} = -4\eta\delta_{10}$$

These shape functions and derivatives may be reduced to those for 4-noded and 10-noded tetrahedron elements. It can be verified that, for any combination of deleted midside nodes, there exist a condition

$$\sum N_i = 1$$

which satisfies the constant strain requirement.

3.6.3 Example element matrix

To illustrate the computational procedure, an element matrix for a linear tetrahedron (4-noded) element is explicitly derived here. The shape functions and the derivatives are tabulated below:

Node	Ni	$\delta N_i/\delta\xi$	$\delta N_i/\delta\eta$	$\delta N_i/\delta\varsigma$
1	ξ	1	0	0
2	η	0	1	0
3	ς	0	0	1
4	1 – ξ – η – ς	–1	-1	–1

Then the Jacobian matrix may be found as

$$[J] = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \end{bmatrix} = \begin{bmatrix} x_{14} & y_{14} & z_{14} \\ x_{24} & y_{24} & z_{24} \\ x_{34} & y_{34} & z_{34} \end{bmatrix}$$

where ($x_{ij} = x_i - x_j$) is used for convenience. Upon inverting the Jacobian matrix, we have

$$[J]^{-1} = \frac{1}{\det J} \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix}$$

where

$$a1 = y_{24}z_{34} - z_{24}y_{34}$$
$$b1 = -\left(x_{24}z_{34} - z_{24}x_{34}\right)$$
$$c1 = x_{24}y_{34} - y_{24}x_{34}$$

$$a2 = -\left(y_{14}z_{34} - z_{14}y_{34}\right)$$
$$b2 = x_{14}z_{34} - z_{14}x_{34}$$
$$c2 = -\left(x_{14}y_{34} - y_{14}x_{34}\right)$$

$$a3 = y_{14}z_{24} - z_{14}y_{24}$$
$$b3 = -\left(x_{14}z_{24} - z_{14}x_{24}\right)$$
$$c3 = x_{14}y_{24} - y_{14}x_{24}$$

The global derivatives of the shape functions are

$$\begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial x} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \frac{\partial N_3}{\partial y} & \frac{\partial N_4}{\partial y} \\ \frac{\partial N_1}{\partial z} & \frac{\partial N_2}{\partial z} & \frac{\partial N_3}{\partial z} & \frac{\partial N_4}{\partial z} \end{bmatrix} = \frac{1}{\det J}\begin{bmatrix} a_1 & a_2 & a_3 & -\left(a_1+a_2+a_3\right) \\ b_1 & b_2 & b_3 & -\left(b_1+b_2+b_3\right) \\ c_1 & c_2 & c_3 & -\left(c_1+c_2+c_3\right) \end{bmatrix}$$

Hence the element matrix will be

$$[B] = \frac{1}{\det J}\left[\begin{array}{ccc:ccc:ccc:ccc} a_1 & 0 & 0 & a_2 & 0 & 0 & a_3 & 0 & 0 & -\sum a_i & 0 & 0 \\ 0 & b_1 & 0 & 0 & b_2 & 0 & 0 & b_3 & 0 & 0 & -\sum b_i & 0 \\ 0 & 0 & c_1 & 0 & 0 & c_2 & 0 & 0 & c_3 & 0 & 0 & -\sum c_i \\ b_1 & a_1 & 0 & b_2 & a_2 & 0 & b_3 & a_3 & 0 & -\sum c_i & -\sum a_i & 0 \\ 0 & c_1 & b_1 & 0 & c_2 & b_2 & 0 & c_3 & b_3 & 0 & -\sum c_i & -\sum b_i \\ c_1 & 0 & a_1 & c_2 & 0 & a_2 & c_3 & 0 & a_3 & -\sum c_i & 0 & -\sum a_i \end{array}\right]$$

Notice that the rank of [*B*] matrix is 6.

3.6.4 Volume integration of element matrices

By virtue of variational principles, the element stiffness matrix is derived as follows:

$$\left[K^e\right] = \int_V B^T D\, B\, dV$$
$$= \int_0^1 \int_0^{1-\varsigma} \int_0^{1-\eta-\varsigma} \left|\det J\right| B^T D\, B\, d\xi\, d\eta\, d\varsigma$$

Equation 3-15.

where *D* is a (6x6) material tangential matrix.

While analytical integration is possible, there are some advantages in using numerical integration. The Gaussian quadrature is used for the tetrahedron as usual. The minimum number of intergration points required for non-singular stiffness matrix may be determined based on

$$\text{minimum n} = \frac{\text{Total dof - rigid body dof}}{\text{No. of strain components for each point}}$$

Equation 3-16.

These numbers are found to be 1 and 4 for linear and quadratic tetrahedron, respectively. Using the integration formulas, shown in Figure 3-4, the element stiffness matrices for linear and quadratic tetrahedron may be computed as

$$\left[K^6\right] = \frac{1}{6}\left|\det J\right| B^T DB$$

for 4–noded TETRA,

$$\left[K^6\right] = \frac{1}{6} \times \frac{1}{4} \sum_{gauss}^{a-d} \left|\det J\right| B^T DB$$

for 5–10 noded TETRA,

Equation 3-17.

where the scaling factor 1/6 is introduced to compensate IJI =6x (Volume). One-point scheme should be used for 4-noded tetrahedron (all the midside nodes deleted) and four-point scheme otherwise.

It is difficult to determine lumped masses when some of the midside nodes are deleted. However, the consistent mass matrix may be obtained using the same shape functions, i.e.,

$$M^e = \int_V \tilde{N}^T \rho \tilde{N} dV$$

Equation 3-18.

where [Ñ] = [N_1I N_2I ..... N_{10}I], with I being (3x3) identity matrix. Again the Gaussian quadrature 4-point formula is used to find

$$\left[M\right]^e = \frac{1}{24} \rho \sum_{gauss} \left|\det J\right| \left[\tilde{N}\right]^T \left[\tilde{N}\right]$$

Equation 3-19.

Notice that 4-point scheme is to be used even for 4-noded tetrahedron. For computational convenience, the consistent mass may be converted to the lumped mass. One way to achieve this is to take the diagonal terms and scale them so that the total mass is preserved.

3.6.5 Element loads and equilibrium

Nodal forces are computed from the element stresses using element matrix, i.e.,

$$\left\{F\right\}^e = \int_V B^T \left\{\sigma\right\} dV$$

Equation 3-20.

The Gaussian integration is performed as

$$\{F\}^e = \frac{1}{6}\left|\det J\right|\left[B\right]^T\{\sigma\}$$

for 4-noded tetrahedron and

$$\{F\}^e = \frac{1}{24}\sum_{gauss}\left|\det J\right|\left[B^T\right]\{\sigma\}$$

for (5-10)-noded tetrahedron.

Equation 3-21.

Thermal load is like an initial strain. Therefore, nodal forces due to thermal load are obtained by

$$\{F\}^e_{thermal} = \int_V B^T\left[D\right]\{\varepsilon_0\}\,dV$$

Equation 3-22.

where $\{\varepsilon_o\}^T = \alpha\Delta T < 1\ 1\ 1\ 0\ 0\ 0 >$. With anisotropic thermal properties, $\{\varepsilon_o\}^T$ becomes $\Delta T < \alpha_1, \alpha_2 \ldots \alpha_6 >$ in general.

The pressure load applied to any surface of the tetrahedron may be distributed to the grid points using the shape functions applicable to the 6-noded triangular element with removable midside nodes, i.e.,

$$\{P\}_{pressure} = \int_A p\left[\tilde{N}'\right]^T\{n\}\,dA$$

Equation 3-23.

where $\{n\}$ is a unit direction vector associated with a pressure (p) and

$$\left[\tilde{N}'\right] = \left[N_1'I \quad N_2'I \quad \ldots \quad N_6'I\right]$$

with I being (3 x 3) identity matrix. Shape functions (N_i') are shown in Figure 3-5. Area integration should be performed using Gaussian quadrature formulas shown in Figure 3-6, i.e.,

$$\{P\}_{pressure} = \frac{1}{6}\sum_{gauss}^{a-c}\left|\det J\right|\tilde{N}'^T\{p\}$$

Equation 3-24.

During the nonlinear iteration process, the equilibrium is not reached until convergence is achieved. The equilibrium is sought in the global level when the residual load R approaches zero. The residual load vector is defined as

$$\{R\} = \{P\} - \sum_e^{linear}\{K^e u^e\} - \sum_e^{nonlinear}\left\{\int B^T\sigma dV + F^e_{thermal}\right\}$$

Equation 3-25.

where {*P*} is the applied load vector including the thermal load and *Σ* implies assemblage in the global coordinate system (coordinate transformation required).

3.6.6 Element coordinates

The element coordinate system for the tetrahedron element is defined with the initial element geometry such that

- The origin is at the first grid point in the connectivity, G1.
- The x-axis is determined by connecting the origin G1 to node G2, i.e.,

$$\vec{i} = \left[\begin{Bmatrix} x_2 \\ y_2 \\ z_2 \end{Bmatrix} - \begin{Bmatrix} x_1 \\ y_1 \\ z_1 \end{Bmatrix} \right]_{normalized}$$

Equation 3-26.

- The y-axis is determined by orthonormalization (Gram-Schmidt process) of the edge direction (V_{13}) with respect to x-axis,

$$\vec{j} = \left[\vec{V}_{13} - \left(\vec{i} \bullet \vec{V}_{13} \right) \vec{i} \right]_{normalized}$$

Equation 3-27.

- The z-axis is orthogonal to x and y according to the right-hand rule, i.e.,

$$\vec{k} = \vec{i} \times \vec{j}$$

Then, the transformation from element coordinates to the basic coordinates is simply

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_{basic} = \left[T_{be} \right] \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_{element} + \begin{Bmatrix} x^e \\ y^e \\ z^e \end{Bmatrix}_{basic}$$

Equation 3-28.

where

$$\left[T_{be} \right] = \begin{bmatrix} i_x & j_x & k_x \\ i_y & j_y & k_y \\ i_z & j_z & k_z \end{bmatrix}$$

Equation 3-29.

and $< x^e, y^e, z^e >^T_{basic}$ is the position vector of the element coordinate system with respect to the basic coordinate system. Notice that $[T_{be}]$ is an orthogonal matrix, i.e.,

$$\left[T_{be}\right]^{-1} = \left[T_{be}\right]^{T}$$

3.6.7 Stress data recovery

In case of linear analysis, the stresses at the integration point are recovered as

$$\left\{\sigma\right\}_{gauss} = \left[D\right]\left\{\left[B\right]\left\{u\right\} - \left\{\alpha\right\}\Delta T\right\}$$

Equation 3-30.

with

$$\Delta T = \left\{N_i\right\}^{T}\left\{T_i - T_{oi}\right\}$$

where {σ}, [B], {u} and {α} are defined in the element coordinate system, and the shape function {N_i} interpolates Gauss point temperatures from the nodal temperatures. In case of nonlinear analysis, stresses are computed again upon convergence, starting from the last converged state (last converged solution of σ and u), i.e.,

$$\left\{\sigma_{new}\right\}_{gauss} = \left\{\sigma_{old}\right\} + \int_{\varepsilon_{old}}^{\varepsilon_{old}+\Delta\varepsilon} D d\varepsilon$$

Given {σ_{old}} and

$$\left\{\Delta\varepsilon\right\} = \left\{\varepsilon\right\} - \left\{\varepsilon_{old}\right\}$$

Equation 3-31.

with

$$\left\{\varepsilon\right\} = \left[B\right]\left\{u\right\} - \left\{\alpha\right\}\Delta T$$

the nonlinear material routine computes updated stresses {σ_{new}}, which are stored in ESTNL.

To output grid point stresses, the stresses, the integration points must be extrapolated to the nodal points. A linear extrapolation can be applied using stresses at the element c.g. and the corner Gauss point associated with a grid point. For a tetrahedron, referring to Figure 3-4,

$$\sigma_1 = \frac{\sigma_a - \sigma_0}{\alpha - 1/4}\left(1 - \frac{1}{4}\right) + \sigma_0$$

Equation 3-32.

where

$$\sigma_0 = \frac{1}{4}\left(\sigma_a + \sigma_b + \sigma_c + \sigma_d\right)$$

This yields

$$\sigma_1 = \frac{1-\beta}{1-4\beta}\sigma_a - \frac{\beta}{1-4\beta}\left(\sigma_b + \sigma_c + \sigma_d\right)$$

Equation 3-33.

In matrix form for all the grid points in the element

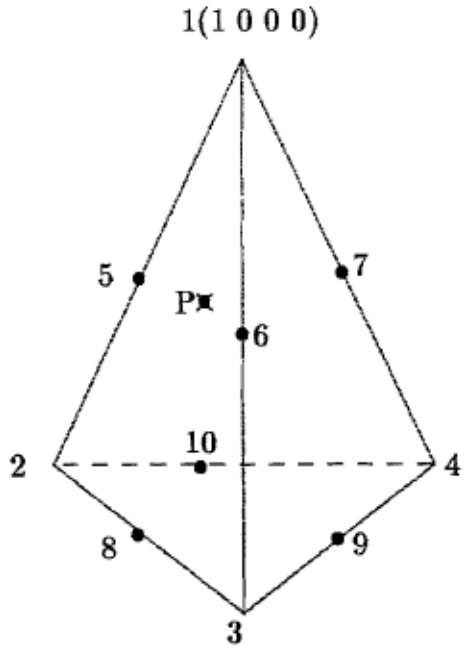
$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \end{Bmatrix}_{grid} = \begin{bmatrix} A & -B & -B & -B \\ & A & -B & -B \\ & & A & -B \\ Sym. & & & A \end{bmatrix} \begin{Bmatrix} \sigma_a \\ \sigma_b \\ \sigma_c \\ \sigma_d \end{Bmatrix}_{gauss}$$

Equation 3-34.

where

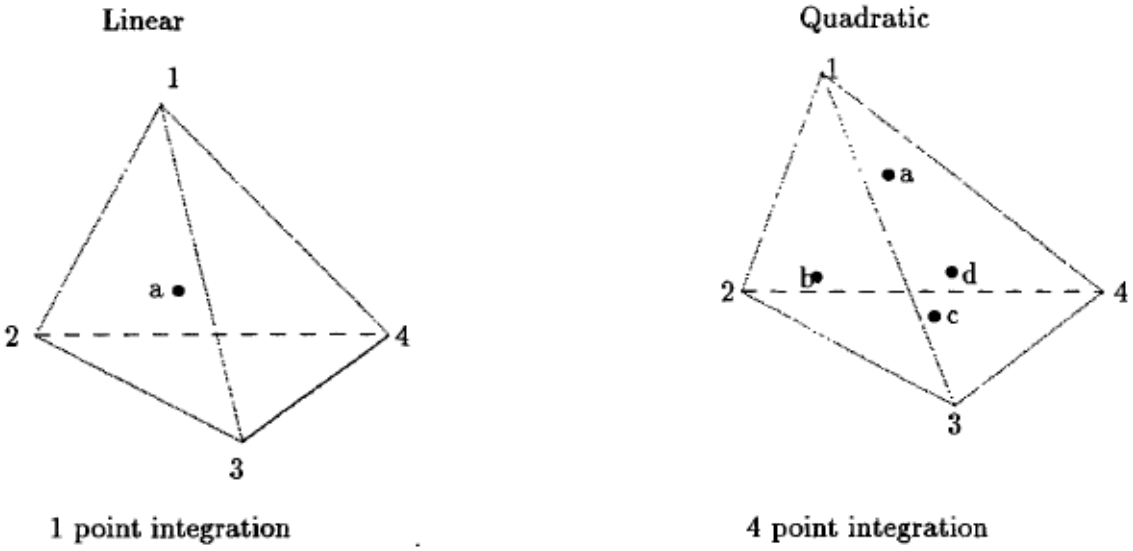
$$A = \frac{1-\beta}{1-4\beta} \quad \text{and} \quad B = \frac{\beta}{1-4\beta}$$

This process must be operated on every component of stress vector. No extrapolation is required for 1-point integration.



GRID	L_1	L_2	L_3	L_4
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	1/2	1/2	0	0
6	1/2	0	1/2	0
7	1/2	0	0	1/2
8	0	1/2	1/2	0
9	0	0	1/2	1/2
10	0	1/2	0	1/2

Figure 3-3. Volume Coordinates for Tetrahedron



1 point integration

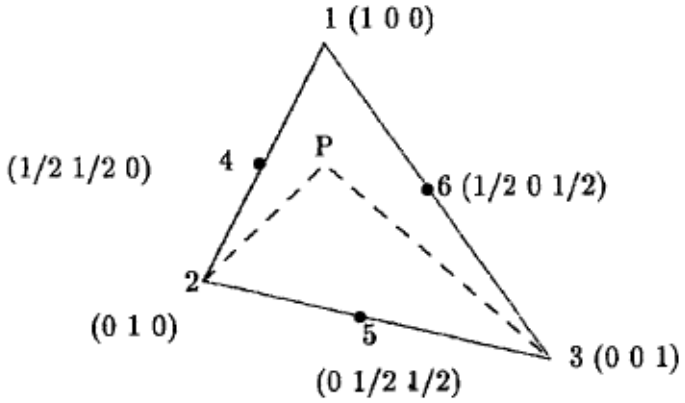
4 point integration

	Point	Weight	ξ	η	ζ	Error
Linear	a	1	1/4	1/4	1/4	$0(h^2)$
Quadratic	a	1/4	α	β	β	$0(h^3)$
	b	1/4	β	α	β	
	c	1/4	β	β	α	
	d	1/4	β	β	β	

$\alpha = 1 - 3\beta = .58541020$

$\beta = 1 / (5 + \sqrt{5}) = 0.13819660$

Figure 3-4. Gaussian Quadrature for Tetrahedron



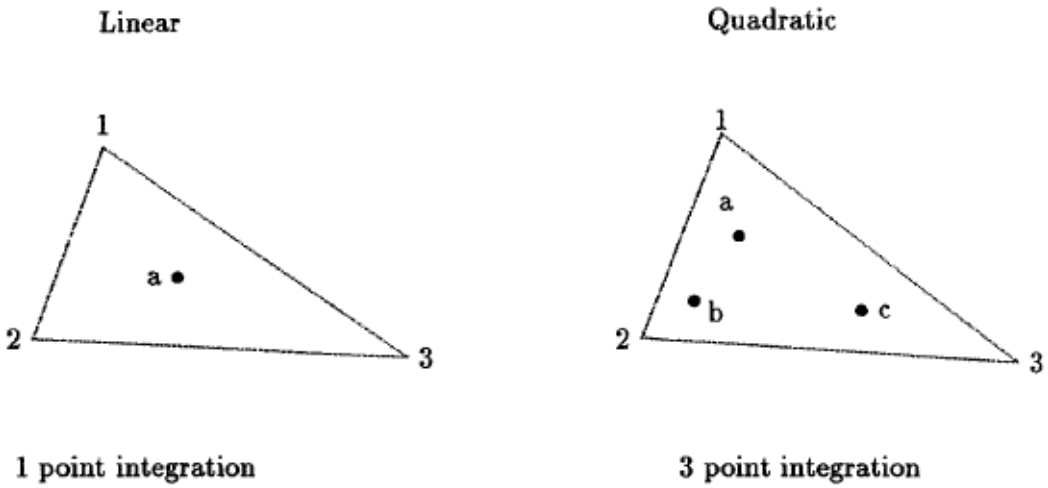
* Area coordinate system ($\xi\ \eta\ \zeta$) such that

$$\xi = \frac{\text{Area}(P23)}{\text{Area}(123)},\ etc.$$

* Shape Functions:

$$\begin{aligned} N_1' &= \xi - \frac{1}{2}N_4 - \frac{1}{2}N_6 \\ N_2' &= \eta - \frac{1}{2}N_4 - \frac{1}{2}N_5 \\ N_3' &= \zeta - \frac{1}{2}N_5 - \frac{1}{2}N_6 \\ N_4' &= 4\xi\eta\delta_4 \\ N_5' &= 4\eta\zeta\delta_5 \\ N_6' &= 4\xi\eta\delta_6 \end{aligned}$$

Figure 3-5. Shape Functions for Pressure Load



	Point	Weight	ξ	η	ζ	Error
Linear	a	1	1/3	1/3	1/3	$0(h^2)$
Quadratic	a	1/3	2/3	1/6	1/6	$0(h^3)$
	b	1/3	1/6	2/3	1/6	
	c	1/3	1/6	1/6	2/3	

Figure 3-6. Gaussian Quadrature for Pressure Load

Chapter 4: Material support

4.1 Material overview

SOL 401 supports the following material types.

- The MAT1 and MATT1 (temperature dependent) bulk entries define isotropic materials.
- The MAT3 and MATT3 (temperature dependent) bulk entries define isotropic materials.
- The MAT9 and MATT9 (temperature dependent) bulk entries define anisotropic materials.
- The MAT11 and MATT11 (temperature dependent) bulk entries define orthotropic materials.
- Plastic and creep materials can optionally be assigned to the 3D solid elements, axisymmetric elements, the plane stress elements, and the plane strain elements. You can enable one or both plasticity/creep in all subcases, or in specific subcases.

4.2 Support for plasticity analysis

You can perform a plasticity analysis in SOL 401. The constitutive model is a standard elastic-plastic model that allows for bilinear and multilinear stress-strain curve representations. For multilinear representations, tabular data is used to define the stress-strain curve. You can specify that the strain values in the tabular data are either total strain or plastic strain. You can also define material properties as temperature-dependent. At present, the von Mises yield criterion is the only yield criterion that is supported.

For bilinear stress-strain representations, you can select either isotropic, kinematic, or combined hardening. For multilinear stress-strain representations, isotropic hardening is the only hardening rule available.

You can selectively enable and disable plasticity effects in subcases. If plasticity-enabled subcases are sequentially dependent, the plasticity state variables at the end of one subcase are used as the plasticity state variables at the beginning of the next subcase. If a plasticity-disabled subcase is placed between plasticity-enabled subcases, and the subcases are all sequentially dependent, the plasticity state variables at the end of the preceding plasticity-enabled subcase are used as the plasticity state variables at the beginning of the later plasticity-enabled subcase.

However, with the exception of special situations, we recommend you avoid placing a sequentially dependent plasticity-disabled subcase after a plasticity-enabled subcase. Doing so eliminates the possibility that the analysis does not account for changes to the plasticity state variables that might result from the loading in the plasticity-disabled subcase.

Creep analysis is also supported in SOL 401. You can enable one or both plasticity/creep in all subcases, or you can enable one or both in specific subcases.

For more information on creep analysis in SOL 401, see Support for creep analysis.

User interface

- The MATNL parameter. With the MATNL parameter, you can globally switch the plasticity analysis capability on or off.
- The PLASTIC parameter is available on the NLCNTL bulk entry to optionally turn off the plasticity capability in a subcase.
- The MATS1 bulk entry allows you to define stress versus plastic strain tabular data.

To activate the plasticity analysis capability in SOL 401, do the following:

1. Reference both the MAT1 and MATS1 bulk entries in the regions where plasticity occurs.
2. Specify PARAM,MATNL,1.
3. Include a NLCNTL case control command that points to a NLCNTL bulk entry.
4. On the NLCNTL bulk entry, specify any applicable parameters.

If your input file contains subcases, and you want to include the effects of plasticity in specific subcases, but not others, you have two options.

Option 1: Use a global NLCNTL case control command

1. Specify PARAM,MATNL,1.
2. Include a NLCNTL case control command above the subcases that points to a NLCNTL bulk entry.
3. On the NLCNTL bulk entry pointed to by the global NLCNTL case control command, specify any applicable parameters.
4. In the subcases that you want to disable the plasticity analysis capability, include a NLCNTL case control command that points to a NLCNTL bulk entry.
5. On the NLCNTL bulk entry pointed to by the NLCNTL case control commands in the subcases, specify “PLASTIC” in a PARAMi field and “NO” in the corresponding VALUEi field.

Option 2: Include NLCNTL case control commands in every subcase

1. Specify PARAM,MATNL,1.
2. Include NLCNTL case control commands in each subcase. Multiple NLCNTL case control commands can point to a single NLCNTL bulk entry.
3. In subcases that you want to enable the plasticity analysis capability, have the NLCNTL case control command point to a NLCNTL bulk entry with any applicable parameters specified.
4. In subcases that you want to disable the plasticity analysis capability, have the NLCNTL case control command point to a NLCNTL bulk entry that has “PLASTIC” specified in a PARAMi field and “NO” specified in the corresponding VALUEi field.

In a SOL 401 plasticity analysis, the property bulk entry referenced by all non-rigid elements must reference a MAT1 bulk entry and a MATS1 bulk entry that have the same material identification

number. If the properties on the MAT1 bulk entry are temperature-dependent, include a MATT1 bulk entry with the same material identification number.

On the MATS1 bulk entry, specify TYPE = “PLASTIC” or “PLSTRN” to select the strain type in tabular data used to describe a multilinear stress-strain curve. Specify TYPE = “PLASTIC” if you want to use total strains. Specify TYPE = “PLSTRN” if you want to use plastic strains. Total and plastic strains are related as follows:

$$\varepsilon_{plastic} = \varepsilon_{total} - \varepsilon_{elastic}$$

where

$$\varepsilon_{elastic} = \frac{\sigma}{E}$$

To describe a bilinear stress-strain curve, specify either TYPE = “PLASTIC” or “PLSTRN” and enter the work hardening slope, H, directly.

For additional information, see the MATS1 bulk entry in the *NX Nastran Quick Reference Guide*.

Time step control

Unlike creep analysis in SOL 401, there is no adaptive time stepping for plasticity analysis. The time steps are defined directly by the solution times. To define solution times for the plasticity analysis, include a TSTEP case control command in your input file that points to a TSTEP1 bulk entry. On the TSTEP1 bulk entry, specify the solution times and the solution times you want results output.

4.3 Overview of Plasticity

For plasticity, SOL 401 includes a von Mises yield function with an associated flow rule. Available options include isotropic hardening, kinematic hardening, and combined hardening. The Ziegler-Prager hardening rules are available for kinematic and combined hardening.

The plasticity model must also be calibrated with uniaxial stress-strain data. The strain information must be in the unitless form of length/length. Classical plasticity models include the following three fundamental ingredients.

1. A yield function or yield criterion defines the limit of elastic behavior for a general state of stress.

   The yield function may be thought of as a surface in a six dimensional stress space. It divides the stress space into two regions. Points inside the yield surface are characterized by elastic stress-strain behavior while stress states on the yield surface are at the limit of elastic behavior. The yield function may be written as a function of stress and a hardening parameter k.

   $$F\left(\sigma_{ij}\right) - k = 0$$

   The von Mises yield function for an isotropic hardening material may be stated as:

   $$F = J_2 - \frac{1}{3}\sigma_y^2 = 0$$

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

   J_2 is the second invariant of the deviatoric stress tensor. The deviatoric stresses, s_{ij}, are given by:

$$s_{ij} = \sigma_{ij} - \sigma_m \delta_{ij}$$

where σ_m is the mean or average normal stress, and δ_{ij} is the Kronecker delta.

Isotropic hardening assumes that the uniaxial yield stress, σ_y, is a function of plastic straining.

For kinematic or combined hardening, the von Mises yield function is

$$F = J_2' - \frac{1}{3}\sigma_y^2 = 0$$

$$J_2' = \frac{1}{2}\left(s_{ij} - \alpha_{ij}\right)\left(s_{ij} - \alpha_{ij}\right)$$

J'_2 is the second invariant of the shift stress that is defined as the deviatoric stress minus the back stress. The back stress may also be thought of as the position of the center of the yield surface in the six-dimensional stress space. For initially isotropic materials, the back stress tensor components are initially zero.

2. A flow rule determines the relative magnitudes of the components of the plastic strain increment tensor.

   The software uses an associated flow rule in which the plastic potential function, g, is the same as the yield function, f, and the components of the plastic strain increment are given by:

$$d\varepsilon_{ij}^p = d\lambda \frac{\partial f}{\partial \sigma_{ij}}$$

   In vector form, for the von Mises yield surface with kinematic hardening, the plastic flow vector is given by:

$$\left\{d\varepsilon^p\right\} = d\lambda \frac{\partial g}{\partial \{\sigma\}} = d\lambda' \begin{Bmatrix} s_{xx} - \alpha_{xx} \\ s_{yy} - \alpha_{yy} \\ s_{zz} - \alpha_{zz} \\ 2\left(\tau_{xy} - \alpha_{xy}\right) \\ 2\left(\tau_{xz} - \alpha_{xz}\right) \\ 2\left(\tau_{yz} - \alpha_{yz}\right) \end{Bmatrix}$$

   For isotropic hardening, this equation reduces to the well-known Prandtl-Reuss equations.

3. A hardening rule defines the changes in the yield function as a result of plastic straining.

   Isotropic Hardening

   When you select isotropic hardening, the software uses a piece-wise linear stress-strain curve. The isotropic hardening assumption isn't very realistic for most materials subjected to cyclic loading. However, it is relatively simple and efficient.

   Isotropic hardening assumes that the yield surface expands uniformly as a result of plastic straining. This assumption is achieved by making the yield stress a function of the integrated effective plastic strain increments, which for a von Mises material is:

$$\sigma_y = \sigma_y\left(\int d\varepsilon_p\right) = \sigma_y\left(\int \sqrt{\frac{2}{3}\, d\varepsilon_{ij}^p\, d\varepsilon_{ij}^p}\right)$$

The slope of the stress plastic strain curve, E_p, is called the plastic modulus. It can be obtained from the uniaxial stress-strain curve and is defined by:

$$E_p = \frac{d\sigma_e}{d\varepsilon_p}$$

For a von Mises material, the effective stress is given by:

$$\sigma_e = \sqrt{3J_2} = \sqrt{\frac{3}{2} s_{ij} s_{ij}}$$

$$= \sqrt{\frac{1}{2}\left\{\left(\sigma_{xx} - \sigma_{yy}\right)^2 + \left(\sigma_{yy} - \sigma_{zz}\right)^2 + \left(\sigma_{zz} - \sigma_{xx}\right)^2 + 6\tau_{xy}^2 + 6\tau_{yz}^2 + 6\tau_{zx}^2\right\}}$$

Kinematic Hardening

When you select kinematic hardening, the software assumes a bilinear stress-strain curve. If the material database contains a multilinear representation, only the yield point and the tangent modulus of the first segment beyond the yield point are used to characterize the stress-strain behavior.

Kinematic hardening assumes that the yield surface translates in the stress space but doesn't change size or shape. The yield stress, σ_y, doesn't change, but the back stress, α_{ij}, is a function of plastic straining. The Ziegler-Prager kinematic hardening is one of the most widely used models. This model assumes that the back stress increment is in the direction of the stress minus the back stress.

$$d\alpha_{ij} = d\mu\left(\sigma_{ij} - \alpha_{ij}\right)$$

The factor, $d\mu$, depends on the plastic strain history.

Ziegler-Prager Combined Kinematic Isotropic Hardening

When you select combined kinematic isotropic hardening, the software assumes a bilinear stress-strain curve. If the material database contains a multilinear representation, only the yield point and the tangent modulus of the first segment beyond the yield point are used to characterize the stress-strain behavior.

Combined hardening assumes that the yield surface both expands and translates in the stress space. The plastic strain increment is composed of two components shown in the following equations:

$$d\varepsilon_{ij}^p = d\varepsilon_{ij}^i + d\varepsilon_{ij}^k$$

$$d\varepsilon_{ij}^i = M\, d\varepsilon_{ij}^p$$

$$d\varepsilon_{ij}^k = \left(1 - M\right) d\varepsilon_{ij}^p$$

$$0 \le M \le 1$$

where,

$d\varepsilon_{ij}^i$ is associated with the isotropic expansion of the yield surface,

and $d\varepsilon_{ij}^k$ is associated with the translation of the yield surface.

The reduced effective plastic strain associated with isotropic hardening is related to the effective plastic strain by the following:

$$\bar{\varepsilon}_p = M\varepsilon_p$$

The back stress increment for Prager combined hardening is:

$$d\alpha_{ij} = c(1-M)d\varepsilon_{ij}^p$$

The back stress increment for Ziegler-Prager hardening is:

$$d\alpha_{ij} = d\mu(1-M)\left(\sigma_{ij} - \alpha_{ij}\right)$$

where M=0.5 in SOL 401.

The software computes plastic strain increments using a backward Euler technique without sub-incrementation. For the isotropic hardening model:

$$\sigma = D_e\left(\varepsilon - \varepsilon^p\right)$$

$$TYF = f\left(\sigma\right) - \sigma_y\left(d\bar{\varepsilon}^p\right)$$

$$d\lambda = \frac{TYF}{\left\{\frac{\delta f}{\delta\sigma}\right\}^T D_e \left\{\frac{\delta f}{\delta\sigma}\right\} + E_p}$$

$$\Delta\varepsilon^p = d\lambda\frac{\delta f}{\delta\sigma}$$

$$\varepsilon_{i+1}^p = \varepsilon_i^p + \Delta\varepsilon^p$$

where D_e is the elastic modulus matrix, ε^p is the plastic strain, $d\bar{\varepsilon}^p$ is the effective plastic strain, and *TYF* is the von Mises yield function.

The software will compute a consistent tangent modulus for use in generating the tangent stiffness matrix. When this procedure is used with a full Newton-Raphson iteration, quadratic convergence can be obtained. A detailed description of this procedure may be found in Crisfield, 1991.

References:

- Crisfield, M. A., *Non-linear Finite Element Analysis of Solids and Structures* (Chichester: John Wiley & Sons, 1991).
- Chakrabarty, J., *Theory Of Plasticity* (New York: McGraw-Hill Book Company, 1987), 55-119.
- Chen, W. F. and Han, D. J., *Plasticity for Structural Engineers*, (New York: Springer-Verlag, 1988), 239-281.
- Lemaitre, J. and Chaboche, J. L., *Mechanics of Solid Materials*, (Cambridge: Cambridge University Press, 1990), 161-240.

4.4 Creep analysis

You can perform creep analysis in SOL 401 using the Bailey-Norton model. All elements supported in SOL 401, except for the rigid elements, support the creep material defined using the MATCRP bulk entry. The Bailey-Norton model represents isotropic creep with optional temperature-dependency.

You can selectively enable and disable creep effects in subcases. If creep-enabled subcases are sequentially dependent, the total accumulated creep strain at the end of one subcase is used as the initial creep strain for the next subcase. If a creep-disabled subcase is placed between creep-enabled subcases, and the subcases are all sequentially dependent, the total accumulated creep strain at the end of the preceding creep-enabled subcase is used as the initial creep strain for the later creep-enabled subcase because no incremental creep strain arises in creep-disabled subcases.

Creep analysis implementation

The Bailey-Norton model relates creep strain to stress and time as follows:

$$\varepsilon_e^c = A\sigma_e^B t^D$$

Equation 4-1.

where $\varepsilon_e{}^c$ is the effective creep strain, σ_e is effective stress, t is time, and A, B, and D are user-defined coefficients. Because the model uses effective stress and effective creep strain, the values for the coefficients are directly relatable to results from uniaxial testing.

In the Bailey-Norton model, temperature is not accounted for explicitly. To account for temperature-dependence, you can define the coefficients as tabular functions of temperature.

For some very simple cases, you can use Equation 4-1 directly to calculate the effective creep strain as a function of time. However, for the general case where temperature and stress vary, and computer simulation is required, Equation 4-1 is applied incrementally over a finite number of time steps.

During the creep analysis, the incremental creep strain for each time step is calculated by numerically integrating the instantaneous creep strain rate. The formula for creep strain rate is obtained from the following flow rule:

$$\dot{\varepsilon}_{ij}^c = \frac{3}{2}\frac{\dot{\varepsilon}_e^c}{\sigma_e}S_{ij}$$

Equation 4-2.

where $\dot{\varepsilon}_{ij}{}^c$ are the components of the creep strain rate tensor, $\dot{\varepsilon}_e{}^c$ is the effective creep strain rate, and S_{ij} are the components of the deviatoric stress tensor.

The effective creep strain rate is obtained by differentiating Equation 4-1 with respect to time.

To evaluate each increment of creep strain, the software performs a numerical integration based on the generalized trapezoidal rule as follows:

$$\left(\Delta\varepsilon_{ij}^c\right)_n = \Delta t_n\left[(1-\beta)\left(\dot{\varepsilon}_{ij}^c\right)_{n-1} + \beta\left(\dot{\varepsilon}_{ij}^c\right)_n\right]$$

Equation 4-3.

where $\Delta t_n = t_n - t_{n-1}$ is the duration of the subinterval, and β is a user-defined numerical integration parameter. Generally, the default value for β of 0.5 is appropriate.

User interface

- With the MATNL parameter, you can globally switch the creep analysis capability on or off.

  For more information, see the MATNL parameter.

- Parameters are available for use with the NLCNTL bulk entry. These parameters allow you to turn off the creep capability in subcases, control adaptive time stepping or define a constant time step, and define the integration factor in Equation 4-3.

  For more information, see the NLCNTL bulk entry in the *NX Nastran Quick Reference Guide*.

- With the MATCRP bulk entry, you define parameters related to the creep constitutive model.

  For more information, see the MATCRP bulk entry in the *NX Nastran Quick Reference Guide*.

To activate the creep analysis capability in SOL 401, do the following:

1. Reference both the MAT1 and MATCRP bulk entries in the regions where creep occurs.
2. Specify PARAM,MATNL,1.
3. Include a NLCNTL case control command that points to a NLCNTL bulk entry.
4. On the NLCNTL bulk entry, specify any applicable parameters.

If your input file contains subcases, and you want to include the effects of creep in specific subcases, but not others, you have two options.

Option 1: Use a global NLCNTL case control command

1. Specify PARAM,MATNL,1.
2. Include a NLCNTL case control command above the subcases that points to a NLCNTL bulk entry.
3. On the NLCNTL bulk entry pointed to by the global NLCNTL case control command, specify any applicable parameters.
4. In the subcases that you want to disable the creep analysis capability, include a NLCNTL case control command that points to a NLCNTL bulk entry.
5. On the NLCNTL bulk entry pointed to by the NLCNTL case control commands in the subcases, specify “CREEP” in a PARAMi field and “NO” in the corresponding VALUEi field.

Option 2: Include NLCNTL case control commands in every subcase

1. Specify PARAM,MATNL,1.
2. Include NLCNTL case control commands in each subcase. Multiple NLCNTL case control commands can point to a single NLCNTL bulk entry.
3. In subcases that you want to enable the creep analysis capability, have the NLCNTL case control command point to a NLCNTL bulk entry with any applicable parameters specified.

4. In subcases that you want to disable the creep analysis capability, have the NLCNTL case control command point to a NLCNTL bulk entry that has “CREEP” specified in a PARAMi field and “NO” specified in the corresponding VALUEi field.

To directly define solution times for the creep analysis, include a TSTEP case control command in your input file that points to a TSTEP1 bulk entry. On the TSTEP1 bulk entry, you can specify the solution times and specify which solution times you want results output.

The solution times you specify on the TSTEP1 bulk entry may result in time steps that are either too coarse to produce accurate results, or too fine to produce results efficiently. To assist you in avoiding such problems, the software uses adaptive time stepping by default. You can tweak the adaptive time stepping algorithm or override adaptive time stepping altogether with new parameters for the NLCNTL bulk entry. For more information on adaptive time stepping, see Time step control.

In a SOL 401 creep analysis, the property bulk entry referenced by all non-rigid elements must reference a MAT1 bulk entry and a MATCRP bulk entry that have the same material identification number. If the properties on the MAT1 bulk entry are temperature-dependent, include a MATT1 bulk entry with the same material identification number.

You use the MATCRP bulk entry to specify:

- The stress threshold below which creep does not occur.
- The hardening rule to apply.
- The coefficients in the Bailey-Norton creep model.

You can specify the coefficients in the Bailey-Norton creep model as either constant or as a function of temperature. To specify a coefficient as temperature-dependent, enter the identification number of a TABLEM1 bulk entry in the corresponding A, B, or D field of the MATCRP bulk entry. On the TABLEM1 bulk entry, enter tabular data that describes how the coefficient varies with temperature. At present, a MATCRP bulk entry cannot reference a TABLEM2, TABLEM3, or TABLEM4 bulk entry.

Time step control

During a SOL 401 creep analysis, the solution times depend on:

- How you specify the TSTEP1 bulk entry.
- How you specify the time stepping parameters.

The time steps that result from the TSTEP1 bulk entry specification may be too coarse to produce accurate results, or too fine to produce results efficiently. By default, the software uses an adaptive time stepping algorithm to avoid such problems.

You can tweak the adaptive time stepping algorithm or override adaptive time stepping altogether and use a constant time step with new parameters for the NLCNTL bulk entry. The new parameters are:

CRCERAT	Ratio of maximum creep increment to elastic strain that is used to adaptively specify the next time step.
CRCINC	Maximum creep increment that is used to adaptively specify the next time step.
CRICOFF	Creep strain increment below which the next time step is the product of the current time step and the maximum time step multiplying factor.

CRINFAC	Numerical integration parameter. See Equation 4-3.
CRINTS	Initial time step for adaptive time stepping, or the constant time step if adaptive time stepping is overridden.
CRMFMN	Minimum time step multiplying factor.
CRMFMX	Maximum time step multiplying factor.
CRSBCDT	Controls whether the first time step in a sequential subcase uses the initial time step or the time step calculated at the end of the previous subcase.
CRTEABS	Maximum absolute truncation error that is used to adaptively specify the next time step.
CRTECO	Effective creep strain below which CRTEABS is used, and above which CRTEREL is used.
CRTEREL	Maximum relative truncation error that is used to adaptively specify the next time step.
CRTSC	Specifies the time stepping method.
CRTSMN	Minimum time step.
CRTSMX	Maximum time step.

The CRTSC parameter controls the overall time stepping behavior. Use the CRTSC parameter to:

- Select the adaptive time stepping algorithm that the software uses to calculate the next time step.
- Override adaptive time stepping altogether and have the software use the value of the CRINTS parameter as a constant time step.

The adaptive time stepping algorithm options include the following:

- The next time step is based on the maximum creep strain increment criterion. You specify the maximum creep strain increment with the CRCINC parameter.
- The next time step is based on the ratio of maximum creep increment to elastic strain criterion. You specify the ratio of maximum creep increment to elastic strain with the CRCERAT parameter.
- The next time step is based on the maximum truncation error criterion. For this option, you have three sub-options.
    - Use the maximum absolute truncation error. You specify the maximum absolute truncation error with the CRTEABS parameter.
    - Use the maximum relative truncation error. You specify the maximum relative truncation error with the CRTEREL parameter.
    - Use the maximum absolute truncation error if the creep strain is less than the value specified by the CRTECO parameter, and use the maximum relative truncation error if the creep strain is greater than the value specified by the CRTECO parameter.

- The next time step is the shortest time step calculated by any combination of the maximum creep strain increment, ratio of maximum creep increment to elastic strain, and maximum truncation error criteria.

When the creep simulation begins, the value of the CRINTS parameter is always used as the first time step. If adaptive time stepping is overridden, the value of the CRINTS parameter is used as a constant time step throughout the simulation.

If adaptive time stepping is not overridden, after each time step the software compares the calculated creep strain increment to the value of the CRICOFF parameter. If the creep strain increment is greater than the value of the CRICOFF parameter, the software uses the adaptive time stepping algorithm to calculate the next time step. If the creep strain increment is less than the value of the CRICOFF parameter, the software uses the product of the current time step and the value of the CRMFMX parameter as the next time step.

If the software uses the adaptive time stepping algorithm to calculate the next time step, the next time step is compared to the product of the current time step and the value for the CRMFMN parameter. If the next time step is smaller than the product of the current time step and the value for the CRMFMN parameter, the software halves the current time step, recalculates the current creep strain increment, and reenters the algorithm at the point the creep strain increment is compared to the value of the CRICOFF parameter. If the next time step is larger than the product of the current time step and the value of the CRMFMN parameter, the software keeps the next time step.

The next time step is then compared against the values of the CRTSMX and CRTSMN parameters. First, the software checks to see if the value of the CRTSMX parameter is 0.0. If so, the software accepts the value for the next time step and uses it to compute the next creep strain increment. If the value of the CRTSMX parameter is not set to 0.0, the next time step is compared to the value of the CRTSMX parameter. If the next time step is larger than the value of the CRTSMX parameter, the software uses the value of the CRTSMX parameter as the next time step and uses it to compute the next creep strain increment. If the next time step is smaller than the value of the CRTSMX parameter, the next time step is compared to the value of the CRTSMN parameter. If the next time step is smaller than the value for the CRTSMN parameter, the software halves the current time step, recalculates the current creep strain increment, and reenters the algorithm at the point the creep strain increment is compared to the CRICOFF parameter. If the next time step is larger than the value of the CRTSMN parameter, the software accepts the value for the next time step and uses it to compute the next creep strain increment.

The adaptive time stepping algorithm is summarized by the following flowchart. In the flowchart, the notation for the value of a parameter is $V_{parameter\ name}$.

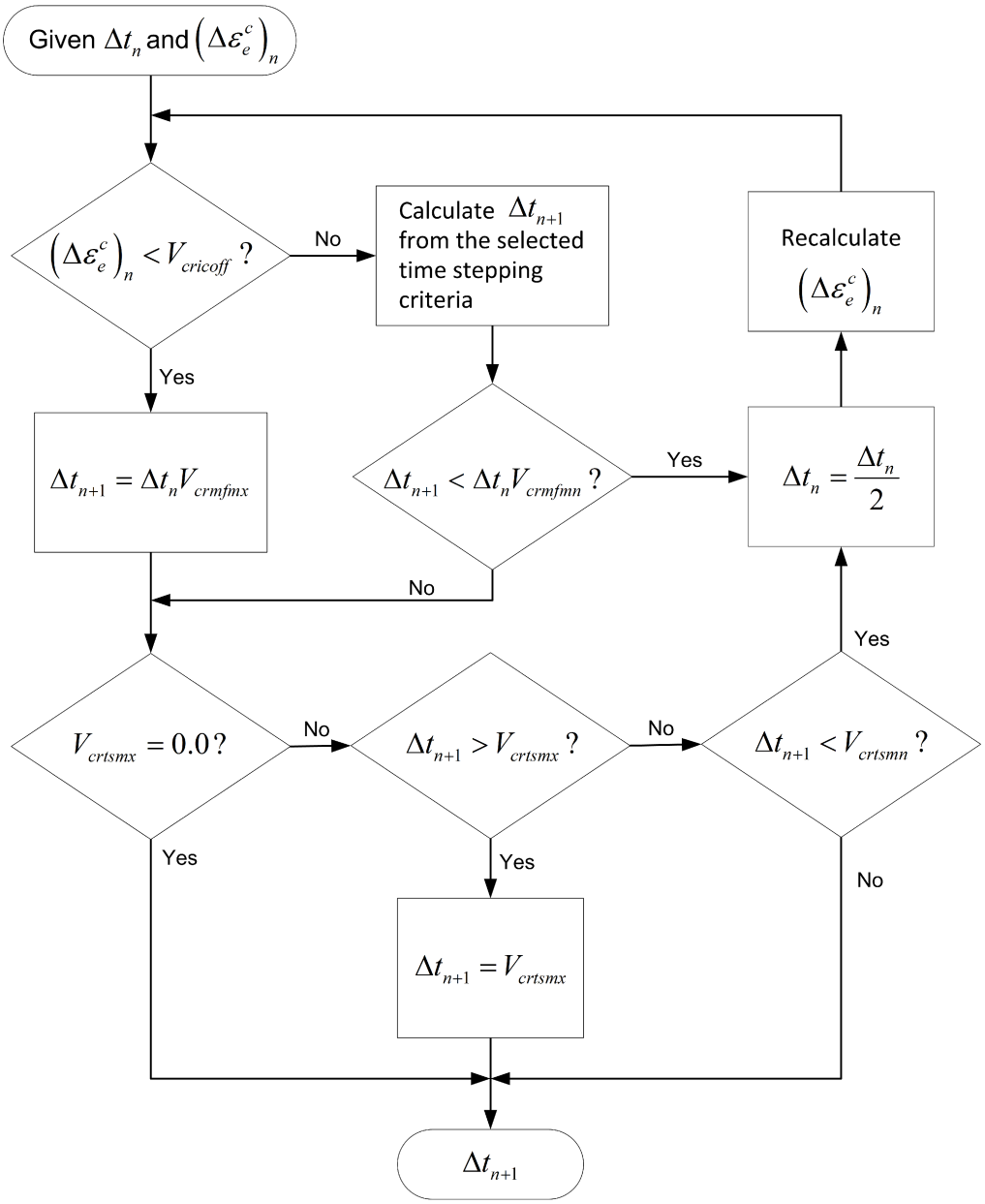


Figure 4-1. Flowchart of adaptive time stepping algorithm

Regardless of whether you are using adaptive time stepping or a constant time step, the solution times you specify with the TENDi and NINCi fields on TSTEP1 bulk entries are always honored. At all times during the creep simulation, if the next time step would result in skipping over a TSTEP1-defined solution time, the software truncates the next time step so that a solve occurs at that solution time.

As a best practice, consider using the TSTEP1 bulk entry to specify only the times you want the results output, and let the adaptive time stepping algorithm determine all the other solution times.

For additional information on the creep-related parameters, see the NLCNTL bulk entry in the NX Nastran Quick Reference Guide.

4.5 Overview of the Creep Model

Classical creep models consist of the following three fundamental parts.

- A uniaxial creep equation expresses the effective creep strain or creep strain rate as a function of effective stress, temperature, and time. In a state of uniaxial stress, the effective creep strain and effective stress reduce to the uniaxial creep strain and the uniaxial stress. The effective creep strain in terms of the creep strain tensor components is given by:

$$\varepsilon_c = \sqrt{\frac{2}{3}\varepsilon_{ij}^c \varepsilon_{ij}^c}$$
$$= \sqrt{\frac{2}{3}\left\{\left\{\varepsilon_{xx}^p\right\}^2 + \left\{\varepsilon_{yy}^p\right\}^2 + \left\{\varepsilon_{zz}^p\right\}^2 + 2\left[\left\{\varepsilon_{xy}^p\right\}^2 + \left\{\varepsilon_{xz}^p\right\}^2 + \left\{\varepsilon_{yz}^p\right\}^2\right]\right\}}$$

  The effective stress in terms of the stress tensor components ij and deviatoric stress tensor components s_{ij} is given by:

$$\sigma_e = \sqrt{\frac{3}{2}s_{ij}s_{ij}}$$
$$= \sqrt{\frac{1}{2}\left\{\left(\sigma_{xx} - \sigma_{yy}\right)^2 + \left(\sigma_{yy} - \sigma_{zz}\right)^2 + \left(\sigma_{zz} - \sigma_{xx}\right)^2 + 6\tau_{xy}^2 + 6\tau_{yz}^2 + 6\tau_{zx}^2\right\}}$$

- A flow rule determines the creep strain rate tensor components for a general state of stress.
- A hardening rule determines creep strain rates from the uniaxial rate equation under changing stress and temperature.

Overview of the Hardening Rules for Creep Models

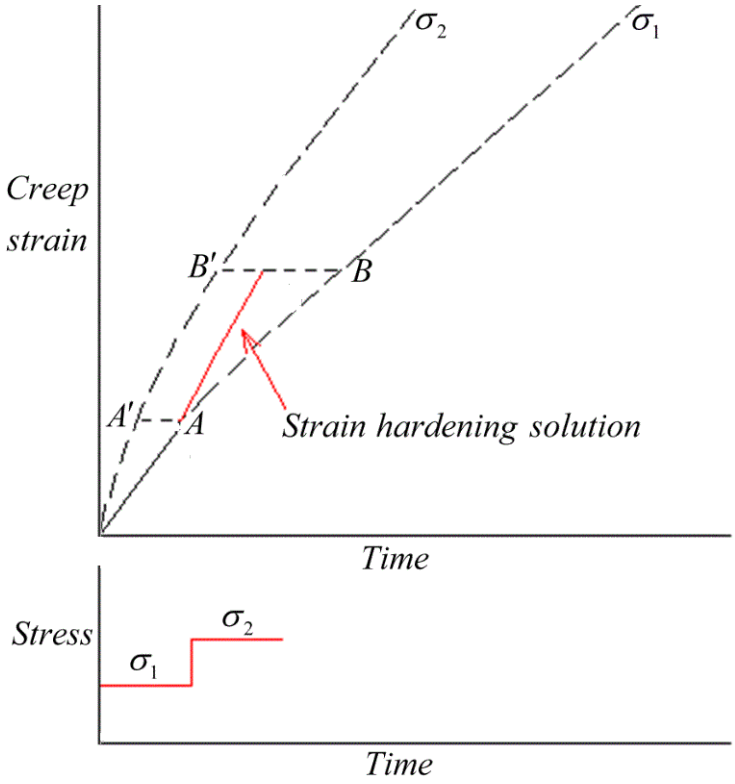
The hardening rule provides the mathematical means to determine the effective creep rate under changing temperature and/or stress. The creep hardening rule is used to determine the current material state relative to further creep straining. It plays a role similar to the yield surface in plasticity theory.

You can select one of the following hardening rules for the creep model using the HARD field on the MATCRP bulk entry.

- HARD = STRAIN (default) selects strain hardening.
- HARD = TIME selects time hardening.

Strain Hardening

The strain hardening rule assumes that the material state is determined by the accumulated effective creep strain. When stress or temperature changes, the shift from one creep curve to another is based on the accumulated effective creep strain. This shift is illustrated in the following figure.

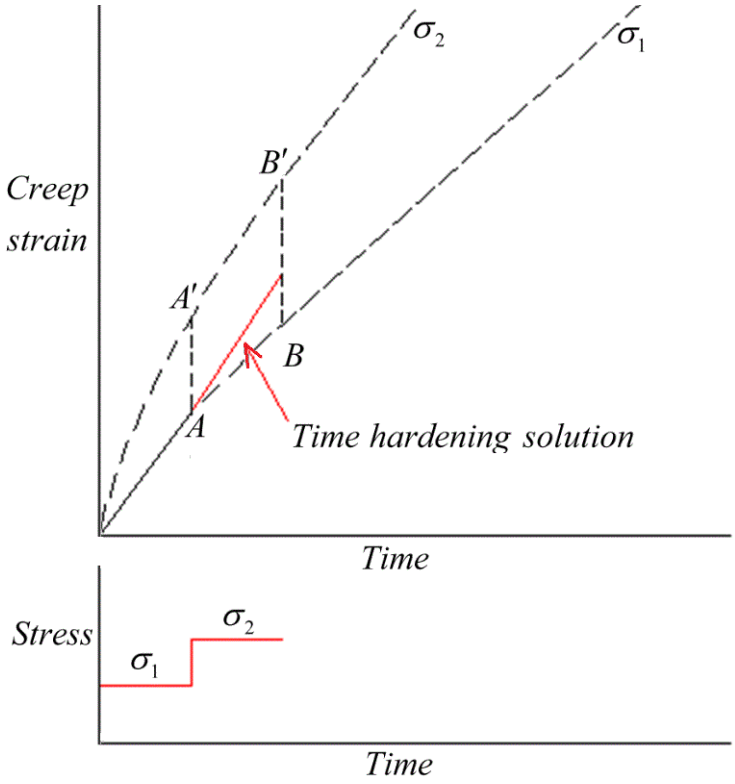


The strain hardening rule is generally considered to be superior to the time hardening rule and is, therefore, the default for the software's creep equations. Under conditions of constant stress and temperature, the time hardening and strain hardening assumptions produce identical results.

Time Hardening

The time hardening rule uses time as a material state variable. It assumes that the material state is determined by the length of time the material has been creeping.

When stress or temperature changes, the shift from one creep curve to another is based on the accumulated creep time. This shift is illustrated in the following figure.



The time hardening rule is usually not very realistic when very large changes in temperature or stress occur over the interval of interest.

Maximum Truncation Error Criterion

With the maximum truncation error criterion, NX Nastran also calculates the next time step by scaling the current time step. The value of the CRTECO parameter on the NLCNTL bulk entry determines whether the maximum absolute truncation error or the maximum relative truncation error is used to adaptively specify the next time step.

- If the effective creep strain is less than the value of the CRTECO parameter, the maximum absolute truncation error is used. The maximum absolute truncation error is specified with the CRTEABS parameter on the NLCNTL bulk entry.

- If the effective creep strain is greater than the value of the CRTECO parameter, the maximum relative truncation error is used. The maximum relative truncation error is specified with the CRTEREL parameter on the NLCNTL bulk entry.

The next time step is calculated by scaling the current time. When the maximum absolute truncation error is applied, the software computes the next time step as:

$$\Delta t_{n+1} = \Delta t_n \frac{CRTEABS}{|e|}$$

When the maximum relative truncation error is applied, the software computes the next time step as:

$$\Delta t_{n+1} = \Delta t_n \left| \frac{\left(\Delta \varepsilon_e^c\right)_n}{e} \right| * \left(CRTEREL\right)$$

where e is the calculated truncation error, and $\Delta \varepsilon_e^c$ is the current effective creep strain increment. There are two approaches to calculate the truncation error e corresponding to the two different scenarios:

1. At the first time step or if β=0 or β=1, the two point rule truncation error is used.

2. From the second time step and 0 < β< 1, the three point rule truncation error is used.

Maximum creep strain increment criterion

With the maximum creep strain increment criterion, NX Nastran calculates the next time step by scaling the current time step as follows:

$$\Delta t_{n+1} = \Delta t_n \left(\frac{CRCINC}{\left| \left(\Delta \varepsilon_e^c\right)_n \right|} \right)$$

Equation 4-4.

where *CRCINC* is the value of the CRCINC parameter and $\Delta\varepsilon_e^c$ is the current effective creep strain increment.

Ratio of maximum creep increment to elastic strain criterion

With the ratio of maximum creep increment to elastic strain criterion, NX Nastran calculates the next time step by scaling the current time as follows:

$$\Delta t_{n+1} = \Delta t_n \left(\frac{CRCERAT}{\left| \left(\Delta \varepsilon_e^c\right)_n / \left(\varepsilon_e^E\right)_n \right|} \right)$$

Equation 4-5.

where *CRCERAT* is the value of the CRCERAT parameter, $\Delta\varepsilon_e^c$ is the current effective creep strain increment, and ε_e^E is the current total effective elastic strain.

4.6 Disable plasticity and creep

The MATNL parameter allows you to switch all creep and/or plasticity effects on/off for all related elements.

When the MATNL parameter is set to 1, PARAM,MATNL,1 is defined to turn creep and/or plasticity effects on, the MATOVR bulk entry allows you to optionally disable the creep and plasticity effects off for the elements selected GROUP entry.

- Use TYPE = “ELEM” to reference a GROUP bulk entry that includes a list of elements. The MATOVR specification applies to the listed elements.

- Use TYPE = “PROP” to reference a GROUP bulk entry that includes a list of properties. The MATOVR specification applies to all elements that reference the properties listed in the GROUP bulk entry.

For more information, see the MATOVR bulk entry in the *NX Nastran Quick Reference Guide*.

Chapter 5: Contact conditions in SOL 401

5.1 Contact Summary

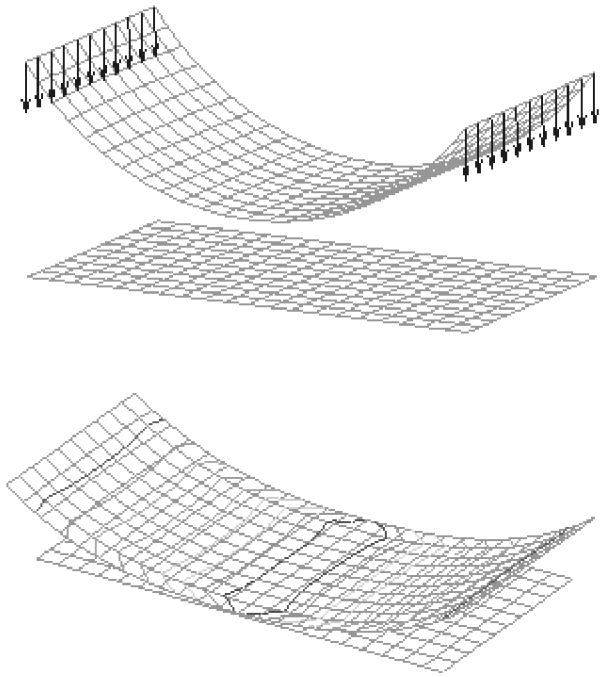
Surface-to-Surface Contact Summary

Surface-to-surface contact can be defined on the faces of the following elements.

- 3-D Solid elements CHEXA, CPENTA, CPYRAM, and CTETRA.

Surface-to-surface contact conditions allow the solution to search and detect when a pair of element faces come into contact. The contact conditions prevent the faces from penetrating and allow finite sliding with optional friction effects. The surface-to-surface contact source and target regions consist of solid element faces. From element faces in the source region, a top and bottom normal is projected. The software creates a contact element if:

- Any of the source element normals intersect with an element in the target region.
- The distance between the two faces is equal to or less than the defined separation distance.



Edge-to-Edge Contact Summary

Edge-to-edge contact can be defined on the edges of the following elements:

- Axisymmetric elements CTRAX3, CQUADX4, CTRAX6, CQUADX8.
- Plane stress elements CPLSTS3, CPLSTS4, CPLSTS6, CPLSTS8.

- Plane strain elements CPLSTN3, CPLSTN4, CPLSTN6, CPLSTN8.

Edge-to-edge contact conditions allow the solution to search and detect when a pair of element edges come into contact. The software creates contact elements if the distance between the edges is equal to or less than the search distance which you specify. The contact elements prevent the edges from penetrating and allow finite sliding with optional friction effects.

The axisymmetric, plane stress, and plane strain elements can be defined in either the XZ plane or in the XY plane. Edge-to-edge contact is supported in either orientation. The generalized plane strain element is not supported by glue or contact regions.

Subcase control

The BCSET case control command selects the contact conditions defined in the bulk data. The BCSET command must be above the subcases. As a result, a contact definition applies to all static subcases. A modal subcase which is sequentially dependent (default) uses the final stiffness from the previous static subcase. When contact is defined, the final stiffness from the static subcase includes the contact stiffness.

Solver options with contact

For models with 3D solid elements and contact, the element iterative solver is generally recommended over the default sparse solver for better performance. The SOL 401 element iterative solver is select by setting the SOLVER parameter on the NLCNTL bulk entry to “ELEMITER”.

5.2 Contact Regions

Edge Contact Regions

An edge contact region is a collection of axisymmetric, plane strain, and plane stress element free edges in a section of the model where you expect contact to occur. You create edge regions with the BEDGE bulk entry.

- To define a contact edge using the BEDGE entry, you enter the element ID along with the corner GRID IDs. You can define the edges on the BEDGE entry in any order.

Surface Contact Regions

A surface contact region is a collection of element free faces in a section of the model where you expect contact to occur. These regions are created using the solid element free faces (BSURFS and BCPROPS).

- The BSURFS entry is defined by its own unique ID and is a list of solid element IDs each followed by 3 grid points defining which face of the 3-D element to include in the contact region.

- The BCPROPS entry is defined by its own unique ID and is a list of solid element property IDs. The free faces of the solid elements selected with a property ID are automatically determined by the software.

A contact surface can be defined with any face of a solid element. Although parabolic faces with omitted midside nodes are permitted, their use could affect accuracy.

All region IDs defined with the BEDGE, BSURFS, or BCPROPS entries must be unique.

5.3 Contact Region Parameters

The contact region parameter OFFSET can be defined using the BCRPARA bulk entry. The CRID field on BCRPARA must match the ID used on one of the BSURFS, BCPROPS, and BEDGE bulk entries to be considered by the solution. The OFFSET parameter is supported with surface-to-surface and edge-to-edge contact definitions.

- Use the OFFSET field to account for a rigid layer between contact face or edge regions. For example, a model which has two metal surfaces coming into contact, and one of these has a ceramic coating. If the ceramic material stiffness is not significant enough to be included in the analysis, it may not have been specifically modeled, but the thickness it adds to the face of the metal may be important when considering the contact problem.

  You can also use the OFFSET field to analyze an interference fit problem if unconnected elements are modeled coincident. The offset value in this example can represent the theoretical interference.

5.4 Contact Pairs

A contact pair combines two contact regions, source and target, in which contact will be analyzed during the solution. Each contact pair can have its own unique friction value (if desired) and search distance.

The BCTSET bulk entry is used to define both edge-to-edge and surface-to-surface contact pairs. The CID field will need to match the value of ‘n' on the BCSET case control entry for the solution to recognize this contact definition. The BCSET case control command must be above the subcases. As a result, a contact definition applies to all static subcases. The SIDi and TIDi fields are used to define source and target regions respectively for a pair. As many pairs as desired can be included on a single BCTSET entry. Each pair can have a unique friction value (optional), a minimum search distance, and a maximum search distance. You can define the optional Coefficient of Friction field (FRICi) for each contact pair if you expect finite sliding to occur. When contact is detected, the solver uses this value to calculate any tangential contact forces by multiplying the normal contact force by FRICi.

The regions you select with SID and TID depend on the type of contact:

- For edge-to-edge contact pairs, SID and TID are contact regions defined with the BEDGE entry.
- For surface-to-surface contact pairs, SID and TID are contact regions defined with the BCPROPS and BSURFS entries.

The minimum and maximum search distance fields (MINDi and MAXDi) define a range in which the solver can initially determine if the distance between element edges or faces in a particular pair are within the threshold for creating contact elements. These values are used to determine where contact elements are created. The minimum distance can be negative if there is an interference condition modeled as overlapping regions.

The contact condition can update when large displacement effects are turned on with PARAM,LGDISP,1. The active and inactive contact elements are updated along with any force adjustments. When sliding occurs, the orientation of the current contact stiffness are updated. If large enough sliding occurs within a user defined tolerance, contact elements are recreated in the current deformed configuration, contact stiffness is recomputed, and contact tractions from the previous iteration are applied to the newly created contact elements.

Combining Contact Sets – BCTADD

You can optionally define multiple BCTSET/BCTPARM bulk entry sets, each set with unique contact set IDs (CSID), and then combine them with a single BCTADD bulk entry. The multiple BCTSET/BCTPARM bulk entry sets are created to adjust certain contact parameters locally. Contact parameters can also be adjusted globally with a BCTPARM bulk entry having the same CSID as the BCSET case control command.

The following example demonstrates the inputs.

```
CASE CONTROL
$CSID on the BCSET case control matches CSID on BCTADD
BCSET = 108
...
BULK DATA
$Local Contact Set definitions
BCTSET 1 1 2 0.0 1.0
BCTSET 2 3 4 0.15 0.0 0.1
5 6 0.15 0.0 0.1
...
$Local Contact Parameters
BCTPARM 1 PENN 10 PENT 1
BCTPARM 2 PENN 1.0 PENT 0.1
...
$Local Contact Sets are combined with BCTADD
BCTADD 108 1 2
...
$Global Contact Parameters
BCTPARM 108 MAXS 30 NCHG 0.02
```

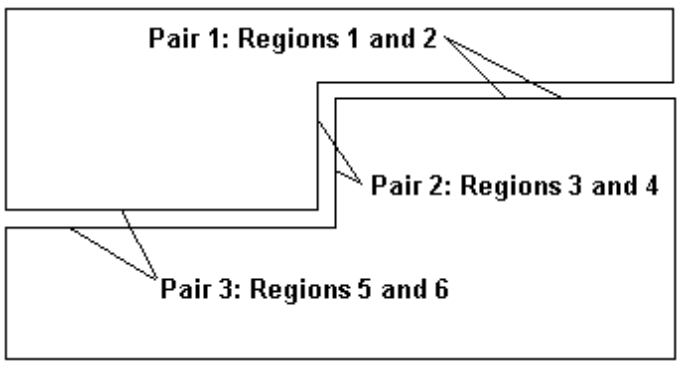
See the section “Contact Control Parameters - BCTPARM” for more information on contact parameters.

Contact with Composite Solid Faces

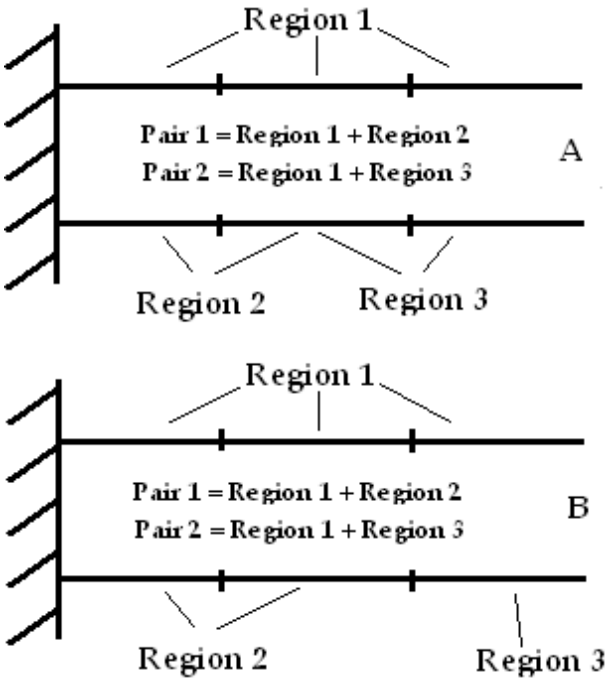
Defining contact regions and pairs on composite solid faces which are perpendicular to the stack direction (edge faces) may produce poor stress continuity. If the contact definition is between edge faces belonging to different PCOMPS definitions, and if the number of plies on each PCOMPS definition is small and the same, and the ply thicknesses are similar, the stress continuity should be fairly smooth. This also applies to the results requested with the BCRESULTS case control command.

Additional Recommendations

When defining contact regions and pairs on geometry which is not tangent continuous, creating single contact regions which cross corner transitions can result in non-uniform stress results around the corners. It is recommended to break these areas into multiple regions and pairs as shown below.



When defining contact regions and pairs, it is recommended to not include the same element face in multiple regions. In “A” below, an element is repeated in regions 2 and 3. In “B”, the same element only exists in region 2. “B” is recommended. Repeating element faces multiple times in the same or different regions can significantly increase memory requirements and degrade performance.



5.5 Contact Control Parameters

The contact control parameters on the BCTPARM bulk entry (optional) can help you adjust the contact algorithm when you are having problems getting a solution to converge and complete, or when the contact results are not as expected. For many solutions, the default settings are appropriate, and the BCTPARM entry is not required.

- For surface-to-surface contact definitions, all of the parameters listed below are supported.
- For edge-to-edge contact definitions, all of the parameters listed below are supported except for REFINE and INTORD.

You can optionally define multiple BCTSET/BCTPARM bulk entry sets, each set with unique contact set IDs (CSID), and then combine them with a single BCTADD bulk entry. The multiple BCTSET/BCTPARM bulk entry sets are created to adjust certain contact parameters locally. Contact parameters can also be adjusted globally with a BCTPARM bulk entry having the same CSID as the BCSET case control command. Also see the section Contact Pairs which includes an input example.

Global and local contact parameters have the following definition and rules:

- Global Contact Parameters

  The BCTPARM bulk entry, which uses the same CSID entered on the BCSET case control command, defines global parameters.

Any of the parameters on the BCTPARM bulk entry can be defined globally. A parameter's default value is used if it is not defined globally or locally.

- Local Contact Parameters

  The BCTPARM bulk entries associated to individual BCTSET bulk entries, which are then combined with a BCTADD bulk entry, define local parameters.

Following is a description of the BCTPARM input parameters. Local contact parameters have "*" next to their names.

Table 5-1. Primary parameters supported by SOL 401:

Name	Description
CTOL	Contact augmentation traction convergence. The augmentation loop convergence criteria can be based on traction convergence. The contact force ratio FRAT is determined as: FRAT = $(\lambda^{k} - \lambda^{k-1}) * (\lambda^{k} - \lambda^{k-1}) / (\lambda^{k} * \lambda^{k})$ where k is the augmentation loop id. If FRAT < CTOL, the contact augmentation loop is considered converged. (Default = 0.05)
PTOL*	Contact penetration tolerance. If the contact penetrations exceed the penetration tolerance, an extra augmentation loop is performed. If the penetrations are below this tolerance, the augmentation loop is considered converged. In addition, if the global solution convergence criteria is satisfied, then the time step is considered converged. PTOL only applies when CNTCONV=0, 1, or 3. If CNTCONV=2, PTOL is ignored and the contact convergence criteria is only based on CTOL. (Default = 1.0E-2 *characteristic length)
CNTCONV	Contact convergence criteria. 0 – The contact convergence criteria is based on the first of PTOL or CTOL. (Default) 1 – The contact convergence criteria is based on PTOL. 2 – The convergence criteria is based on CTOL. 3 – The convergence criteria is based on both CTOL and PTOL.
MAXS	Maximum number of augmentation (outer) loops. If the augmentation loop has not converged in MAXS number of iterations, the solution will proceed to the next step if the global convergence criteria has been met. (Default = 20)

Table 5-1. Primary parameters supported by SOL 401:

Name	Description
INIPENE*	Use when the goal is for a pair of contact regions to be touching without interference, but due to the faceted nature of finite elements around curved geometry, some of the element edges or faces may have a slight gap or penetration. 0 or 1 - Contact is evaluated exactly as the geometry is modeled. No corrections will occur for gaps or penetrations (Default). 2 - Penetrations will be reset to a new initial condition in which there is no interference. 3 - Gaps and penetrations are both reset to a new initial condition in which there is no interference. INIPENE is applied when contact elements are initially created, and if they are recreated as a result of large displacement effects when PARAM,LGDISP,1 is defined.
OPNSTF*	Open contact stiffness scale factor. The open contact stiffness is computed by OPNSTF * closed stiffness. (OPNSTF default = 1.0E-6)
OPNTOL*	Open gap tolerance scale factor. The open contact stiffness (OPNSTF * closed stiffness) is applied to the contact elements that have a gap value less than or equal to OPNTOL * characteristic length, but greater or equal than GAPTOL * characteristic length. The contact element stiffness is 0.0 if the gap is greater than OPNTOL * characteristic length. (OPNTOL default = 1.0)
GAPTOL*	Closed gap tolerance scale factor. The closed contact stiffness is applied to the contact elements that have a gap less than GAPTOL * characteristic length. (Default = 1.0E-6)
NOSEP*	No separation contact option. NOSEP=0 (default): When contact stiffness is recomputed in a consecutive nonlinear iteration, contact elements which are inactive as a result of normal tractions=0.0 and no penetration, and which have a gap greater than GAPTOL * characteristic length will remain inactive in the consecutive iteration. NOSEP=1: The open contact stiffness (OPNSTF * closed stiffness) is applied to the inactive contact elements that have a gap value less than or equal to OPNTOL * characteristic length, but greater or equal than GAPTOL * characteristic length.. The contact elements with a gap greater than OPNTOL * characteristic length remain inactive. While sliding is permitted with this option, the magnitude of the sliding can be controlled by the tangential penalty factor. To define frictionless sliding, set the coefficient of friction=0.0 or tangential penalty factor (PENT)=0.0. (Default=0)

Table 5-1. Primary parameters supported by SOL 401:

Name	Description
GUPDATE	Geometry update flag 0 – Contact geometry updates will not be done during the analysis. 1 – Geometry update will be done for large deflection analysis whenever the relative tangential sliding between source and target regions in a pair exceeds the tolerance set by the GUPTOL parameter (Default for geometric nonlinear analysis).
GUPTOL*	Geometry update tolerance. If the relative sliding distance between the source and target regions exceeds this tolerance, a geometry update will be initiated with large displacement. (Default = 0.1 * characteristic length)
DISCAL	Displacement scaling option 0 – No scaling will be done. 1 – Scaling will be done if required during every iteration. A check will be performed after every displacement increment to see if the incremental displacements would cause penetration between the source and target regions. If the penetrations exceed DISTOL, the entire incremental displacements will be scaled back to limit the penetrations in the model. (Default)
DISTOL	Tolerance for displacement scaling feature. (Default = 0.5* characteristic length)
KSTAB	Stiffness stabilization for contact. 0 – Stiffness stabilization is off. (Default) 1 – The stiffness matrix is stabilized when it is singular due to inactive contact constraints. The stabilization adds a factor (1.0) to the diagonal terms of the stiffness matrix. KSTAB=1 is only supported with the sparse solver, and will disable any open contact stiffness specified through the OPNSTF parameter.

Table 5-2. Secondary parameters supported by SOL 401:

The following parameters are available for special cases.

Name	Description
PENN*	Penalty factor for normal direction. PENN and PENT are automatically calculated by default. When PENT is defined but PENN is undefined, PENN = 10 * PENT.
PENT*	Penalty factor for transverse direction. PENN and PENT are automatically calculated by default. When PENN is defined but PENT is undefined, PENT = PENN / 10.

Table 5-2. Secondary parameters supported by SOL 401:

The following parameters are available for special cases.

Name	Description
PENTYP*	Changes how contact element stiffness is calculated (Default=1). 1- PENN and PENT are entered as units of 1/Length. 2 - PENN and PENT are entered as units of Force/(Length x Area).
AUTOSCAL*	Scales the automatically calculated penalty factors PENN and PENT either up or down. AUTOSCAL can be used to scale the stiffness of specific contact pairs if convergence issues occur (0<Real; Default=1.0).
REFINE	Requests that the software refine the mesh on the source region during the solution to be more consistent with the target side mesh. 0 - Refinement does not occur. 2 - Refinement occurs (default).
INTORD	Determines the number of contact evaluation points for a single element edge or face on the source region. The number of contact evaluation points is dependent on the value of INTORD, and on the type of element face. 1 – The reduced number of contact evaluation points is used. 2 – Use an increased number of contact evaluation points (default). 3 – Use a high number of contact evaluation points.

5.6 Contact Output Request

Contact results can be requested with the BCRESULTS case control command. Forces, tractions, separation distance, total and incremental slide distance, and contact status can be output.

The separation distance is based on the current deformed configuration, and is output for both active an inactive contact elements. The slide distance is a relative displacement in the tangential direction between the source and target faces. The total and incremental slide distance are also output when the separation distance is requested. This total distance is the summation of the incremental sliding which occurred from all previous solution steps from all static subcases. The incremental slide distance is the sliding which occurred since the last output step. Separation distance and slide distance are output for grids on both the source and the target.

When the contact status is requested with the new STATUS describer on the BCRESULTS command, an integer value indicating the contact status is output on each grid point included in a contact source or target region. The status values are:

0: No contact exists.

1: A sticking contact condition exist.

2: A sliding contact condition exist.

5.7 SOL 401 contact algorithm

The example below represents solid element faces included in a source and a target region.

S1, S2, S3, S4,...Sj are the grid points defining one solid element face which is included in the source region.

T1, T2, T3, T4,...Tj are the grid points defining one solid element face which is include in the target region.

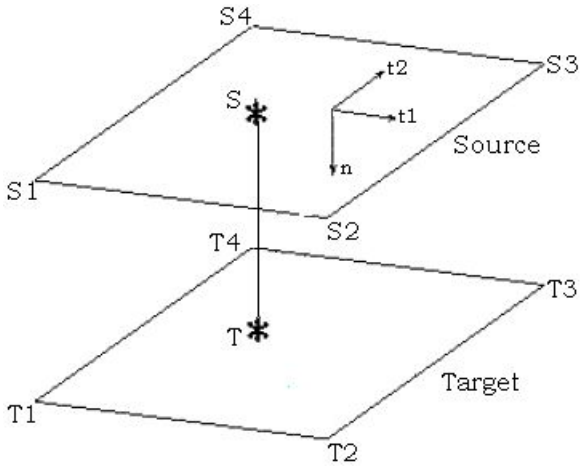


Figure 5-1. Contact Source and Target Example

Global equations including contact:

$$(K + K_c)\Delta u = P - F_c - F_{\text{int}}$$

Equation 5-1.

where

K_c= Contact stiffness assembled for all active contact elements.

F_c= Contact forces resulting from normal and tangential components of the tractions.

$$F_c = \sum_{i=1}^{N_S}(G_{ni}^S - G_{ni}^T)\lambda_{ni} + \sum_{i=1}^{N_T}(G_{ti}^S - G_{ti}^T)\lambda_{ti}$$

Equation 5-2.

where

N_s = number of grid points on an element source face

N_t = number of grid points on an element target face

λ_{ni} = normal traction for contact element i

λ_{ti} = tangential traction for contact element i

$$G_n^S = \begin{bmatrix} N_{S1} & N_{S2} & N_{S3} & N_{S4} & \dots & N_{Sj} \end{bmatrix} \cdot \vec{\eta}$$

$$G_n^T = \begin{bmatrix} N_{T1} & N_{T2} & N_{T3} & N_{T4} & \dots & N_{Tj} \end{bmatrix} \cdot \vec{\eta}$$

$$G_t^S = \begin{bmatrix} N_{S1} & N_{S2} & N_{S3} & N_{S4} & \dots & N_{Sj} \end{bmatrix} \cdot \vec{t}_d \quad [d = 1,2]$$

$$G_t^T = \begin{bmatrix} N_{T1} & N_{T2} & N_{T3} & N_{T4} & \dots & N_{Tj} \end{bmatrix} \cdot \vec{t}_d \quad [d = 1,2]$$

$\begin{bmatrix} N_{S1} \dots N_{Sj} \end{bmatrix}$ = source shape functions evaluated at the contact points S_j.

$\begin{bmatrix} N_{T1} \dots N_{Tj} \end{bmatrix}$ = target shape functions evaluated at contact points T_j,

$\vec{\eta}$, $\vec{t}_d$ = normal and tangential base vectors evaluated at the source locations S_j.

The normal traction λ_n^i is evaluated for iteration i as

$$\lambda_n^i = \varepsilon_n \cdot g_n^i$$

where

ε_n = normal penalty stiffness

g_n^i = gap at contact element = $\left(G_n^S - G_n^T\right) \cdot u^i$

where u^i = grid point displacements at iteration i.

To compute the tangential traction, we need the slip increment over the current step.

$$\Delta g_t = g_{t(n+1)}^i - g_{t(n)}$$

where

$g_{t(n+1)}^i$ = relative tangential displacement at iteration i of (n+1)th time step.

$g_{t(n)}$ = relative tangential displacement from the last converged time step.

The relative tangential displacement for iteration i is computed as

$$g_t^i = \left(G_t^s - G_t^T\right) \cdot u^i$$

The tangential traction is computed as:

$$\lambda_t^{trial} = \varepsilon_t \Delta g_t + \lambda_{t(n)}$$

$$\phi = \left\| \lambda_t^{trial} \right\| - \mu \lambda_n$$

where μ is the Coulomb friction coefficient,

and ε_t = tangential penalty stiffness.

If $\Phi \le 0$; the contact element is sticking.

If Φ > 0; the contact element is slipping.

If sticking:

$$\lambda_t = \lambda_t^{trial}$$

If slipping:

$$\lambda_t = \frac{\mu\lambda_n\lambda_t^{trial}}{\left\|\lambda_t^{trial}\right\|}$$

The equivalent grid point forces resulting from an active contact element are then computed from equation 6-2.

1. The stiffness matrix contribution of an active sticking contact element is:

$$K_{sticking} = \left(G_n^s - G_n^T\right)^{\mathrm{T}} E_n \left(G_n^s - G_n^T\right) + \left(G_t^s - G_t^T\right)^{\mathrm{T}} E_T \left(G_t^s - G_t^T\right)$$

2. The stiffness matrix contribution of an active slipping contact element is:

$$K_{slipping} = \left(G_n^s - G_n^T\right)^{\mathrm{T}} E_n \left(G_n^s - G_n^T\right) + \left(G_n^s - G_n^T\right)^{\mathrm{T}} \mu E_n \frac{\lambda_t}{\left\|\lambda_t\right\|}\left(G_t^s - G_t^T\right)$$

$$+\left(G_t^s - G_t^T\right)^{\mathrm{T}} \mu E_t \frac{\lambda_n}{\left\|\lambda_t\right\|}\begin{bmatrix} 1-\frac{\lambda_{t1}\lambda_{t2}}{\left\|\lambda_t\right\|^2} & -\frac{\lambda_{t1}\lambda_{t2}}{\left\|\lambda_t\right\|^2} \\ -\frac{\lambda_{t1}\lambda_{t2}}{\left\|\lambda_t\right\|^2} & 1-\frac{\lambda_{t2}\lambda_{t2}}{\left\|\lambda_t\right\|^2} \end{bmatrix}\left(G_t^s - G_t^t\right)$$

   For slipping contact elements, the second stiffness term is unsymmetric. If the normal traction is assumed to be fixed during the Newton-Raphson iterations, the second term vanishes, and a symmetric stiffness matrix is retained.

Flow chart of contact algorithm

Steps 1 through 4 occur for each time step (n+1).

1. Loop over contact augmentations (k=1 to MAXS). MAXS is defined on the BCTPARM bulk entry.
   a. Initialize λ_n^k and λ_t^k
   b. Compute k and k_c
2. Do Newton-Raphson iterations (i=1 to MAXITER). MAXITER is defined on the NLCNTL bulk entry.
   a. Compute internal forces (F_{int})
   b. Compute contact tractions and forces (Fc)

$$T_n^k = \lambda_n^k + \varepsilon_n g_n^i$$

$$T_t^{k,trial} = \lambda_t^k + \varepsilon_t (g_{t(n+1)}^i - g_{t(n)})$$

$$\phi = \left\|T_t^{k,trial}\right\| - \mu\lambda_n^k$$

Assuming that the normal traction is fixed during Newton-Raphson iterations, λ_n^k is used as opposed to T_n^k resulting in symmetrization.

If $\phi \le 0;\ T_t^k = T_t^{k,trial}$ (sticking)

else; $T_t^k = \dfrac{\mu\lambda_n^k}{\left\|T_t^{k,trial}\right\|} T_t^{k,trial}$ (slipping)

Compute contact forces F_c using T_n^k and T_t^k.

c. Compute Residual $R = P - F_c - F_{int}$

d. Solve $(K + K_c)\Delta u = R$

e. Scale back Δu if large incremented penetrations are detected.

f. Check for the solution convergence on the residual. The solution convergence criteria is defined by the CONV parameter on the NLCNTL bulk entry.
(i) If the solution convergence criteria is satisfied; augment tractions by step 3.
(ii) If the solution convergence criteria is not satisfied; do more iterations, go to 2a, update stiffness if NEWK=TRUE, and go to step 1b.

3. $k = k + 1;\quad \lambda_n^k = T_n^k;\quad \lambda_t^k = T_t^k$

Check for maximum penetration convergence and/or force convergence. The contact convergence tolerances are defined by the PTOL and CTOL parameters on the BCTPARM bulk entry. The penetration convergence is satisfied if the maximum penetration (P_{max}) is smaller than the penetration tolerance (PTOL). The force convergence is satisfied if the contact force ratio (FRAT) is smaller than the contact force tolerance (CTOL), where

FRAT = $(\lambda^{k} - \lambda^{k-1}) * (\lambda^{k} - \lambda^{k-1}) / (\lambda^{k} * \lambda^{k})$.

The CNTCONV parameter on the BCTPARM bulk entry detemines which tolerance (PTOL, CTOL) is used to satisfied contact convergence.

- If CNTCONV=0 (default), contact convergence is satisfied if either the penetration tolerance (PTOL) or the contact force tolerance (CTOL) are satisfied:

  If (P_{max} < PTOL) or (FRAT < CTOL), go to step 4.

- If CNTCONV=1, contact convergence is satisfied if the penetration tolerance (PTOL) is satisfied:

  If (P_{max} < PTOL), go to step 4.

- If CNTCONV=2, contact convergence is satisfied if the contact force tolerance (CTOL) is satisfied:

  If (FRAT < CTOL), go to step 4.

- If CNTCONV=3, contact convergence is satisfied if both the penetration tolerance (PTOL) and the contact force tolerance (CTOL) are satisfied:

If (P_{max}<PTOL) and (FRAT<CTOL), go to step 4.

- Else, go to step 1b.

4. End.

Chapter 6: Bolt preload in SOL 401

6.1 Bolt preload

The bolt preload capability allows you to model bolts with either 3D solid elements or 2D plane stress elements. The 3D solid elements you can use are the CHEXA, CPENTA, and CTETRA elements. The 2D plane stress elements you can use are the CPLSTS3, CPLSTS4, CPLSTS6, and CPLSTS8 elements. A common scenario for using 2D plane stress elements to model bolts is in an axisymmetric analysis. A bolt preload can be combined with geometric nonlinear and plasticity conditions.

The procedure to model bolts with preload in SOL 401 is similar to other solution sequences.

- You use the BOLT bulk entry to select the elements that represent the bolt.
- You use the BOLTFOR bulk entry to define the bolt preload force.
- You use the BOLTLD bulk entry to optionally combine and scale bolt preload forces.
- You use the BOLTLD case control command in your bolt preload subcase to invoke the capability. The BOLTLD command references a BOLTLD bulk entry or BOLTFOR bulk entries. Bolt preload can only be invoked in a single subcase.

Bolts in SOL 401 are defined with the ETYPE = 3 format on the BOLT bulk entry. This format requires that you list all of the elements that are used to model the bolt shaft on the BOLT entry. The grid point you enter in the GP field on the BOLT entry indicates to the software where to calculate the cross sectional area of the bolt. For this calculation, NX Nastran uses the direction that you define with the CSID and IDIR fields on the BOLT entry as the bolt axis. To avoid any cross sectional effects at the bolt ends, it is best to select a grid point in the GP field closer to the middle of bolt length.

NX Nastran uses the cross sectional area and the bolt preload force to estimate the initial bolt stress and strain. To account for compliance in the structure that is being bolted together, the software iterates on the bolt strain until convergence is satified.

You can adjust the bolt preload convergence tolerance with the EPSBOLT parameter (default=1.0E-3), which is defined on the NLCNTL bulk entry. For each bolt preload iteration, the software computes the difference between the current bolt preload and the user-defined preload. If the difference is less than value of the EPSBOLT parameter, the bolt preload calculation is considered converged. If the difference is greater than EPSBOLT, the preload strain is recomputed for the next bolt preload iteration. The iterations continue until either convergence is satisfied, or the number of iterations reaches the value of the ITRBOLT parameter (default=20). The ITRBOLT parameter is also defined on the NLCNTL bulk entry.

You must define TEND1 = 0.0 on the TSTEP1 bulk entry selected for the bolt preload subcase. You can optionally specify the number of increments on the TSTEP1 entry even though the duration of the time steps are zero. If you do not specify the number of increments, the software applies the preload force in a single increment.

The bolt preload subcase includes the bolt preload force and can optionally include a temperature load. No other loads are supported in the bolt preload subcase. The consecutive sequentially

dependent subcases can then include service loads. The resulting bolt strain is included in the consecutive sequentially dependent subcases until a non-sequentially dependent static subcase occurs.

If your input file includes globally-defined loads, you must include DLOAD = 0 in the bolt preload subcase. Doing so instructs the software to ignore the global loads during the bolt preload solve. Otherwise, a fatal error is issued and the run terminates.

Often, the bolt preload subcase is the first subcase, although this is not required. In addition, a bolt preload subcase which is not the first can be defined as sequentially dependent. For example, you can define an initial static subcase in order to resolve an interference condition using contact. Then a sequentially dependent bolt preload subcase can follow. The final stiffness from the initial subcase, which includes the contact stiffness, along with the contact forces, all are used in the bolt preload subcase. Note that loads defined in the initial subcase cannot be included in the bolt preload subcase. Also note that the TSTEP1 in the initial subcase must also have TEND1 = 0.0.

If the bolt preload subcase is the first, you do not need to specify SEQDEP = NO since the first subcase is always non-sequentially dependent.

The convergence information related to bolt preload is listed in the f06 file. For example,

```
...
PRELOAD BOLT CALCULATIONS FOR BOLT ID           1
-------------------------------------
ITERATION              =           3
USER PRELOAD           =   200000000.000000
INTERNAL FORCE         =   199999616.790662
PERCENTAGE DIFFERENCE  =  1.916046692281961E-004
BOLT PRELOAD CONVERGENCE ATTAINED
---------------------------------
...
```

The software issues a fatal error message if the bolt preload iterative solution fails to converge.

See the BOLT bulk entry.

Chapter 7: Considerations for nonlinear analysis

7.1 Discrete system for a nonlinear continuum model

Theories in solid mechanics are dictated by three governing relationships:

- The state equilibrium that

$$\frac{\partial \sigma_{ij}}{\partial x_j} + b_i = 0$$

Equation 7-1.

  requires where σ_{ij} are stress tensor components, b_i are body forces, and x_j are space coordinates.

- The constitutive relations represented by stress-strain relations, e.g., for linear elasticity,

$$\sigma_{ij} = D_{ijkl}\varepsilon_{kl}$$

Equation 7-2.

  where ε_{kl} are strain tensor components and D_{ijkl} are elastic constants.

- The compatibility represented by strain-displacement relations, e.g., for a small deformation,

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

Equation 7-3.

  where u_i are displacements.

These systems of governing differential equations must be satisfied for every infinitesimal element throughout the domain of the continuum. The complete set of state variables, namely displacements, may be determined by solving these systems of equations supplemented by boundary conditions, and in dynamic situations by initial conditions as well. For the nonlinear problems, the governing equations should be satisfied throughout the history of load application. The material nonlinearity is manifested in the constitutive relations. The geometric nonlinearity is pronounced in the strain-displacement relations, but it also affects the equilibrium equation by changing applied loads. Changes in constraints affect the boundary conditions, which constitute contact problems.

Most of the known solutions for the solid mechanics problems are based on ideal geometry and linear approximations. However, the real nature is more complicated and inherently nonlinear. The linear system is a very particular case of a general problem. Even the nonlinear solutions that we seek deal with only a small subset of special cases in a general category of nonlinear problems. When the

nonlinear system is confronted, no general mathematical solutions exist and superposition no longer applies. The system may even be non-conservative.

The first phase of the structural analysis is the idealization of a physical system into a simpler and more manageable engineering problem. The idealization process involves simplifications of the geometry, boundary and joint conditions, and loading conditions, etc. using engineering intuitions, experimental data, empirical observations, and classical solutions. If the idealized structural system renders a problem that cannot be resorted to a classical method of analysis, further idealization is required, namely discretization, for numerical analysis.

Finite elements represent spatial discretization of a continuum. As such, however, they do not immediately impose nonlinearity. When nonlinearity has to be taken into account for large displacements and/or stresses, a numerical model poses new dimensions to the discretization in addition to the n-dimensional Euclidean space. That is, the discretization is applied to time, load, and material properties by using piece wise linear curves. While discretization allows approximate solutions by numerical methods, it introduces numerous mathematical singularities which may complicate computational processes. Fortunately, the efficiency of modern digital computers makes it feasible to apply complicated computational procedures to the complex systems of engineering problems.

For the discrete system, governing differential equations are converted to algebraic equations. The finite element model represents a structure by an assemblage of finite elements interconnected at nodal points. State variables are the displacements (displacement method or stiffness approach.) of the nodal points which carry fictitious forces representing distributed stresses actually acting on the element boundaries. The equilibrium requirements are satisfied at nodal points by the nodal force balance. The material constitutive laws are satisfied at the integration points of the element. The compatibility is ensured by the displacement continuity between elements. It is noted, however, that the compatibility of the nonconforming elements is ensured by a patch test.

7.2 Finite element formulation for equilibrium equations

The variational principle renders the system governing equilibrium equations when applied to a functional ll, representing a total potential of a continuum, i.e.,

$$\Pi = U - W$$

Equation 7-4.

where *U* is the strain energy of the system and *W* is the potential energy of the external loads. The equilibrium equations can be obtained by invoking the principle of virtual work or the Ritz method, i.e.,

$$\partial\Pi = 0 \quad or \quad \frac{\partial\Pi}{\partial\{u\}} = 0$$

Equation 7-5.

which implies that the total potential of the system must be stationary with respect to the state variables (displacement) for equilibrium to be ensured. The functional IT is so called because it involves the integral of implicit functions of the state variables, {*u*}.

Considering a three-dimensional continuum for a nonlinear problem, the stationarity condition results in

$$\int_V \sigma_{ij}\delta\,\dot{\varepsilon}_{ij}\,dV = \int_V b_i\delta\dot{u}_i\,dV + \int_S t_i\delta\dot{u}_i\,dS + \sum_i p_i\delta\dot{u}_i$$

Equation 7-6.

where the dots and δ denote infinitesimal increments and arbitrary variations, respectively. The left-hand side represents variations in the strain energy increment and the right-hand side represents variations in the external work which consists of body forces b_i (such as a gravity load), traction forces t_i at the boundary surface (such as pressure loads), and concentrated forces p_i. Now it remains to determine admissible functions expressing the arguments of the functional II in terms of state variables {*u*}, which are valid throughout the whole region and satisfy the boundary conditions.

The finite element method can be characterized by the following features distinguished from the conventional Ritz methods or the matrix method for frame structures:

- The whole region of the system is divided into numerous subdomains, called finite elements, which have simple geometrical shapes.
- The variational process is limited to each finite element, which aggregates into a whole region when assembled.
- The admissible displacement field within each element, $\{\tilde{u}\}$, can be expressed in terms of nodal displacements using interpolation functions known as shape functions, *N*, i.e.,

$$\{\tilde{u}\} = [N]\{u\}$$

Equation 7-7.

  where {*u*} is a displacement vector consisting of all nodal points of the element.

The strain-displacement relations for the element can then be established in terms of nodal displacements using the shape functions in Equation 7-7, i.e.,

$$\{\dot{\varepsilon}\} = [B]\{\dot{u}\}$$

Equation 7-8.

where

$$\{\dot{\varepsilon}\}^T = \langle \dot{\varepsilon}_x \;\; \dot{\varepsilon}_y \;\; \dot{\varepsilon}_z \;\; \dot{\gamma}_{xy} \;\; \dot{\gamma}_{yz} \;\; \dot{\gamma}_{xz} \rangle$$

Equation 7-9.

and the element matrix [*B*] consists of derivatives of the shape functions, evaluated at the current deformed geometry. Notice that the geometric linear problem requires that the element matrix be evaluated only at the initial geometry. The software employs an approximate updated Lagrangian approach for geometric nonlinear problems, by which linear strains are computed in the updated element coordinate system in order to eliminate the effects of the rigid body rotation but the

equilibrium is established at the final position in the stationary coordinate system. This method does not require reevaluation of the element matrix [*B*] (constant in the absence of large strains) while the element coordinates are reevaluated continuously.

Equilibrium equations for an element may be obtained by reducing Equation 7-6 after the substitution of Equations 7-7 and 7-8, based on the small deformation theory. Then the element boundary stresses are statically equivalent to the nodal forces which balance the applied external loads, i.e.,

$$\{F\}^e = \{P\}^e$$

Equation 7-10.

with

$$\{F\}^e = \int_V [B]^T \{\sigma\} dV$$

Equation 7-11.

and

$$\{P\}^e = \int_V [N]^T \{b\} dV + \int_S [N_S]^T \{t\} dS + \{p\}$$

Equation 7-12.

where [N_S] is an appropriate interpolation function for the traction force. Notice that the equilibrium equation for an incremental load may be expressed as

$$\{\dot{F}\} = \int_V [B]^T \{\dot{\sigma}\} dV = \{\dot{P}\}$$

Equation 7-13.

where $\{\dot{\sigma}\}$ should be components of co-rotational stress which is independent of a rigid body rotation.

The element stiffness matrix can be obtained by substituting the constitutive relations into Equation 7-13, i.e.,

$$\{\dot{\sigma}\} = [D]\{\dot{\varepsilon}\}$$

Equation 7-14.

where

$$\{\sigma\}^T = \langle \sigma_x \quad \sigma_y \quad \sigma_z \quad \tau_{xy} \quad \tau_{yz} \quad \tau_{zx} \rangle$$

Equation 7-15.

and [*D*] is the material tangent matrix. The nodal forces of an element can then be expressed as

$$\left\{\dot{F}\right\}^e = \int_V \left[B\right]^T \left\{\dot{\sigma}\right\} dV = \left[K\right]^e \left\{\dot{u}\right\}$$

Equation 7-16.

where the element stiffness is

$$\left[K\right]^e = \int_V \left[B\right]^T \left[D\right]\left[B\right] dV$$

Equation 7-17.

Notice that this expression represents an element stiffness due to the material stiffness without geometric nonlinear effects. As will be shown later, an additional stiffness [J(d)] due to initial stresses should be included for an incremental process because the initial stresses exist from the second increment.

The equilibrium must be satisfied in the whole region throughout the complete history of load application. Equilibrium equations for the global discrete system are obtained when all the elements are assembled, i.e.,

$$\sum_m \int_V \left[B\right]^T \left\{\sigma\right\} dV = \sum_m \left\{P\right\}^e$$

Equation 7-18.

where Σ over m denotes a summation over all elements. For the incremental process, the equilibrium equation may be rewritten as

$$\sum_m \int_V \left[B\right]^T \left\{\sigma - \sigma^0\right\} dV = \left\{\Delta P\right\}$$

Equation 7-19.

with

$$\left\{\Delta P\right\} = \sum_m \left\{P\right\}^e - \sum_m \int_V \left[B\right]^T \left\{\sigma^0\right\} dV$$

Equation 7-20.

where $\{\sigma^0\}$ represents an initial stress or the stress state at the preceding load step.

Because of the approximations involved in the interpolation functions, the finite element model provides an approximate solution even if the equilibrium Equation 7-18 is satisfied exactly. Consequently, the differential equations of equilibrium are not satisfied exactly even for linear problems, but the error decreases as the finite element mesh is refined. This convergence condition is required and ensured by element formulations with regard to the element convergence criteria.

7.3 Coordinate transformations

The coordinate transformation is one of the most frequent operations in the finite element method. Vectors and matrices defined in a particular coordinate system can be transformed into another system. Coordinate systems involved are:

- Basic coordinate system: a Cartesian coordinate system on which local coordinate systems are defined.
- Local coordinate system: defined by the user in the Bulk Data, which may include special coordinates such as cylindrical and spherical coordinate systems.
- Global coordinate system: a collective coordinate system which comprises all the local coordinate systems specified for output quantities.
- Element coordinate system: a Cartesian coordinate system unique to each element.
- Displaced element coordinate system: similar to element coordinate system but defined in the displaced position.
- Material coordinate system: a Cartesian coordinate system used to orient anisotropic material properties.
- Modal coordinate system: a generalized coordinate system defined for each eigenmode.

It is noted that the global system is a Cartesian coordinate system, although non-Cartesian coordinate systems are adopted to orient the local Cartesian coordinates for output quantities. In the software, all the displacements and forces, hence the system matrices, such as the stiffness matrix, are expressed in the global coordinates. This implies that all the major computations involved in the analysis are processed in Cartesian coordinates. Element and material coordinate systems are defined in the element connectivity description. Now we only have to consider linear transformations between Cartesian coordinate systems.

Let us consider a coordinate transformation between the primed and unprimed systems which are right-handed Cartesian coordinates. The transformation matrix T consists of direction cosines of unit vectors of the unprimed coordinate system, i.e.,

$$u' = Tu$$

Equation 7-21.

where

$$T = \begin{bmatrix} i_x & j_x & k_x \\ i_y & j_y & k_y \\ i_z & j_z & k_z \end{bmatrix}$$

Equation 7-22.

Notice that *T* is an orthogonal matrix and thus

$$[T]^{-1} = [T]^{T}$$

Equation 7-23.

Because the work and energy are invariants with respect to coordinate transformation, i.e.,

$$P^{T} u = P'^{T} u'$$

Equation 7-24.

it follows that

$$P' = T\,P$$

Equation 7-25.

Then the equilibrium equation,

$$K' u' = P'$$

Equation 7-26.

may be expressed in the unprimed coordinate system by

$$K\,u = P$$

Equation 7-27.

with

$$K = T^{T} K' T$$

Equation 7-28.

It is noted that the modal matrix Φ is used as a transformation matrix for a modal transformation which is not elaborated here.

$$\{u_e\} = T_{be}^{T} T_{bg} \{u_g\}$$

Equation 7-29.

$$\{F_g\} = T_{bg}^{T} T_{be} \{F_e\}$$

The forces and displacements are transformed from element to global coordinates and vice versa, i.e.

where T_{be} transforms from element to basic coordinates and T_{bg} transforms from global to basic coordinates.

It is noted that T_{be} is identical for all the nodes of an element but T_{bg} may vary from node to node in the same element. The element stiffness matrix is transformed into global coordinates by

$$K_{gg} = T_{bg}^{T} T_{be} K_{ee} T_{be}^{T} T_{bg}$$

Equation 7-30.

for which the building blocks of T_{be} and T_{bg} are (3x3) matrices formed for each nodal point and have to be assembled for an entire element, e.g. for a three-noded triangular shell element

$$T_{bg} = \begin{bmatrix} T_{bg}^{1} & 0 & & & & 0 \\ 0 & T_{bg}^{1} & & & & \\ & & T_{bg}^{2} & 0 & & \\ & & 0 & T_{bg}^{2} & & \\ & & & & T_{bg}^{3} & 0 \\ 0 & & & & 0 & T_{bg}^{3} \end{bmatrix}$$

Equation 7-31.

where the superscript is used to associate each (3x3) matrix with the nodal point and is repeated for the rotational degrees of freedom.

7.4 Displacement sets and reduction of system equations

The equilibrium equations, and thus system matrices, are reduced in size using the displacement set, which is a unique feature of this software. Mutually exclusive subsets of the global displacement set, $\{u_g\}$, are defined as follows:

u_m	Degrees-of-freedom eliminated by multipoint constraints
u_s	Degrees-of-freedom eliminated by single-point constraints
u_o	Degrees-of-freedom omitted by static condensation
u_r	Degrees-of-freedom eliminated by a bulk data SUPORT to suppress rigid body motion
u_i	Degrees-of-freedom which remain for solution after reductions

For convenience, complementary sets are defined as follows:

$u_n = u_g - u_m$

$u_f = u_n - u_s$

$u_a = u_f - u_o$

$u_l = u_a - u_r$

The subsets are defined by the user with a possible exception in the s-set if PARAM,AUTOSPC,YES is used. Notice that the rigid elements are equivalent to the multipoint constraints internally in the program, but they are not selectable in the subcases as for MPCs. Because the set-reduction operations involve many basic modules and DMAP blocks, the mathematics for elimination of constraints and static condensation is reviewed here.

The multipoint constraint equations are formed in the module GP4 as follows:

$$\left[R_{mg}\right]\begin{Bmatrix}u_m\\u_n\end{Bmatrix}=\{0\}$$

Equation 7-32.

where

$$\left[R_{mg}\right]=\left[R_{mm}\mid R_{mn}\right]$$

The module MCE1 partitions [R_{mg}] and solves for a transformation matrix [G_{mn}], i.e.,

$$\{u_m\}=\left[G_{mn}\right]\{u_n\}$$

Equation 7-33.

where

$$\left[G_{mn}\right]=-\left[R_{mm}\right]^{-1}\left[R_{mn}\right]$$

Then the module MCE2 partitions the global stiffness matrix, [K_{gg}], and reduce it to the n-set, i.e.,

$$\begin{bmatrix}K_{mm} & K_{mn}\\K_{nm} & K'_{nn}\end{bmatrix}\begin{Bmatrix}u_m\\u_n\end{Bmatrix}=\begin{Bmatrix}P_m\\P'_n\end{Bmatrix}+\begin{Bmatrix}Q_m\\Q'_n\end{Bmatrix}$$

Equation 7-34.

from which the system is reduced to

$$K_{nn}u_n=P_n+Q_n$$

Equation 7-35.

where

$$K_{nn}=G_{mn}^T\left[K_{mm}G_{mn}+K_{mn}\right]+\left[K_{mn}^TG_{mn}+K'_{nn}\right]$$

$$P_n=G_{mn}^TP_m+P'_n$$

and

$$Q_n=G_{mn}^TQ_m+Q'_n$$

The primes are used in K'_{nn}, P'_n, and Q'_n to distinguish from K_{nn}, P_n, and Q_n, which are resulting matrices after the reduction.

Equations in the n-set can be further reduced by eliminating single-point constraints, i.e.,

$$\begin{bmatrix}K_{ss} & K_{sf}\\K_{fs} & K_{ff}\end{bmatrix}\begin{Bmatrix}u_s\\u_f\end{Bmatrix}=\begin{Bmatrix}P_s\\P'_f\end{Bmatrix}+\begin{Bmatrix}Q_s\\0\end{Bmatrix}$$

Equation 7-36.

which is reduced to

$$K_{ff}u_f = P_f$$

Equation 7-37.

where

$$P_f = P'_f - K_{fs}Y_s$$

with

$$u_s = Y_s$$

Notice that the effects of constraint forces (Q_s and Q_m) are not visible in Equation 7-37. The single-point constraint forces are recovered by

$$Q_s = -P_s + K_{fs}^T u_f + K_{ss}Y_s$$

Equation 7-38.

Further reduction of equations in the f-set is performed by an elimination of the o-set, known as static condensation. The f-set is partitioned by the UPARTN module as follows:

$$\begin{bmatrix} K_{oo} & K_{oa} \\ K_{ao} & K'_{aa} \end{bmatrix} \begin{Bmatrix} u_o \\ u_a \end{Bmatrix} = \begin{Bmatrix} P_o \\ P'_a \end{Bmatrix}$$

Equation 7-39.

from which

$$\{u_0\} = \{u_o^o\} + [G_{oa}]\{u_a\}$$

Equation 7-40.

where

$$u_o^o = K_{oo}^{-1}P_o$$

and

$$G_{oa} = -K_{oo}^{-1}K_{oa}$$

Then the reduced system of equations in the a-set is obtained as

$$K_{aa}u_a = P_a$$

Equation 7-41.

where

$$K_{aa} = K'_{aa} + K_{oa}^T G_{oa}$$

and

$$P_a = P'_a + G_{oa}^T P_o$$

There are some rules to remember regarding the displacement sets in SOL 401. They are:

- The r-set is not supported. Do not use the SUPORT bulk entry.
- PARAM, AUTOSPC is not supported.
- PARAM, AUTOSPCR is not supported.
- Rigid elements are formulated with linear multipoint constraint equations and do not have large displacement capability. Consequently, erroneous results will be obtained if the rigid element undergoes a large rotation. To avoid this, stiff elements should be used in place of rigid elements for large displacement analysis.

7.5 Nonlinear solution procedure

The general-purpose program developer faces the task of providing the best workable solution method for a wide spectrum of problems, while maintaining flexibility by allowing the user to specify optional parameters. Based on the extensive numerical experiments, an attempt was made to establish a general strategy suitable for most problems without requiring insight or experience. Variations in combining theories, algorithms, criteria and parameter values with numerous test problems resulted in a succinct implementation.

The major feature of the nonlinear analysis is the requirement for the incremental and iterative processes to obtain a solution. The main issue is how to choose the most efficient method from the options available for the incremental and iterative processes in the solution of nonlinear equilibrium equations. The increment size for time steps has the most significant effect on the efficiency and the accuracy of the computation, particularly in the path-dependent problems. The incremental and iterative processes are complementary to each other because the larger the increment size the more iterations the solution requires. While an excessively small increment reduces the computing efficiency without any significant improvement in accuracy, a large increment may deteriorate the efficiency as well as the accuracy; it may even cause divergence.

It is impossible to optimize the incremental step size in the absence of prior knowledge of the structural response. The best engineering judgement should be exercised to determine the increment size based on the severity of the nonlinearity. Needless to say, no incremental load steps are required when the response is linear. In principle, the size of the load increment (or time increment) should be chosen to yield a uniform rate of change in strains or stresses for the material nonlinear problems and a uniform rate of change in displacements for geometric nonlinear problems.

User specifications for solution methods in nonlinear analyses are allowed via:

- The NLCNTL bulk entry for the static analysis. It is selected by the NLCNTL case control command.
- The TSTEP1 bulk entry for the load increment (time based). It is selected by the TSTEPNL case control command.
- The EIGRL bulk entry for the modal analysis. It is selected by the METHOD case control command.

The increment size can vary from subcase to subcase by specifying different TSTEPNL. It is recommended to define separate TSTEPNL for every subcase even if the same values are specified, so that changes can be accommodated in the subcase level as needed.

Chapter 8: Geometric nonlinearity

8.1 Overview and user interface

Geometric nonlinearities are manifested in problems involving large rotations and large deformation. The characteristics are follower forces due to large rotations, geometric stiffening due to initial stress effect (as a result of large rotations), and large strains due to large deformation.

Geometric nonlinear effects should be significant if the deformed shape of the structure appears distinctive from the original geometry by a visual inspection. A more rigorous and quantitative definition for the large displacements can be derived from the plate theory of Kirchhoff and Love: the small deflection theory is valid for a maximum deflection of less than 20% of the plate thickness or 2% of the small span length. However, this definition seems to be a little conservative for numerical analysis, and there is no distinct limit for large displacements because geometric nonlinear effects are related to the boundary conditions as well as the dimensions of the structure. If the load-deflection curve of the critical point can be estimated, the loading point should be in the nonlinear portion of the curve.

Geometric nonlinear effects in the structure involving large rotations, whether rigid body rotations or deformation induced rotations, are self-evident. Stiffening of a membrane, stiffness in a pendulum or snap-through of an arch belong to this category. The motion of a pendulum under gravity is caused by geometric (differential) stiffness. Follower forces are manifested when the applied loads are displacement dependent, such as pressure load and thermal load applied on the surface that rotates. Centrifugal force is another example of follower forces. Large strain effects are pronounced in metal forming problems which could have strains exceeding 100%. Finite strain formulation is required to treat the problems in this category. The software does not currently support the large strain capability. In most structural applications, however, moderately large strains (20 to 30%) appear in local areas if there is any large deformation. The software can be used for that category of problems. Other geometric nonlinear effects are treated by updated element coordinates, gimbal angles (or rotation vector), and the differential stiffness [K^d].

The geometric nonlinearity is controlled by the parameter LGDISP with the following values in SOL 401:

- = 0 for geometrically linear analysis
- = 1 for geometrically nonlinear analysis

With values of 1 or 2 for LGDISP, all the potentially nonlinear elements become actively nonlinear elements unlike the material nonlinear model.

This solver has a distinct approach to the large rotation, for which the element coordinates are continuously updated to the current configuration during the iteration. The equilibrium is sought in the deformed position. Consider the internal force computation as follows:

$$F = \int_V \overline{B}^T \sigma dV$$

Equation 8-1.

The element matrix $\left[\overline{B}\right]$ is defined from the strain definition as

$$\{d\varepsilon\} = \left[\overline{B}\right]\{du\}$$

Equation 8-2.

in which $\left[\overline{B}\right]$ could be divided into two parts (linear and nonlinear), i.e.,

$$\overline{B} = B_L + B_N$$

Equation 8-3.

Upon differentiation of Equation 8-1, we have

$$dF = \int_V \overline{B}^T d\sigma dV + \int_V d\overline{B}^T \sigma dV$$

Equation 8-4.

where {σ} represents stresses with reference to the original coordinates. Substituting Equation 8-3 and

$$d\sigma = D\, d\varepsilon = D\, \overline{B}\, du$$

Equation 8-5.

Equation 8-4 becomes

$$dF = \underbrace{\left[K_L + K_R + K_\sigma\right]}_{\text{tangential matrix}} du$$

Equation 8-6.

with

$$K_L = \int_V B_L^T D\, B_L\, dV$$

$$K_R = \int_V \left[B_L^T D\, B_N + B_N^T D\, B_N + B_N^T D\, B_L \right] dV$$

Equation 8-7.

and

$$K_\sigma \, du = \int_V dB_N^T \, \sigma \, dV$$

Equation 8-8.

in which K_L represents the usual linear stiffness matrix, K_R a stiffness due to large rotation, and K_g a geometric stiffness dependent on the initial stress level.

Now it remains to define the nonlinear part of the element matrix (B_N). The definition of finite strains based on the Lagrangian formulation (referred to the initial configuration) is as follows:

$$\varepsilon_x = \frac{\partial u}{\partial x} + \frac{1}{2}\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial x}\right)^2\right]$$

Equation 8-9.

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} + \left[\frac{\partial u}{\partial x}\cdot\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\cdot\frac{\partial v}{\partial y} + \frac{\partial w}{\partial x}\cdot\frac{\partial w}{\partial y}\right]$$

Equation 8-10.

with other components obtained similarly. In matrix notation

$$\{\varepsilon\} = \{\varepsilon_L\} + \{\varepsilon_N\}$$

where $\{\varepsilon_L\}$ is the usual infinitesimal strain vector and $\{\varepsilon_N\}$ is the nonlinear strain vector consisting of the second order terms, i.e.,

$$\{\varepsilon_L\} = [B_L]\{u\}$$

$$\{\varepsilon_N\} = \frac{1}{2}[A]\{\theta\}$$

Equation 8-11.

where

$$\underbrace{[A]}_{6x9} = \begin{bmatrix} a_x^T & 0 & 0 \\ 0 & a_y^T & 0 \\ 0 & 0 & a_z^T \\ a_y^T & a_x^T & 0 \\ 0 & a_z^T & a_y^T \\ a_z^T & 0 & a_x^T \end{bmatrix}$$

Equation 8-12.

and

$$\underbrace{\{\theta\}}_{9x1} = \begin{Bmatrix} a_x \\ a_y \\ a_z \end{Bmatrix} \text{ with } \underbrace{\{a_x\}}_{3x1} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial x} \\ \frac{\partial w}{\partial x} \end{Bmatrix}$$

Equation 8-13.

Introducing shape functions (N_i) and nodal displacements {u} (using an example of a 10-noded tetrahedron), displacement derivatives are expressed by

$$\{a_x\} = \begin{bmatrix} \frac{\partial N_1}{\partial x} I & \frac{\partial N_2}{\partial x} I & \cdots & \frac{\partial N_{10}}{\partial x} I \end{bmatrix} \{u\}$$

Equation 8-14.

and

$$\{\theta\} = [G]\{u\}$$

Equation 8-15.

where

$$\underbrace{[G]}_{9x30} = \begin{bmatrix} \frac{\partial N_1}{\partial x} I & \frac{\partial N_2}{\partial x} I & \cdots & \frac{\partial N_{10}}{\partial x} I \\ \frac{\partial N_1}{\partial y} I & \frac{\partial N_2}{\partial y} I & \cdots & \frac{\partial N_{10}}{\partial y} I \\ \frac{\partial N_1}{\partial z} I & \frac{\partial N_2}{\partial z} I & \cdots & \frac{\partial N_{10}}{\partial z} I \end{bmatrix}$$

Equation 8-16.

and

$$\underbrace{\{u\}}_{30x1} = \begin{Bmatrix} a_1 \\ \vdots \\ a_i \\ \vdots \\ a_{10} \end{Bmatrix} \quad with \quad \{a_i\} = \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix}$$

Equation 8-17.

From the properties of matrices A and θ, it can be shown that

$$d\varepsilon_N = \frac{1}{2} dA\ \theta + \frac{1}{2} A\ d\theta = A\, d\theta = AG\, du$$

from which

$$\underbrace{[B_N]}_{6x30} = \underbrace{[A]}_{6x9}\ \underbrace{[G]}_{9x30}$$

Equation 8-18.

The initial stress stiffness [K_σ] can be derived as follows:

$$K_\sigma\, da = \int_V dB_N^T \sigma\, dV = \int_V G^T dA^T \sigma\, dV$$

in which

$$dA^T \sigma = \begin{bmatrix} da_x & 0 & 0 & da_y & 0 & da_z \\ 0 & da_y & 0 & da_x & da_z & 0 \\ 0 & 0 & da_z & 0 & da_y & da_x \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{Bmatrix}$$

$$= M\, d\theta = M\, G\, du$$

where

$$\underbrace{[M]}_{9x9} = \begin{bmatrix} \sigma_x I & \tau_{xz} I & \tau_{xz} I \\ & \sigma_y I & \tau_{yz} I \\ Sym & & \sigma_z I \end{bmatrix}$$

Equation 8-19.

with *I* being the (3x3) identity matrix. Finally the geometric stiffness is

$$\underbrace{[K_\sigma]}_{30x30} = \int_V G^T M\, G\, dV$$

Equation 8-20.

It has been found that stiffness matrices caused by geometric nonlinearity (K_R and K_σ) can be computed from the matrices [*A*], [*G*], and [*M*] with the following observations:

- [*G*] is dependent upon the initial geometry, hence stays constant unless the geometry is updated. This matrix is used in forming [K_R] and [K_σ].
- [*A*] is used in forming [K_R]. [*A*] is dependent on the rotations and should be updated continuously.
- [*M*] is used in forming [K_σ]. [*M*] is dependent on the stresses and should be updated continuously.

The primary functions of nonlinear stiffness matrices can be interpreted as follows:

- The matrix [K_R] takes into account the effects of large rotations. The large displacement effects, due to rigid body translation and rotation, are treated effectively in the absence of large strains by updating element coordinates in the software.

- Geometric stiffness matrix [K_σ] takes into account the effects of the initial stresses. This effect becomes important with geometric stiffening, and is used for instability analysis. The geometric stiffness matrix [K_σ] is equivalent to the differential stiffness [K^d] in the software.

8.2 Updated element coordinates

When the large displacement effect is included in the nonlinear analysis, the solver employs a method of displaced element coordinate system. This method allows large rotations by updating element coordinates to the deformed geometry, and the equilibrium is computed in the deformed configuration.

8.2.1 Concept of convective coordinates

The concept is based on the fact that the rigid body motion does not contribute to the strain energy and is eliminated from the internal force computation. Consider a rod which underwent rigid body motion as well as deformation as shown below:

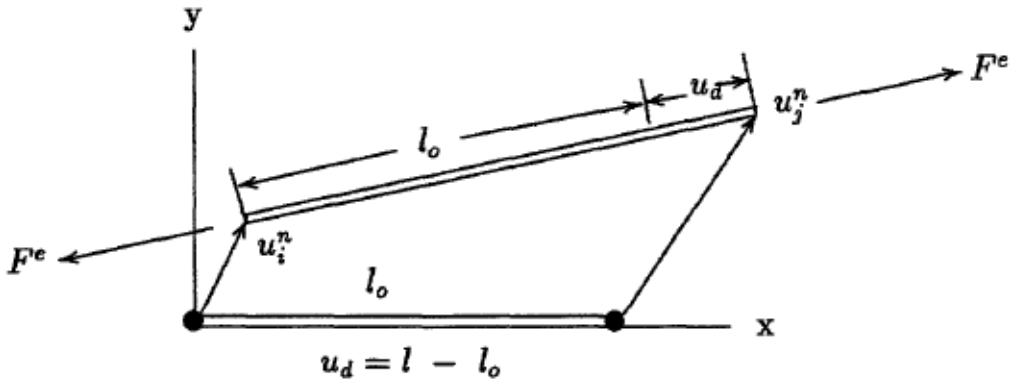


Figure 8-1. Net Deformation of a Rod

The net displacement u_d is measured in the displaced element coordinate system by overlaying the original element on top of the deformed element. The element force can simply be computed by

$$\{F_d\}^e = [K]^e \{u_d\}$$

Equation 8-21.

where the superscript *e* denotes an elemental operation and the subscript *d* denotes the vectors in the displaced element coordinate system. Then the element forces should be transformed into the common coordinate system (namely global coordinate system denoted by a subscript *g*) before assembly for global operations, i.e.,

$$\{F_g\} = \sum T_{bg}^T T_{bd} \{F_d\}^e$$

Equation 8-22.

where the summation sign implies an assembly operation, and T_{bd} and T_{bg} are transformation matrices from displaced to basic and from global to basic coordinate systems, respectively.

8.2.2 Updated coordinates and net deformation

Referring to Figure 8-2, a quadrilateral element is shown in its original and deformed positions (denoted by subscripts *e* and *d*, respectively) with reference to the basic coordinate system (denoted by a subscript *b*).

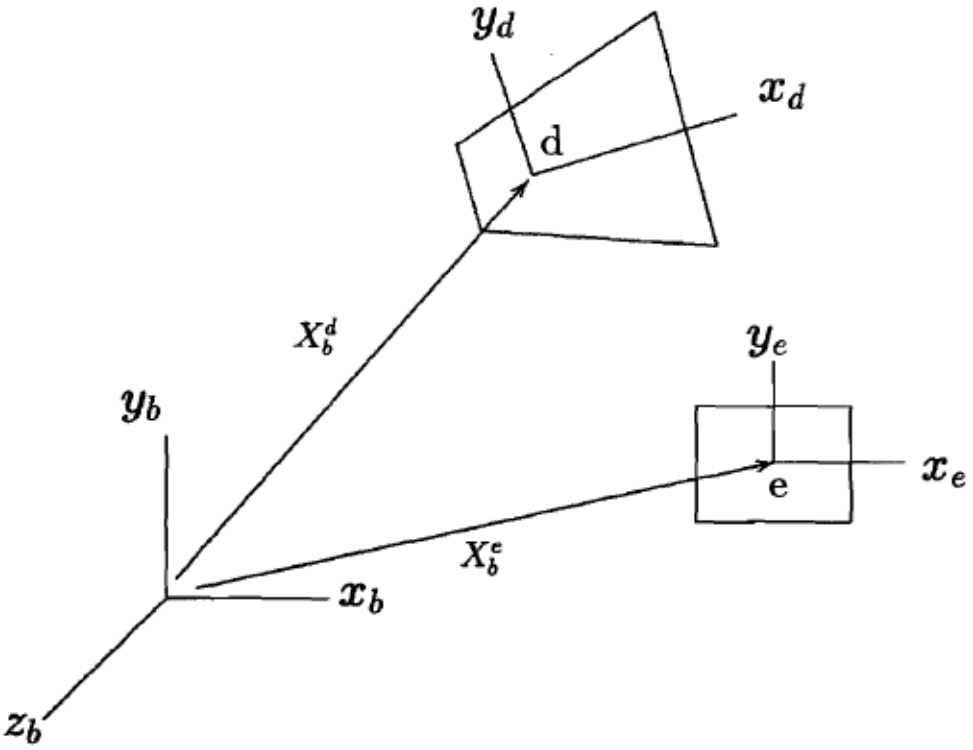


Figure 8-2. Element Coordinates vs. Displaced Coordinates

The element coordinate system is established by bisecting the diagonals of the quadrilateral. Transformation from the element coordinate system to basic coordinates is simply

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_{basic} = [T_{be}] \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_{element} + \begin{Bmatrix} x^e \\ y^e \\ z^e \end{Bmatrix}_{basic}$$

Equation 8-23.

where the position vector (X^e_b in Figure 8-2) of the element coordinate system with respect to the basic coordinate system is denoted by $< x^e, y^e, z^e >^T_{basic}$ and transformation matrix $[T_{be}]$ is

composed of direction cosines of unit vectors of the element coordinate system with respect to the basic coordinate system, i.e.,

$$[T_{be}] = \begin{bmatrix} i_x & j_x & k_x \\ i_y & j_y & k_y \\ i_z & j_z & k_z \end{bmatrix}$$

Equation 8-24.

As the element deforms or displaces, the element coordinate system moves and this is defined as a displaced coordinate system. The displaced coordinate system is established in the same manner as the element coordinate system. Again the transformation should be performed similarly, i.e.,

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_{basic} = [T_{bd}] \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_{displacement} + \begin{Bmatrix} x^d \\ y^d \\ z^d \end{Bmatrix}_{basic}$$

Equation 8-25.

where $< x^d, y^d, z^d >^T_{basic}$ is the position vector of the displaced element coordinate system with respect to the basic coordinate system (X^d_b in Figure 8-2) and $[T_{bd}]$ is formed simliarly to $[T_{be}]$.

In order to isolate the deformation from the rigid body displacements, nodal displacements are computed in the displaced element coordinate system by overlaying the original element as shown in Figure 8-3.

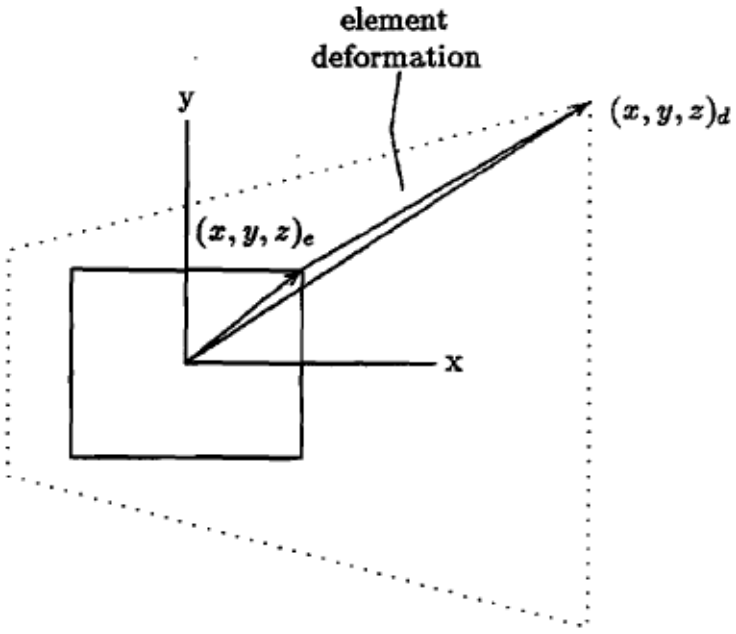


Figure 8-3. Computation of Net Deformation

The net displacements can be computed by subtracting the original nodal coordinates in the element coordinate system from the displaced nodal coordinates in the displaced element coordinate system, i.e.,

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix}_d = \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_d - \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_e$$

Equation 8-26.

in which the nodal coordinates in the element and displaced element coordinate systems can be computed by the following transformations:

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_e = \left[T_{be}\right]^T \left[\begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_{basic} - \begin{Bmatrix} x^e \\ y^e \\ z^e \end{Bmatrix}_{basic} \right]$$

Equation 8-27.

and

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_d = \left[T_{bd}\right]^T \left[\begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_{basic} + \left[T_{bg}\right] \begin{Bmatrix} u \\ v \\ w \end{Bmatrix}_{global} - \begin{Bmatrix} x^d \\ y^d \\ z^d \end{Bmatrix}_{basic} \right]$$

Equation 8-28.

Substitution of Equations 8-27 and 8-28 into Equation 8-26 results in

$$\left\{u_d\right\} = T_{bd}^T \left\{X_b + T_{bg} u_g - X_b^d\right\} - T_{be}^T \left\{X_b - X_b^e\right\}$$

Equation 8-29.

where $\{u_g\}$ is a total displacement (translational components only) in the global coordinates. In the absence of the large displacement effect, the net displacement u_d in Equation 8-29 is reduced to:

$$\left\{u_e^i\right\} = T_{be}^T T_{bg} \left\{u_g^i\right\}$$

8.2.3 Provisions for global operation

It is noted that the net rotations (θ^x, θ^y, and θ^z of each node associated with the shell and beam elements) are computed by a gimbal angle approach (or rotation vector approach) before computing element forces. Subsequently, the element forces have to be transformed to the global coordinate system before assembly for equilibrium check. The internal forces are computed using net displacements and rotations, u_d, i.e.,

$$\{F_g\} = \sum T_{bg}^T T_{bd} [K]^e \{u_d\}$$

Equation 8-30.

if the material is linear or

$$\{F_g\} = \sum T_{bg}^T T_{bd} \int B^T \sigma(u_d)\, dV$$

Equation 8-31.

if nonlinear material is involved. Consequently, the tangent stiffness matrix is formed in the global system by assembling the element stiffness matrices transformed into the global coordinate system from the displaced coordinate system, i.e.,

$$[K] = \sum T_{bg}^T T_{bd} [K]^e T_{bd}^T T_{bg}$$

Equation 8-32.

The update process is performed at every iteration and the updated nodal displacements u_d are used whenever strains and stresses are computed. Effectively, the second order effect due to large rigid body motion is eliminated. However, the displacement output shows the total displacements in the global coordinates, i.e.,

$$\{u_g^i\} = T_{bg}^T T_{bd} \{u_d^i\} + T_{bg}^T T_{bd} \left[T_{be}^T \{x_b^i - x_b^e\} - T_{bd}^T \{x_b^i - x_b^d\} \right]$$

Equation 8-33.

where the subscript *i.* denotes operations on each nodal point. The transformation matrix $[T_{bd}]$ is computed for each element after each iteration and stored in the ESTNL data block for stiffness matrix update when requiredby the stiffness matrix update strategy. On the other hand, the transformation $[T_{bg}]$ is computed for each nodal point and it is not stored but recomputed whenever it is needed. The nodal coordinates in the undeformed geometry, X_b, are available from the data block BGPDT.

This approach can be interpreted as approximate updated Lagrangian method, since the motion of the body follows Lagrangian description. Stresses are computed in the deformed geometry just like Cauchy stress. However, this method of displaced coordinate system is a unique and salient feature in the software. The referential geometry in the updated Lagrangian method is brought up-to-date at every incremental step upon convergence but fixed during the iterative process, which is inherently different from the current method of updating the coordinate system.

8.3 Follower forces

The term "follower force" usually refers to the applied loads that change direction and magnitude with structural displacements and rotations, e.g.,

$$P(u) = \int_A N^T p \hat{n}\, dA$$

Equation 8-34.

where *p* is the magnitude of the pressure on the surface *A*, interpolated by a shape function *N*, and *ñdA* changes as a function of *u*. They generally occur with fluid pressures such as the pressurized balloon, inflated tire, or the lift load on the airplane wing. Other physical applications involve kinematics such as the classical fire hose instability problem or inertia loads on spinning bodies. In the software, the term applies to specific load inputs as defined below.

8.3.1 Basic definition

For geometrical nonlinear analysis, static loads belong to one of two categories, namely:

- Loads defined by fixed vector inputs, which may be calculated once per run and cannot change direction or magnitude.
- Loads defined by the location of one or more GRID points.

The first category includes simple forces, and enforced displacements. The second, follower force category, includes the following Bulk Data inputs:

FORCE1, FORCE2	The direction changes with displacements of the referenced GRID points. The magnitudes of these concentrated loads are constant.
PLOAD, PLOAD4	The pressure loads follow the surface of the solid elements (HEXA, PENTA, PYRAM and TETRA).
RFORCE	Centrifugal loads change in magnitude and direction with motion of the masses attached to the GRID points. The effect may be destabilizing if large motions occur. It is recommended that lumped masses be used with these loads.

Also note that upstream superelements are assumed to be linear and therefore the upstream loads will remain fixed in magnitude and direction. In addition, forces on omitted degrees of freedom (when ASET or OMIT data are present) should not be follower forces.

8.3.2 Implementation

The follower forces depend on the GRID displacements and therefore must be recalculated for each nonlinear iteration and line search. The basic equation for residual error, defined by Equation 9-2, becomes:

$$\left\{R_a^i\right\} = \left\{P_a\left(u^i\right)\right\} - \left\{F_a\left(u^i\right)\right\}$$

Equation 8-35.

where the applied load vector {Pal is now a variable. Corrective Loads are computed based on the updated geometry and added to the initially applied loads to account for the follower forces, i.e.,

$$P\left(u\right) = P\left(0\right) + f\left(u\right)$$

Equation 8-36.

where

$$f(u) = P(u) - P(0)$$

Equation 8-37.

Note that thermal effects are included in the vector {*F*}.

In turn, the tangent matrix could be calculated from Equation 9-3 using derivatives of the loads, which is termed follower matrix. However, the nonlinear solution process ignores the stiffness effects of the changing loads and use the approximation:

$$[K_T] = -\left[\frac{\partial R}{\partial u}\right] \simeq \left[\frac{\partial F}{\partial u}\right]$$

Equation 8-38.

The effect of the approximation is minor in most cases. However, it could become a major concern in thin shell models with pressure loads causing large rotations, where the converged solutions will be correct but the rate of convergence may be slow or cause divergence. Also the buckling solutions or modal analysis on preloaded structure with pressure load may not be correct due to the approximate tangent matrix if the effect of the follower matrix is significant.

The follower force effects in the analysis can be controlled by the parameter LGDISP. Three options are available in PARAM LGDISP:

- = 0 for no geometric nonlinearity
- = 1 for full geometric nonlinearity (including follower forces)

Chapter 9: Solution methods

9.1 Solution Algorithm

Let n represent the current time step, i the current iteration, (n-1) will be the last converged time step. If n=1, the previous converged time step will be the initial conditions.

U represents the displacement vector.

P represents the external force vector.

F represents the internal force vector.

R represents the residual (R= P-F) vector.

t represents time.

Δt represents time step size.

1. Assemble stiffness matrix for the structure if this is the first step, or if a stiffness update is requested any time during the solution. Decompose the stiffness matrix.
2. Save last converged displacements (U_{n-1}) and external forces (P_{n-1})
3. $t_n=t_{n-1}+\Delta t$
4. Obtain the external force vector P_n^1 for the first iteration of current time step.
   a. If i>1: Include follower force effects to obtain P_n^1.
5. Compute internal forces $F_n^1(U_n^{i-1})$
6. Compute residual: $R = P_n^i - F_n^i$
7. If i>1:
   a. Compute: $E_2=\Delta U_n^{i-1}$. R
   b. Compute $E_r=E_2/E_1$
   c. If $E_r>1$ or $E_r<-100$, ndiv=ndiv+1
   d. Write quasi Newton vectors for BFGS.
8. Compute displacement increment: $\Delta U_n^i=K^{-1}$ R
9. Compute $E_1=\Delta U_n^i.R$
10. Compute $U_n^i=U_n^{i-1}+\Delta U_n^i$
11. Compute current stiffness parameter

12. Check for convergence.
    a. If solution has converged and stiffness update is needed, go to 1.
    b. If solution has converged and stiffness update isn't necessary, go to 2.
    c. If solution hasn't converged and stiffness update is needed, go to 1 (skip 3).
    d. If solution hasn't converged and stiffness update isn't needed, go to 4.

9.2 Adaptive Solution Strategies

Nonlinear finite element computations comprise material processes, element force computations, and various global solution strategies. The computational procedure involves incremental and iterative processes ranging from local subincrements to global solution processes. Performance of the finite element program can be scrutinized from three different perspectives: computational efficiency, solution accuracy, and effectiveness. All of these attributes of the nonlinear program can be improved by adaptive algorithms.

9.3 Newton's method of iteration

The equilibrium equations in the g-set may be written as

$\{P_g\} + \{Q_g\} - \{F_g\} = \{0\}$ where $\{P_g\}$, $\{Q_g\}$, and $\{F_g\}$ represent vectors of applied loads, constraint forces, and element nodal forces, respectively. Element nodal forces are nonlinear functions of displacements for nonlinear elements. Since the equilibrium condition is not immediately attained in the presence of nonlinear elements, an iterative scheme such as the Newton-Raphson method is required. Since the error vanishes at constrained points and the constraint forces vanish at free points, the unbalanced forces acting at nodal points at any iteration step are conveniently defined as an error vector by

Equation 9-1.

$$\{R_a\} = \{P_a\} - \{F_a\}.$$

Equation 9-2.

Notice that the a-set is equivalent to the l-set in the nonlinear analysis because the r-set does not exist. The subscript a will be dropped for simplicity in the following discussion.

Based on Newton's method, a linearized system of equations is solved for incremental displacements by Gaussian elimination in succession. The Jacobian of the error vector emerges as the tangential stiffness matrix. The equation to solve at the i-th iteration is

$[K_T]\ \{\Delta u^i\} = \{R^{i-1}\}$

Equation 9-3.

where

$$[K_T] = -\left[\frac{\partial R}{\partial u}\right]_{u=u^i} = \left[\frac{\partial F}{\partial u}\right]_{u=u^i}$$

and

$\{\Delta u^i\} = \{u^i\} - \{u^{i-1}\}$,

$\{R^i\} = \{P\} - \{F(u^i)\}$.

The iteration continues until the residual error {R} and the incremental displacements {Δu} become negligible, which is signified by the convergence criteria.

The tangential stiffness consists of the geometric stiffness in addition to the material stiffness, i.e., without regard to the coordinate transformation,

$$\left[K_T\right] = \frac{\partial}{\partial u}\int_V \left[B\right]^T \left\{\sigma\right\} dV = \left[K^m + K^d\right]$$

Equation 9-4.

where [K^m] and [K^d] refer to the material and the differential stiffness, respectively. The material stiffness is given in Equation 4-17 with a material tangential matrix for [*D*]. The differential stiffness, which is caused by the initial stress, is defined as follows:

$$\left[K^d\right] = \int_V \left[\frac{\partial B_N}{\partial u}\right]^T \left\{\sigma\right\} dV = \int_V \left[G\right]^T \left[M\right]\left[G\right] dV$$

Equation 9-5.

where [B_N] represents the second order effects in the strain-displacement relations, [*G*] consists of derivatives of shape functions and [*M*] is a function of stresses. Notice that the initial displacement stiffness is not included in [K_T] because its effects are already eliminated in the element formulation.

Newton's procedure was previously implemented using a corrective force. Recalling that the element forces for linear elements are expressed as

{F} = [K] {u},

a corrective force vector may be defined as

{C} = {F} – [K] {u}

Equation 9-6.

The iteration starts with initial values

$\{u^0\}$ = last converged displacement;

$\{R^0\} = P - F(u^0) = \{\Delta P\} + \{Re\}$;

where *{ΔP}* is an incremental load vector and $\{R_e\}$ is a residual load error carried over from the last converged solution. Then the successive error vectors can be evaluated by

$R^{i+1} = R^i - K(u^{i+1} - u^i) - (C^{i+1} - C^i)$.

Equation 9-7.

It is noted that the corrective force vector vanishes for linear elements. The corrective force calculation has been removed, and the error vector is computed directly from the internal forces, i.e.,

$$\{R^{i+1}\} = \{P\} - \{F^i\}$$

Equation 9-8.

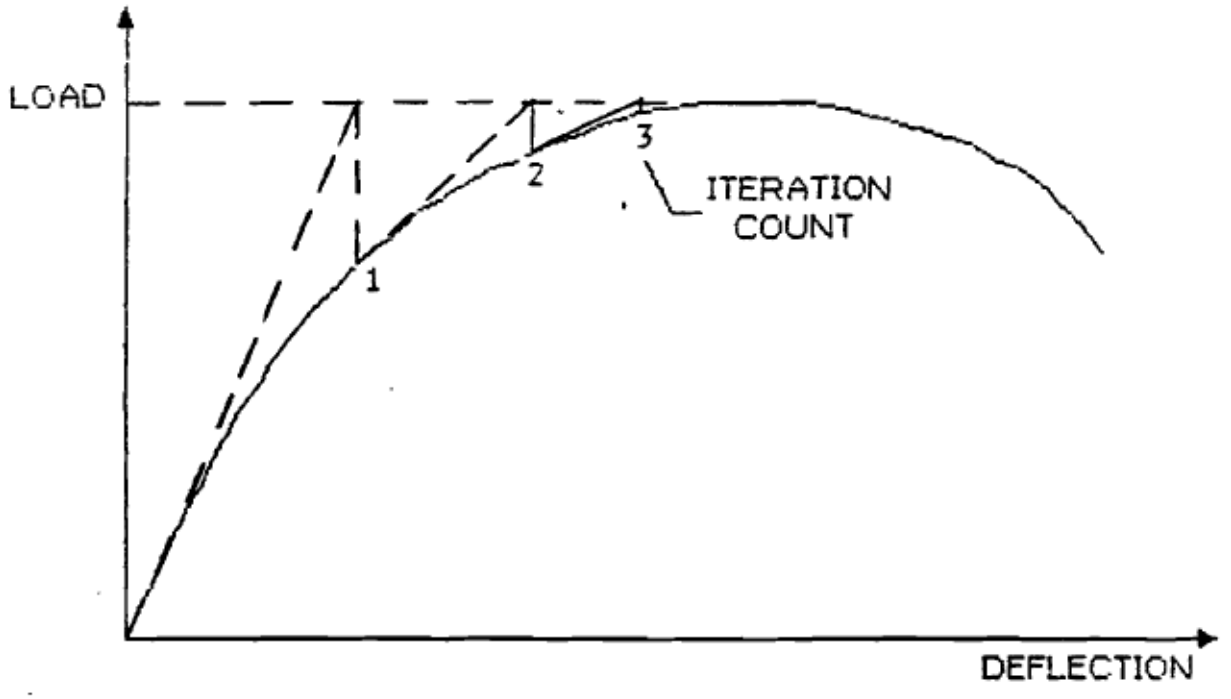
Then the residual load error is automatically carried over to the next incremental process.

The merit of the Newton-Raphson method is the quadratic rate of convergence, i.e.,

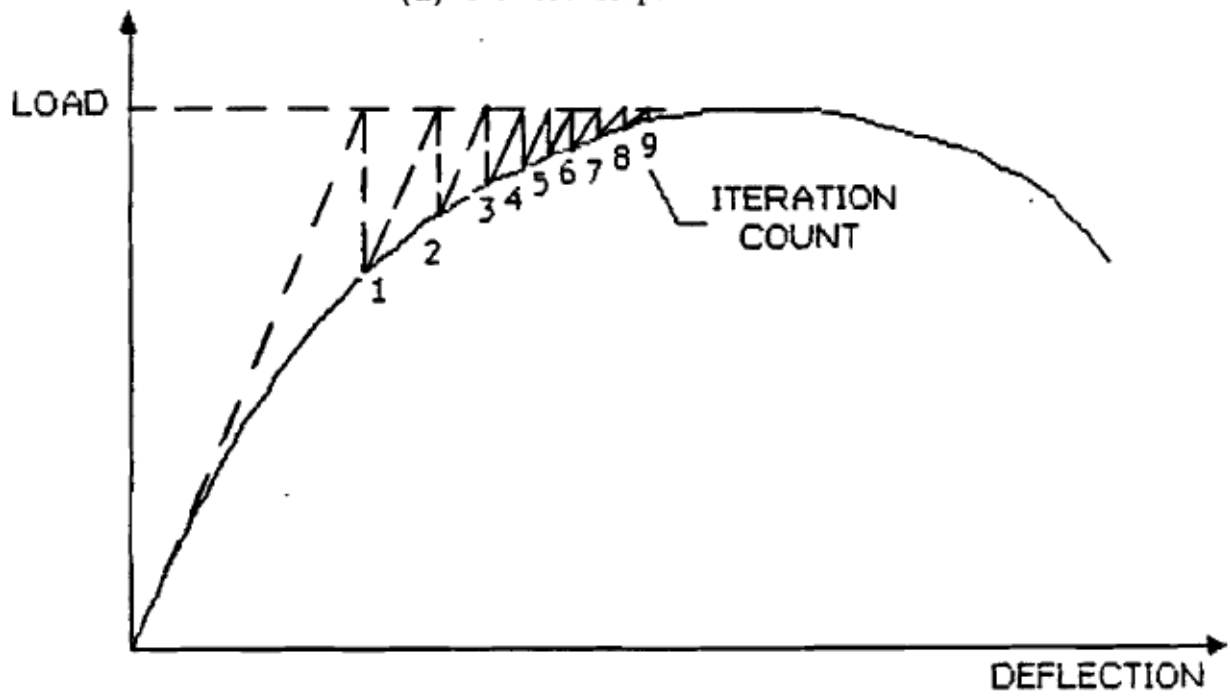
$$||u^* - u^{i+1}|| \leq q||u^* - u^i||^2$$

Equation 9-9.

where u^* is a true value of $\{u\}$, q is a constant, and 1111 represents a vector norm. From a practical standpoint, however, determination of the tangential stiffness and its inverse at each iteration entails a considerable amount of computation. As Figure 9-1 suggests, one may resort to the modified Newton's method which requires the tangential stiffness to be evaluated just once at the initial position, $\{u^o\}$, and used thereafter to solve for $\{\Delta u^i\}$. However, more iterations are required for a given accuracy by the modified Newton's method. The Gaussian elimination method is better suited for this approach than the iterative descent method because the decomposition is performed only once.



(a) Newton-Raphston Method



(b) Modified Newton's Method

Figure 9-1. Newton's Methods for Iteration

9.4 Stiffness update strategies

Among other features of the solution algorithm, the stiffness matrix update has probably the most profound effect on the success of the nonlinear solution. In spite of its significance, however, it is very difficult to implement a robust algorithm for update strategy due to the lack of a prior information regarding the right timing for an update. In this section, stiffness update strategies are reviewed with respect to the static analysis. Variations of the modified Newton's method are adopted in the software.

However, the modified Newton's method could lead to divergence when the stiffness changes drastically, as demonstrated in Figure 9-1, unless the tangential stiffness is reevaluated at the critical point. To this end, an adaptive matrix update method is unavoidable. Stiffness update strategies are established to update the stiffness matrix on an as needed bases such as probable divergence.

9.4.1 Update principles

Newton's method could be trapped in an infinite loop, oscillating about the local maximum as illustrated in Figure 9-1(a). This difficulty is overcome during the Newton's iteration by discarding the differential stiffness, $[K^d]$, when the tangential stiffness is not positive definite as shown in Figure 9-1(b).

Control over the stiffness update method is achieved with the KUPDATE parameter on the NLCNTL entry:

1. If KUPDATE=1 (Full Newton method), stiffness is updated after each iteration.
2. If KUPDATE=0 (Auto method, default), stiffness is updated based on the change in the value of current stiffness parameter since the last stiffness update.
3. If KUPDATE=-1 (Initial stiffness method), no stiffness update is made during the solution.
4. If KUPDATE=N (N>1,Quasi Newton method), if solution time step doesn't converge in N iterations, stiffness is updated at the iterated configuration. If the parameter TSTEPK is set to YES (Default=NO) on the NLCNTL entry, then stiffness update is also performed before beginning a new time step.

Current stiffness parameter approach for the automatic stiffness update method (KUPDATE=0)

The current stiffness parameter approach proposed by Bergan and Crisfield [*] is used for automatic stiffness update method. The current stiffness parameter gives a scalar measure for the stiffness of the structure at the current loading condition.

$$S_n^i = \frac{\left(\Delta P_n^i \cdot \Delta P_n^i\right)\left(\Delta P_1^1 \cdot \Delta U_1^1\right)}{\left(\Delta P_1^1 \cdot \Delta P_1^1\right)\left(\Delta P_n^i \cdot \Delta U_n^i\right)}$$

Where, n is the time step index,

I is the teration index,

$\Delta P_n^i = P_n^i - P_{n-1}^i$ is the incremental applied load between time step n-1 and n,

$\Delta P_1^1 = P_1^1 - P_0^1$ is the initial applied load (for the first iteration of the first time step),

P_1^1 is the applied load in the first iteration of the first time step of the first subcase,

$\Delta U_n^i = U_n^i - U_{n-1}^i$ is the incremental displacement between time step n-1 and n,

$\Delta U_1^1 = U_1^1 U_0^1$ is the initial displacement increment (after the first iteration of first step).

After the first iteration of the first step, $S_1^1=1$, that is, the initial value of the current stiffness parameter is one. After each iteration, the value of current stiffness parameter is recomputed. The change in current stiffness parameter is computed for each iteration as:

$$\Delta S = \frac{100\left(\left|S_n^i - S_{ref}^i\right|\right)}{S_{ref}}$$

Where, S_{ref} is the reference value for current stiffness parameter. At the start of the solution, S_{ref} is set to 1.0. Stiffness is updated if $\Delta S \geq \alpha$, where $\alpha=5.0$ for a problem with structural loading only, $\alpha=20.0$ for a problem with pure thermal load, and $\alpha=10.0$ for a problem with a combination of structural and thermal load. The value for α can be defined by the user on the NLCNTL card through parameter CSTFPAR. Valid input for CSTFPAR is a real number.

S_{ref} is updated upon updating stiffness to correspond to the value of S_n^i.

1. Bergan. P, Horrigmoe. G, Krakeland . B, and Soreide T., SOLUTION TECHNIQUES FOR NON-LINEAR FINITE ELEMENT PROBLEMS, International Journal for Numerical Methods in Engineering, Vol. 12, 1677-1696 (1978)

2. Crisfield M. A. , Non-linear Finite Element Analysis of Solids and Structures, Volume 1: Essentials, John Wiley & Sons, Chichester, 1991

9.4.2 Divergence criteria

The MAXDIV parameter in the NLCNTL entry requires an integer to specify a limit on the probable divergence conditions allowed for each time step to continue. In each iteration of a time step, the following three quantities are computed:

$$E_\gamma = \frac{\Delta U_n^i \cdot R_n^i}{\Delta U_n^i \cdot R_n^{i-1}}$$

Equation 9-10.

$$\lambda_p^i = \frac{\varepsilon_p^i}{\varepsilon_p^{i-1}}$$

Equation 9-11.

$$\lambda_w^i = \frac{\varepsilon_w^i}{\varepsilon_w^{i-1}}$$

Equation 9-12.

where,

$R_n^{i-1}=P_n^{i-1}-F_n^{i-1}$ (U_n^{i-2}) is the residual for iteration i-1 for time step n,

$\Delta U_n^i=K^{-1}$ R_n^{i-1} is the displacement increment computed in iteration i-1 for time step n,

$R_n^i=P_n^i-F_n^i$ (U_n^{i-1}) is the residual for iteration i for time step n,

ε_p^i is the computed error for the force norm for iteration i for time step n,

ε_p^{i-1} is the computed error for the force norm for iteration i-1 for time step n,

ε_w^i is the computed error for the energy norm for iteration i for time step n,

ε_w^{i-1} is the computed error for the energy norm for iteration i-1 for time step n.

For the first iteration of every new time step, the value for λ_p^i and λ_w^i is set to 0.9999. The variable NDIV is initiated to 0 in the first iteration of each time step. NDIV is incremented if:

1. If E_r>1 or E_r<-100, NDIV=NDIV+1

2. If λ_w^i>1.00 and λ_p^i>1.00, NDIV=NDIV+1

NDIV is reset to 0 if neither of the two conditions above are met in any given iteration. The solution is considered to have diverged if NDIV>MAXDIV. MAXDIV is a NLCNTL parameter which controls maximum allowable divergences in a time step (Default=3).

Upon divergence, the software will:

1. Revert the solution to the last converged point.

2. Reform stiffness at the last converged configuration (unless KUPDATE=-1, in that case no stiffness update is performed).

3. Perform bisection, and attempt to solve the time step with a reduced size from the last converged point.

   For a timestep of size Δt, on bisection the time step is reduced to:

   $\Delta t_1 = 1/2\ \Delta t$

   Bisection continues until the solution converges, that is,

   $\Delta t_k = 1/2\ \Delta t_{k-1} = 1/2^k\ \Delta t$

   where k is a bisection count. Once the bisection is successful (rendering a converged solution), the integration proceeds to the next time step. If k = 1, the same time step size is used for the next time step.

   If k > 1, an effort is made to accelerate the solution process by increasing the time step size for the next step. The time step size for the next time step is influenced by all of the following factors:

   a. The number of iterations (i) that were required to reach convergence with time step size of Δt_k.

   b. Number of bisections (k) performed.

   c. Remaining time left (δt) to complete the original time step of size Δt.

      The time step size used for the next time step is m Δt_k, where $1 \le m \le k - 1$. The largest value of m that satisfied the following conditions is used:

      A. $I \le 3$ or $I \le MAXITR/2^m$, where MAXITR is the maximum permissible iterations for a given timestep.

      B. $\delta t/m\ \Delta tk$ is an integer greater than 0.

Although the goal of increasing time step size is to reduce the number of time steps till completion of solution, for some problems, further bisection may be required in subsequent steps.

The maximum number of bisections is limited by the parameter MAXBIS (default=5). The bisection process is activated on an as-needed basis. You also have an option to suppress bisection by specifying MAXBIS=0.

d. If the solution fails to converge even after attempting maximum permissible bisections, the solution is terminated with a fatal message. Results corresponding to the last converged user requested output time are printed.

9.5 Convergence criteria

The convergence test is an important factor that affects accuracy and overall efficiency in nonlinear finite element analysis. Out-of-balance forces and changes in displacements should vanish upon convergence in an iterative process. The energy error accommodates both quantities and is usually adequate for most problems. However, the displacements could be in gross error while the residual load error is negligible, or vice versa.

In order to ensure accurate and consistent convergence, multiple criteria with errors measured in terms of displacements, loads, and energy should be combined. It is the error function and the convergence tolerance that characterize the criteria. Error functions are formulated using the weighted normalization so that the error measures are dimensionless. Tolerances should be realistic for the solution scheme to be efficient. In this context, variations are considered in search of the best workable combinations of error functions and tolerances for a wide class of structural problems.

9.5.1 Rudimentary considerations

The convergence test is a decision-making process, on which termination of the iterative process is based, while the true solution is not known. The convergence criteria are extremely important for the incremental/iterative solution strategy to be effective and efficient, because improper criteria could cause inefficiency as well as inaccuracy. It is rather astounding to:find a scarcity of publications on this subject, considering the significant impact of the convergence criteria on the accuracy and the efficiency of the computation. Two distinct aspects are involved in the convergence criteria:

- Error functions to be minimized by the iteration.
- Tolerances of error functions within which errors are acceptable.

Both aspects must be defined properly for the criteria to be effective, for the solution scheme to be efficient, and for the solution to be accurate.

There are no universally accepted convergence criteria to date in the field of finite element analysis. Conditions to be met by ideal convergence criteria for a general-purpose finite element analysis have been contemplated. The convergence criteria should:

- be satisfied for linear cases at all times.
- be independent of structural units.
- be reliable (cancellation of errors are not acceptable).

- render consistent accuracy.
- be independent of structural characteristics (stiffening or softening).
- be able to handle all the loading cases including constant loading, unloading, and no external loading (applicable to creep analysis).
- have smooth transitions after the stiffness updates and loading changes.

These conditions dictate the formulation of error functions to be discussed.

9.5.2 Convergence conditions

The iteration continues until the convergence is attained by satisfying the convergence criteria and the residual error vector at convergence is carried over to the next incremental step. When the convergence criteria are satisfied, the out-of-balance forces and the changes in displacements should be sufficiently small so that the remaining error is not physically significant nor will it cause any detrimental effects, numerically or physically, on the succeeding incremental steps. Convergence tolerances have the following effects:

- excessively tight tolerances cause a waste of computing resources for unnecessary accuracy.
- excessively loose tolerances cause not only inaccuracy but convergence difficulties in the subsequent steps due to cumulative errors.

The fundamental difficulty of the convergence tests for a structural analysis lies in the fact that the base vectors (forces and displacements) involve inconsistent units, namely, combinations of forces and moments or translations and rotations. Indiscriminate use of these vectors will cause unit-dependent convergence criteria. For example, while an error in forces is dominant when the model is expressed in newton-meter, the error would be dominated by moments if the same model is described in newton-millimeter.

The most natural and reasonable criterion for the convergence test is formulated in terms of an energy error. The energy error is the logical choice because both the out-of-balance forces $\{R\}$ and the change in displacements $\{\Delta u\}$ should be minimized by the iteration process. Furthermore, energy quantities do not pose problems of inconsistent units due to mixed units associated with translations and rotations.

Although the convergence test in terms of energy errors is usually adequate, some distinct errors are not detected with this criterion; i.e., displacements are in gross error while the residual load error is negligible, or vice versa. This would be the case if the degrees-of-freedom in error have a very small or a very large stiffness. Such cases compel the need for criteria in terms of loads and displacements. Nominally, by visualizing the load-deflection curve for a one-dimensional case, it can be noticed that the convergence criterion in terms of loads governs the stiffening structure and the criterion in terms of displacements governs the softening structure. Scalar error functions for these criteria are formulated to be dimensionless by introducing the weighted normalization.

9.5.3 Error functions and weighted normalization

Error functions are defined for errors in displacement, residual force and energy criterion. For displacement and residual force error criterion, L2 vector norms are used. The L2 norm is indicated by $||V||_2$, where V represents a vector (for example, ΔU). In solution 401, error criterion corresponding

to displacement, force and energy (work) are computed in every iteration. Error in energy (work) is used as the default convergence criterion. These criteria are computed as follows:

- EPSU (displacement criterion)

$$\varepsilon_U^i = \frac{\left\| U_n^i - U_n^{i-1} \right\|_2}{\left\| U_n^i - U_{n-1} \right\|_2}$$

Equation 9-13.

where,

U_n^i is the displacement vector for iteration i of time step n,

U_n^{i-1} is the displacement vector for iteration i-1 of time step n, and

U_{n-1} is the displacement vector at convergence for time step n-1.

The displacement error computation can be summarized as the ratio of L2 norm of the incremental displacement in the current iteration and the L2 norm of the incremental displacement in the current time step. The error in displacement in the first iteration for every time step should be 1.000.

- EPSP (force criterion)

$$\varepsilon_p^i = \frac{\left\| R_n^i \right\|_2}{\max\left(\left\| P_n^i \right\|_2, \left\| F_n^i \right\|_2, \left\| R_n^1 \right\|_2 \right)}$$

where,

P_n^i is the external force vector for iteration i of time step n,

F_n^i is the internal force vector for iteration i of time step n,

R_n^i is the residual force vector for iteration i of time step n, and

R_n^1 is the residual force vector for iteration 1 of time step n.

Equation 9-14.

- EPSW (work criterion)

$$\varepsilon_W^i = \frac{ABS\left(\Delta U_n^i . R_n^i \right)}{\max\left(ABS\left(U_n^i . P_n^i \right), ABS\left(U_n^i . F_n^i \right), ABS\left(U_n^1 . R_n^1 \right) \right)}$$

Equation 9-15.

Where,

$\Delta U_n^i$is the incremental displacement vector computed for iteration i of time step n,

R_n^i is the residual force vector for iteration i of time step n,

U_n^i is the total displacement vector for iteration i of time step n,

P_n^i is the external force vector for iteration i of time step n,

F_n^i is the internal force vector for iteration i of time step n,

R_n^1 is the residual force vector for iteration 1 of time step n, and

U_n^1 is the total displacement vector for iteration 1 of time step n.

Default tolerances for these error criterion are:

EPSU (Displacement): –1.0E-2

EPSP (Force): –1.0E-2

EPSW (Work): –1.0E-6

You can specify custom convergence tolerances on the NLCNTL bulk entry. The solution is considered to have converged if the computed error criteria are less than the tolerance specified.

9.5.4 Implementation

The convergence tolerance determines the efficiency of the solution scheme as well as the accuracy of the solution. The tolerance should be realistic, not too tight nor too loose. It is difficult to choose optimal default values for the convergence tolerances. However, efforts have been made to set the default values to provide reliable solutions to the general class of problems. Thus, default tolerances should be adhered to until good reasons are found to change them.

The following three error functions (in terms of displacements, loads, and energy) are computed and compared to tolerances.

E_u < EPSU (=10^{-2} by default)

E_p < EPSP (=10^{-2} by default)

E_w < EPSW (=10^{-6} by default)

where EPSU, EPSP, and EPSW are tolerances specified in the NLCNTL entry. However, only those criteria chosen by the user (combinations of U,P, and/or W) are designed to be satisfied for convergence.

It is noted that divergence conditions are established independent of convergence criteria.

Siemens Industry Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific
Suites 4301-4302, 43/F
AIA Kowloon Tower, Landmark East
100 How Ming Street
Kwun Tong, Kowloon
Hong Kong
+852 2230 3308

About Siemens PLM Software

Siemens PLM Software, a business unit of the Siemens Industry Automation Division, is a leading global provider of product lifecycle management (PLM) software and services with 7 million licensed seats and 71,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with companies to deliver open solutions that help them turn more ideas into successful products. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.