NX Nastran 12

Advanced Nonlinear Theory and Modeling Guide

Proprietary & Restricted Rights Notice

© 2017 Siemens Product Lifecycle Management Software Inc. All Rights Reserved.

This software and related documentation are proprietary to Siemens Product Lifecycle Management Software Inc. Siemens and the Siemens logo are registered trademarks of Siemens AG. NX is a trademark or registered trademark of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries.

NASTRAN is a registered trademark of the National Aeronautics and Space Administration. NX Nastran is an enhanced proprietary version developed and maintained by Siemens Product Lifecycle Management Software Inc.

MSC is a registered trademark of MSC.Software Corporation. MSC.Nastran and MSC.Patran are trademarks of MSC.Software Corporation.

All other trademarks are the property of their respective owners.

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.

HDF5 (Hierarchical Data Format 5) Software Library and Utilities Copyright 2006-2016 by The HDF Group

NCSA HDF5 (Hierarchical Data Format 5) Software Library and Utilities Copyright 1998-2006 by the Board of Trustees of the University of Illinois. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted for any purpose (including commercial purposes) provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the following disclaimer in the documentation and/or materials provided with the distribution.

3. In addition, redistributions of modified forms of the source or binary code must carry prominent notices stating that the original code was changed and the date of the change.

4. All publications or advertising materials mentioning features or use of this software are asked, but not required, to acknowledge that it was developed by The HDF Group and by the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign and credit the contributors.

5. Neither the name of The HDF Group, the name of the University, nor the name of any Contributor may be used to endorse or promote products derived from this software without specific prior written permission from The HDF Group, the University, or the Contributor, respectively.

DISCLAIMER: THIS SOFTWARE IS PROVIDED BY THE HDF GROUP AND THE CONTRIBUTORS "AS IS" WITH NO WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED.

In no event shall The HDF Group or the Contributors be liable for any damages suffered by the users arising out of the use of this software, even if advised of the possibility of such damage

Table of contents

1. Introduction

1.1 Objective of this manual

This Theory and Modeling Guide serves two purposes:

- To provide a concise summary of the theoretical basis of Advanced Nonlinear Solution as it applies to Solution 601 and Solution 701. This includes the finite element procedures used, the elements and the material models. The depth of coverage of these theoretical issues is such that the user can effectively use Solutions 601 and 701. A number of references are provided throughout the manual which give more details on the theory and procedure used in the program. These references should be consulted for further details. Much reference is made however to the book *Finite Element Procedures* (ref. KJB).

  ref. K.J. Bathe, *Finite Element Procedures, 2nd ed.*, Cambridge, MA, Klaus-Jürgen Bathe, 2014.

- To provide guidelines for practical and efficient modeling using Advanced Nonlinear Solution. These modeling guidelines are based on the theoretical foundation mentioned above, and the capabilities and limitations of the different procedures, elements, material models and algorithms available in the program. NX Nastran commands and parameter settings needed to activate different analysis features are frequently mentioned.

It is assumed that the user is familiar with NX Nastran fundamentals pertaining to linear analysis. This includes general knowledge of the NX Nastran structure, commands, elements, materials, and loads.

We intend to update this report as we continue our work on Advanced Nonlinear Solution. If you have any suggestions regarding the material discussed in this manual, we would be glad to hear from you.

1.2 Overview of Advanced Nonlinear Solution

- Advanced Nonlinear Solution is a Nastran solution option focused on nonlinear problems. It is capable of treating geometric and material nonlinearities as well as nonlinearities resulting from contact conditions. State-of-the-art formulations and solution algorithms are used which have proven to be reliable and efficient.

- Advanced Nonlinear Solution supports static and implicit dynamic nonlinear analysis via Solution 601, and explicit dynamic analysis via Solution 701. Solution 601 also supports heat transfer analysis and coupled structural heat transfer analysis.

- Advanced Nonlinear Solution supports many of the standard Nastran commands and several commands specific to Advanced Nonlinear Solution that deal with nonlinear features such as contact. The NX Nastran Quick Reference Guide provides more details on the Nastran commands and entries that are supported in Advanced Nonlinear Solution.

- Advanced Nonlinear Solution supports many of the commonly used features of linear Nastran analysis. This includes most of the elements, materials, boundary conditions, and loads. Some of these features are modified to be more suitable for nonlinear analysis, and many other new features are added that are needed for nonlinear analysis.

- The elements available in Advanced Nonlinear Solution can be broadly classified into rods, beams, 2-D solids, 3-D solids, shells, 2-D fluids, 3-D fluids, scalar elements and rigid elements. The formulations used for these elements have proven to be reliable and efficient in linear, large displacement, and large strain analyses. Chapter 2 provides more details on the elements.

- The material models available in Advanced Nonlinear Solution are elastic isotropic, elastic orthotropic, plastic bilinear/multilinear, plastic-cyclic, hyperelastic, gasket, nonlinear elastic isotropic, shape memory alloy, viscoelastic and fluid. Thermal and creep effects can be added to some of these materials. Chapter 3 provides more details on these material models.

- Advanced Nonlinear Solution has very powerful features for contact analysis. These include several contact algorithms and different contact types such as single-sided contact, double-sided contact, self-contact, and tied contact. Chapter 4 provides more details on contact.

- Loads, boundary conditions and constraints are addressed in Chapter 5. Time varying loads and boundary conditions are common to nonlinear analysis and their input in Advanced Nonlinear Solution is slightly different from other Nastran solutions, as discussed in Chapter 5.

- Solution 601 of Advanced Nonlinear Solution currently supports two nonlinear structural analysis types: static and implicit transient dynamic. Details on the formulations used are provided in Chapter 6. Other features of nonlinear analysis, such as time stepping, load displacement control (arc length method), line search, and available solvers are also discussed in Chapter 6.

- Solution 701 of Advanced Nonlinear Solution is dedicated to explicit transient dynamic analysis. Details on the formulations used are provided in Chapter 7. Other features of explicit analysis, such as stability and time step estimation, are also discussed in Chapter 7.

- Solution 601 of Advanced Nonlinear Solution also supports two heat transfer or coupled structural heat transfer analysis types. The first type 153 is for static structural with steady state heat transfer, or just steady state heat transfer. The second analysis type 159 is for cases when either the structural or heat transfer models are transient (dynamic). This type can also be used for just transient heat transfer analysis. Details of the heat transfer analysis are provided in Chapter 8, and details of the thermo-mechanical coupled (TMC) analysis are provided in Chapter 9.

- Additional capabilities present in Advanced Nonlinear Solution such as restarts, stiffness stabilization, initial conditions, and parallel processing are discussed in Chapter 10.

- Most of the global settings controlling the structural solutions in Advanced Nonlinear Solution are provided in the NXSTRAT bulk data entry. This includes parameters that control the solver selection, time integration values, convergence tolerances, contact settings, etc. An explanation of these parameters is found in the NX Nastran Quick Reference Guide.

- Similarly, most of the global settings controlling the heat transfer or coupled solutions in Advanced Nonlinear Solution are provided in the TMCPARA bulk data entry.

1.2.1 Choosing between Solutions 601 and 701

- The main criterion governing the selection of the implicit (Solution 601) or explicit (Solution 701) formulations is the time scale of the solution.

ref. KJB Section 9.2

- The implicit method can use much larger time steps since it is unconditionally stable. However, it involves the assembly and solution of a system of equations, and it is iterative. Therefore, the computational time per load step is relatively high. The explicit method uses much smaller time steps since it is conditionally stable, meaning that the time step for the solution has to be less than a certain critical time step, which depends on the smallest element size and the material properties. However, it involves no matrix solution and is non-iterative. Therefore, the computational time per load step is relatively low.

- For both linear and nonlinear static problems, the implicit method is the only option.

- For heat transfer and coupled structural heat transfer problems, the implicit method is the only option.

- For low-speed dynamic problems, the solution time spans a period of time considerably longer than the time it takes the wave to propagate through an element. The solution in this case is dominated by the lower frequencies of the structure. This class of problems covers most structural dynamics problems, certain metal forming problems, crush analysis, earthquake response and biomedical problems. When the explicit method is used for such problems the resulting number of time steps will be excessive, unless mass-scaling is applied, or the loads are artificially applied over a shorter time frame. No such modifications are needed in the implicit method. Hence, the implicit method is the optimal choice.

- For high-speed dynamic problems, the solution time is comparable to the time required for the wave to propagate through the structure. This class of problems covers most wave propagation problems, explosives problems, and high-speed impact problems. For these problems, the number of steps required with the explicit method is not excessive. If the implicit method uses a similar time step it will be much slower and if it uses a much larger time step it will introduce other solution errors since it will not be capturing the pertinent features of the solution (but it will remain stable). Hence, the explicit method is the optimal choice.

- A large number of dynamics problems cannot be fully classified as either low-speed or high-speed dynamic. This includes many crash problems, drop tests and metal forming problems. For these problems both solution methods are comparable. However, whenever possible (when the time step is relatively large and there are no convergence difficulties) we recommend the use of the implicit solution method.

- Note that the explicit solution provided in Solution 701 does not use reduced integration with hour-glassing. This technique reduces the computational time per load step. However, it can have detrimental effect on the accuracy and reliability of the solution.

- Since the explicit time step size depends on the length of the smallest element, one excessively small element will reduce the stable time step for the whole model. Mass-scaling can be applied to these small elements to increase their stable time step. The implicit method is not sensitive to such small elements.

- Since the explicit time step size depends on the material properties, a nearly incompressible material will also significantly reduce the stable time step. The compressibility of the material can be increased in explicit analysis to achieve a more acceptable solution time. The implicit method is not as sensitive to highly incompressible materials (provided that a mixed formulation is used).

- Higher order elements such as the 10-node tetrahederal, 20 and 27 node brick elements are only available in implicit analysis. They are not used in explicit analysis because no suitable mass-lumping technique is available for these elements.

- Model nonlinearity is another criterion influencing the choice between implicit and explicit solutions. As the level of nonlinearity increases, the implicit method requires more time steps, and each time step may require more iterations to converge. In some cases, no convergence is reached. The explicit method however, is less sensitive to the level of nonlinearity.

  Note that when the implicit method fails it is usually due to non-convergence within a time step, while when the explicit method fails it is usually due to a diverging solution.

- The memory requirements is another factor. For the same mesh, the explicit method requires less memory since it does not store a stiffness matrix and does not require a solver. This can be significant for very large problems.

- Since Advanced Nonlinear Solution handles both Solution 601 and Solution 701 with very similar inputs, the user can in many cases restart from one analysis type to the other. This capability can be used, for example, to perform implicit springback analysis following an explicit metal forming simulation, or to perform an explicit analysis following the implicit application of a gravity load.

  It can also be used to overcome certain convergence difficulties in implicit analyses. A restart from the last converged implicit solution to explicit can be performed, then, once that stage is passed, another restart from explicit to implicit can be performed to proceed with the rest of the solution.

1.2.2 Units

In Advanced Nonlinear Solution, it is important to enter all physical quantities (lengths, forces, masses, times, etc.) using a consistent set of units. For example, when working with the SI system of units, enter lengths in meters, forces in Newtons, masses in kg, times in seconds. When working with other systems of units, all mass and mass-related units must be consistent with the length, force and time units. For example, in the USCS system (USCS is an abbreviation for the U.S. Customary System), when the length unit is inches, the force unit is pound and the time unit is second, the mass unit is lb-sec^2/in, not lb.

Rotational degrees of freedom are always expressed in radians.

1.3 Structure of Advanced Nonlinear Solution

- The input data for Advanced Nonlinear Solution follows the standard Nastran format consisting of the following 5 sections:

1. Nastran Statement (optional)
2. File Management Statements (optional)
3. Executive control Statements
4. Case Control Statements
5. Bulk Data Entries

The first two sections do not involve any special treatment in Advanced Nonlinear Solution. The remaining three sections involve some features specific to Advanced Nonlinear Solution, as described below.

1.3.1 Executive Control

- Solution 601 is invoked by selecting solution sequence 601 in the SOL Executive Control Statement. This statement has the following form:

```
SOL 601,N
```

where N determines the specific analysis type selected by Solution 601.

Currently, static and direct time-integration implicit dynamic structural analyses are available as shown in Table 1.3-1. In addition, two analysis types are available for thermal or coupled thermal-mechanical problems.

- Solution 701 is invoked by selecting solution sequence 701 in the SOL Executive Control Statement. This statement has the following form:

```
SOL 701
```

and is used for explicit dynamic analyses.

- In many aspects Solution 601,106 is similar to Solution 106 for nonlinear static analysis. However, it uses the advanced nonlinear features of Solution 601. Likewise, Solution 601,129 is similar to Solution 129 for nonlinear transient response analysis. Solution 701 provides an alternative to the implicit nonlinear dynamic analysis of Solution 601,129.

Table 1.3-1: Solution 601 Analysis Types

N	Solution 601 Analysis Type
106	Static
129	Transient dynamic
153	Steady state thermal + static structural
159	Transient thermal + dynamic structural[1]

[1] N = 159 also allows either of the structure or the thermal parts to be static or steady state.

1.3.2 Case Control

- The Case Control Section supports several commands that control the solution, commands that select the input loads, temperatures and boundary conditions, commands that select the output data and commands that select contact sets. Table 1.3-2 lists the supported Case Control Commands.

Table 1.3-2: Case Control Commands

Case Control Command	**Description**
Solution control	
SUBCASE[1]	Subcase delimiter
TSTEP[2]	Time step set selection
ANALYSIS[3]	Subcase analysis type solution
Loads and boundary conditions	
LOAD	Static load set selection
DLOAD[4]	Dynamic load set selection
SPC	Single-point constraint set selection
MPC	Multipoint constraint set selection

TEMPERATURE[5]
TEMPERATURE(LOAD)
TEMPERATURE(INITIAL)
	Temperature set selection
	Temperature load
	Initial temperature

IC	Transient initial condition set selection
BGSET	Glue contact set selection
BOLTLD	Bolt preload set selection
DTEMP	Time-dependent temperature set selection
DMIG related	
B2GG	Selects direct input damping matrices
K2GG	Selects direct input stiffness matrices
M2GG	Selects direct input mass matrices
Element related	
EBDSET	Element birth/death selection

Table 1.3-2: Case Control Commands (continued)

Output related	
SET	Set definition
DISPLACEMENT	Displacement output request
VELOCITY	Velocity output request
ACCELERATION	Acceleration output request
STRESS	Element stress/strain output request
SPCFORCES	Reaction force output request
GPFORCE	Nodal force output request
GKRESULTS	Gasket results output request
TITLE	Output title
SHELLTHK	Shell thickness output request
THERMAL	Temperature output request
FLUX	Heat transfer output request
OLOAD	Applied load output request
BGRESULTS	Glue result output request
Contact related	
BCSET	Contact set selection
BCRESULTS	Contact results output request

Notes for Table 1.3-2:

1. Only one subcase is allowed in structural analysis Advanced Nonlinear Solution (N = 106, 129). In coupled TMC analyses (N = 153, 159), two subcases are required, one for the structural and one for the thermal submodel.
2. TSTEP is used for all analysis types in Advanced Nonlinear Solution. In explicit analysis with automatic time stepping it is used for determining the frequency of output of results.
3. Supports ANALYSIS = STRUC and ANALYSIS = HEAT for SOL 601,153 and SOL 601,159.
4. DLOAD is used for time-varying loads for both static and transient dynamic analyses.
5. TEMPERATURE, TEMPERATURE(BOTH) and TEMPERATURE(MAT) are not allowed for Advanced Nonlinear Solution. Use TEMPERATURE(INIT) and TEMPERATURE(LOAD) instead.

1.3.3 Bulk Data

- The Bulk Data section contains all the details of the model. Advanced Nonlinear Solution supports most of the commonly used Bulk Data entries. In many cases, restrictions are imposed on some of the parameters in a Bulk Data entry, and in some other cases, different interpretation is applied to some of the parameters to make them more suitable for nonlinear analysis. Several Bulk Data entries are also specific to Advanced Nonlinear Solution.

- Table 1.3-3 lists the supported Bulk Data entries.

Table 1.3-3: Bulk Data entries supported by Advanced Nonlinear Solution

Element Connectivity				
CBAR	CMASS1	CPLSTS3	CQUADX4	CTRAX3
CBEAM	CMASS2	CPLSTS4	CQUADX8	CTRAX6
CBUSH	CONM1	CPLSTS6	CROD	CTETRA
CBUSH1D	CONM2	CPLSTS8	CTRIA3	RBAR
CDAMP1	CONROD	CPYRAM	CTRIA6	RBE2
CDAMP2	CPENTA	CQUAD4	CTRIAR	RBE3
CELAS1	CPLSTN3	CQUAD8		
CELAS2	CPLSTN4	CQUADR		
CGAP	CPLSTN6			
CHEXA	CPLSTN8			
Element Properties				
EBDSET	PBCOMP	PCOMP	PLPLANE	PROD
EBDADD	PBEAM	PCOMPG	PLSOLID	PSHELL
PBAR	PBEAML	PDAMP	PMASS	PSHL3D
PBARL	PBUSH	PELAS	PPLANE	PSOLID
	PBUSH1D	PELAST		
		PGAP		
Material Properties				
CREEP	MAT8	MATHE	MATT1	PCONV
MAT1	MAT9	MATHEV	MATT2	PLCYISO
MAT2	MAT10	MATHEM	MATT3	PLCYKIN
MAT3	MAT11	MATHP	MATT4	PLCYRUP
MAT4	MATCID	MATPLCY	MATT5	RADM
MAT5	MATCRP	MATS1	MATT8	RADMT
	MATG	MATSMA	MATT9	TABLEM1
		MATSR	MATT11	TABLES1
			MATTC	TABLEST
			MATVE	

Table 1.3-3: Bulk Data entries supported by Advanced Nonlinear Solution (continued)

Loads, Boundary Conditions and Constraints				
BCPROPS BEDGE BFLUID BGSET BOLT BOLTFOR BSURFS DLOAD DTEMP	FORCE FORCE1 FORCE2 GRAV LOAD MOMENT MOMENT1 MOMENT2	MPC MPCADD PLOAD PLOAD1 PLOAD2 PLOAD4 PLOADE1	PLOADX1 RFORCE RFORCE1 SPC SPC1 SPCADD SPCD	TABLED1 TABLED2 TEMP TEMPD TIC TLOAD1
Heat Transfer Loads and Boundary Conditions				
BDYOR CHBDYE CHBYDG	CONV QBDY1 QBDY2	QHBDY QVOL RADBC	TEMPBC	
Contact				
BCPROP BCPROPS	BCRPARA BCTADD	BCTPARA BCTSET	BEDGE BLSEG	BSURF BSURFS
Direct Matrix Input				
DMIG				
Other Commands				
CORD1C CORD1R	CORD1S CORD2C	CORD2R CORD2S	GRID GROUP NXSTRAT[1]	PARAM[2] TMCPARA[3] TSTEP[4]

Notes:

1. NXSTRAT is the main entry defining the solution settings for Advanced Nonlinear Solution.
2. Only a few PARAM variables are supported. Most are replaced by NXSTRAT variables.
3. TMCPARA is the main entry defining the solution settings for heat transfer and TMC models.
4. TSTEP is used for both static and dynamic analyses.

1.3.4 Terminology used in Advanced Nonlinear Solution

The terminology used in Advanced Nonlinear Solution is for the most part the same as that used in other Nastran documents.

2. Elements

- Advanced Nonlinear Solution supports most of the commonly used elements in linear Nastran analyses. Some of these elements are modified to be more suitable for nonlinear analysis.

- The Advanced Nonlinear Solution elements are generally classified as line, surface, solid, scalar, R-type or potential-based fluid.

  - Line elements are divided into 2 main categories – rod elements and beam elements. Rod elements only possess axial stiffness, while beam elements also possess bending, shear and torsional stiffness.
  - Surface elements are also divided into 2 main categories – 2-D solids and shell elements.
  - 3-D solid elements are the only solid elements in Advanced Nonlinear Solution.
  - The scalar elements are spring, mass, damper elements and 6-DOF spring elements.
  - R-type elements impose constraints between nodes, such as rigid elements.
  - Potential-based fluid elements are used in static analysis to model the compressibility of fluids.
  - Other element types available in Advanced Nonlinear Solution are the gap element, concentrated mass element, and the bushing element.

- This chapter outlines the theory behind the different element classes, and also provides details on how to use the elements in modeling. This includes the materials that can be used with each element type, their applicability to large displacement and large strain problems, their numerical integration, etc.

- More detailed descriptions of element input and output are provided in several other manuals, including:

  - NX Nastran Reference Manual
  - NX Nastran Quick Reference Guide
  - NX Nastran DMAP Programmer’s Guide

- Table 2-1 below shows the different elements available in Advanced Nonlinear Solution, and how they can be obtained from Nastran element connectivity and property ID entries. Restrictions related to Solution 701 are noted.

Table 2-1: Elements available in Advanced Nonlinear Solution

Element Connectivity Entry	**Property ID Entry**	**Advanced Nonlinear Solution Element**
Rod Elements		
CROD	PROD	2-node rod element
CONROD	None	2-node rod element
Beam Elements		
CBAR	PBAR, PBARL	2-node beam element
CBEAM	PBEAM, PBEAML, PBCOMP	2-node beam element
Shell Elements[3]		
CQUAD4	PSHELL[1], PCOMP[2], PSHL3D[7], PCOMPG	4-node quadrilateral shell element
CQUAD8	PSHELL[1], PCOMP[2], PCOMPG	4-node to 8-node quadrilateral shell element
CQUADR	PSHELL, PCOMP[2], PCOMPG	4-node quadrilateral shell element
CTRIA3	PSHELL[1], PCOMP[2], PSHL3D[7], PCOMPG	3-node triangular shell element
CTRIA6	PSHELL[1], PCOMP[2], PCOMPG	3-node to 6-node triangular shell element
CTRIAR	PSHELL, PCOMP[2], PSHL3D[7], PCOMPG	3-node triangular shell element

Table 2-1: Elements available in Advanced Nonlinear Solution (continued)

2D Solid and Fluid Elements[4]		
CPLSTN3	PPLANE, PLPLANE	3-node triangular 2D plane strain element
CPLSTN4	PPLANE, PLPLANE	4-node quadrilateral 2D plane strain element
CPLSTN6	PPLANE, PLPLANE	6-node triangular 2D plane strain element
CPLSTN8	PPLANE, PLPLANE	8-node quadrilateral 2D plane strain element
CPLSTS3	PPLANE, PLPLANE	3-node triangular 2D plane stress element
CPLSTS4	PPLANE, PLPLANE	4-node quadrilateral 2D plane stress element
CPLSTS6	PPLANE, PLPLANE	6-node triangular 2D plane stress element
CPLSTS8	PPLANE, PLPLANE	8-node quadrilateral 2D plane stress element
CQUAD[6]	PLPLANE	4-node to 9-node quadrilateral 2D plane strain element with hyperelastic material
CQUAD4	PLPLANE, PSHELL[1]	4-node quadrilateral 2D plane strain element
CQUAD8	PLPLANE, PSHELL[1]	4-node to 8-node 2D plane strain element
CTRIA3	PLPLANE, PSHELL[1]	3-node triangular 2D plane strain element
CTRIA6	PLPLANE, PSHELL[1]	3-node to 6-node triangular 2D plane strain element

Table 2-1: Elements available in Advanced Nonlinear Solution (continued)

2D Solid and Fluid Elements (continued)		
CQUADX4	PSOLID, PLSOLID	4-node quadrilateral 2D axisymmetric element
CQUADX8	PSOLID, PLSOLID	8-node quadrilateral 2D axisymmetric element
CTRAX3	PSOLID, PLSOLID	3-node triangular 2D axisymmetric element
CTRAX6	PSOLID, PLSOLID	6-node triangular 2D axisymmetric element
3D Solid and Fluid Elements[5]		
CHEXA	PSOLID, PLSOLID	8-node to 20-node brick 3D solid element
CPENTA	PSOLID, PLSOLID	6-node to 15-node wedge 3D solid element
CTETRA	PSOLID, PLSOLID	4-node to 10-node tetrahedral 3D solid element
CPYRAM	PSOLID, PLSOLID	5-node to 13-node pyramid 3D solid element
Scalar Elements		
CELAS1; **CELAS2**	PELAS; None	Spring element
CDAMP1; **CDAMP2**	PDAMP; None	Damper element
CMASS1; **CMASS2**	PMASS; None	Mass element
R-Type Elements		
RBAR	None	Single rigid element
RBE2	None	Multiple rigid elements
RBE3	None	Interpolation constraint element

Table 2-1: Elements available in Advanced Nonlinear Solution (continued)

Other Elements		
CGAP	PGAP	2-node gap element
CONM1, CONM2	None	Concentrated mass element
CBUSH1D	PBUSH1D	Rod Type Spring-and-Damper Connection
CBUSH	PBUSH	Generalized Spring-and-Damper Connection

Notes:

1. CQUAD4, CQUAD8, CTRIA3, and CTRIA6 with a PSHELL property ID are treated as either 2D plane strain elements or shell elements depending on the MID2 parameter.
2. Elements with PCOMP or PCOMPG property ID entries are treated as multi-layered shell elements. These elements are not supported in Solution 701.
3. Only 3-node and 4-node single layer shells are supported in Solution 701.
4. 2-D solid elements are not supported in Solution 701.
5. Only 4-node tetrahedral, 6-node wedge and 8-node brick 3-D solid elements are supported in Solution 701.
6. CQUAD cannot be used for potential-based fluid elements.
7. PSHL3D is used to specify a 3D-shell element.

- Table 2-2 lists the acceptable combination of elements and materials for Solution 601. Thermal effects in this table imply temperature dependent material properties. Thermal strains are usually accounted for in isothermal material models.

  For potential-based fluid elements, only the fluid material can be used.

Table 2-2: Element and material property combinations in Solution 601

	Rod	**Beam**	**Shell**	**2D Solid**	**3D Solid**
Elastic isotropic	✓	✓	✓	✓	✓
...Thermal	✓		✓	✓	✓
...Creep	✓		✓	✓	✓
Elastic orthotropic			✓	✓	✓
...Thermal			✓	✓	✓
Plastic bilinear/ multilinear	✓	✓	✓	✓	✓
Plastic-cyclic	✓	✓	✓	✓	✓
Plastic thermal	✓		✓	✓	✓
Plastic creep	✓		✓	✓	✓
Hyperelastic			✓[2]	✓	✓
Gasket					✓
Nonlinear elastic isotropic	✓[1]		✓[1]	✓[1]	✓[1]
Shape memory alloy	✓		✓	✓	✓
Viscoelastic	✓		✓	✓	✓

Notes:

1. No thermal strains in these material models.
2. 3D-shell element only.

- Table 2-3 lists the acceptable combination of elements and materials for Solution 701. Thermal effects in this table imply temperature dependent material properties. Thermal strains are usually accounted for in isothermal material models. Note that interpolation of temperature dependent material properties is only performed at the start of the analysis in Solution 701.

  The potential-based fluid elements are not available in Solution 701.

Table 2-3: Element and material property combinations in Solution 701

	Rod	**Beam**	**Shell**	**2D Solid**	**3D Solid**
Elastic isotropic	✓	✓	✓		✓
...Thermal	✓		✓		✓
...Creep					
Elastic orthotropic			✓		✓
...Thermal			✓		✓
Plastic bilinear/ multilinear	✓	✓	✓		✓
Plastic-cyclic	✓	✓	✓		✓
Plastic thermal	✓		✓		✓
Plastic creep					
Hyperelastic			✓[2]		✓
Gasket					✓
Nonlinear elastic isotropic	✓[1]				
Shape memory alloy					
Viscoelastic					

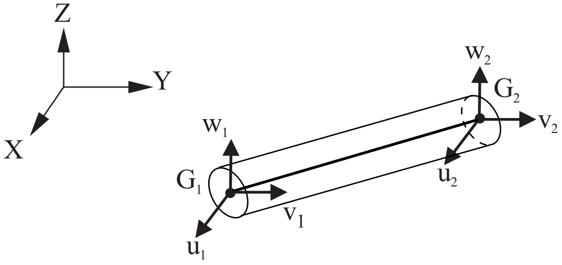
Notes:

1. No thermal strains in these material models.
2. 3D-shell element only.

2.1 Rod elements

2.1.1 General considerations

- Rod elements are generated using the CONROD and CROD entries. These line elements only possess axial stiffness. Fig. 2.1-1 shows the nodes and degrees of freedom of a rod element. Note that the rod element only has 2 nodes.

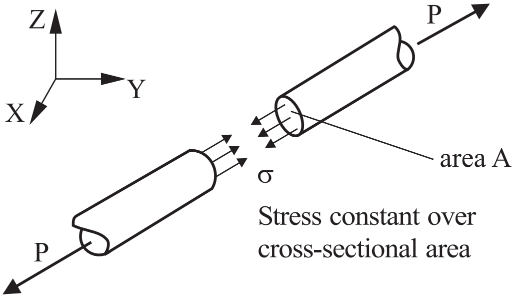


(u, v, w) are nodal translational degrees of freedom

Fig. 2.1-1: Rod element

ref. KJB Sections 5.3.1, 6.3.3

- Note that the only force transmitted by the rod element is the longitudinal force as illustrated in Fig. 2.1-2. This force is constant throughout the element.



Stress constant over cross-sectional area

Fig. 2.1-2: Stresses and forces in rod elements

2.1.2 Material models and formulations

- See Tables 2-2 and 2-3 for a list of the material models that are compatible with rod elements.

- The rod elements can be used with **small displacement/small strain** or **large displacement/small strain** kinematics. In the small displacement case, the displacements and strains are assumed infinitesimally small. In the large displacement case, the displacements and rotations can be very large. In all cases, the cross-sectional area of the element is assumed to remain unchanged, and the strain is equal to the longitudinal displacement divided by the original length.

  All of the compatible material models listed in Tables 2-2 and 2-3 can be used with both the small and large displacement formulations.

2.1.3 Numerical integration

- The rod elements use one point Gauss integration.

2.1.4 Mass matrices

- The consistent mass matrix is calculated using Eq. (4.25) in ref. KJB, p. 165.

- The lumped mass matrix for the rod element is formed by dividing the element's mass *M* among its nodes. The mass assigned to each node is $M \cdot \left(\frac{\ell_i}{L}\right)$, in which L = total element length, ℓ_i = fraction of the total element length associated with element node *i* (i.e., for the 2-node rod element, $\ell_1 = \frac{L}{2}$ and $\ell_2 = \frac{L}{2}$). The element has no rotational mass.

- The same lumped mass matrix is used for both Solution 601 and Solution 701.

2.1.5 Heat transfer capabilities

- The rod element supports 1-D heat conductivity, heat capacity and heat generation features in heat transfer and coupled TMC analyses.

- One temperature degree of freedom is present at each node.

- The heat capacity matrix can be calculated based on a lumped or consistent heat capacity assumption.

- In the lumped heat capacity assumption, each node gets a heat capacity of $c\rho AL/2$.

- This element can also be used as a general thermal link element between any two points in space.

2.2 Beam elements

- The beam element is a 2-node Hermitian beam with a constant cross-section. The element is initially straight.

- The beam element can be employed in the following analysis conditions:

  - Linear analysis, in which case the displacements, rotations and strains are infinitesimally small, and the material is linear elastic.
  - Materially nonlinear analysis, in which case the displacements, rotations and strains are infinitesimally small, but the material is nonlinear.
  - Large displacement/large rotation analysis, in which case the displacements and rotations can be large, but the strains are small. The material can either be linear or nonlinear.

- The beam element can be used in Solution 601 (statics and implicit dynamics) and in Solution 701 (explicit dynamics).

- The beam element can optionally include a bolt feature. The bolt feature is fully described in Section 10.7.

- Throughout this section, the element formulations of the current version of Advanced Nonlinear Solution are described. It is, however, possible to choose the element formulations used in Advanced Nonlinear Solution of NX 8.5, using the BEAMALG parameter of the NXSTRAT entry. (However, if the elasto-plastic material model is chosen, then the current beam element algorithms are always used.) BEAMALG=0 corresponds to the current version (the default) and BEAMALG=1 corresponds to NX 8.5.

  The results obtained will vary depending upon whether the current formulations or the NX 8.5 formulations are used. The major improvements are in the large displacement/large rotation formulation and in the elastic-plastic formulation.

- Beam elements are generated using the CBAR and CBEAM entries. The properties for a CBAR entry are defined using PBAR or PBARL entries while the properties for CBEAM are defined using the PBEAM, PBEAML or PBCOMP entries. See Tables 2-2 and 2-3 for a list of the material models that are compatible with the beam element.

2.2.1 Beam geometry and cross-sections

- Figs 2.2-1 and 2.2-2 show the beam element along with its local coordinate system (*r,s,t*). The *r* direction always lies along the neutral line of the beam (line connecting nodes GA and GB). The orientation of the *s* and *t* directions is defined using the **v** vector defined in the CBAR or CBEAM entries.

  Notice that, for the Hermitian beam element, (*r,s,t*) are not isoparametric coordinates, rather (*r,s,t*) have the same units as the global coordinates.

  Fig 2.2-2 also shows the degrees of freedom at the local nodes. These degrees of freedom are defined in the local coordinate system.

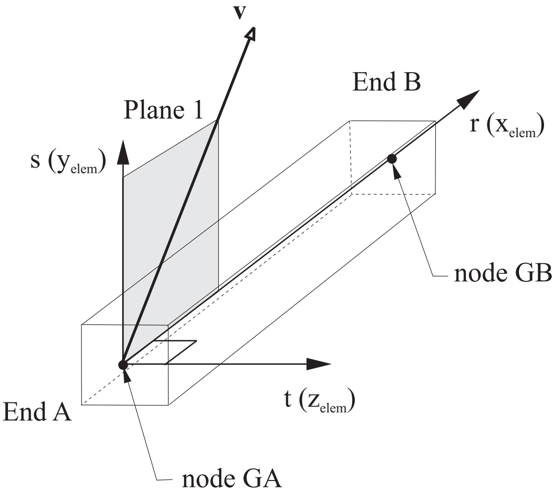


Fig. 2.2-1: Beam element

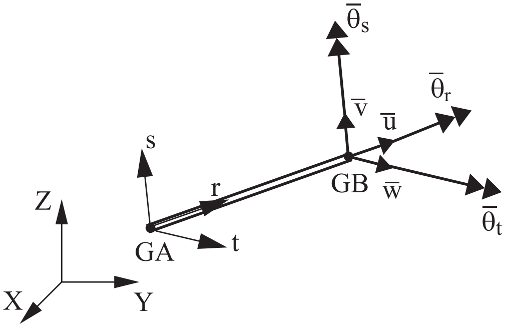
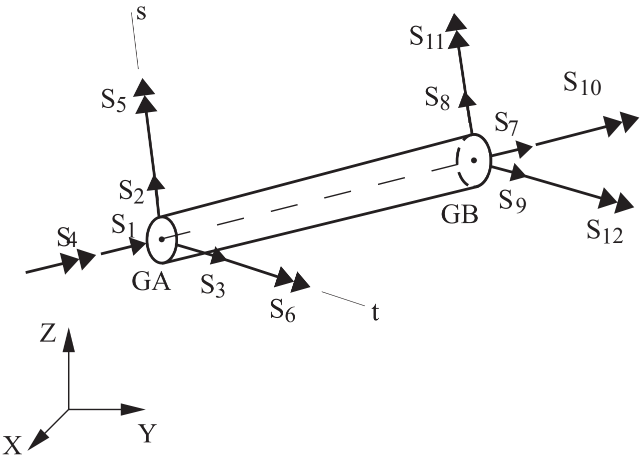


Fig. 2.2-2: Degrees of freedom and local axes for beam element

- The forces / moments in the beam element are shown in Fig. 2.2-3. These forces / moments are also defined in the element local coordinate system.

- The *s* and *t* directions give the orientation of the beam element cross-section. Care must be used in defining the *s* and *t* directions so that the beam element cross-section has the desired orientation.



S_1 = r-direction force at node GA (axial force, positive in compression)
S_2 = s-direction force at node GA (shear force)
S_3 = t-direction force at node GA (shear force)
S_4 = r-direction moment at node GA (torsion)
S_5 = s-direction moment at node GA (bending moment)
S_6 = t-direction moment at node GA (bending moment)

S_7 = r-direction force at node GB (axial force, positive in tension)
S_8 = s-direction force at node GB (shear force)
S_9 = t-direction force at node GB (shear force)
S_{10} = r-direction moment at node GB (torsion)
S_{11} = s-direction moment at node GB (bending moment)
S_{12} = t-direction moment at node GB (bending moment)

Fig. 2.2-3: Element end forces/moments

- The basic beam geometric properties are:

A : Cross-sectional area

L : Beam length

A_s^{sh} , A_t^{sh} , effective shear cross-section areas in the s and t directions. These are calculated as $A_s^{sh} = K1 \times A$, $A_t^{sh} = K2 \times A$.

I_{rr} : Polar moment of inertia: $I_{rr} = \int_A s^2 + t^2 dA$

(do not confuse the polar moment of inertia I_{rr} with the Saint-Venant torsional constant J)

I_{ss} : Inertia for bending about the s-axis:

$$I_{ss} = \int_A t^2 dA$$

I_{tt} : Inertia for bending about the t-axis:

$$I_{tt} = \int_A s^2 dA$$

I_{st} : Product of inertia:

$$I_{st} = \int_A st dA$$

If s-t axes are coincident with the principal axes of the section then $I_{st} = 0$.

J : Saint-Venant torsional constant

Cross-sections

- The following PBARL and PBEAML cross-sections are supported by Advanced Nonlinear Solution: BAR, BOX, CHAN, CHAN1, CHAN2, H, I, I1, ROD, T, T1, T2, TUBE

- Table 2.2-1 shows the beam capabilities available for the available cross sections.

Table 2.2-1: Table of beam capabilities for the available cross-sections

Formulation and material model	General cross-section entered using PBAR, PBEAM or PBCOMP	Cross-sections BOX, CHAN, CHAN1, CHAN2, H, I, I1, T, T1, T2, entered using PBARL or PBEAML	Cross-sections BAR, ROD, TUBE, entered using PBARL or PBEAML
Linear elastic	✓	✓	✓
Large displacement elastic	✓	✓	✓
MNO plastic	---	---	✓
Large displacement plastic	---	---	✓

2.2.2 Beam element formulations

2.2.2.1 Kinematics

- The element is formulated based on the Bernoulli-Euler beam theory, corrected for shear deformation effects if requested.

- The beam displacements in the beam local coordinate system are $\overline{u}$ (axial r-direction displacement), $\overline{\mathrm{v}}$ (transverse s-direction displacement) and $\overline{w}$ (transverse t-direction displacement).

- It is assumed that the cross-section rotates rigidly. Therefore the displacements at an arbitrary point on the beam cross-section can be written in terms of the displacements and rotations of the beam neutral axis:

$$
\begin{aligned}
\overline{u} &= \overline{u}_n - \overline{\theta}_t s + \overline{\theta}_s t \\
\overline{v} &= \overline{v}_n - \overline{\theta}_r t \\
\overline{w} &= \overline{w}_n + \overline{\theta}_r s
\end{aligned}
$$

In addition, the transverse rotations of the beam neutral axis are assumed to be equal to the slopes of the transverse displacements:

$$\bar{\theta}_s = -\frac{d\bar{w}_n}{dr}$$

$$\bar{\theta}_t = \frac{d\bar{v}_n}{dr}$$

The neutral axis displacements and rotations are interpolated from the nodal displacements and rotations using

$$\begin{aligned}
\bar{u}_n &= L_1\bar{u}^1 + L_2\bar{u}^2 \\
\bar{v}_n &= H_1\bar{v}^1 + H_2\bar{\theta}_t^1 + H_3\bar{v}^2 + H_4\bar{\theta}_t^2 \\
\bar{w}_n &= H_1\bar{w}^1 - H_2\bar{\theta}_s^1 + H_3\bar{w}^2 - H_4\bar{\theta}_s^2 \\
\bar{\theta}_r &= L_1\bar{\theta}_r^1 + L_2\bar{\theta}_r^2 \\
\bar{\theta}_s &= -\frac{dH_1}{dr}\bar{w}^1 + \frac{dH_2}{dr}\bar{\theta}_s^1 - \frac{dH_3}{dr}\bar{w}^2 + \frac{dH_4}{dr}\bar{\theta}_s^2 \\
\bar{\theta}_t &= \frac{dH_1}{dr}\bar{v}^1 + \frac{dH_2}{dr}\bar{\theta}_t^1 + \frac{dH_3}{dr}\bar{v}^2 + \frac{dH_4}{dr}\bar{\theta}_t^2
\end{aligned}$$

where the nodal displacements and rotations are $\left(\bar{u}^1, \bar{v}^1, \bar{w}^1, \bar{\theta}_r^1, \bar{\theta}_s^1, \bar{\theta}_t^1\right)$ for node GA and $\left(\bar{u}^2, \bar{v}^2, \bar{w}^2, \bar{\theta}_r^2, \bar{\theta}_s^2, \bar{\theta}_t^2\right)$ for node GB, and in which

$$L_1 = 1 - \frac{r}{L},\; L_2 = \frac{r}{L}$$

$$H_1 = 1 - 3\frac{r^2}{L^2} + 2\frac{r^3}{L^3},\; H_2 = r - 2\frac{r^2}{L} + \frac{r^3}{L^2},$$

$$H_3 = 3\frac{r^2}{L^2} - 2\frac{r^3}{L^3},\; H_4 = -\frac{r^2}{L} + \frac{r^3}{L^2}$$

are the linear and cubic interpolation functions (Hermitian displacement functions). It is seen that the transverse neutral axis displacements $\bar{v}_{\mathrm{n}}$ and $\bar{w}_n$ are cubic and that the axial neutral axis displacement $\overline{u}_n$ and torsional rotation $\overline{\theta_r}$ are linear.

- In all cases, the centroid of the cross-section is located at the origin of the cross-section axes (*s*,*t*), and, in addition, the principal axes of the cross-section are aligned with the cross-section axes (*s*,*t*). Thus $I_{st} = 0$.

- Note that axial forces applied at nodes GA or GB are assumed to be acting along the beam's centroid and hence cause no bending. Also shear forces applied at nodes GA or GB are assumed to be acting through the beam's shear center and hence cause no twisting.

- The kinematics given above do not account for the torsional warping of non-circular cross-sections, or for shear deformations. For the linear elastic material model, the torsional warping is accounted for by choice of the torsional constant *J*, and shear deformations are accounted for by choice of the shear areas. For the elasto-plastic material model with BAR cross-section, the torsional warping is approximated as described in Section 2.2.2.7.

- The stiffness matrix, mass matrix and force vector are formulated in the local degrees of freedom (in the r, s, t axes). These matrices and vectors are then transformed to the nodal degrees of freedom (either global or skew) and assembled into the global system matrices.

2.2.2.2 Linear formulation

- It is assumed that the displacements, rotations, and strains are infinitesimally small, and the elastic-isotropic material is used.

2.2.2.3 Materially-nonlinear-only formulation

- It is assumed that the displacements, rotations, and strains are infinitesimally small. Either the elastic-isotropic material is used (for example, in conjunction with element birth-death) or the plastic material model is used.

2.2.2.4 Large displacement formulation

- It is assumed that large displacements/rotations can occur, but only small strains. Any beam material model, linear or nonlinear, can be used. The input of beam cross-sections and material data is exactly the same as when using the linear or materially-nonlinear-only beam elements.

- If a stressed large displacement beam element is subjected to a rigid body rotation, the stresses/forces (expressed in the local coordinate system) do not change during the rigid body rotation.

- The geometry of the large displacement beam element is shown in Fig 2.2-4. The shape of the beam neutral axis is completely specified by the positions of the end-nodes, and by the orientation of the end-node triads. Each end-node triad consists of three orthogonal unit vectors $\mathbf{V}_r, \mathbf{V}_s, \mathbf{V}_t$. Initially the triads are identical to the element coordinate axes unit vectors, e.g, $\mathbf{V}_r$ is a unit vector in the *r* direction, etc.

  During the deformations, the positions of the end-nodes are updated, as usual, by the nodal displacements. The orientations of the end-node triads are updated incrementally by the increments in nodal rotations. The end-node triads can rotate independently of each other, however it is assumed that the relative rotation of the end-node triads remains small.

  The shape of the beam neutral axis becomes a curved space curve. The bending and torsion in the beam is obtained from the relative orientations of the end-node triads.

- In the large displacement formulation, the coordinate system in which the beam local displacements and rotations are measured is updated during the solution. In the current version of Advanced Nonlinear Solution, this coordinate system is taken from the orientations of the end-node triads. In Advanced Nonlinear Solution of NXN 8.5, this coordinate system is taken from the coordinates of the end-nodes. Hence the solution output in the current version of Advanced Nonlinear Solution will in general be different than the solution output in Advanced Nonlinear Solution of NXN 8.5.

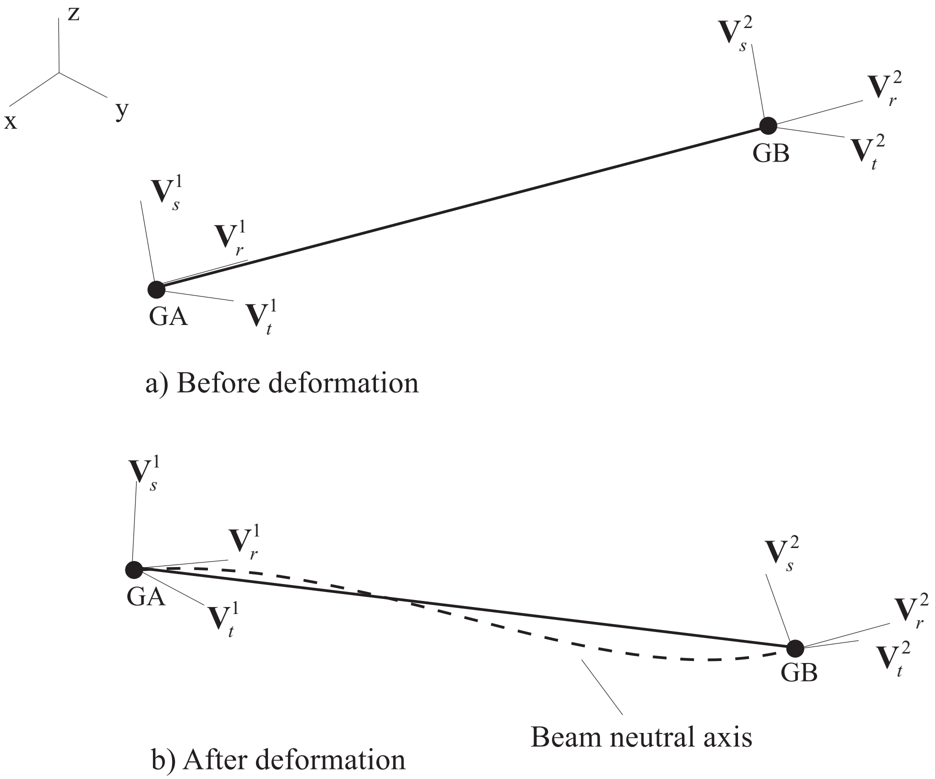
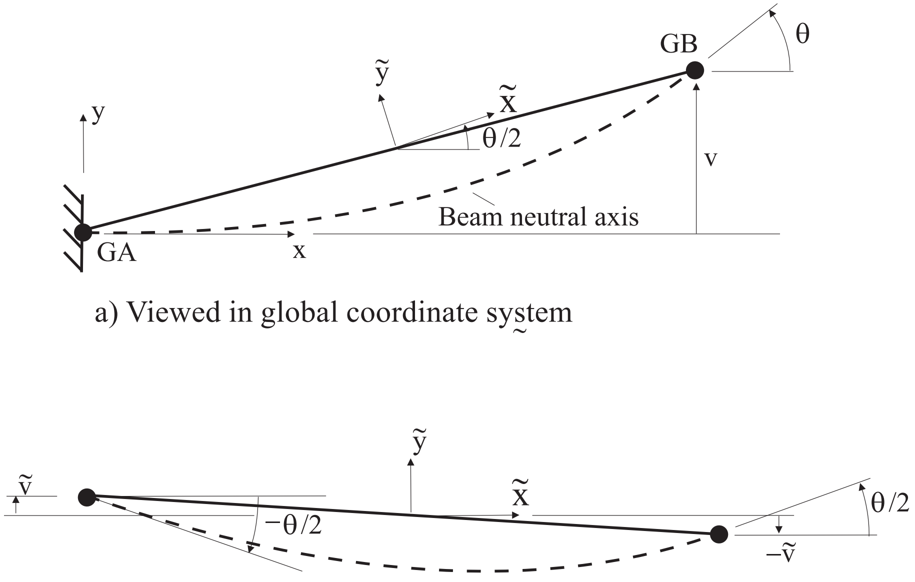


Fig. 2.2-4: Geometry of large displacement beam element

Fig 2.2-5 shows an example for in-plane bending. A single beam element is clamped at node GA and subjected to a prescribed rotation θ at node GB. The coordinate system $\tilde{x}, \tilde{y}$ used to measure beam local displacements and rotations is located halfway between the nodes, and is rotated an angle $\theta / 2$ with respect to the global coordinate system x, y. Notice that the $\tilde{x}$ direction does not coincide with the line between the end-nodes.

The rotation at local node GB with respect to the local coordinate system is $\theta / 2$, and the rotation at local node GA with respect to the local coordinate system is $-\theta / 2$. Also, the transverse displacement at local node GB with respect to the local coordinate system is $-\tilde{v}$ and the transverse displacement at local node GA with respect to the local coordinate system is $\tilde{v}$.



a) Viewed in global coordinate system

b) Viewed in local coordinate system

Fig. 2.2-5: Coordinate system update for large displacement beam element, in-plane bending example

The advantage of using a coordinate system defined from the orientations of the end-node triads is that this coordinate system is uniquely defined even for general 3-D deformations including torsion. A coordinate system defined from the end-nodes coordinates is not uniquely defined for 3-D deformations including torsion.

2.2.2.5 Mass matrices

- The beam element can be used with a lumped or a consistent mass matrix, except for explicit dynamic analysis (Solution 701) which always uses a lumped mass matrix.

- The consistent mass matrix of the beam element is evaluated in closed form, and does not include the effect of shear deformations. The matrix is defined in the local coordinate system using

$$\mathbf{M}=\rho AL\left[\begin{array}{cccccccccccc}
\frac{1}{3} & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 \\
 & \frac{13}{35}+\frac{6I_{tt}}{5AL^2} & 0 & 0 & 0 & \frac{11L}{210}+\frac{I_{tt}}{10AL} & 0 & \frac{9}{70}-\frac{6I_{tt}}{5AL^2} & 0 & 0 & 0 & -\frac{13L}{420}+\frac{I_{tt}}{10AL} \\
 & & \frac{13}{35}+\frac{6I_{ss}}{5AL^2} & 0 & -\frac{11L}{210}-\frac{I_{ss}}{10AL} & 0 & 0 & 0 & \frac{9}{70}-\frac{6I_{ss}}{5AL^2} & 0 & \frac{13L}{420}-\frac{I_{ss}}{10AL} & 0 \\
 & & & \frac{I_{rr}}{3A} & 0 & 0 & 0 & 0 & 0 & \frac{I_{rr}}{6A} & 0 & 0 \\
 & & & & \frac{L^2}{105}+\frac{2I_{ss}}{15A} & 0 & 0 & 0 & -\frac{13L}{420}+\frac{I_{ss}}{10AL} & 0 & -\frac{L^2}{140}-\frac{I_{ss}}{30A} & 0 \\
 & & & & & \frac{L^2}{105}+\frac{2I_{tt}}{15A} & 0 & \frac{13L}{420}-\frac{I_{tt}}{10AL} & 0 & 0 & 0 & -\frac{L^2}{140}-\frac{I_{tt}}{30A} \\
 & & & & & & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & \frac{13}{35}+\frac{6I_{tt}}{5AL^2} & 0 & 0 & 0 & -\frac{11L}{210}-\frac{I_{tt}}{10AL} \\
 & & & & & & & & \frac{13}{35}+\frac{6I_{ss}}{5AL^2} & 0 & \frac{11L}{210}+\frac{I_{ss}}{10AL} & 0 \\
 & & & & & & & & & \frac{I_{rr}}{3A} & 0 & 0 \\
 & & & & & & & & & & \frac{L^2}{105}+\frac{2I_{ss}}{15A} & 0 \\
 & & \textit{symmetric} & & & & & & & & & \frac{L^2}{105}+\frac{2I_{tt}}{15A}
\end{array}\right]$$

This matrix is derived in the following reference.

ref. J.S. Przemieniecki, *Theory of Matrix Structural Analysis,* McGraw-Hill Book Co., 1968.

- The lumped mass for translational degrees of freedom is *M / 2* where *M* is the total mass of the element.

- The rotational lumped mass for static analysis is 0.

The rotational lumped mass for implicit dynamic analysis (Solution 601) is $M_{rr}=\frac{2}{3}\cdot\frac{M}{2}\cdot\frac{I_{rr}}{A}$. This lumped mass is applied to all rotational degrees of freedom in order to obtain a diagonal mass matrix in any coordinate system.

The rotational lumped mass for explicit dynamic analysis (Solution 701) is $M_{rr} = 3 \cdot \frac{M}{2} \cdot \frac{I_m}{A}$ where $I_m = \max\left(I_{ss}, I_{tt}\right)$ is the maximum bending moment of inertia of the beam. This lumped mass is applied to all rotational degrees of freedom. Note that this scaling of rotational masses ensures that the rotational degrees of freedom do not affect the critical stable time step of the element.

2.2.2.6 Elastic beam element

- The beam element stiffness matrix is evaluated in closed form. The stiffness matrix used is discussed in detail in the following reference:

ref. J.S. Przemieniecki, *Theory of Matrix Structural Analysis,* McGraw-Hill Book Co., 1968.

The stiffness matrix, in the local coordinate system, is

$$\mathbf{K} = \begin{bmatrix}
\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & -k_{11} & 0 & 0 & 0 & 0 & 0 \\
 & \frac{12EI_{tt}}{L^3(1+\phi_2)} & 0 & 0 & 0 & \frac{6EI_{tt}}{L^2(1+\phi_2)} & 0 & -k_{22} & 0 & 0 & 0 & k_{26} \\
 & & \frac{12EI_{ss}}{L^3(1+\phi_3)} & 0 & \frac{-6EI_{ss}}{L^2(1+\phi_3)} & 0 & 0 & 0 & -k_{33} & 0 & k_{35} & 0 \\
 & & & \frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & -k_{44} & 0 & 0 \\
 & & & & \frac{EI_{ss}(4+\phi_3)}{L(1+\phi_3)} & 0 & 0 & 0 & -k_{35} & 0 & \frac{EI_{ss}(2-\phi_3)}{L(1+\phi_3)} & 0 \\
 & & & & & \frac{EI_{tt}(4+\phi_2)}{L(1+\phi_2)} & 0 & -k_{26} & 0 & 0 & 0 & \frac{EI_{tt}(2-\phi_2)}{L(1+\phi_2)} \\
 & & & & & & k_{11} & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & k_{22} & 0 & 0 & 0 & -k_{26} \\
 & & & & & & & & k_{33} & 0 & -k_{35} & 0 \\
 & & & & & & & & & k_{44} & 0 & 0 \\
 & & & & & & & & & & k_{55} & 0 \\
 & & \textit{symmetric} & & & & & & & & & k_{66}
\end{bmatrix}$$

in which E = Young's modulus, ν = Poisson's ratio, G = shear modulus $= \frac{E}{2(1+\nu)}$. Also

$$\begin{aligned}\phi_2 &= \frac{24(1+\nu)}{L^2 A_s^{sh}} I_{tt}, \quad A_s^{sh} > 0 \\ &= 0, \qquad\qquad\quad A_s^{sh} = 0\end{aligned}$$

$$\begin{aligned}\phi_3 &= \frac{24(1+\nu)}{L^2 A_t^{sh}} I_{ss}, \quad A_t^{sh} > 0 \\ &= 0, \qquad\qquad\quad A_t^{sh} = 0\end{aligned}$$

are factors that are used to approximate the effects of shear deformations.

- Note that torsional warping effects are only taken into account by the selection of the torsional constant *J*.

- The coefficient of thermal expansion can be specified as a material property. The coefficient of thermal expansion is constant (independent of the temperature). In addition, the beam temperature is taken as the average of the temperatures of the beam end-nodes. Temperature gradients at beam nodes are ignored.

- Stress and strain output is not supported in elastic beam elements.

.

2.2.2.7 Elastic-plastic beam element

- The element is used with the plastic-cyclic material model described in Section 3.4.2. The element can also be used with a plastic-bilinear material with isotropic hardening, but then the program automatically converts the material input into the equivalent plastic-cyclic material input.

  The stress-strain law used incorporates the assumptions that the stresses τ_{ss} , τ_{tt} , τ_{st} are zero.

- The beam element, using an elastic-plastic material model, can only be employed for the BAR, ROD and TUBE cross-sections.

- The material model can be used either with the materially-nonlinear-only formulation or with the large displacement formulation (in which case the displacements/rotations can be large). In all cases, the strains are assumed to be small.

- Shear deformation effects are not included.

- For beams with a BAR cross-section, warping effects are included by modifying the longitudinal displacement as follows:

$$\bar{u} = ... + st\bar{\alpha}_1 + \left(s^3 t - st^3\right)\bar{\alpha}_2$$

where $\bar{\alpha}_1, \bar{\alpha}_2$ are additional degrees of freedom. The additional degrees of freedom are condensed out at the element level.

This procedure is described in the following reference:

ref. K.J. Bathe and A. Chaudhary, "On the Displacement Formulation of Torsion of Shafts with Rectangular Cross-Sections", *Int. Num. Meth. in Eng.,* Vol. 18, pp. 1565-1568, 1982.

ref. KJB Section 6.6.3

- All element matrices in elasto-plastic analysis are calculated using numerical integration. The default integration orders are given in Tables 2.2-2 and 2.2-3. The locations and the labeling of the integration points are given in Fig. 2.2-6 and 2.2-7. For the ROD and TUBE sections, polar coordinates are used:

$$R = \sqrt{s^2 + t^2} \ , \ \psi = \tan^{-1}\frac{t}{s} \, .$$

Table 2.2-2: Integration orders in elasto-plastic beam analysis, BAR section

Coordinate	Integration order[1]
r	5
s,t	7

1) Newton-Cotes integration is used in all coordinate directions

Table 2.2-3: Integration orders in elasto-plastic beam analysis, ROD and TUBE sections

Coordinate	Integration scheme	Integration order
r	Newton-Cotes	5
radius R	Newton-Cotes	3
tangential angle ψ	Composite trapezoidal rule	8

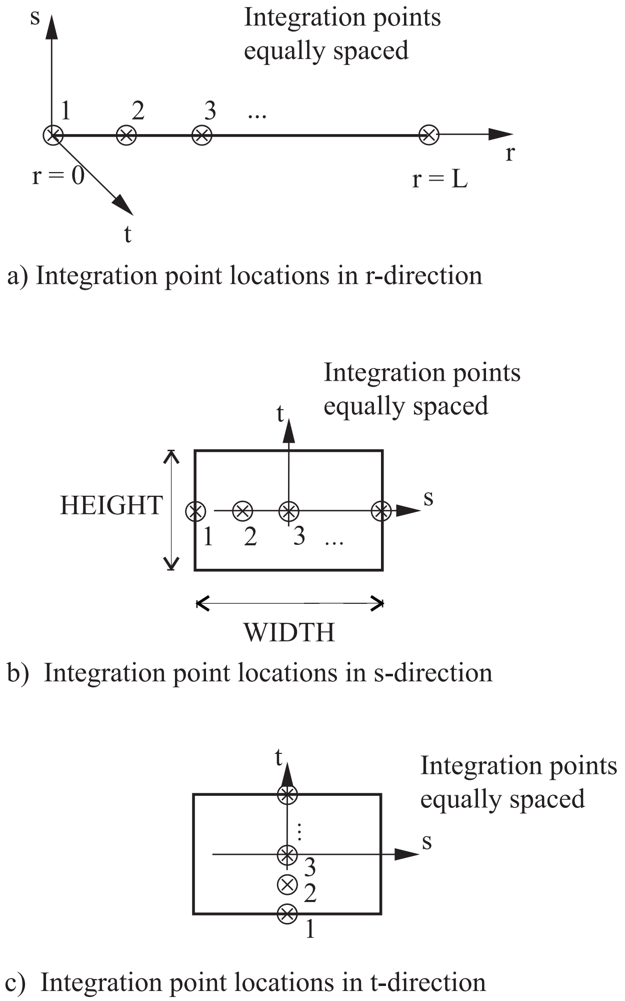
- Note that the stiffness matrix is identical to the one used in linear elastic analysis if

  ▸ The cross-section of the beam is circular (or rectangular) with $b = a$ or $b \gg a$, (because the exact warping functions are employed for $b = a$ and $b \gg a$, and the appropriate torsional rigidity is used).

  ▸ The numerical integration is of high enough order

  ▸ The material is (still) elastic

- Stress and strain output is not supported in elastic-plastic beam elements.

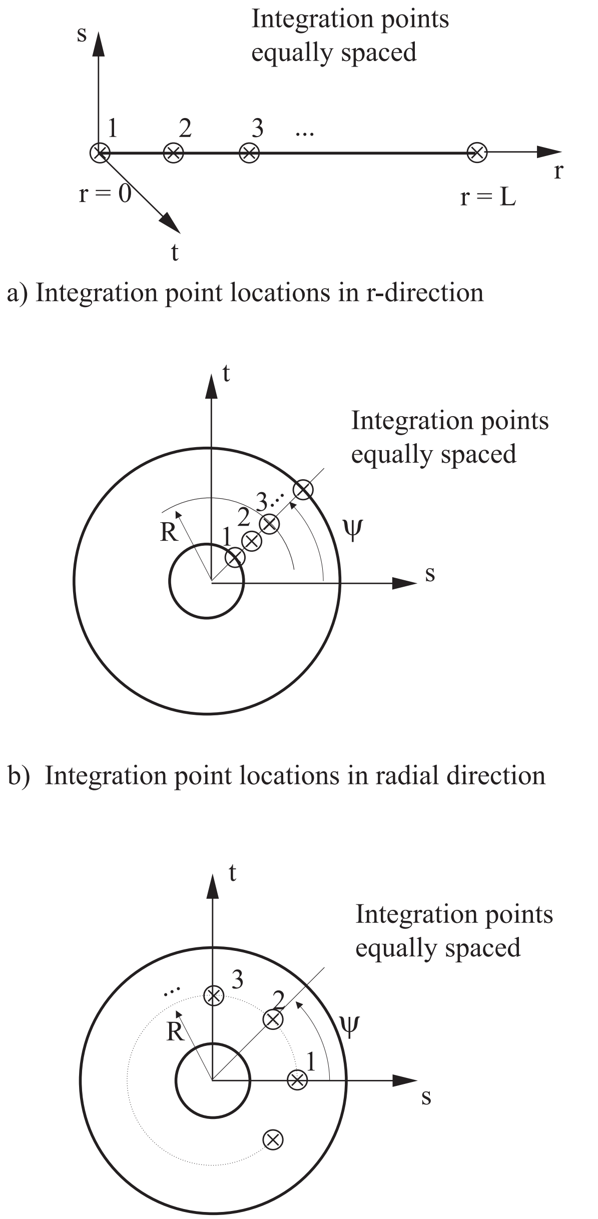


a) Integration point locations in r-direction

b) Integration point locations in s-direction

c) Integration point locations in t-direction

Fig. 2.2-6: Integration point locations in elasto-plastic beam analysis, BAR cross-section



a) Integration point locations in r-direction

b) Integration point locations in radial direction

c) Integration point locations in tangential direction

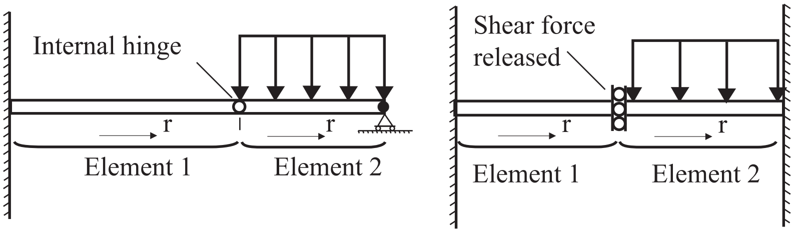
Fig. 2.2-7: Integration point locations in elasto-plastic beam analysis, ROD and TUBE sections

2.2.3 Heat transfer capabilities

- The beam element has the same heat transfer capabilities of the rod element. See Section 2.1.5 for details.

2.2.4 Pin flag option

To model beam internal hinges, a moment and force release option can be used (see Fig. 2.2-8). Twisting moments and axial forces can also be released.



Specify 456 in the PB field in CBAR for element 1, or specify 456 in the PA field in CBAR for element 2

Specify 23 in the PB field in CBAR for element 1, or specify 23 in the PA field in CBAR for element 2

(a) Moment to be released at internal hinge

(b) Shear force to be released

Fig. 2.2-8: Use of pin flag options

- A brief description of the theory of pin flags follows:

For static linear elastic analysis without warping degrees of freedom, the stiffness matrix and internal force vector for a beam element can be written as

$$\mathbf{K}\mathbf{u}=\mathbf{F}$$

In this expression, $\mathbf{u}$ contains the displacements and rotations of the beam nodes, in the beam local coordinate system:

$$\mathbf{u}^T=\begin{bmatrix} u_r^1 & u_s^1 & u_t^1 & \theta_r^1 & \theta_s^1 & \theta_t^1 & u_r^2 & u_s^2 & u_t^2 & \theta_r^2 & \theta_s^2 & \theta_t^2 \end{bmatrix},$$

$\mathbf{F}$ contains the forces and moments of the beam nodes,

$$\mathbf{F}^T = \begin{bmatrix} S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 & S_9 & S_{10} & S_{11} & S_{12} \end{bmatrix}$$

and $\mathbf{K}$ is the stiffness matrix. These expressions use the same notations as Figs. 2.2-1 and 2.2-2.

Now suppose that one or more of the local displacements or rotations is to be released. “Releasing” means that the corresponding force is set equal to zero. For example, if the axial displacement at local node GB is to be released, then force S_7 is set to zero. The corresponding pin flag is PB=1 in the CBAR bulk data entry.

In order to release the selected local displacements or rotations, the following procedure is used:

1) $\mathbf{u}$ is rearranged and partitioned: $\mathbf{u}^T = \left[\mathbf{u}_A^T \mid \mathbf{u}_B^T \right]$, so that $\mathbf{u}_A$ contains all of the local displacements / rotations to be retained, and $\mathbf{u}_B$ contains all of the local displacements / rotations to be released. For example, if the axial displacement at local node 2 is to be released, then

$$\mathbf{u}_A^T = \begin{bmatrix} u_r^1 & u_s^1 & u_t^1 & \theta_r^1 & \theta_s^1 & \theta_t^1 & u_s^2 & u_t^2 & \theta_r^2 & \theta_s^2 & \theta_t^2 \end{bmatrix}$$
$$\mathbf{u}_B^T = \begin{bmatrix} u_r^2 \end{bmatrix}$$

2) $\mathbf{F}$ and $\mathbf{K}$ are similarly rearranged and partitioned. The resulting system of equations is

$$\left[\begin{array}{c|c} \mathbf{K}_{AA} & \mathbf{K}_{AB} \\ \hline \mathbf{K}_{AB}^T & \mathbf{K}_{BB} \end{array}\right] \left[\begin{array}{c} \mathbf{u}_A \\ \hline \mathbf{u}_B \end{array}\right] = \left[\begin{array}{c} \mathbf{F}_A \\ \hline \mathbf{F}_B \end{array}\right]$$

3) The pin flag condition is now expressed as $\mathbf{F}_B = \mathbf{0}$, and the resulting system of equations becomes

$$\left[\begin{array}{c|c} \mathbf{K}_{AA} & \mathbf{K}_{AB} \\ \hline \mathbf{K}_{AB}^T & \mathbf{K}_{BB} \end{array}\right] \left[\begin{array}{c} \mathbf{u}_A \\ \hline \mathbf{u}_B \end{array}\right] = \left[\begin{array}{c} \mathbf{F}_A \\ \hline \mathbf{0} \end{array}\right]$$

4) This system of equations is satisfied by choosing

$$\left(\mathbf{K}_{AA} - \mathbf{K}_{AB}\mathbf{K}_{BB}^{-1}\mathbf{K}_{AB}^{T}\right)\mathbf{u}_A = \mathbf{F}_A$$

$$\mathbf{u}_B = -\mathbf{K}_{BB}^{-1}\mathbf{K}_{AB}^{T}\mathbf{u}_A$$

Therefore the stiffness matrix that is assembled into the global system of equations is $\mathbf{K}_{AA} - \mathbf{K}_{AB}\mathbf{K}_{BB}^{-1}\mathbf{K}_{AB}^{T}$.

- Any external forces acting onto the released degrees of freedom are ignored.

- The pin flag procedure is also implemented for nonlinear analysis, both for materially nonlinear elements and also for geometrically nonlinear elements.

- The pin flag procedure is available for static, implicit dynamic and explicit dynamic analysis.

- Pin flags only affect the stiffness matrix and force vector, not the mass matrix. Therefore inertial forces and moments (forces and moments due to mass matrix effects) are not released in the above procedure.

- If pin flags are used in dynamic analysis, the elements in which pin flags are specified should be very short, in order to minimize the inertial forces and moments.

- Pin flags are applied to the element local nodes (not to the global nodes). Therefore, to model the hinge shown in Fig. 2.2-8(a), pin flags that release the moments can be applied to local node GB of element 1, or to local node GA of element 2 (but not to both local nodes).

- When the beam elements are geometrically nonlinear, the global directions corresponding to the released degrees of freedom change as the model deforms. By default, this effect is fully included in the pin flag calculations. However, this effect can occasionally slow down convergence.

  The pin flags are evaluated in a local coordinate system corresponding to the configuration of the beam in the previous equilibrium iteration. Therefore, at convergence, the pin flags are evaluated in (almost) the current configuration of the beam. (At convergence, the difference between the current equilibrium iteration and the previous equilibrium iteration is very small.)

In implicit static and implicit dynamic analysis, when the elements are geometrically nonlinear, the program performs at least two equilibrium iterations in each time step.

- In explicit dynamic analysis, the use of pin flags will cause the program to run considerably more slowly than if this option were not used.

2.2.5 Beam element modeling hints

- For modeling tapered beams or curved beams, the user needs to divide such members into several elements. For the tapered beams the user needs to divide the beam into several elements and use an appropriate constant cross-section for each of these elements.

- In order to model the bending due to an off-centroidal axial force or a shear force applied away from the shear center, the resulting moments can be applied directly or the forces can be applied at an offset location using rigid elements.

- Off-centered beam elements can be modeled using rigid elements (see Fig. 2.2-9 and Section 2.7-1).

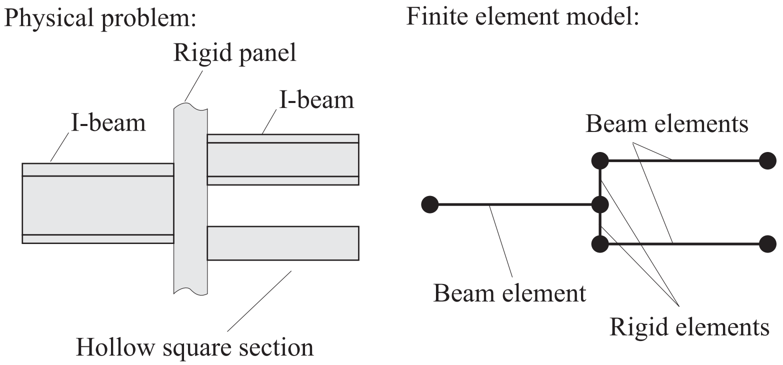


Fig. 2.2-9: Use of rigid elements for modeling off-centered beams

2.3 Shell elements

- Shell elements in Advanced Nonlinear Solution are generated when a PSHELL, PSHL3D, PCOMP or PCOMPG property ID entry is referenced by one of the following Nastran shell entries: CQUAD4, CTRIA3, CQUAD8, CTRIA6, CQUADR, or CTRIAR. The elements are shown in Fig. 2.3-1.

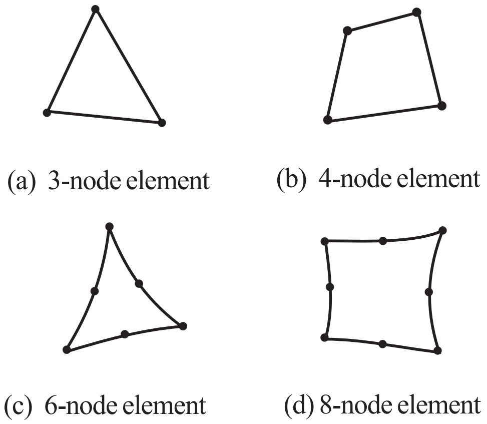


Fig. 2.3-1: Shell elements in Advanced Nonlinear Solution

- The PSHELL entry results in a single-layered shell, while PCOMP and PCOMPG produces a composite shell.

- The PSHL3D entry results in a single-layered 3D-shell. 3D-shell elements are described in Section 2.3.9.

- Shell elements are classified based on the number of nodes in the element. Table 2.3-1 shows the correspondence between the different shell elements and the NX element connectivity entries.

- Solution 701 only supports 3-node and 4-node single-layered shell elements.

Table 2.3-1: Correspondence between shell elements and NX element connectivity

Shell element	NX element connectivity entry
3-node	CTRIA3, CTRIAR
4-node	CQUAD4, CQUADR
6-node[1]	CTRIA6
8-node[1]	CQUAD8
9-node[1]	CQUAD8[2]

Notes:
1. Only for Solution 601
2. With ELCV = 1 in NXSTRAT entry

- The extra middle node in the 9-node shell element is automatically added by the program when ELCV is set to 1 in the NXSTRAT entry. This extra node improves the performance of the shell element. The boundary conditions at the added node are predicted from the neighboring nodes.

- Incompatible modes (bubble functions) can be used with 4-node shell elements. Additional displacement degrees of freedom are introduced which are not associated with nodes; therefore the condition of displacement compatibility between adjacent elements is not satisfied in general. The addition of the incompatible modes (bubble functions) increases the flexibility of the element, especially in bending situations. For theoretical considerations, see reference KJB, Section 4.4.1. Note that these incompatible-mode elements are formulated to pass the patch test. Also note that element distortions deteriorate the element performance when incompatible modes are used.

The incompatible modes feature can only be used with 4-node single layer shell elements. The feature is available in linear and nonlinear analysis.

The incompatible modes feature is set through ICMODE in the NXSTRAT entry.

- Table 2.3-2 lists the features and capabilities available for the shell element types mentioned above.

Table 2.3-2: Features available for shell elements

Shell element	Large displacement/ small strain	Large strain ULJ formulation	Large strain ULH formulation	Bubble functions	3D-shell	Soln 701
3-node	✓	✓	✓		✓	✓
4-node	✓	✓	✓	✓	✓	✓
6-node	✓	✓	✓			
8-node	✓					
9-node	✓	✓	✓			

2.3.1 Basic assumptions in element formulation

- The basic equations used in the formulation of the shell elements in Advanced Nonlinear Solution are given in ref. KJB. These elements are based on the Mixed Interpolation of Tensorial Components (MITC). Tying points are used to interpolate the transverse shear strain and the membrane strains if necessary. These elements show excellent performance.

- The shell element formulation treats the shell as a three-dimensional continuum with the following two assumptions used in the Timoshenko beam theory and the Reissner-Mindlin plate theory:

Assumption 1: Material particles that originally lie on a straight line "normal" to the midsurface of the structure remain on that straight line during deformation.

Assumption 2: The stress in the direction normal to the midsurface of the structure is zero.

For the Timoshenko beam theory, the structure is the beam, and for the Reissner/Mindlin plate theory, the structure is the plate under consideration. In shell analysis, these assumptions correspond to a very general shell theory. See the reference below for more details:

ref. D. Chapelle and K.J. Bathe, *The Finite Element Analysis of Shells — Fundamentals*, 2nd ed, Springer, 2011.

- In the calculations of the shell element matrices the following geometric quantities are used:

  ▸ The coordinates of the node *k* that lies on the shell element midsurface at ${}^t x_i^k , i = 1, 2, 3$ (see Fig. 2.3-2); (the left superscript denotes the configuration at time *t*)

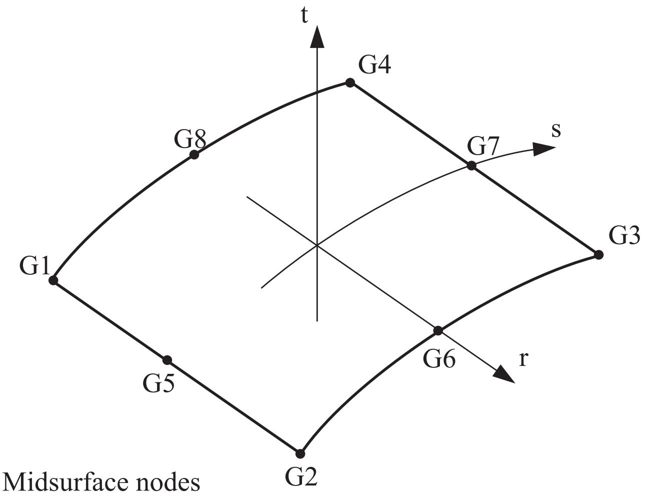


Fig. 2.3-2: Some conventions for the shell element; local node numbering; local element coordinate system

  ▸ The director vectors ${}^t\mathbf{V}_n^k$ pointing in the direction "normal" to the shell midsurface

ref. KJB Fig. 5.33 page 437

  ▸ The shell thickness, a_k, at the nodal points measured in the direction of the director vectors ${}^t\mathbf{V}_n^k$ (see Fig. 2.3-3).

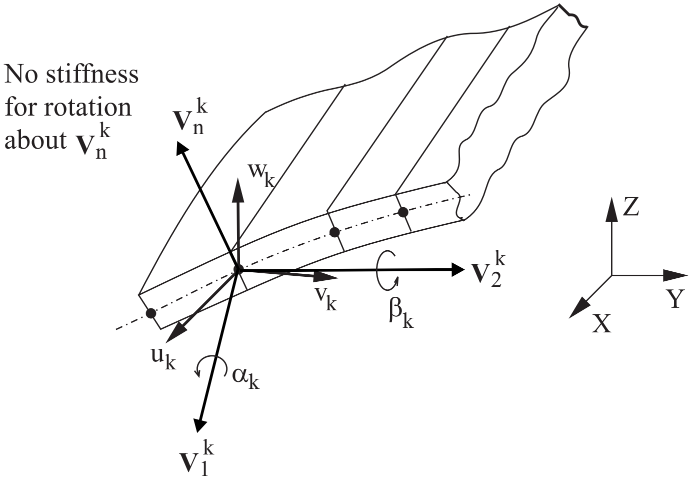


Fig. 2.3-3: Shell degrees of freedom at node k

- Based on these quantities the geometry of the shell is interpolated as follows:

$$ {}^t x_i = \sum_{k=1}^{q} h_k \, {}^t x_i^k + \frac{t}{2} \sum_{k=1}^{q} a_k h_k \, {}^t V_{ni}^k \quad (i = 1,2,3) $$

where q is the number of element nodes, ${}^t V_{ni}^k \; (i = 1,2,3)$ are the components of the shell director vector ${}^t \mathbf{V}_n^k$ and $h_k(r,s)$ are the 2-D interpolation functions.

- At the element level the shell has 5 independent degrees of freedom per node: 3 displacements about the displacement coordinate system resulting from the displacement of the shell midsurface and 2 rotations resulting from the motion of the shell direction vector $\mathbf{V}_n^k$:

$$ {}^t u_i = \sum_{k=1}^{q} h_k \, {}^t u_i^k + \frac{t}{2} \sum_{k=1}^{q} a_k h_k \left({}^t V_{ni}^k - {}^0 V_{ni}^k \right) $$

The motion of the director vector at node *k* is described using 2 rotational degrees of freedom about $\mathbf{V}_1^k$ and $\mathbf{V}_2^k$ which are 2 axes perpendicular to the shell director $\mathbf{V}_n^k$ as shown in Fig. 2.3-3.

$$\mathbf{V}_1^k = \frac{\mathbf{Y} \times \mathbf{V}_n^k}{\left\| \mathbf{Y} \times \mathbf{V}_n^k \right\|_2}$$

$$\mathbf{V}_2^k = \mathbf{V}_n^k \times \mathbf{V}_1^k$$

For the special case when the $\mathbf{V}_n^k$ vector is parallel to the **Y** axis, the program uses the following conventions:

$$\mathbf{V}_1^k \equiv \mathbf{Z} \qquad \mathbf{V}_2^k \equiv \mathbf{X} \quad \text{when} \quad \mathbf{V}_n^k \equiv +\mathbf{Y}$$

and

$$\mathbf{V}_1^k \equiv -\mathbf{Z} \qquad \mathbf{V}_2^k \equiv \mathbf{X} \quad \text{when} \quad \mathbf{V}_n^k \equiv -\mathbf{Y}$$

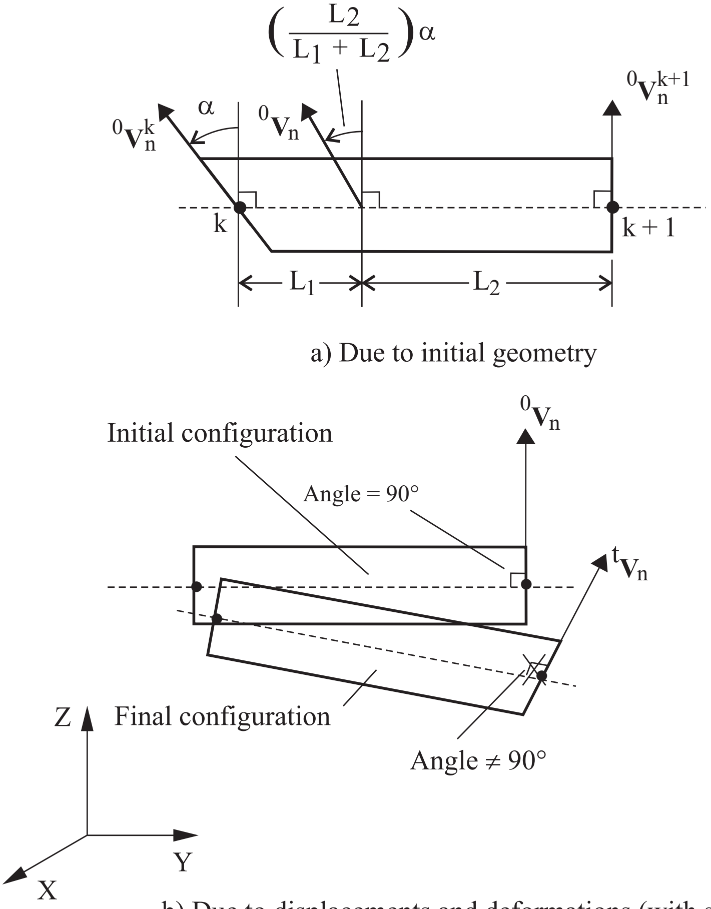
The two rotational degrees of freedom named α_k and β_k are about axes $\mathbf{V}_1^k$ and $\mathbf{V}_2^k$ respectively.

When using the large displacement formulations, the definitions of $\mathbf{V}_1^k$ and $\mathbf{V}_2^k$ are only used at time = 0 (in the initial configuration) after which the vectors ${}^t\mathbf{V}_n^k$ and ${}^t\mathbf{V}_1^k$ are updated using incremental rotations at the nodal points, and ${}^t\mathbf{V}_2^k$ is calculated by the cross-product ${}^t\mathbf{V}_2^k = {}^t\mathbf{V}_n^k \times {}^t\mathbf{V}_1^k$.

Note that a shell node may however be assigned 3 rotational degrees of freedom. In this case, the element's two rotational degrees of freedom are transformed to the displacement coordinate system before assembly.

- Assumption 1 on the kinematic behavior of the shell enters the finite element solution in that the particles along the director vector ${}^t\mathbf{V}_n$ (interpolated from the nodal point director vectors ${}^t\mathbf{V}_n^k$) remain on a straight line during deformation.

Note that in the finite element solution, the vector ${}^{t}\mathbf{V}_{n}$ is not necessarily exactly normal to the shell midsurface. Fig. 2.3-4(a) demonstrates this observation for a very simple case, considering the shell initial configuration. Furthermore, even if ${}^{t}\mathbf{V}_{n}$ is originally normal to the shell midsurface, after deformations have taken place this vector will in general not be exactly perpendicular to the midsurface because of shear deformations (see Fig. 2.3-4(b)).



a) Due to initial geometry

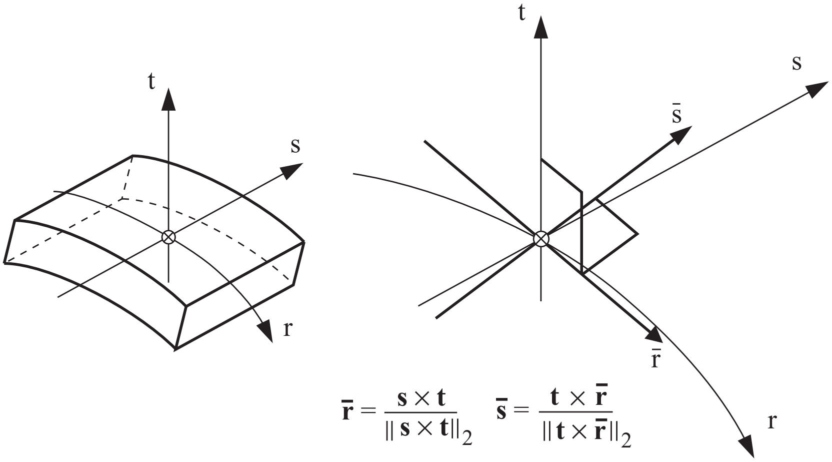
b) Due to displacements and deformations (with shear)

Fig. 2.3-4: Examples of director vectors not normal to the shell midsurface

ref. KJB Section 5.4.2 page 440

- The assumption 2 on the stress situation enters the finite element solution in a manner that is dependent on the formulation employed:

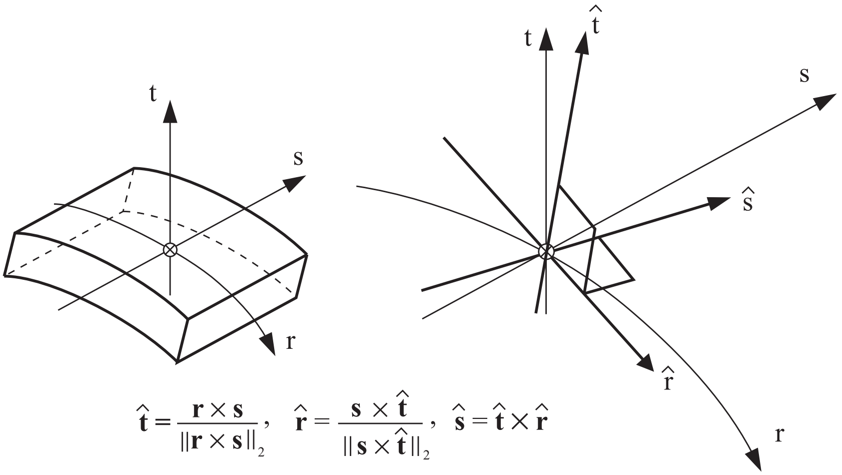
All formulations except for the large displacement/large strain shell element: The stress in the t-direction (i.e., in the direction of ${}^{t}\mathbf{V}_n$) is imposed to be zero. This is achieved by using the stress-strain relationship in the $\bar{r}, \bar{s}, t$ coordinate system, shown in Fig. 2.3-5(a), with the condition that the stress in the direction t is zero.


t
s
r
t
s
s̄
r̄
r
$\bar{\mathbf{r}} = \dfrac{\mathbf{s}\times\mathbf{t}}{\|\mathbf{s}\times\mathbf{t}\|_2}$
$\bar{\mathbf{s}} = \dfrac{\mathbf{t}\times\bar{\mathbf{r}}}{\|\mathbf{t}\times\bar{\mathbf{r}}\|_2}$

(a) Definition of the local Cartesian system at an integration point in the shell

Fig. 2.3-5: Local coordinate systems in shell element

Large displacement/large strain shell element: The stress in the $\hat{t}$ -direction (not necessarily in the direction of ${}^{t}\mathbf{V}_n$) is imposed to be zero. This is achieved by using the stress-strain relationship in the $\hat{r}, \hat{s}, \hat{t}$ coordinate system, shown in Fig. 2.3-5(b), with the condition that the stress in the direction $\hat{t}$ is zero.



$$\hat{\mathbf{t}} = \frac{\mathbf{r} \times \mathbf{s}}{\|\mathbf{r} \times \mathbf{s}\|_2}, \quad \hat{\mathbf{r}} = \frac{\mathbf{s} \times \hat{\mathbf{t}}}{\|\mathbf{s} \times \hat{\mathbf{t}}\|_2}, \quad \hat{\mathbf{s}} = \hat{\mathbf{t}} \times \hat{\mathbf{r}}$$

b) Definition of the midsurface Cartesian system ($\hat{\mathbf{r}}$, $\hat{\mathbf{s}}$, $\hat{\mathbf{t}}$)
at an integration point

Fig. 2.3-5: (continued)

- The interpolation of the geometry of the shell element is always as described above, but for a specific solution time the current coordinates of the midsurface nodal points are used, and the current director vectors are employed. The midsurface nodal point coordinates are updated by the translational displacements of the nodes and the director vectors are updated using the rotations at the nodes (rotation increments in large displacement analysis).

ref. KJB
pp. 399, 440

- The transverse shear deformations are assumed to be constant across the shell thickness.

- In large displacement analysis, the midsurface nodal point coordinates are updated by adding the translational displacements of the nodes, and the director vectors are updated using the incremental rotations at the nodes by applying the large rotation update transformation described in p. 580 of ref. KJB (Exercise 6.56).

2.3.2 Material models and formulations

- See Tables 2-2 and 2-3 for a list of the material models that are compatible with shell elements**.**

- The shell element can be used with

  - **small displacement/small strain** kinematics,
  - **large displacement/small strain** kinematics, or
  - **large displacement/large strain** kinematics.

ref. KJB Section 6.6

In the small displacement/small strain case, the displacements and rotations are assumed to be infinitesimally small. Using a linear material results in a linear element formulation, and using a nonlinear material results in a materially-nonlinear-only formulation.

In the large displacement/small strain case, the displacements and rotations can be large, but the strains are assumed to be small. In this case, a TL formulation is used.

The large displacement/large strain formulation for shells can be either a ULJ (updated Lagrangian Jaumann) formulation or a ULH formulation (updated Lagrangian Hencky) depending on the ULFORM parameter in the NXSTRAT entry. In the ULJ formulation, the total strains can be large, but the incremental strain for each time step should be small (< 1%). The ULH formulation requires more computations, however, it has no such restriction on the size of the incremental strains.

The large displacement/large strain kinematics can be only used with single layer shell elements with a plastic bilinear/multilinear or plastic-cyclic material. See Table 2.3-2 for a list of the supported shell elements.

2.3.3 Shell nodal point degrees of freedom

- Shell nodes can have either 2 or 3 rotational degrees of freedom which results in nodes having either 5 or 6 degrees of freedom.

- The criterion for determining whether a shell node is assigned 5 or 6 degrees of freedom is as follows. 5 degrees of freedom are initially assigned to all shell midsurface nodes. The following cases change the node to 6 degrees of freedom:

▸ **Geometry**. Shell elements at that node intersecting at an angle greater than a specified tolerance (SDOFANG parameter in the NXSTRAT entry).

▸ **Other elements**. If the node also has other elements with rotational degrees of freedom, i.e., beam elements, rotational springs, rotational masses or rotational dampers.

▸ **Rotational loads, constraints or boundary conditions**. This includes the following cases:

- applied moment at the node
- rotational fixed boundary condition at the node
- rigid link connected to the node
- constraint equation involving constrained rotations connected to the node
- enforced rotations at the node

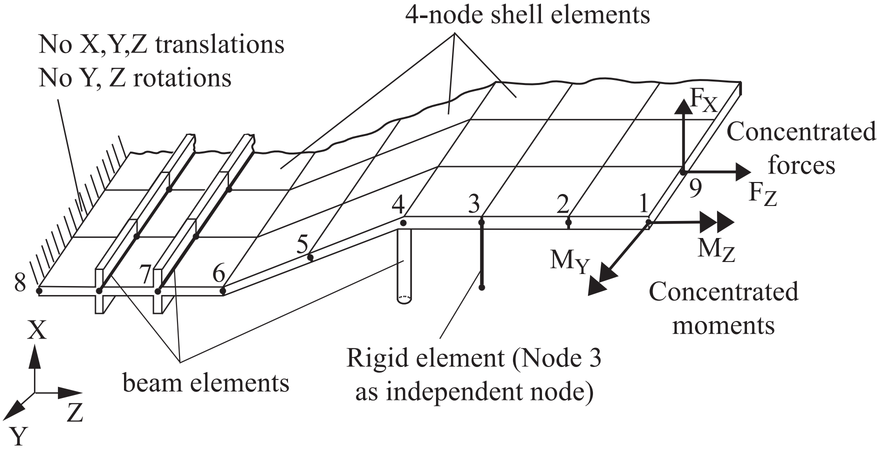
- Shell nodes with 6 degrees of freedom may be a potential source for singularity. In this case, very weak springs are automatically added to prevent the singularity. The cases in which this happens are discussed later in this section.

- Fig. 2.3-6 shows examples of 5 and 6 degree of freedom shell nodes.

- Note that for both 5 and 6 degree of freedom shell nodes, the translations u_k, v_k, w_k are referred to the chosen displacement coordinate system.

5 degrees of freedom node: A node "*k*" that is assigned 5 degrees of freedom incorporates the following assumption:

▸ Only one director vector (denoted at time = 0 as ${}^{0}\mathbf{V}_{n}^{k}$) is associated with the node. The program calculates the director vector by taking the average of all normal vectors (one normal vector is generated per shell element attached to node *k*) at the node. This is illustrated in Fig. 2.3-7.



Node	Number of DOFs	Potential singularity
1	6	Yes
2	5	No
3	6	Yes
4	6	No
5	5	No
6	6	No
7	6	No
8	6	Yes
9	5	No

Fig. 2.3-6: Examples of shell nodes with 5 or 6 degrees of freedom

If two (or more) elements attached to the node have oppositely directed normals, the program reverses the oppositely directed normals, so that all normals attached to the node have (nearly) the same direction.

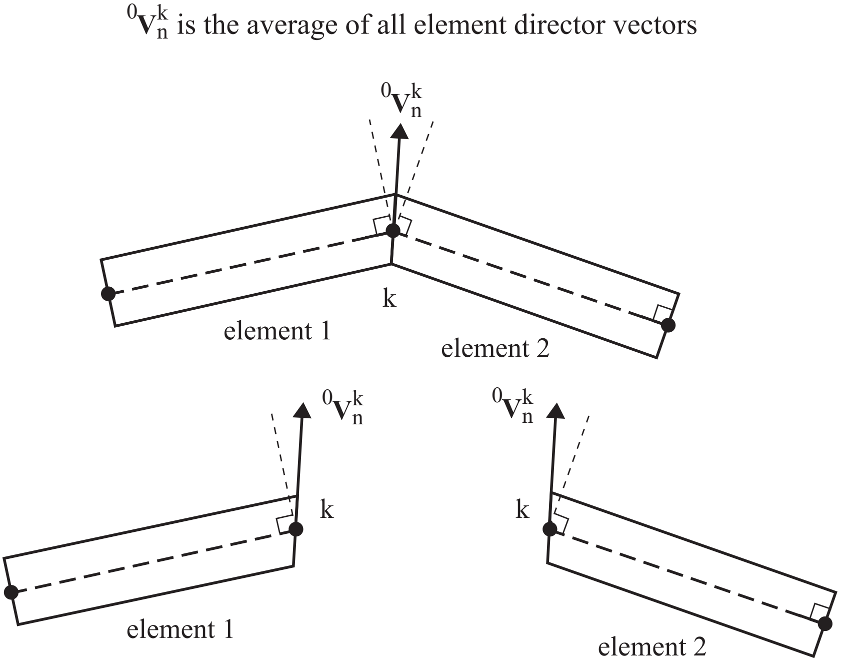


Fig. 2.3-7: 5 degree of freedom shell node with unique vector at node k

6 degrees of freedom node: A node "*k*" that is assigned 6 degrees of freedom incorporates the following assumption:

- The program generates as many normal vectors at node *k* as there are shell elements attached to the node. Hence each individual shell element establishes at node *k* a vector normal to its midsurface. This is illustrated in Fig. 2.3-8. The components of the shell element matrices corresponding to the rotational degrees of freedom at this node are first formulated in the local midsurface system defined by the normal vector and then rotated to the displacement coordinate system.

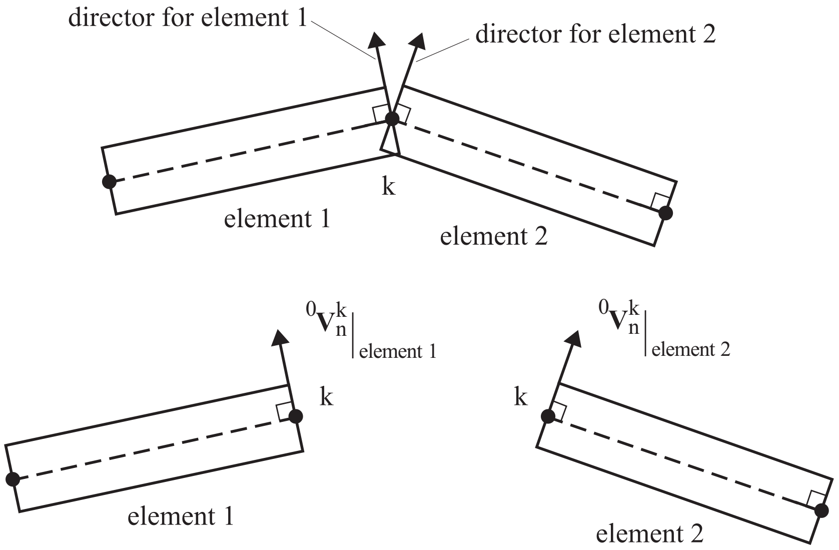

director for element 1
director for element 2
k
element 1
element 2
${}^{0}\mathbf{V}_{n}^{k}\Big|_{\text{element 1}}$
${}^{0}\mathbf{V}_{n}^{k}\Big|_{\text{element 2}}$
k
k
element 1
element 2

Fig. 2.3-8: 6 degree of freedom shell node with separate director vectors at node k (each vector is used as a director vector for the respective element)

- The three rotational degrees of freedom at node *k* referred to the displacement coordinate system can be free or constrained.

Singularity at 6 degree of freedom shell nodes

- When a shell node is forced to have 6 degrees of freedom due to the reasons explained above, there may be a singularity at one of the rotational degrees of freedom. In this case, a weak rotational spring is added to the 3 rotational degrees of freedom. This is done automatically by Advanced Nonlinear Solution and usually does not require user intervention. The stiffness of the spring is set to be a small fraction of the average rotational stiffnesses at the shell node. This fraction is can be changed via the DRILLKF parameter in NXSTRAT.

- Not all the cases that lead to a shell node possessing 6 degrees of freedom (listed at the beginning of this section) may introduce a singularity at the node.

▸ **Geometry**. No potential singularity exists in this case, since the shell is curved.

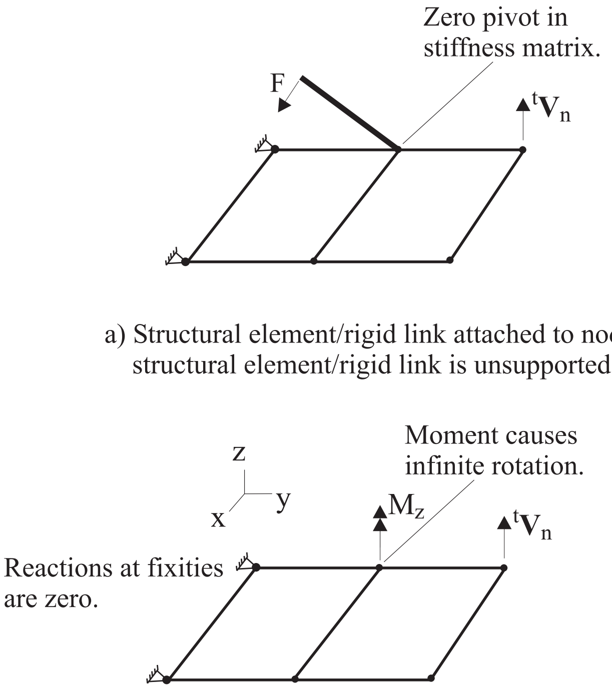
▸ **Other elements**. Beam-stiffened shells will have a singularity at the shell nodes only if the beam is perpendicular to a flat shell surface. Otherwise, a singularity can still exist in the model if the beam is not properly restrained (see Fig. 2.3-9 (a)). The same applies to rotational springs, masses and dampers.

▸ **Rotational loads, constraints or boundary conditions**. All the items listed earlier for this feature result in a potential singularity (see Fig. 2.3-9 (b)) except when all rotational degrees of freedom at the node are fixed.

- If multiple factors lead to the presence of 6 degrees of freedom at a shell node, no singularity is present if any of the factors eliminates the singularity. For example, if a shell node has an applied moment and is attached to non-perpendicular beam elements there is no singularity.

- Fig. 2.3-6 shows examples where shell singularity may or may not occur.

- The singularity that may result from beams attached to shells requires some clarification. If a beam connects two shell structures as shown in Fig. 2.3-10, and it is perpendicular to both shells, then the beam is free to rotate about this perpendicular direction (the z-direction in this example). If the beam intersects the shells at an angle, this singularity is not present.



Zero pivot in stiffness matrix.

a) Structural element/rigid link attached to node, structural element/rigid link is unsupported.

Moment causes infinite rotation.

Reactions at fixities are zero.

b) Moment applied in shell normal direction.

Fig. 2.3-9: Flat shell with 6 DOFs at a node with singularity

- An alternative to using the drilling stiffness option is to connect the 6 DOF nodes on flat shells to neighboring shell nodes using soft beam elements (so-called “weld elements”). This idea is shown in Fig 2.3-11. Then moments applied into the ${}^t\mathbf{V}_n$ direction will be taken by the weld elements, and these moments will cause equilibrating reactions at the fixities. The weld elements also provide stiffness in the ${}^t\mathbf{V}_n$ direction, so that there will be no zero pivots.

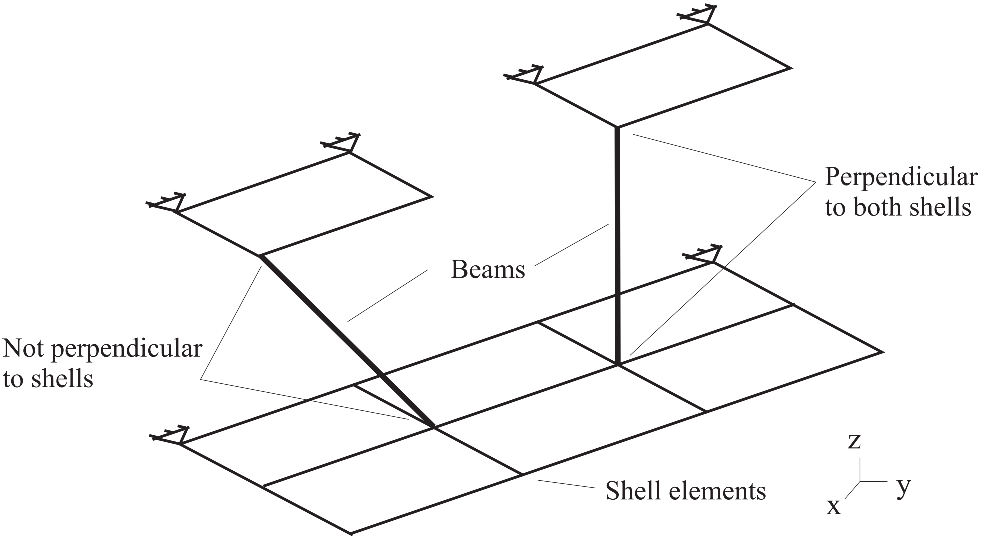


Fig. 2.3-10: Beams intersecting shell elements

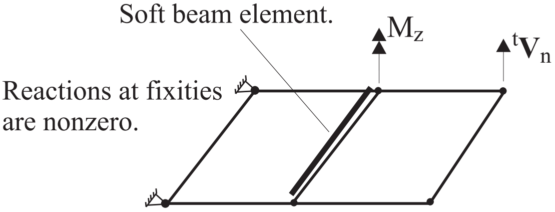


Fig. 2.3-11: Soft beam element takes applied moment

2.3.4 Composite shell elements (Solution 601 only)

- Composite shell elements are generated when a PCOMP or PCOMPG property ID is referenced by one of the following Nastran shell connectivity entries: CQUAD4, CTRIA3, CQUAD8, CTRIA6.

- The composite shell elements are kinematically formulated in the same way as the single layer shell elements, but

  - ▸ An arbitrary number of layers can be used to make up the total thickness of the shell, and each layer can be assigned a different thickness.
  - ▸ Each layer can be assigned one of the different material models available. The element is nonlinear if any of the material models is nonlinear, or if the large displacement formulation is used.
  - ▸ Large displacement/large strain kinematics are not supported for composite shell elements.

- The conventions for defining the director vectors, the local axes $\mathbf{V}_1$ and $\mathbf{V}_2$, and the 5/6 degree of freedom selection are all the same as those for the single layer shell.

- In order to take into account the change of material properties from one layer to another, numerical integration of the mass and stiffness matrices is performed layer by layer using reduced natural coordinates through the thickness of the element (see Figs. 2.3-12 and 2.3-13). The relation between the element natural coordinate t and the reduced natural coordinate t^n of layer n is:

$$t = -1 + \frac{1}{a}\left[2\left(\sum_{i=1}^{n} \ell^i \right) - \ell^n \left(1 - t^n \right) \right] \qquad (2.3\text{-}1)$$

with

t = element natural coordinate through the thickness
t^n = layer n natural coordinate through the thickness
ℓ^i = thickness of layer i
a = total element thickness

a and ℓ^i are functions of r and s.

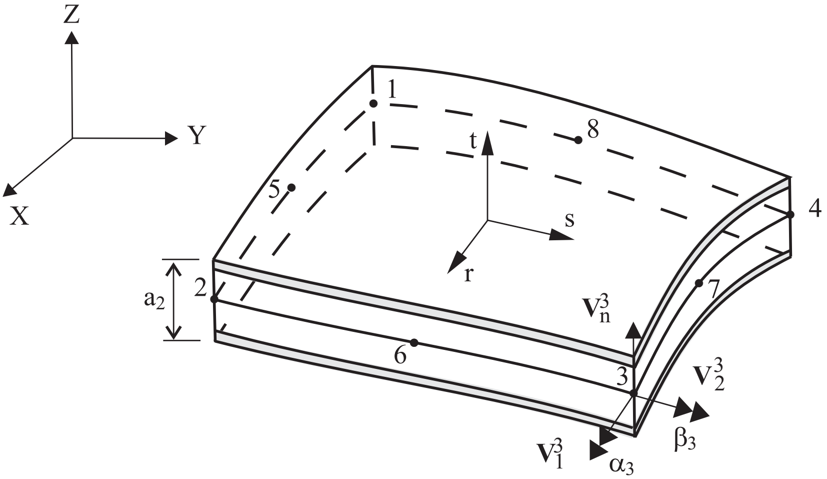


Fig. 2.3-12: 8-node composite shell element

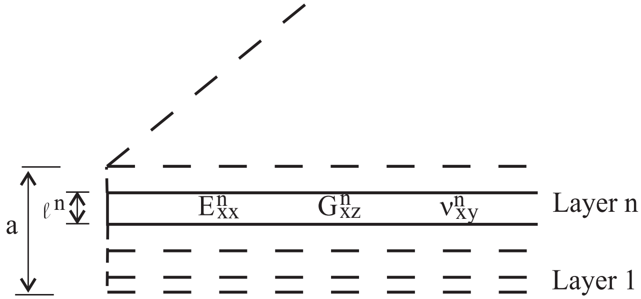


Fig. 2.3-13: Multilayered shell

The geometry of layer *n* is given by:

$$ {}^t x_i = \sum_{k=1}^{N} h_k \, {}^t x_i^k + \sum_{k=1}^{N} \left[m_k^n + t^n \frac{\ell_k^n}{2} \right] h_k \, {}^t V_{ni}^k \qquad (2.3\text{-}2) $$

with

$^{t}x_i$ = coordinate of a point inside layer *n* in direction *i*

N = number of nodes

h_k = interpolation functions

$^{t}x_i^k$ = Cartesian coordinates of node *k*

$^{t}V_{ni}^k$ = component of normal vector $^{t}\mathbf{V}_n^k$ at node *k*

a_k = total element thickness at node *k*

ℓ_k^j = thickness of layer *j* at node *k*

m_k^n = distance between element midsurface and midsurface of layer *n* at node *k*

In the above formula, m_k^n is given by

$$m_k^n = -\frac{a_k}{2} + \sum_{j=1}^{n} \ell_k^j - \frac{\ell_k^n}{2} \tag{2.3-3}$$

2.3.5 Numerical integration

- Gauss numerical integration is used in the in-plane directions of the shell. For the 4-node shell element, 2×2 integration is used. For the 8-node and 9-node elements, 3×3 point integration is used. The 3-node triangular shell element uses 4-point Gauss integration in the in-plane directions, and the 6-node triangular shell element uses 7-point Gauss integration.

ref. KJB Section 6.8.4

- Numerical integration through the shell thickness is as follows:

  ▸ For elastic materials, 2-point Gaussian integration is always used.

  ▸ For elasto-plastic materials and the nonlinear elastic material, 5-point Newton-Cotes integration is the default. Although using 5-point integration is computationally more expensive, it gives much more accurate results for elasto-plastic shells.

- For composite shells with elasto-plastic materials, 3-point Newton-Cotes integration is the default.

The order of through-thickness integration can be modified via the TINT parameter in the NXSTRAT entry. If TINT is specified, it will be applied to both single-layered and composite elasto-plastic shells.

- The same integration order is used for both Solution 601 and 701.

2.3.6 Mass matrices

- In Solution 601 shell elements can be employed with a lumped or a consistent mass matrix. Only a lumped mass matrix is allowed in Solution 701.

- The consistent mass matrix is calculated using the isoparametric formulation with the shell element interpolation functions.

- The lumped mass for translational degrees of freedom of midsurface nodes is M / n where M is the total element mass and n is the number of nodes. No special distributory concepts are employed to distinguish between corner and midside nodes, or to account for element distortion.

The rotational lumped mass for implicit analysis (Solution 601) is $\frac{M}{n} \cdot \frac{1}{12}\left(t_{\text{av}}^2\right)$, where t_{av} is the average shell thickness. The same rotational mass matrix is assumed for 5- and 6-degree of freedom nodes, and is applied to all rotational degrees of freedom.

The rotational lumped mass for explicit analysis (Solution 701) is $\frac{M}{n} \cdot \frac{1}{12}\left(t_{\text{av}}^2 + A\right)$, where t_{av} is the average shell thickness and A is the cross-sectional area. The rotational masses are scaled up to ensures that the rotational degrees of freedom will not reduce the critical time step for shell elements. The same rotational mass matrix is assumed for 5- and 6-degree of freedom nodes and is applied to all rotational degrees of freedom.

2.3.7 Heat transfer capabilities

- Heat transfer capabilities are available for all supported shell elements, including composite shells.

ref KJB Section 5.4.2

- The shell heat transfer capabilities are formulated by assuming that the temperature varies linearly through the shell thickness direction. Two degrees of freedom are therefore assigned at each shell node, one for the top shell surface and one for the bottom shell surface.

- The element matrices are integrated numerically by Gauss integration using the same integration order as the structural matrices.

- In the calculation of the top and bottom shell surfaces, the following geometric quantities are used:

  ▸ The coordinates of the nodes that lie on the shell element midsurface.
  ▸ The director vectors $\mathbf{V}_n$ normal to the shell midsurface.
  ▸ The shell thicknesses a at the nodal points measured in the direction of the vector $\mathbf{V}_n^k$ (see Fig. 2.3-14)

- Fig. 2.3-14 shows a 4-node thermal shell element with the shell midsurface nodes, the nodal director vectors and constructed top and bottom nodes. The director vectors are automatically calculated by the program, see Fig. 2.3-15.

- In the calculation of the shell element matrices, i.e., conductivity, heat capacity, and heat generation, the top and bottom shell surfaces are used instead of the midsurface.

- The shell heat capacity matrix can be calculated based on a lumped or a consistent formulation, similar to the mass matrix in structural analysis.

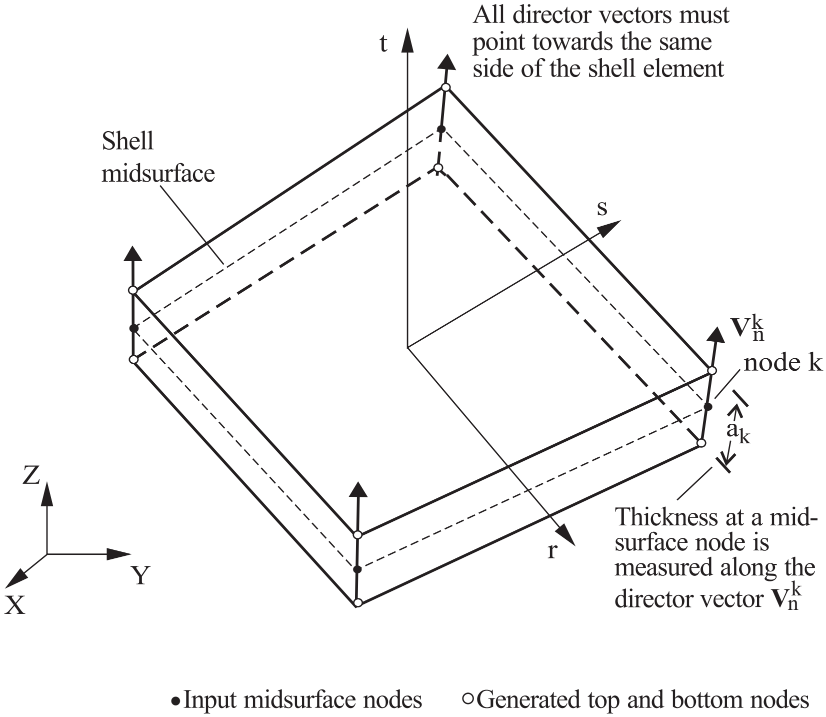

All director vectors must point towards the same side of the shell element
t
Shell midsurface
s
$\mathbf{V}_n^k$
node k
a_k
Z
Y
X
r
Thickness at a mid-surface node is measured along the director vector $\mathbf{V}_n^k$
•Input midsurface nodes
○Generated top and bottom nodes

Fig. 2.3-14: Description of the thermal shell element

- Thermal loads and boundary conditions such as applied temperatures, heat flux, convection and radiation can all be applied to either the top or bottom shell surfaces.

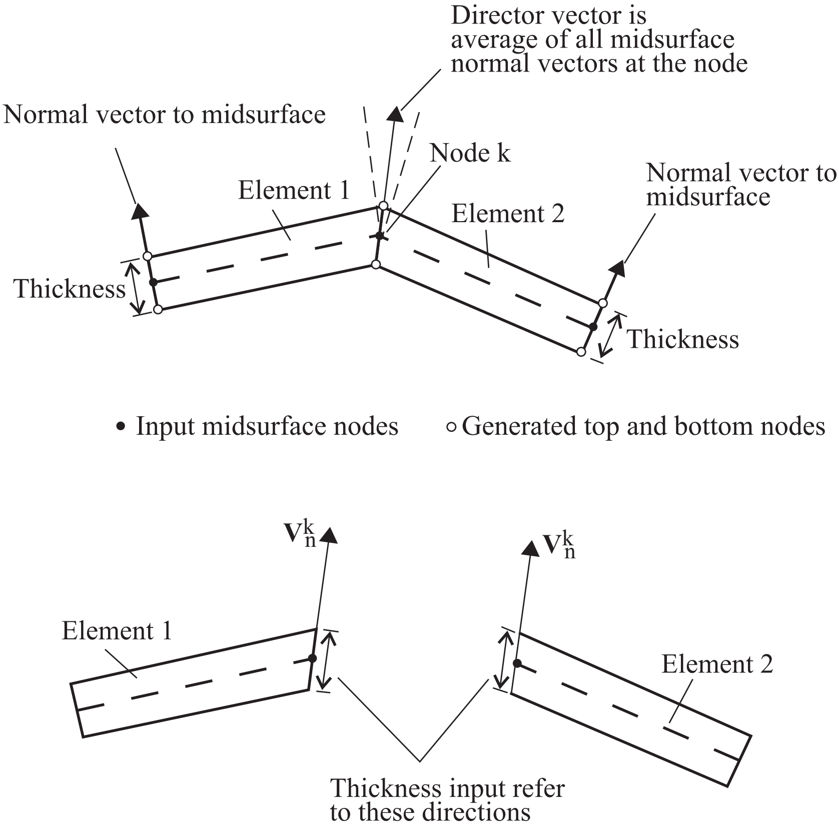


Fig. 2.3-15: Program-calculated director vector at thermal shell nodes

2.3.8 Selection of elements for analysis of thin and thick shells

- The most effective element for analysis of general shells is usually the 4-node element. This element does not lock and has a high predictive capability and hence can be used for effective analysis of thin and thick shells.

ref. KJB
pp. 403-408

- The phenomenon of an element being much too stiff is, in the literature, referred to as element locking. In essence, the phenomenon arises because the interpolation functions used for an element are not "abundantly" able to represent zero (or very small) shearing or membrane strains. If the element cannot represent zero shearing strains, but the physical situation corresponds to zero (or very small) shearing strains, then the element becomes very stiff as its thickness over length ratio decreases. The MITC elements are implemented to overcome the locking problem. More details on the interpolations used for the transverse and membrane terms are provided in ref. KJB, pp. 403 – 406.

- The 8-node element is not as effective as the other MITC elements, and its use is not recommended in general.

- In order to arrive at an appropriate element idealization of a thin shell, it may be effective to consider the behavior of a single element in modeling a typical part of the shell. As an example, if a shell of thickness h and principal radii of curvatures R_1 and R_2 is to be analyzed, a single element of this thickness and these radii and supported as a cantilever could be subjected to different simple stress states. The behavior of the single element when subjected to the simple stress states (e.g., constant bending moments) tells what size of element, and hence element idealization, can be used to solve the actual shell problem.

- For cases where the ratio thickness/radius of curvature is large (in the original configuration or in the deformed configuration in large deformation analysis) it is best to use 5 dofs at each shell node. Then all elements will represent the hyperbolic stress distributions through the element thickness. If 6 dofs per node are used, then only the higher order elements will represent the hyperbolic stress distribution because the nodal director vectors are constructed from the geometry of the elements.

- Geometrically nonlinear incompatible modes elements with large aspect ratio should not be used, because spurious modes may be present in the finite element solution.

2.3.9 3D-shell element

Overview

- One characteristic of the shell elements described earlier is that the change in thickness of the element is not explicitly calculated from the element degrees of freedom. This is because the zero stress through the shell thickness assumption is used in the material descriptions.

- However, in large strain analysis, the change in thickness can become important. For example, during out-of-plane bending, the material in compression thickens, and the material in tension thins (for nonzero Poisson's ratio). Hence the midsurface is no longer exactly halfway between the top and bottom surfaces. This effect is shown in Fig 2.3-16.

- This change in thickness is naturally modeled when quadratic 3D solid elements are used, because the nodes on the top and bottom surfaces can move relative to the nodes on the midsurface, as shown in Fig 2.3-17. However 3D solid elements tend to lock when they are very thin, so that they are unsuitable for bending analysis of thin structures.

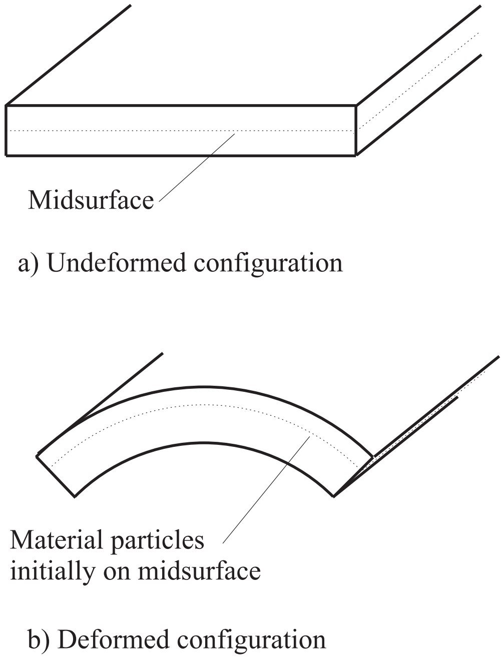
- In the 3D-shell element, the change in thickness of the element is modeled using control vectors, as shown for the case of pure bending in Fig 2.3-18. The motion of the control vectors is controlled by element degrees of freedom at the shell midsurface, as discussed in more detail below.

- Because the change in thickness is explicitly calculated from element degrees of freedom, the assumption of zero stress through the shell thickness is not used in the 3D-shell element.

- In addition, the 3D-shell elements use MITC tying rules to relieve shear locking. Therefore these elements are suitable for out-of-plane bending analysis of thin structures, even for large bending strains.

- The 3D-shell elements can be used with 3 or 4 nodes (CTRIA3 or CQUAD4). The 4-node element is recommended for general use.

- The 3D-shell elements can be used in static, implicit dynamic or explicit dynamic analysis.



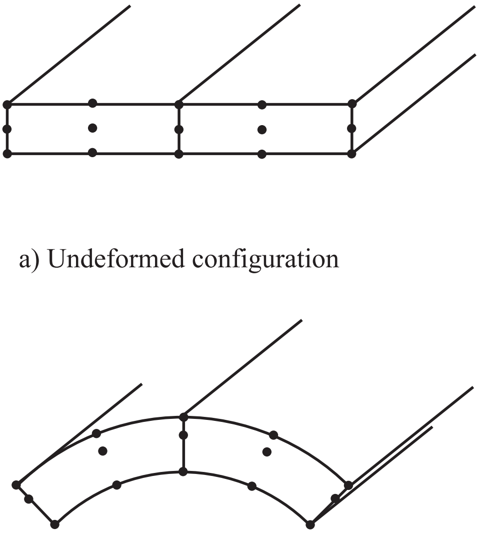
a) Undeformed configuration

b) Deformed configuration

Fig. 2.3-16: Kinematics of pure bending

- The following features available for other shell elements are not available for the 3D-shell element: 6-, 8-node elements, composite (multilayer) shells.

- When using contact groups with true offsets (BCTPARA entry, OFFTYPE=2), the 3D-shell elements contribute to the offsets. However, the offset at a 3D-shell element node is computed as half the current shell thickness at the node, and this offset is used regardless of whether the contact occurs on the shell top surface or on the shell bottom surface.



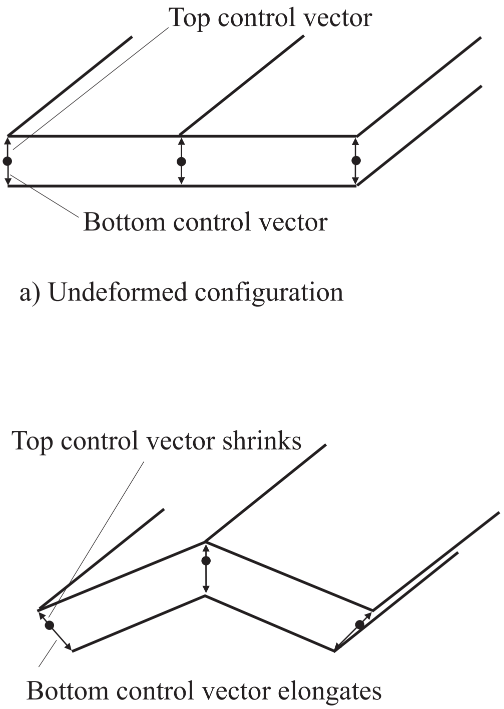
a) Undeformed configuration

b) Deformed configuration

Fig. 2.3-17: Pure bending modeled with quadratic 3D elements

- The 3D-shell is more fully described in the following reference:

ref. T. Sussman & K.J. Bathe, 3D-shell elements for structures in large strains, *Computers & Structures*, **122**, 2-12, 2013.



a) Undeformed configuration

b) Deformed configuration

Fig. 2.3-18: Pure bending modeled with 3D-shell elements

Kinematics and degrees of freedom of the 3D-shell element

Fig. 2.3-19 shows the corner of a 3D-shell element, with its top and bottom surfaces described by control vectors. Initially the control vectors are equal and opposite. During deformations, the control vectors can evolve independently. Thus in the deformed configuration, the control vectors are in general not equal and opposite, as shown in Fig 2.3-19(b).

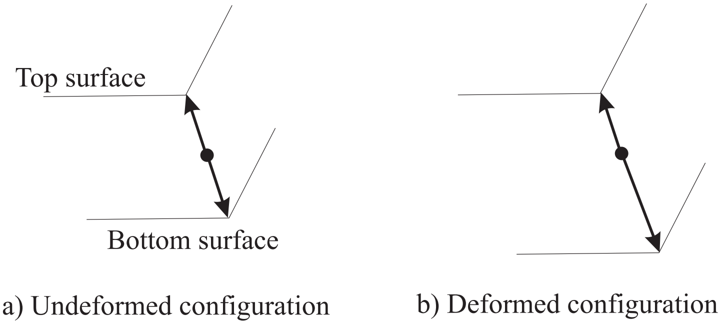


Fig. 2.3-19: Control vectors at 3D-shell element node

The control vector motions are governed by element degrees of freedom. For ease of use of the element, the element degrees of freedom include the same degrees of freedom as for the usual shell elements:

x, y, z translations
α , β rotations (5 DOF node), or θ_x , θ_y , θ_z rotations (6 DOF node)

and extra degrees of freedom:

constant and linear thickness incremental strains

The elongations of the control vectors are governed by the constant and linear thickness incremental strains. The control vectors always point in opposite directions, but in general have different lengths after deformations. And material particles that were initially on a straight line remain on a straight line after deformations.

All of the considerations for selection of 5 and 6 DOF nodes discussed in Section 2.7.3 directly apply to the 3D-shell element nodes.

It is not allowed to prescribe or fix any of the extra degrees of freedom.

Material models and formulations for the 3D-shell element

Material models for the usual shell elements are developed using the assumption of zero stress through the shell thickness. Hence these models do not directly apply to the 3D-shell element.

The following material models are implemented for the 3D-shell element: **elastic-isotropic (MAT1)**, **plastic-cyclic (MATPLCY), hyperelastic (MATHE, MATHP).**

Small displacement/small strain, **large displacement/small strain** or **large displacement/large strain formulations** can be used with these material models, as described under the material models descriptions.

Material model notes

The thermal strains are assumed to be small in all material models, including the hyperelastic material models.

When stresses are output in the shell local coordinate system, the coordinate system is the $\hat{r},\hat{s},\hat{t}$ system shown in Fig. 2.3-5(b).

Mixed u/p formulation

The mixed u/p formulation is used for the plastic-cyclic material and all of the hyperelastic materials except for the hyperfoam material. When the u/p formulation is used, the assumed pressure field is $p = p_0 + p_1 t$ where p_0 and p_1 are the pressure degrees of freedom, and t is the isoparametric coordinate through the shell thickness. Note that it is necessary to allow a linear variation of pressure through the shell thickness in order to model out-of-plane bending.

Incompatible modes

Incompatible modes can be used in the 4-node 3D-shell element. However incompatible modes and the u/p formulation cannot be used together, therefore incompatible modes are not used for the materials in the last paragraph.

2.4 Surface elements – 2-D solids (Solution 601 only)

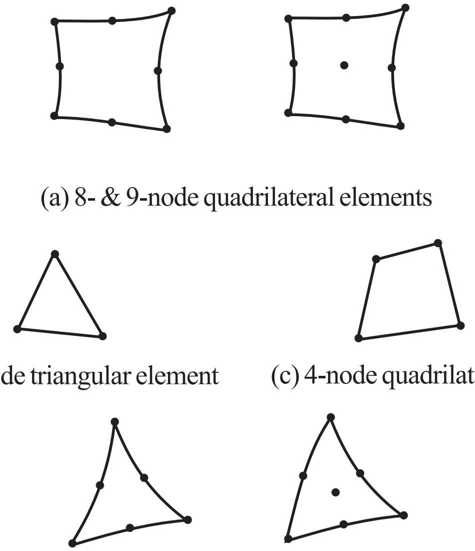
- 2-D solid elements are obtained in the following cases:

  - PPLANE property ID that is referenced by CPLSTS3, CPLSTS4, CPLSTS6 or CPLSTS8 plane stress elements, or the CPLSTN3, CPLSTN4, CPLSTN6 or CPLSTN8 plane strain elements. This leads to 2D plane strain or plane stress elements that must be oriented in either the X-Y or X-Z plane.
  - PSOLID or PLSOLID property ID entry that is referenced by the axisymmetric elements CQUADX4, CQUADX8, CTRAX3, or CTRAX6. This leads to an axisymmetric 2-D element which must be oriented in either the X-Y or X-Z plane. This is the preferred form for axisymmetric elements since elastic, plastic and hyperelastic materials can be used with these elements. Contact analysis can also be performed with these elements.
  - PLPLANE property ID that is referenced by the CPLSTS3, CPLSTS4, CPLSTS6 or CPLSTS8 plane stress elements, or the CPLSTN3, CPLSTN4, CPLSTN6 or CPLSTN8 plane strain elements. This leads to a hyperelastic 2-D plane strain or plane stress element which must be oriented in either the X-Y or X-Z plane.
  - PLPLANE property ID entry that is referenced by the CQUAD, CQUAD4, CQUAD8, CTRIA3 or CTRIA6 shell elements. This leads to a hyperelastic plane strain or axisymmetric 2-D element which must be oriented in the X-Y plane.
  - PSHELL property ID entry with MID2 = -1 that is referenced by the shell elements CQUAD4, CQUAD8, CTRIA3, or CTRIA6. This leads to a plane strain 2-D element which must be oriented in the X-Y plane.

- 2-D elements are not supported in Solution 701.

2.4.1 General considerations

- The following kinematic assumptions are available for two-dimensional elements in Solution 601: **plane strain**, **plane stress** and **axisymmetric**. Fig. 2.4-1 and Fig. 2.4-2 show some typical 2-D elements and the assumptions used in the formulations.



(a) 8- & 9-node quadrilateral elements

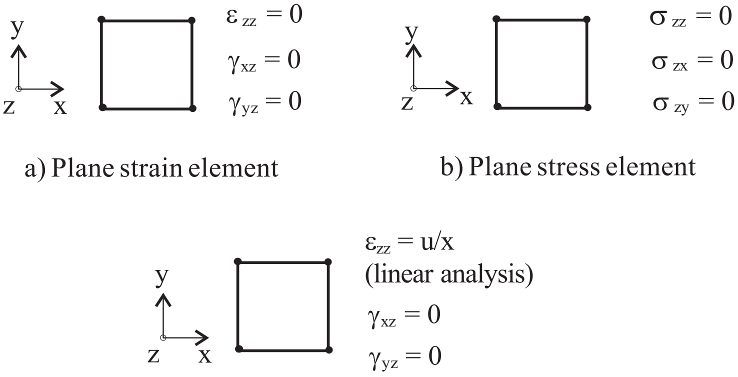
(b) 3-node triangular element

(c) 4-node quadrilateral element

(d) 6- & 7-node triangular elements

Fig. 2.4-1: 2-D solid elements

- 2-D solid elements in Solution 601 are classified based on the number of nodes in the element and the element shape. Table 2.4-1 shows the correspondence between the different 2-D solid elements and the NX element connectivity entries.



a) Plane strain element

b) Plane stress element

c) Axisymmetric element

Fig. 2.4-2: Basic assumptions in 2-D analysis (assuming element lies in X-Y plane)

- Note that the extra middle node in the 7-node and 9-node 2-D elements is automatically added by the program when ELCV is set to 1 in the NXSTRAT entry. These extra nodes improve the performance of the 2-D elements as explained later in this section. The boundary conditions at the added node are predicted from the neighboring nodes.

- The axisymmetric element must lie in the +X half plane.

- 2-D solid elements can be combined with any other elements available in Advanced Nonlinear Solution.

ref. KJB Sections 5.3.1 and 5.3.2

- The axisymmetric element represents one radian of the structure, and defines the stiffness, mass and forces accordingly. Hence, when this element is combined with other elements, or when concentrated loads are defined, these must also refer to one radian, see ref. KJB, Examples 5.9 and 5.10, p. 356.

- The plane strain element provides for the stiffness of a unit thickness of the structure, and defines the stiffness, mass and forces accordingly.

Table 2.4-1: Correspondence between 2-D solid elements and NX element connectivity entries

2-D solid element	NX element connectivity entry
3-node triangle	CPLSTN3[1], CPLSTS3[2] CTRIA3[4], CTRAX3[5]
4-node quadrilateral	CPLSTN4[1], CPLSTS4[2] CQUAD4[4], CQUADX4[5] CQUAD[3] (with 4 input nodes)
6-node triangle	CPLSTN6[1], CPLSTS6[2] CTRIA6[4], CTRAX6[5]
7-node triangle	CPLSTN6[1,6], CPLSTS6[2,6] CTRIA6[4,6], CTRAX6[5,6]
8-node quadrilateral	CPLSTN8[1], CPLSTS8[2] CQUAD8[4], CQUADX8[5] CQUAD[3] (with 8 input nodes)
9-node quadrilateral	CPLSTN8[1,6], CPLSTS8[2,6] CQUAD8[4,6], CQUADX8[5,6] CQUAD[3,6] (with 8 input nodes) CQUAD[3] (with 9 input nodes)

Notes:

1. Plane strain
2. Plane stress
3. Plane strain hyperelastic only
4. Plane strain hyperelastic
5. Axisymmetric with no restriction on material
6. With ELCV = 1 in NXSTRAT entry

- The plane stress 2-D element has an element thickness that is defined either in the CPLSTSi element entry or in the PPLANE or PLPLANE entry. The element can have a varying thickness, that is, the thickness can be different at each node. However, the thickness at a mid-side node is always taken as the average of the thickness of the corresponding corner nodes.

- The basic 2-D elements used in Solution 601 are isoparametric displacement-based elements, and their formulation is described in detail in ref. KJB, Section 5.3.

- The basic finite element assumptions for the coordinates are (see Fig. 2.4-3):

$$x = \sum_{i=1}^{q} h_i x_i \; ; \quad y = \sum_{i=1}^{q} h_i y_i$$

and for the displacements:

$$u = \sum_{i=1}^{q} h_i u_i \; ; \quad \mathrm{v} = \sum_{i=1}^{q} h_i \mathrm{v}_i$$

where

$h_i(r,s)$ = interpolation function corresponding to node i
(r,s) = isoparametric coordinates
q = number of element nodes
x_i, y_i = nodal point coordinates
u_i, v_i = nodal point displacements

The equations above are for 2-D solid elements that lie in the X-Y plane. A simple change of variable from y to z describes the 2-D solid elements in the X-Z plane.

- In addition to the displacement-based elements, special mixed-interpolated elements are also available, in which the displacements and pressure are interpolated separately. These elements are effective and should be preferred in the analysis of incompressible media and inelastic materials (elastic materials with Poisson's ratio close to 0.5, rubber-like materials, creep and elasto-plastic materials). The mixed formulation is only available for plane strain and axisymmetric 2-D elements. It is not available (and not needed) for plane stress 2-D elements.

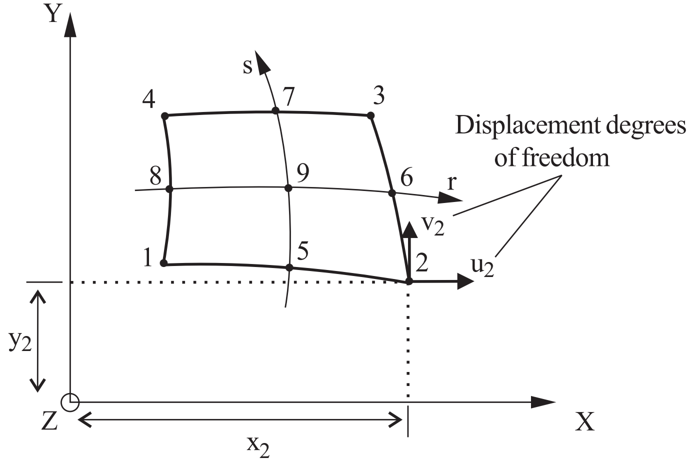


Fig. 2.4-3: Conventions used for the 2-D solid element (assuming element lies in X-Y plane)

- Table 2.4-2 shows the number of pressure degrees of freedom used for each 2-D element type. For more details on the number of degrees of freedom ideal for each element, see the ref. KJB, Section 4.4.3 and Table 4.6, pp. 292-295.

Table 2.4-2: Mixed formulation settings for 2-D solid elements

2-D solid element	Number of pressure DOFs
3-node triangle	-
4-node quadrilateral	1
6-node triangle	3
7-node triangle	3
8-node quadrilateral	3
9-node quadrilateral	3

- The mixed interpolation is the default setting for hyperelastic materials. It can be activated for other materials, such as elastic with Poisson’s ratio close to 0.5, elasto-plastic, and creep, via the UPFORM flag in the NXSTRAT entry.

- The 4-node element (1 pressure degree of freedom) and 9-node element (3 pressure degrees of freedom) are recommended for use with the mixed formulation.

ref. T. Sussman and K.J. Bathe, "A Finite Element Formulation for Nonlinear Incompressible Elastic and Inelastic Analysis," *J. Computers and Structures*, Vol. 26, No. 1/2, pp. 357-409, 1987.

- In addition to the displacement-based and mixed-interpolated elements, Advanced Nonlinear Solution also includes the possibility of including incompatible modes (bubble functions) in the formulation of the 4-node 2-D solid element. Within this element, additional displacement degrees of freedom are introduced. These additional displacement degrees of freedom are not associated with nodes; therefore the condition of displacement compatibility between adjacent elements is not satisfied in general. The addition of the incompatible modes (bubble functions) increases the flexibility of the element, especially in bending situations. For theoretical considerations, see reference KJB, Section 4.4.1. Note that these incompatible-mode elements are formulated to pass the patch test. Also note that element distortions deteriorate the element performance when incompatible modes are used.

For plane stress and plane strain elements, the incompatible modes feature is activated by settting ICMODE=1 in the NXSTRAT entry. For axisymmetric elements, the incompatible modes feature is activiated by setting IN=BUBBLE in the PSOLID entry.

The incompatible modes feature should be used with caution when using large displacement/small strain or large displacement/ large strain kinematics, in conjunction with large aspect ratio elements, because meshes of incompatible modes elements can contain spurious modes under these conditions.

The incompatible modes feature cannot be used in conjunction with the mixed-interpolation formulation.

- The interpolation functions used for 2-D solid elements are defined in ref. KJB, Fig. 5.4, p. 344.

ref. KJB Section 5.3.2

- The 6-node spatially isotropic triangle is obtained by correcting the interpolation functions of the collapsed 8-node element. It then uses the same interpolation functions for each of the 3 corner nodes and for each of the midside nodes.

  The 3-node triangular element is obtained by collapsing one side of the 4-node element. This element exhibits the constant strain conditions (except that the hoop strain in axisymmetric analysis varies over the element).

- The stresses/strains can be output either at the center and corner grid points (PSOLID STRESS=blank or GRID), or at the center and corner Gauss points (PSOLID STRESS=1 or GAUSS). The option for output at the Gauss points is only available for axisymmetric elements.

2.4.2 Material models and formulations

- See Tables 2-2 and 2-3 for a list of the material models that are compatible with 2-D solid elements.

- Advanced Nonlinear Solution automatically uses the mixed interpolation formulation for hyperelastic materials. The mixed formulation is also recommended for elastic-plastic materials and also elastic materials with a Poisson ratio close to 0.5. For these materials, the u/p mixed formulation can be activated by setting UPFORM = 1 in the NXSTRAT entry.

- The two-dimensional elements can be used with

  - **small displacement/small strain** kinematics**,**

  - **large displacement/small strain** kinematics, or

  - **large displacement/large strain** kinematics.

  ▸ The small displacement/small strain and large displacement/small strain kinematics can be used with any of the compatible material models, except for the hyperelastic material. The use of a linear material with small displacement/small strain kinematics corresponds to a linear formulation, and the use of a nonlinear material with the small displacement/small strain kinematics corresponds to a materially-nonlinear-only formulation.

- ▸ The program uses the TL (total Lagrangian) formulation when the large displacement/small strain formulation is selected.
- ▸ The large displacement/large strain kinematics can be used with plastic materials including those with thermal and creep effects, as well as hyperelastic materials. The ULH (updated Lagrangian Hencky) formulation or ULJ (updated Lagrangian Jaumann) formulation can be used for all compatible material models except the hyperelastic material. For the hyperelastic material, the TL (total Lagrangian) formulation is used. The ULFORM parameter in the NXSTRAT entry determines the ULH/ULJ setting.

ref. KJB Sections 6.2 and 6.3.4

• The basic continuum mechanics formulations of 2-D solid elements are described in ref. KJB, pp. 497-537, and the finite element discretization is given in ref. KJB pp. 538-542, 549-555.

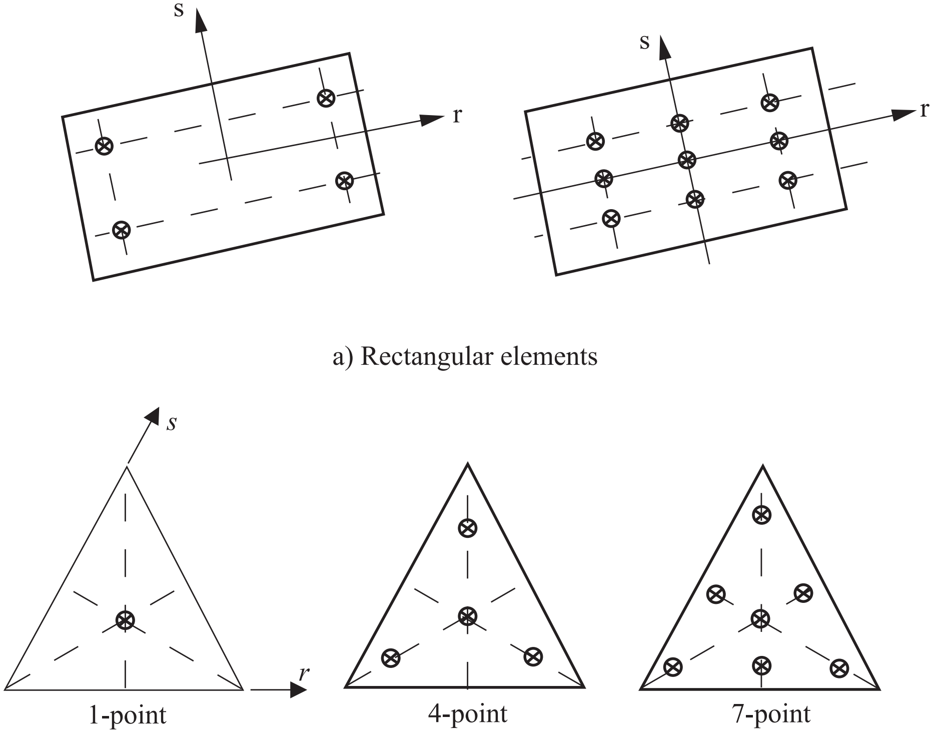
ref. KJB Section 6.8.1

• Note that all these formulations can be mixed in the same finite element model. If the elements are initially compatible, then they will remain compatible throughout the analysis.

2.4.3 Numerical integration

ref. KJB Sections 5.5.3, 5.5.4 and 5.5.5

• The 4-node quadrilateral element uses 2×2 Gauss integration for the calculation of element matrices. The 8-node and 9-node elements use 3×3 Gauss integration. See Fig 2.4-4(a).



a) Rectangular elements

b) Triangular elements

Fig. 2.4-4: Integration point positions for 2-D solid elements

- The 3-node, 6-node and 7-node triangular elements are spatially isotropic with respect to integration point locations and interpolation functions (see Section 5.3.2, ref. KJB). The 3-node element uses a single point integration in plane strain and 4-point Gauss integration in the axisymmetric case. The 6-node and 7-node triangular elements use 7-point Gauss integration. See Fig 2.4-4(b).

- Note that in geometrically nonlinear analysis, the spatial positions of the Gauss integration points change continuously as the element undergoes deformations, but throughout the response the same material particles are at the integration points.

2.4.4 Mass matrices

- The consistent mass matrix is always calculated using either 3×3 Gauss integration for rectangular elements or 7-point Gauss integration for triangular elements.

- The lumped mass matrix of an element is formed by dividing the element's mass M equally among its n nodes. Hence, the mass assigned to each node is M / n. No special distributory concepts are employed to distinguish between corner and midside nodes, or to account for element distortion.

2.4.5 Heat transfer capabilities

- Heat transfer capabilities are available for all 2-D solid elements.

- The planar 2-D solid heat transfer elements may be defined using any of the 2-D solid plane strain or plane stress element entries. There is no difference between plane strain and plane stress for heat transfer analysis, except for the element thickness, see below.

- One temperature degree of freedom is present at each node.

- The axisymmetric elements must be defined using the CQUADX$_i$ or CTRAX$_i$ entries, and they cover one radian of the physical domain.

- The element matrices are integrated numerically by Gauss integration using the same integration order as the structural matrices.

- The planar 2-D heat transfer element assumes the same thickness as the underlying plane stress or plane strain element. The axisymmetric element always extends one radian in the circumferential direction.

- The heat capacity matrix can be calculated based on a lumped or consistent heat capacity assumption.

- The lumped heat capacity matrix of an element is formed by dividing the element's total heat capacity *C* equally among its *n* nodes. Hence, the mass assigned to each node is C/n. No special distributory concepts are employed to distinguish between corner and midside nodes, or to account for element distortion.

2.4.6 Recommendations on use of elements

- The 9-node element is usually the most effective.

- The linear interpolation elements (3-node and 4-node) should only be used in analyses when bending effects are not dominant. If the 4-node element is used in problems where bending effects are significant, incompatible modes should be activated.

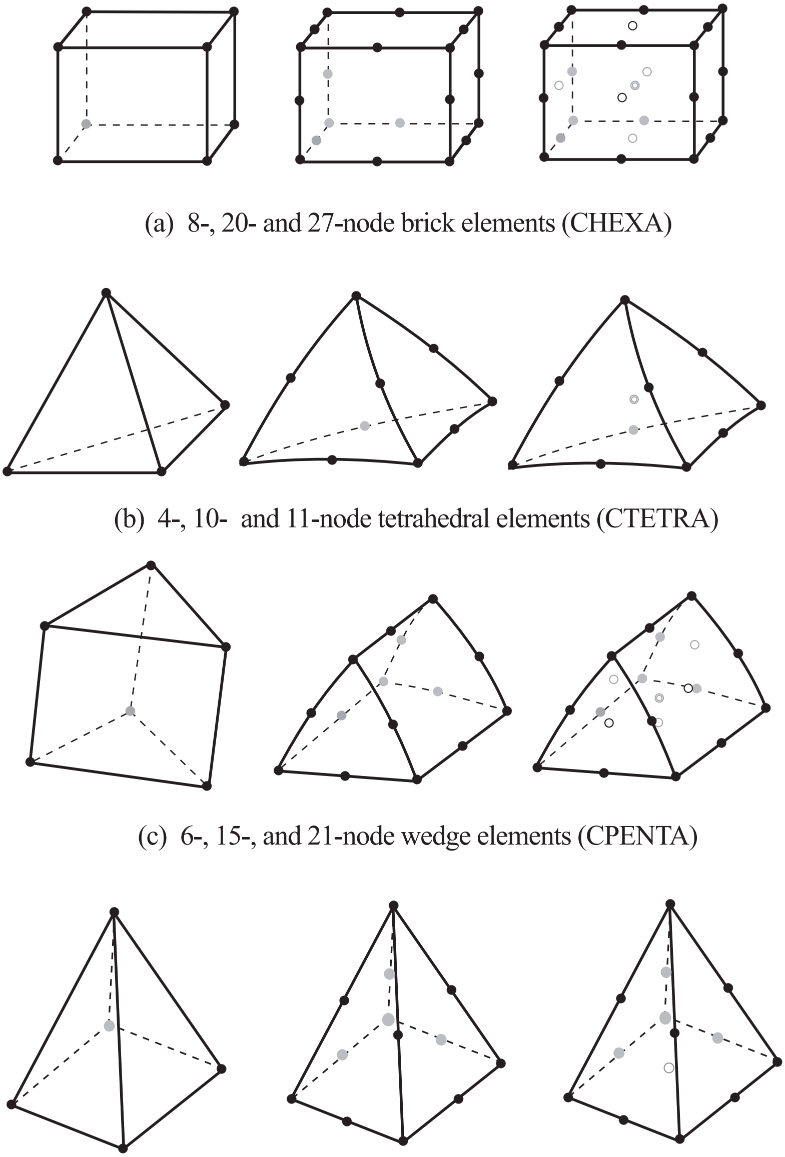
- For nearly incompressible elastic materials, elasto-plastic materials and creep materials, and when using plane strain or axisymmetric elements, the use of the u/p mixed formulation elements is recommended.

2.5 Solid elements – 3-D

2.5.1 General considerations

- 3-D solid elements are generated using the CHEXA, CPENTA , CTETRA and CPYRAM element connectivity entries. They generate 6-, 5- and 4-sided 3-D elements. Typical 3-D solid elements are shown in Fig 2.5-1.

- The PSOLID property ID entry is used for all of the supported materials, except hyperelastic, which uses PLSOLID.



(a) 8-, 20- and 27-node brick elements (CHEXA)

(b) 4-, 10- and 11-node tetrahedral elements (CTETRA)

(c) 6-, 15-, and 21-node wedge elements (CPENTA)

(d) 5-, 13-, and 14-pyramid elements (CPYRAM)

Fig. 2.5-1: 3-D solid elements

- 3-D solid elements in Advanced Nonlinear Solution are classified based on the number of nodes in the element, and the element shape.

- Table 2.5-1 shows the correspondence between the different 3-D solid elements and the NX element connectivity entries. Note that the elements are frequently referred to just by their number of nodes.

- Solution 701 only supports linear elements (4-node tetrahedron, 6-node wedge and 8-node brick elements).

Table 2.5-1: Correspondence between 3-D solid elements and NX element connectivity entries

3-D solid element	NX element connectivity entry
4-node tetrahedron	CTETRA
10-node tetrahedron[1]	CTETRA
11-node tetrahedron[1]	CTETRA and ELCV = 1 in NXSTRAT
6-node wedge	CPENTA
15-node wedge[1]	CPENTA
21-node wedge[1]	CPENTA and ELCV = 1 in NXSTRAT
8-node brick	CHEXA
20-node brick[1]	CHEXA
27-node brick[1]	CHEXA and ELCV = 1 in NXSTRAT
5-node pyramid	CPYRAM
13-node pyramid[1]	CPYRAM
14-node pyramid[1]	CPYRAM and ELCV = 1 in NXSTRAT

Note:

1. Only for Solution 601

- Advanced Nonlinear Solution supports incomplete quadratic 3-D elements for tetrahedral and pyramid elements. Incomplete quadratic elements are not supported for brick and wedge elements. For example, a CHEXA entry can only have 8 nodes or 20 nodes. Anything in between is not supported. Also, a CTETRA can have any of its midside nodes removed.

- For nonlinear analysis, stress/strain results for 3-D solid elements are output in the element coordinate system. ELRESCS = 1 in NXSTRAT may be used to request output of nonlinear stress/strain results in the material coordinate system. The option is useful for post-processors that do not perform any transformation of the stress/strain coordinate system when importing the op2 file.

- Note that the mid-volume and midsurface nodes in the 27-node, 21-node, 14-node and 11-node elements are automatically added by Advanced Nonlinear Solution when ELCV is set to 1 in the NXSTRAT entry. The boundary conditions at the added nodes are predicted from the neighboring nodes.

ref. KJB Section 5.3

- The elements used in Advanced Nonlinear Solution are isoparametric displacement-based elements, and their formulation is described in ref. KJB, Section 5.3.

- The basic finite element assumptions for the coordinates are (see Fig. 2.5-2, for the brick element):

$$x = \sum_{i=1}^{q} h_i x_i \qquad y = \sum_{i=1}^{q} h_i y_i \qquad z = \sum_{i=1}^{q} h_i z_i$$

and for the displacements:

$$u = \sum_{i=1}^{q} h_i u_i \qquad \mathrm{v} = \sum_{i=1}^{q} h_i \mathrm{v}_i \qquad w = \sum_{i=1}^{q} h_i w_i$$

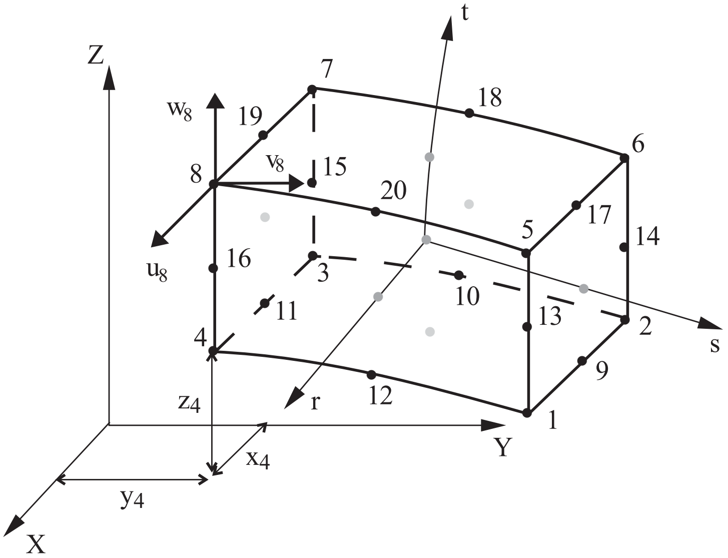


Fig. 2.5-2: Conventions used for the nodal coordinates and displacements of the 3-D solid element

where

$h_i\ (r, s, t)$	=	interpolation function corresponding to node *i*
r, s, t	=	isoparametric coordinates
q	=	number of element nodes
x_i, y_i, z_i	=	nodal point coordinates
u_i, v_i, w_i	=	nodal point displacements

- In addition to the displacement-based elements, special mixed-interpolated elements are also available, in which the displacements and pressure are interpolated separately. These elements are effective and should be preferred in the analysis of incompressible media and inelastic materials (specifically for materials in which Poisson's ratio is close to 0.5, for rubber-like materials and for elasto-plastic materials). Table 2.5-2 shows the number of pressure degrees of freedom for each 3-D element type. For more details on the mixed interpolation of pressure and displacement degrees of freedom for 3-D solids, see Section 4.4.3, p. 276, and Tables 4.6 and 4.7, pp. 292 - 295 in ref. KJB.

Table 2.5-2: Mixed u/p formulations available for 3-D solid elements

3-D solid element	Number of pressure DOFs
4-node tetrahedron	-
10-node, 11-node tetrahedron	4
6-node wedge	1
15-node, 21-node wedge	4
8-node brick	1
20-node, 27-node brick	4
5-node pyramid	1
13-node, 14-node pyramid	1

- The mixed interpolation is the default setting for hyperelastic materials. It can be activated for other materials, such as elastic with Poisson's ratio close to 0.5, elasto-plastic, and creep, via the UPFORM flag in the NXSTRAT entry.

- The 8-node element (1 pressure degree of freedom) and 27-node element (4 pressure degrees of freedom) are recommended for use with the mixed formulation.

- Note that 4 pressure degrees of freedom are used for the 10-node tetrahedron, the 15-node wedge and the 20-node brick element. Even though this setting does not satisfy the inf-sup test, the elements generally perform better than with a single pressure degree of freedom. Still, it is better to add the midside nodes if possible. This is done by setting ELCV = 1 in the NXSTRAT entry.

- In addition to the displacement-based and mixed-interpolated elements, Advanced Nonlinear Solution also includes the possibility of including incompatible modes (bubble functions) in the formulation of the 5-node pyramid, 6-node wedge and the 8-node brick element. Within this element, additional displacement degrees of freedom are introduced. These additional displacement degrees of freedom are not associated with nodes; therefore the condition of displacement compatibility between adjacent elements is not satisfied in general. The addition of the incompatible modes (bubble functions) increases the flexibility of the element, especially in bending situations. For theoretical considerations, see reference KJB, Section 4.4.1. Note that these incompatible-mode elements are formulated to pass the patch test. Also note that element distortions deteriorate the element performance when incompatible modes are used.

The incompatible modes feature is activated by setting IN=BUBBLE in the PSOLID entry.

The incompatible modes feature should be used with caution when using large displacement/small strain or large displacement/ large strain kinematics, in conjunction with large aspect ratio elements, because meshes of incompatible modes elements can contain spurious modes under these conditions.

The incompatible modes feature cannot be used in conjunction with the mixed-interpolation formulation.

- Table 2.5-3 shows which elements support incompatible modes (bubble functions). The incompatible modes feature is only available for the 5-node pyramid, 6- node wedge and the 8-node brick elements.

Table 2.5-3: Incompatible modes (bubble functions) available for 3-D solid elements

3-D solid element	Support for incompatible modes
4-node tetrahedron	No
5- to 11-node tetrahedron	No
6-node wedge	Yes
15-node, 21-node wedge	No
8-node brick	Yes
20-node, 27-node brick	No
5-node pyramid	Yes
6- to 14-node pyramid	No

- The interpolation functions used for 3-D brick solid elements for $q \leq 20$ are shown in Fig. 5.5, ref. KJB, p. 345 (note that ref KJB uses a different local node numbering convention).

- The 10-node tetrahedron (see Fig. 2.5-1(c)) is obtained by collapsing nodes and sides of rectangular elements. Spatially isotropic 10-node and 11-node tetrahedra are used in Solution 601.

  The 4-node tetrahedron (see Fig. 2.5-1(c)) is obtained by collapsing nodes and sides of the 8-node brick element. This element exhibits constant strain conditions.

- The stresses/strains can be output either at the center and corner grid points (PSOLID STRESS=blank or GRID), or at the center and corner Gauss points (PSOLID STRESS=1 or GAUSS).

2.5.2 Material models and nonlinear formulations

- See Tables 2-2 and 2-3 for a list of the material models that are compatible with 3-D solid elements.

- Advanced Nonlinear Solution automatically uses the mixed interpolation formulation for hyperelastic materials. The mixed formulation is also recommended for elastic-plastic materials and elastic materials with a Poisson ratio close to 0.5. It can be activated by setting UPFORM = 1 in the NXSTRAT entry.

- The 3-D elements can be used with

  - **small displacement/small strain** kinematics,

  - **large displacement/small strain** kinematics, or

  - **large displacement/large strain** kinematics.

  ▸ The small displacement/small strain and large displacement/small strain kinematics can be used with any of the compatible material models, except for the hyperelastic material. The use of a linear material with small displacement/small strain kinematics corresponds to a linear formulation, and the use of a nonlinear material with the small displacement/small strain kinematics corresponds to a materially-nonlinear-only formulation.

- ▸ The program uses the TL (total Lagrangian) formulation when large displacement/small strain kinematics is selected.
- ▸ The large displacement/large strain kinematics can be used with plastic materials including thermal and creep effects, as well as hyperelastic materials. The ULH (updated Lagrangian Hencky) formulation or the ULJ (updated Lagrangian Jaumann) formulation can be used for all compatible material models except the hyperelastic material. For the hyperelastic material models, a TL (total Lagrangian) formulation is used. The ULFORM parameter in the NXSTRAT entry determines the ULH/ULJ setting.

ref. KJB Sections 6.2 and 6.3.5

- The basic continuum mechanics formulations are described in ref. KJB, pp. 497-568. The finite element discretization is summarized in Table 6.6, p. 555, ref. KJB.

ref. KJB Section 6.8.1

- Note that all these formulations can be used in the same finite element model. If the elements are initially compatible, they will remain compatible throughout the analysis.

2.5.3 Numerical integration

ref. KJB Sections 5.5.3, 5.5.4 and 5.5.5

- The 8-node brick element uses 2×2×2 Gauss integration for the calculation of element matrices. The 20-node and 27-node elements use 3×3×3 Gauss integration.

- Tetrahedral elements are spatially isotropic with respect to integration point locations and interpolation functions. By default, for the 4-node tetrahedral element, 1-point Gauss integration is used, for the 10-node tetrahedral element, 5-point Gauss integration is used, and 17-point Gauss integration is also used for the 11-node tetrahedral element. The Gauss integration order for tetrahedral elements can also be chosen using TETINT in the NXSTRAT entry.

- Note that in geometrically nonlinear analysis, the spatial positions of the Gauss integration points change continuously as the element undergoes deformations, but throughout the response the same material particles are at the integration points.

- The same integration order is used for both Solution 601 and 701.

2.5.4 Mass matrices

- The consistent mass matrix is always calculated using 3×3×3 Gauss integration except for the tetrahedral 4-node, 10-node and 11-node elements which use a 17-point Gauss integration.

- The lumped mass matrix of an element is formed by dividing the element's mass *M* equally among each of its *n* nodal points. Hence the mass assigned to each node is M / n. No special distributory concepts are employed to distinguish between corner and midside nodes, or to account for element distortion.

- The same lumped matrix is used for both Solution 601 and Solution 701.

2.5.5 Heat transfer capabilities

- Heat transfer capabilities are available for all 3-D solid elements.

- One temperature degree of freedom is present at each node.

- The element matrices are integrated numerically by Gauss integration using the same integration order as the structural matrices.

- The heat capacity matrix can be calculated based on a lumped or consistent heat capacity assumption.

- The lumped heat capacity matrix of an element is formed by dividing the element's total heat capacity *C* equally among each of its *n* nodal points. Hence the mass assigned to each node is C / n. No special distributory concepts are employed to distinguish between corner and midside nodes, or to account for element distortion.

2.5.6 Recommendations on use of elements

- The linear interpolation elements (4- to 8-node) usually perform better in contact problems.

- The linear interpolation elements (5-node, 6-node and 8-node brick elements, without incompatible modes) should only be used in analyses when bending effects are not dominant. If bending effects are insignificant, it is usually best to not use incompatible modes.

- Since the 4-node tetrahedron is a constant strain element, many elements (fine meshes) must usually be used in analyses.

- For nearly incompressible elastic materials, elasto-plastic materials and creep materials, the use of the u/p mixed formulation elements is recommended.

- When the structure to be modeled has a dimension which is extremely small compared with the others, e.g., thin plates and shells, the use of the 3-D solid element usually results in too stiff a model and a poor conditioning of the stiffness matrix. In this case, the use of shell elements, particularly the 4-node shell element (see Section 2.3), is more effective.

Recommendations specific to Solution 601

ref. KJB Page 383

- The 27-node element is the most accurate among all available elements. However, the use of this element can be costly.

- The 20-node element is usually the most effective, especially if the element is rectangular (undistorted).

2.6 Scalar elements – Springs, masses and dampers

2.6.1 CELAS1, CELAS2, CMASS1, CMASS2, CDAMP1, CDAMP2

- Scalar elements in Advanced Nonlinear Solution either connect 2 degrees of freedom together or just a single degree of freedom to the ground. There are three forms of scalar elements: springs, masses, and dampers.

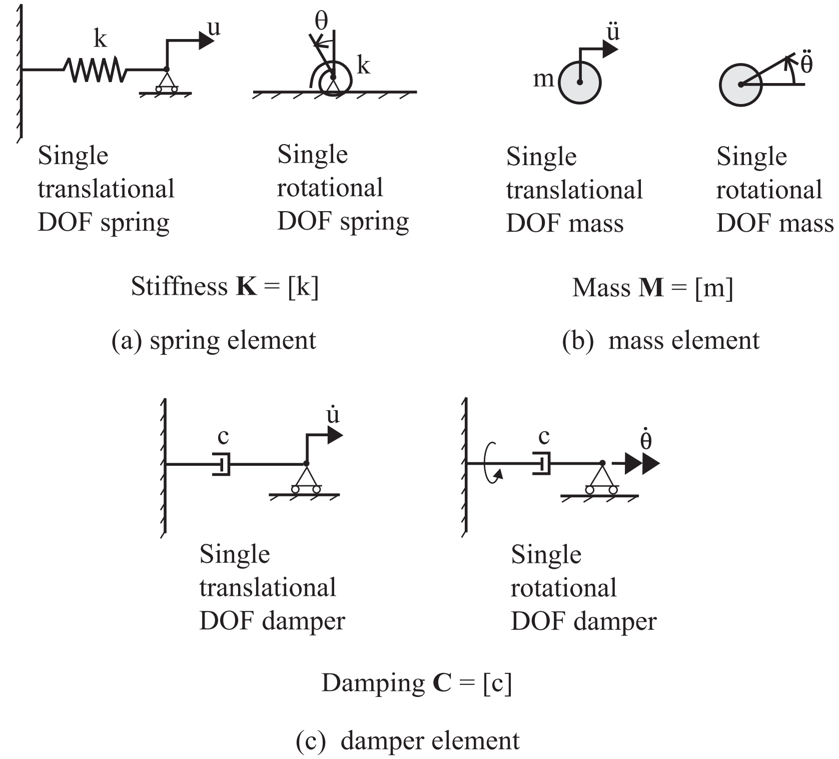
  - Spring elements are defined using the CELAS1 and CELAS2 element connectivity entries.
  - Mass elements are defined using the CMASS1 and CMASS2 element connectivity entries.
  - Damper elements are defined using the CDAMP1 and CDAMP2 element connectivity entries.

- Fig. 2.6-1 shows the spring, mass and damper single degree of freedom elements available in Advanced Nonlinear Solution. They correspond to a grounded spring, a concentrated mass, and a grounded damper, respectively.

- Fig. 2.6-2 shows the available scalar elements connecting two degrees of freedom. Only the translational version of the spring and damper are shown in the figure, but they can connect rotational degrees of freedom as well.

2.6.2 6-DOF spring element (Solution 601 only)

- The 6-DOF spring element is a generalized spring-damper element which can be linear or materially-nonlinear only (MNO). This element is defined using the CBUSH element connectivity entries. It can have single node, two coincident or two non-coincident nodes. In each degree of freedom, the element stiffness can be defined as a constant or using a force-displacement curve in the element coordinate system. The damping coefficients are always constants in units of force per unit velocity.



Single translational DOF spring

Single rotational DOF spring

Single translational DOF mass

Single rotational DOF mass

Stiffness **K** = [k]

(a) spring element

Mass **M** = [m]

(b) mass element

Single translational DOF damper

Single rotational DOF damper

Damping **C** = [c]

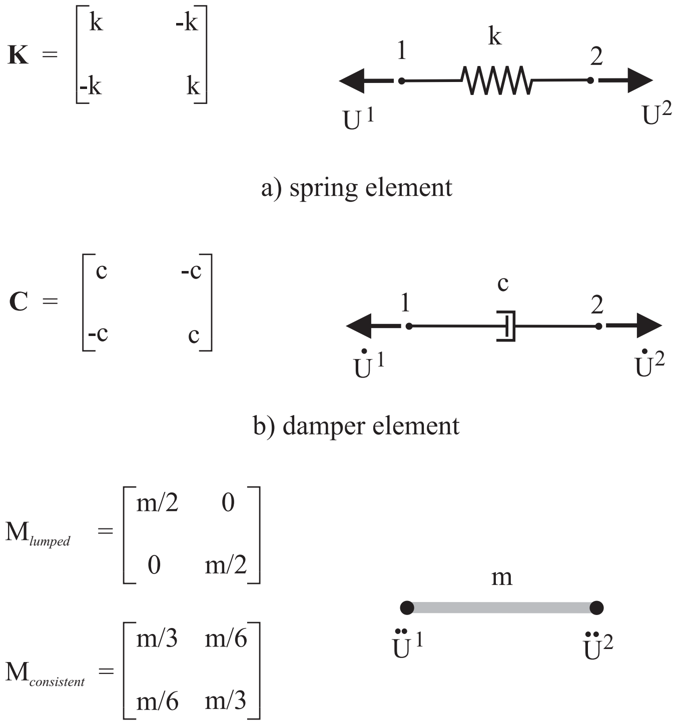
(c) damper element

Fig. 2.6-1: Single degree of freedom scalar elements

- A displacement (skew) system can be used in the 6-DOF spring element to prescribe loads and constraints. Element birth/death is also supported.

- Currently, the 6-DOF spring element is not supported in Solution 701.

- If a 6-DOF spring element has single node or two coincident nodes, its element coordinate system must be defined using a CID as shown in Fig. 2.6-3 and Fig. 2.6-4. A single node 6-DOF spring element corresponds to a grounded spring acting in the user-specified degree of freedom.



$$\mathbf{K} = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix}$$

a) spring element

$$\mathbf{C} = \begin{bmatrix} c & -c \\ -c & c \end{bmatrix}$$

b) damper element

$$M_{lumped} = \begin{bmatrix} m/2 & 0 \\ 0 & m/2 \end{bmatrix}$$

$$M_{consistent} = \begin{bmatrix} m/3 & m/6 \\ m/6 & m/3 \end{bmatrix}$$

c) mass element

Fig. 2.6-2: Two-degrees-of-freedom scalar elements

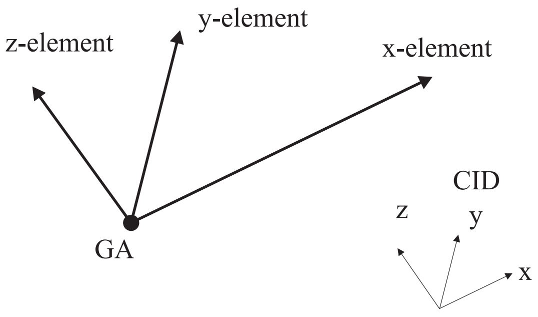


Fig. 2.6-3: 6-DOF spring element with single node

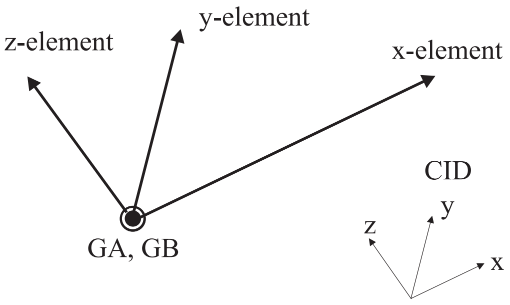


Fig. 2.6-4: 6-DOF spring element with two coincident nodes

- If a 6-DOF spring element has two non-coincident nodes, its element coordinate system can be defined using a CID, an orientation vector or its axial direction as shown in Figs. 2.6-5 to 2.6-7.

In Fig. 2.6-5, the element coordinate system is defined by a CID. Note that GA and GB might or might not have displacement (skew) coordinate systems.

`In Fig. 2.6-6, the element coordinate system is defined by an orientation vector using GO or X1, X2, X3. Note that X1, X2, X3 refers to the displacement (skew) coordinate system of GA.

In Fig. 2.6-7, a 6-DOF spring element is defined with two non-coincident nodes without GO, X1, X2, X3 or CID. This defines a 1-D axial/torsional spring/damper. In this case, axial stiffness (or damping) or torsional stiffness (or damping) or both must be specified but all other stiffness (or damping) must not be specified.

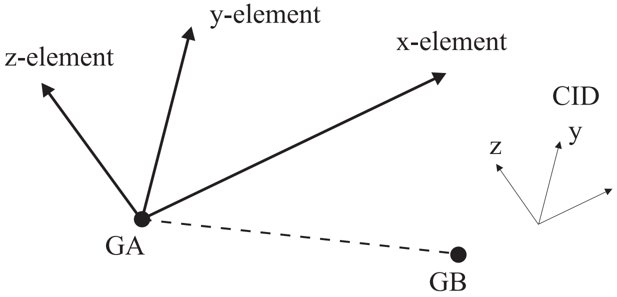


Fig. 2.6-5: 6-DOF spring element with two non-coincident nodes

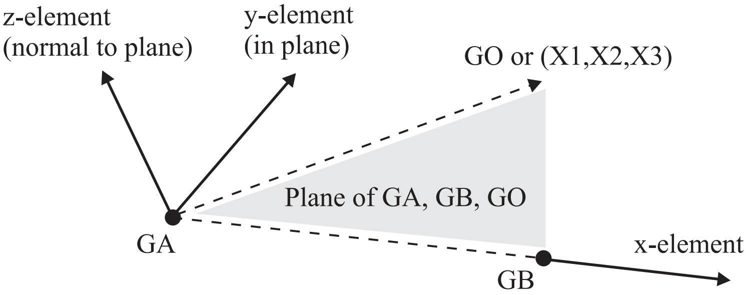


Fig. 2.6-6: 6-DOF spring element with two non-coincident nodes

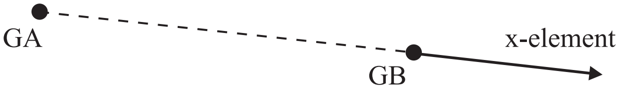


Fig. 2.6-7: 1-D spring element with two non-coincident nodes

2.7 R-type elements

- R-type elements impose multipoint constraints on one or more nodes. The constraints are created automatically by the program based on the element's input. The following R-type elements are supported in Advanced Nonlinear Solution: RBAR, RBE2 and RBE3.

- Rigid elements are a subset of R-type elements that include RBAR and RBE2.

- RBE3 is an interpolation constraint element which also produces constraint equations.

2.7.1 Rigid elements

- Solution 601 provides several options for modeling the Rigid elements. They can be modeled as perfectly rigid elements using constraint equations or as flexible (but stiff) elements. The EQRBAR or EQRBE2 parameters in the NXSTRAT entry determine how the Rigid elements are treated.

- Solution 701 does not support the flexible option.

- The RBAR entry generates a single Rigid element between two nodes.

- The RBE2 entry generates multiple Rigid elements. They connect one independent node to several nodes.

- If the perfectly rigid option is selected, Rigid elements are internally represented either as standard multipoint constraints, or as rigid links (see Section 5.8 for enforcement of constraint equations). Multipoint constraints have constant constraint coefficients and therefore do not give accurate results in large displacements (unless the 2 nodes are coincident or the constraints do not involve rotational degrees of freedom). Rigid links also create multipoint constraints but with variable coefficients that are updated based on the deformation of the structure. This is illustrated in Fig. 2.7-1. Therefore, whenever possible, large-displacement rigid links are used.

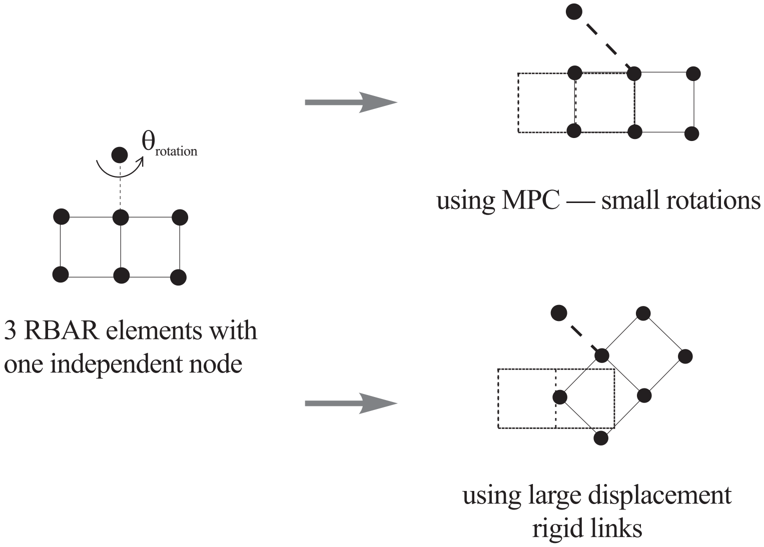

θ rotation
3 RBAR elements with one independent node
using MPC — small rotations
using large displacement rigid links

Fig. 2.7-1: Difference between small displacement MPC and large displacement rigid links

- Rigid elements that are internally represented as multipoint constraints are affected by the general constraint setting (GENMPC parameter in NXSTRAT). If constraints are set to general constraints (GENMPC=1), the constraint is enforced using Lagrange multipliers. Rigid elements represented by rigid links (which have variable constraint coefficients) are not influenced by the general constraint flag. They are always enforced using the default master-slave constraint approach.

- If the flexible option is selected for Rigid elements, Solution 601 internally generates beam or spring elements depending on the Rigid element parameters and the distance between the nodes (RBLCRIT parameter in NXSTRAT), or a spring element translation can be always requested (in the EQRBAR parameter in NXSTRAT).

- The stiffness of the internal springs and the Young’s modulus and cross-sectional area of the internal beams can be automatically determined by Solution 601 or set by the user (see SPRINGK, BEAME and BEAMA parameters in NXSTRAT entry).

- The rigid option results in more accurate enforcement of the constraint. However, the compliance introduced in the model when using the flexible option can lead to easier convergence in nonlinear problems.

- The flexible option results in none of the degrees of freedom becoming dependent. This allows multiple constraints to be defined at a node, and it is sometimes beneficial for contact.

- A dependent degree of freedom of a constraint (standard, not general constraint) or rigid link cannot be used in another constraint or rigid link as an independent degree of freedom. Hence, chaining of constraints is not allowed. Chaining of rigid links is enabled by internally replacing the dependent node of each rigid link by the first node in the chain (to avoid the restriction mentioned above).

Classification of Rigid elements

- The internal representation of an RBAR rigid element depends on the options present in CNA, CNB, CMA and CMB, as shown in Fig. 2.7-2.

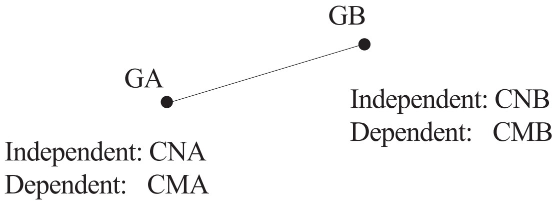


Fig. 2.7-2: Relevant parameters in the RBAR rigid element

- Currently, Advanced Nonlinear Solution identifies 5 classes of RBAR settings. Each class gets a different internal representation. Checking for each class is done in sequence starting with Class 1.

Class 1:
All 6 degrees of freedom of one point are dependent on those of the other point. In other words,

CNA = 123456, CNB = 0, CMA = 0, CMB = 123456 or

CNA = 0, CNB = 123456, CMA = 123456, CMB = 0

Class 2:
One point has all the dependent degrees of freedom (but not all 6 of them), and all those that are not dependent (missing terms in CMA or CMB) involve degrees of freedom that do not exist at the slave node. For example,

CNA = 123456, CNB = 0, CMA = 0, CMB = 123

where node B is attached only to 3D solid elements (so degrees of freedom 456 do not exist).

Another example,

CNA = 0, CNB = 123456, CMA = 12, CMB = 0

where node A is attached only to 2D solid elements (so degrees of freedom 3456 do not exist)

Note that this only applies to non-existent degrees of freedom (not fixed ones). If an excluded DOF is fixed then the rigid element does not belong to this Class.

Class 3:
One point has all the dependent degrees of freedom (but not all 6 of them). In other words,

CNA = 123456, CNB = 0, CMA = 0, CMB = Q

or
CNA = 0, CNB = 123456, CMA = Q, CMB = 0

where Q is any combination of the 6 DOFs except “0” and “123456” (“0” is not allowed, and “123456” belongs to Class 1). Note that if the degrees of freedom not included in Q are all non-existent at the node, then the rigid element belongs to Class 2.

Class 4:
All 6 degrees of freedom active but not all dependent degrees of freedom belong to 1 point. For example,

CNA	CNB	CMA	CMB
123	456	0	0
12346	5	5	12346

Class 5:
Not all the 6 degrees of freedom are active in the constraint and rigid element fails criteria for Classes 2 and 3. For example,

CNA	CNB	CMA	CMB
123	456	4	3

Note that there are some other valid settings for RBAR that are not supported in Advanced Nonlinear Solution.

- The internal representation of Rigid elements for each class is described in Table 2.7-1.

- RBE2 is interpreted in the same manner as RBAR except that it produces multiple Rigid elements. These elements can only belong to Class 1 or 3, and their internal representation is dictated by the EQRBE2 parameter in NXSTRAT.

Table 2.7-1: Internal representation of Rigid elements

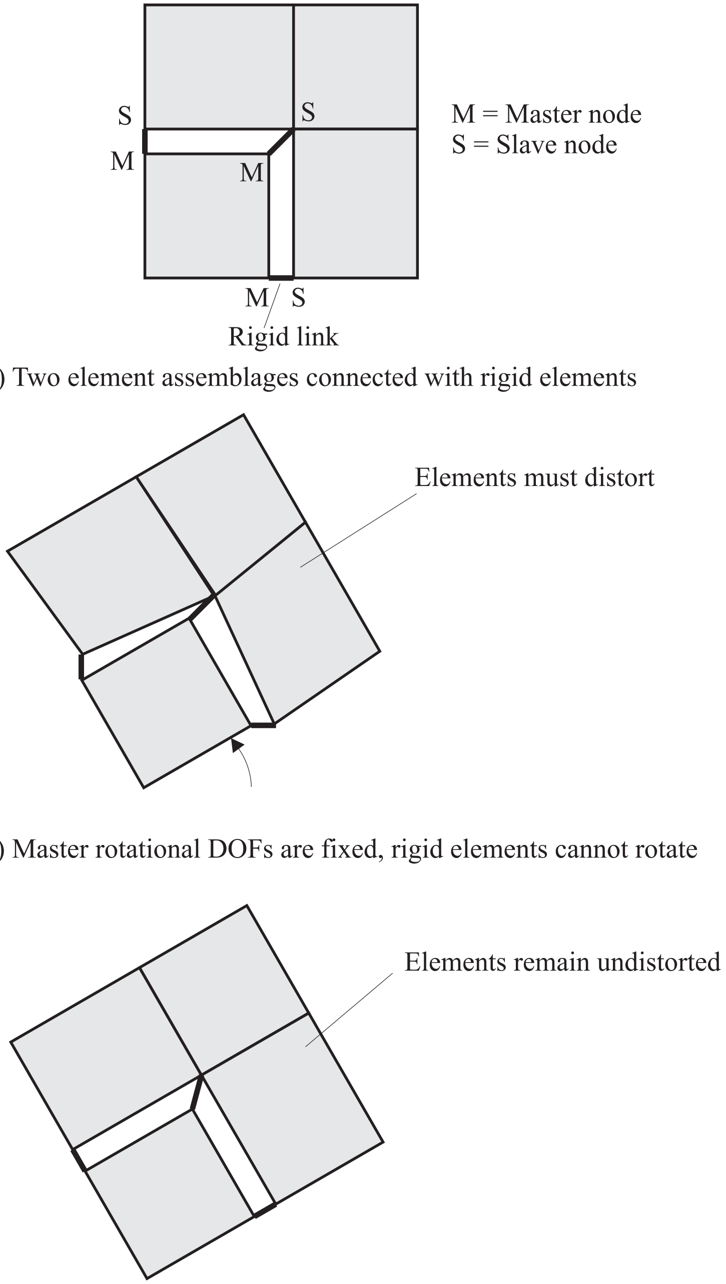
	Rigid option		Flexible option	
	$L < L_{crit}$	$L > L_{crit}$	$L < L_{crit}$	$L > L_{crit}$
Class 1	MPC	Rigid link[1]	Springs	Beam[1]
Class 2	MPC	Rigid link[1]	Springs	Beam[1]
Class 3	MPC	Rigid link[1]	Springs	Springs
Class 4	MPC	MPC	Springs	Beam[1]
Class 5	MPC	MPC	Springs	Springs

[1] This constraint is accurate in large displacement analysis

Rigid elements and continuum elements: Care must be used when connecting continuum elements (2-D and 3-D solid elements) with rigid elements. In order for the rigid element to rotate, the master DOF of the rigid element must have its rotational DOFs free.

This situation is illustrated in Fig. 2.7-3. Two assemblages of 2-D solid elements are connected together with large displacement rigid elements. If the z rotational DOFs are fixed, then when the lower element is rotated, the rigid elements maintain both their length and their angle, and the upper element assemblage distorts as shown in Fig 2.7-3(b). If the z rotational DOFs are free, then when the lower element is rotated, the rigid elements are free to rotate and the upper element assemblage can also rotate without distortion as shown in Fig 2.7-3(c).

Solution 601 does not fix the rotational DOFs of all nodes connected to continuum elements. Thus, the model in Fig. 2.7-3 will behave as shown in Fig 2.7-3(c), as expected.



a) Two element assemblages connected with rigid elements

b) Master rotational DOFs are fixed, rigid elements cannot rotate

c) Master rotational DOFs are free, rigid elements can rotate

Fig. 2.7-3: Rigid elements used with continuum elements

2.7.2 RBE3 element

- The RBE3 R-type element defines the motion of a reference node as a weighted average of the motion of a set of other nodes. This element is a useful tool for distributing applied load and mass in a model.

- The reference node is denoted REFGRID, and only the components REFC of this node are connected to the RBE3 element. The other nodes in the RBE3 element are denoted Gi,j. The double indexing i,j is used so that weights WTi and components Ci can be associated with sets of nodes Gi,j. Only the components Ci of nodes Gi,j are connected to the RBE3 element.

- The element is implemented as a set of constraint equations in which the motions of the reference node REFGRID are constrained to the motions of the other nodes Gi,j.

- In the following, we briefly outline the derivation of the RBE3 element. This derivation is based upon TAN 4494 "Mathematical Specification of the RBE3 Element". For simplicity, we consider just the case in which all components are given in the basic coordinate system, but the RBE3 element allows for the components of the reference node, and for the components of each of the connected nodes, to all be in a different coordinate system.

Let superscript q denote the reference node REFGRID and superscript k denote one of the connected nodes Gi,j, with associated weight WTi.

The derivation is based on the transmission of forces/moments from the connected nodes k to the reference node q, and on the transmission of forces/moments from the reference node q to the connected nodes k.

Firstly, if the forces/moments at a given connected node k are known, then these forces are transmitted to reference node q using the equilibrium equation

$$\begin{bmatrix} F_x^q \\ F_y^q \\ F_z^q \\ M_x^q \\ M_y^q \\ M_z^q \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -(z^k - z^q) & (y^k - y^q) & 1 & 0 & 0 \\ (z^k - z^q) & 0 & -(x^k - x^q) & 0 & 1 & 0 \\ -(y^k - y^q) & (x^k - x^q) & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_x^k \\ F_y^k \\ F_z^k \\ M_x^k \\ M_y^k \\ M_z^k \end{bmatrix}$$

or, in matrix form,

$$\mathbf{F}^q = (\mathbf{S}^k)^T \mathbf{F}^k \qquad (2.7\text{-}1)$$

where $\mathbf{F}^q$ and $\mathbf{F}^k$ are column vectors containing the forces and moments, and $\mathbf{S}^k$ is the transpose of the square matrix in the above equation.

Therefore if the forces/moments at all connected nodes k are known, then the forces/moments at reference node q is found using

$$\mathbf{F}^q = \sum_k (\mathbf{S}^k)^T \mathbf{F}^k \qquad (2.7\text{-}2)$$

Secondly, we now postulate that, if the forces/moments at reference node q are known, that these forces/moments are transmitted to connected node k using

$$\mathbf{F}^k = \mathbf{W}^k \mathbf{S}^k \mathbf{X} \mathbf{F}^q \qquad (2.7\text{-}3)$$

where $\mathbf{W}^k = \begin{bmatrix} w_1^k &&&&& \\ & w_2^k &&&& \\ && w_3^k &&& \\ &&& w_4^k && \\ &&&& w_5^k & \\ &&&&& w_6^k \end{bmatrix}$ is a diagonal

matrix with weighting factors given by WTi and Ci as follows:

```
For each component l, l=1,2,3,4,5,6, then
    If component l is not in the Ci list, then
        $w_l^k = 0$
    else
        if l=1,2,3, then
            $w_l^k = \mathrm{WTi}$
        else
            $w_l^k = \mathrm{WTi} \bullet L_c^2$
        endif
    endif
end
```

In this calculation, L_c is the average distance between the reference node and all of the connected nodes. If this average distance is zero, then $L_c = 1$. L_c is introduced into the weighting matrix so that the element has dimensional independence.

Matrix $\mathbf{X}$ is a 6x6 matrix that is the same for all of the connected nodes. Matrix $\mathbf{X}$ is determined as follows. Combining (2.7-2) and (2.7-3) gives

$$\mathbf{F}^q = \sum_k (\mathbf{S}^k)^T \mathbf{W}^k \mathbf{S}^k \mathbf{X} \mathbf{F}^q \qquad (2.7\text{-}4)$$

and for this to be satisfied for all $\mathbf{F}^q$,

$$\mathbf{X} = \left(\sum_k (\mathbf{S}^k)^T \mathbf{W}^k \mathbf{S}^k \right)^{-1} \qquad (2.7\text{-}5)$$

Now let

$$\mathbf{G}^k = \mathbf{W}^k \mathbf{S}^k \mathbf{X} \qquad (2.7\text{-}6)$$

From (2.7-3),

$$\mathbf{F}^k = \mathbf{G}^k \mathbf{F}^q \qquad (2.7\text{-}7)$$

Finally, we apply the principle of virtual work to the RBE3 element:

$$\begin{aligned}(\mathbf{F}^q)^T \delta\mathbf{u}^q &= \sum_k (\mathbf{F}^k)^T \delta\mathbf{u}^k \\ &= \sum_k (\mathbf{F}^q)^T (\mathbf{G}^k)^T \delta\mathbf{u}^k\end{aligned} \qquad (2.7\text{-}8)$$

(in which $\delta\mathbf{u}$ are the virtual displacements/rotations), and for (2.7-8) to be satisfied for all $\mathbf{F}^q$,

$$\delta\mathbf{u}^q = \sum_k (\mathbf{G}^k)^T \delta\mathbf{u}^k \qquad (2.7\text{-}9)$$

(2.7-9) shows that the virtual displacements/rotations of the reference node must be related to the virtual displacements/ rotations of the connected nodes. This can only occur if the actual displacements/rotations of the reference node are related to the actual displacements/rotations of the connected nodes:

$$\mathbf{u}^q = \sum_k (\mathbf{G}^k)^T \mathbf{u}^k \qquad (2.7\text{-}10)$$

(2.7-10) has six rows. Each row *l* selected by a component in REFC corresponds to a multipoint constraint equation, in which u_l^q is dependent and $\mathbf{u}^k$ is independent. Rows not selected by a component in REFC are discarded.

- The RBE3 element assumes that the displacements of the reference and connected nodes are small. This is seen in the definition of matrix $\mathbf{S}^k$, since the original coordinates of the nodes are used in this matrix.

- If a point mass is attached to the reference node, and the mass matrix is consistent, the point mass will be coupled to all connected nodes. In the worst case, when the reference node is connected to all nodes in the model, the mass matrix will be full, leading to a very slow solution and high memory usage. To avoid this, use a lumped mass matrix.

- Similarly, if a nonzero stiffness is attached to the reference node, the stiffness will be coupled to all connected nodes.

2.8 Potential-based fluid elements (Solution 601,106 only)

- The elements discussed in this section incorporate the following assumptions:

  - Inviscid, irrotational medium with no heat transfer
  - Compressible or almost incompressible medium
  - Relatively small displacements of the fluid boundary

- The potential-based fluid elements can be used in static analyses, where the pressure distribution in the fluid and the displacement and stress distribution in the structure is of interest.

- The potential-based fluid elements can be employed in 2-D and 3-D analyses. Two-dimensional elements can be employed in planar and axisymmetric analyses. Two-dimensional elements must be defined in the XZ plane, and axisymmetric elements must lie in the +X half plane (all nodal point coordinates must have non-negative x values).

- The potential-based fluid elements can be coupled with structural elements, as described in detail below. The structural motions cause fluid pressure, and the fluid pressure causes additional forces to act on the structure.

- The potential-based fluid elements can be coupled to a pressure boundary condition (i.e., no structure adjacent to the potential-based fluid element boundary). This feature can be used to model free surfaces.

2.8.1 Theory

Fluid

Fig. 2.8-1 shows a generic fluid region. It is assumed that the fluid is inviscid and irrotational with no heat transfer, and that the velocities and density changes are infinitesimally small. Under these assumptions, the continuity equation in the fluid is

$$\dot{\rho} + \nabla \bullet (\rho \nabla \phi) \approx \dot{\rho} + \rho_0 \nabla^2 \phi \approx \frac{\rho_0 \dot{p}}{\kappa} + \rho_0 \nabla^2 \phi = 0 \tag{2.8-1}$$

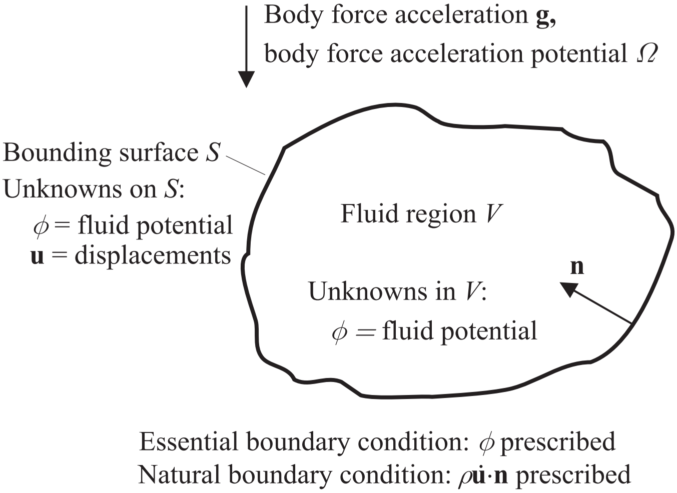
the momentum/equilibrium equation is

$$h \approx \frac{p}{\rho} \approx \Omega(\mathbf{x}) - \dot{\phi} \tag{2.8-2}$$

and the pressure-density relationship is

$$\frac{\rho}{\rho_0} = 1 + \frac{p}{\kappa} \tag{2.8-3}$$

where ρ is the density, ϕ is the velocity potential ($\mathbf{v} = \nabla \phi$ where $\mathbf{v}$ is the fluid velocity), h is the specific enthalpy (defined as $h = \int \frac{dp}{\rho}$), p is the pressure, κ is the bulk modulus and ρ_0 is the nominal density. Also $\Omega(\mathbf{x})$ is the potential of the (conservative) body force accelerations at position $\mathbf{x}$. For example, when the body forces are due to gravity, $\nabla \Omega = \mathbf{g}$, where $\mathbf{g}$ is the acceleration due to gravity.



Essential boundary condition: ϕ prescribed
Natural boundary condition: $\rho \ddot{\mathbf{u}} \cdot \mathbf{n}$ prescribed

Fig. 2.8-1: Fluid region

The momentum/equilibrium equation can also be written

$$p = \rho_0 \left(\Omega(\mathbf{x}) - \dot{\phi} \right) \qquad (2.8\text{-}4)$$

Substituting (2.8-4) into (2.8-3) gives

$$-\rho_0 \ddot{\phi} + \kappa \nabla^2 \phi = -\rho_0 \dot{\Omega} \qquad (2.8\text{-}5)$$

Equation (2.8-5) is a special form of the wave equation. It is linear in the solution variable ϕ. (2.8-5) can be written in variational form using standard techniques. The result is

$$-\int_V \rho_0 \ddot{\phi}\, \delta\phi dV - \int_V \kappa \nabla \phi \cdot \delta \nabla \phi \, dV - \int_S \kappa \, \dot{\mathbf{u}} \cdot \mathbf{n} \, \delta\phi \, dS$$
$$= -\int_V \rho_0 \dot{\Omega} \, \delta\phi \, dV \qquad (2.8\text{-}6)$$

Structure

- We assume that part of the boundary S is adjacent to the structure (Fig. 2.8-2). The part of the boundary adjacent to the structure is denoted S_1.

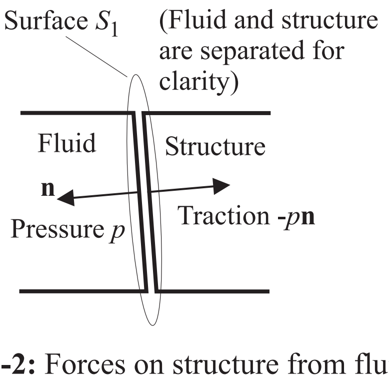


Fig. 2.8-2: Forces on structure from fluid

The fluid pressure on S_1 provides additional forces on the structure adjacent to S_1:

$$\delta F_u = \int_{S_1} p\mathbf{n}\bullet\delta\mathbf{u}\,dS_1 = \int_{S_1}\left(\rho_0\Omega + \rho_0\frac{\partial\Omega}{\partial\mathbf{x}}\bullet\mathbf{u} - \rho_0\dot{\phi}\right)\mathbf{n}\bullet\delta\mathbf{u}\,dS_1 \qquad (2.8\text{-}7)$$

The finite element contributions to the system matrices corresponding to (2.8-6) and (2.8-7) are

$$\begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & -\mathbf{M}_{\mathrm{FF}}\end{bmatrix}\begin{bmatrix}\ddot{\mathbf{U}}\\ \ddot{\boldsymbol{\phi}}\end{bmatrix} + \begin{bmatrix}\mathbf{0} & \mathbf{C}_{\mathrm{FU}}^{\mathrm{T}}\\ \mathbf{C}_{\mathrm{FU}} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{U}}\\ \dot{\boldsymbol{\phi}}\end{bmatrix} + \begin{bmatrix}\left(\mathbf{K}_{\mathrm{UU}}\right)_{\mathrm{S}} & \mathbf{0}\\ \mathbf{0} & -\mathbf{K}_{\mathrm{FF}}\end{bmatrix}\begin{bmatrix}\mathbf{U}\\ \boldsymbol{\phi}\end{bmatrix}$$
$$= \begin{bmatrix}\left(\mathbf{R}_{\mathrm{UB}}\right)_{\mathrm{S}}\\ \mathbf{0}\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ -\dot{\mathbf{R}}_{\mathrm{FB}}\end{bmatrix} \qquad (2.8\text{-}8)$$

where

$\mathbf{M}_{\mathrm{FF}}$ = matrix from $\ddot{\phi}\,\delta\phi$ term in (2.8-6)
$\mathbf{K}_{\mathrm{FF}}$ = matrix from $\nabla\phi\cdot\delta\nabla\phi$ term in (2.8-6)
$\mathbf{C}_{\mathrm{FU}}$ = matrix from $\dot{\mathbf{u}}\cdot\mathbf{n}\delta\phi$ term in (2.8-6)
$\left(\mathbf{K}_{\mathrm{UU}}\right)_{\mathrm{S}}$ = matrix from $\left(\rho_0\frac{\partial\Omega}{\partial\mathbf{x}}\bullet\mathbf{u}\right)\mathbf{n}\bullet\delta\mathbf{u}$ term in (2.8-7)
$\left(\mathbf{R}_{\mathrm{UB}}\right)_{\mathrm{S}}$ = loads vector from $\left(\rho_0\Omega\right)\mathbf{n}\bullet\delta\mathbf{u}$ term in (2.8-7)
$\dot{\mathbf{R}}_{\mathrm{FB}}$ = loads vector from $\rho_0\dot{\Omega}\,\delta\phi$ term in (2.8-6)
$\mathbf{U}$ = vector containing unknown nodal displacements
$\boldsymbol{\phi}$ = vector containing unknown nodal fluid potentials.

We note that the term $\left(\mathbf{K}_{\mathrm{UU}}\right)_{\mathrm{S}}$ is numerically very small compared with the rest of the structural stiffness matrix, when there is a structure adjacent to the fluid. But $\left(\mathbf{K}_{\mathrm{UU}}\right)_{\mathrm{S}}$ is important in the case when there is no structure adjacent to the fluid.

- The left-hand-side of equation (2.8-8) with the exception of the term $\left(\mathbf{K}_{\mathrm{UU}}\right)_{\mathrm{S}}$, is identical to the formulation presented in the following reference:

  ref. L.G. Olson and K.J. Bathe, "Analysis of fluid-structure interactions. A direct symmetric coupled formulation based on the fluid velocity potential", *J. Computers and Structures*, Vol 21, No. 1/2, pp 21-32, 1985.

Static conditions

The static equations of motion can be formally derived from (2.8-8) by applying the Laplace transform to both sides of (2.8-8) and applying the final value theorem. The result is

$$\begin{bmatrix} \left(\mathbf{K}_{\mathrm{UU}}\right)_{\mathrm{S}} & \mathbf{C}_{\mathrm{FU}}^{\mathrm{T}} \\ \mathbf{C}_{\mathrm{FU}} & -\mathbf{M}_{\mathrm{FF}} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \dot{\boldsymbol{\phi}} \end{bmatrix} = \begin{bmatrix} \left(\mathbf{R}_{\mathrm{UB}}\right)_{\mathrm{S}} \\ -\mathbf{R}_{\mathrm{FB}} \end{bmatrix} \qquad (2.8\text{-}9)$$

together with the condition

$$\mathbf{K}_{\mathrm{FF}}\boldsymbol{\phi} = \mathbf{0} \qquad (2.8\text{-}10)$$

- There are a number of unusual characteristics of (2.8-9) and (2.8-10):

  - The solution involves $\dot{\boldsymbol{\phi}}$ instead of $\boldsymbol{\phi}$. This makes sense as (2.8-4) then implies that p is constant (in time) in a static solution.
  - The condition (2.8-10) must be satisfied. This condition is satisfied whenever $\dot{\boldsymbol{\phi}} = \text{constant}$ within each separate fluid region. Hence the number of unknown potential degrees of freedom in static analysis is equal to the number of separate fluid regions in the analysis.
  - The condition $\dot{\boldsymbol{\phi}} = \text{constant}$ within each separate fluid region implies that $p = \rho_0\Omega + C$ where C is a constant determined from the solution. Hence the variation of pressure within each separate fluid region is contained

within C and any choice of constant of integration within $\rho_0\Omega$ (recall that Ω is a potential and therefore includes an arbitrary constant of integration) is balanced by an opposite change in C.

- ▸ It is necessary to enter the density of the fluid in static analysis, even when the solution does not depend on the density.
- ▸ The 2nd row of (2.8-9) represents mass conservation within each separate fluid region. Any structural motion that does not change the fluid volume of the fluid region is not given stiffness by the fluid. As examples, consider motions tangential to the fluid, and also consider a "ripple" on a free surface in which the fluid volume is not changed.

2.8.2 Elements

- The volume *V* of the fluid domain is modeled using two-dimensional or three-dimensional fluid elements. These elements are analogous to the two-dimensional or three-dimensional solid elements and the nodal point numbering of the fluid elements is the same as the nodal point numbering of the solid elements. Thus for example, CHEXA, CTETRA, CPENTA, CPYRAM entries are used to define three-dimensional fluid elements.

- The two-dimensional elements are either planar (unit thickness of fluid assumed) or axisymmetric (1 radian of fluid assumed). CPLSTNi entries are used to define planar elements and CQUADXi, CTRAXi entries are used to define axisymmetric elements. The elements must lie in the X-Z plane and the axisymmetric elements must lie in the +X half plane.

- The bounding surface *S* of the fluid domain is modeled with fluid boundaries, as discussed in detail below.

- It is required that each separate fluid domain be modeled with separate fluid element PIDs. This is because Solution 601 constrains the potential degrees of freedom of the elements with the same fluid element PIDs together in static analysis during phi model completion, step 7, see Section 2.8.5.

- It is not permitted to have fluid regions of different densities sharing the same potential degrees of freedom. This is because the nodal pressure would be different as computed from the fluid regions connected to the node.

- The fluid elements and adjacent structure elements must be compatible.

- It is recommended that the fluid and adjacent structure *not* share the coincident nodes. This allows Solution 601 to construct constraint equations between the fluid and structural degrees of freedom that are most appropriate during phi model completion, see Section 2.8.5.

- The fluid material properties are specified via the MAT10 entry.

2.8.3 Fluid boundary conditions

For ease of modeling, fluid boundary conditions of various types can be defined along the surface of the fluid domain. A fluid boundary is specified using the BFLUID entry, by referencing a BSURFS, BCPROPS or BEDGE entry.

There are several types of fluid boundary:

Fluid-structure: Place a fluid-structure fluid boundary on the boundary between a potential-based fluid and the adjacent structure.

In many cases, Solution 601 can automatically generate fluid-structure interface elements along the boundary between the fluid and structure during phi model completion, step 1, see Section 2.8.5. So fluid-structure boundaries typically need not be defined.

We emphasize that the potential-based fluid elements must be compatible with the adjacent structural elements.

It is assumed that the structure provides stiffness to all translational degrees of freedom, because the fluid-structure boundary does not provide stiffness to the translational directions that are tangential to the fluid boundary.

Free surface: Place a free surface fluid boundary on the boundary where the pressures are to be prescribed and the displacements are desired, for example, on the free surface of a fluid.

It is necessary to fix all displacements that are tangential to the free surface fluid boundary, because the free surface fluid bondary does not provide stiffness to the translational directions that are tangential to the fluid boundary.

In many cases, Solution 601 can generate skew systems and fixities corresponding to the tangential directions during phi model generation, see Section 2.8.5.

Fluid-fluid: Place a fluid-fluid boundary on the boundary between two potential-based fluid elements of two different fluid regions.

We emphasize that the fluid elements must be compatible between the two different fluid regions.

Note that only one fluid-fluid boundary need be defined for each boundary. Solution 601 generates a fluid-fluid interface element for each of the two elements that share a common boundary during phi model completion, step 1, see Section 2.8-5.

Rigid-wall: Place a rigid-wall potential interface wherever the fluid is not to flow through the boundary.

Interface elements

Each fluid boundary type has an associated interface element type.

The difference between fluid boundaries and interface elements is that a fluid boundary is specified as part of the model definition on a surface, but an interface element covers only the edge or face of a single fluid element. Thus a single fluid boundary corresponds to many interface elements.

Interface elements are automatically generated by Solution 601 as part of the "phi model completion" process.

A rigid-wall fluid boundary suppresses any automatic generation of interface elements along the rigid-wall boundary. Solution 601 uses the rigid-wall boundary during phi model completion, step 2, in constructing structural normals, see Section 2.11.15.

2.8.4 Loads

Concentrated forces, pressure loads, prescribed displacements

Concentrated forces, pressure loads and/or prescribed displacements can be applied directly to any part of the fluid boundary on which there are fluid-structure, free surface or fluid-fluid boundaries. However, when applying concentrated forces, remember that Solution 601 can apply skew systems to certain nodes on the fluid boundary during phi model completion, see Section 2.8.5. Therefore, make sure that the nodes on which you apply concentrated forces have the anticipated degree of freedom directions.

Mass-proportional loads

Mass-proportional loads applied to fluid elements are interpreted as physical body forces (and not, for example, as ground accelerations). These loads are used in the construction of Ω, and therefore these loads must be constant in time.

Centrifugal loads

Centrifugal load effects are not included in the potential-based fluid elements.

2.8.5 Phi model completion

As can be seen above, there are many restrictions and conditions that must be considered when specifying boundary conditions on potential-based fluid elements. These conditions have been automated in Solution 601 in the following way. Solution 601 performs “phi model completion” before beginning an analysis in which potential-based fluid elements are used. The steps in phi model completion are:

1) Solution 601 loops over all fluid element sides on the boundary of each fluid element region. If the fluid element side has a fluid boundary, an interface element of the appropriate type is generated. Otherwise, the side is checked to see if it is attached to a structure (shares structural degrees of freedom with structural

elements) or is close to a structure (nodes coincident with nodes of a structural element); and, if any of the above conditions are met, an interface element of the appropriate type is generated. The intent of this step is to cover as much of the fluid boundary as possible with interface elements.

2) Solution 601 loops over all nodes attached to interface elements. If the node is attached to structural elements, the node is skipped. Otherwise the types of the attached interface elements are determined. Then

a) If the node is attached only to a free surface interface or fluid-fluid interface, then the node has a free normal direction (normal to the interface) and zero stiffness directions that are tangential to the free normal. The free normal and zero stiffness directions are identified, and if they are not aligned with the global directions, a skew system is generated that is aligned with the free normal and zero stiffness directions.

b) If the node is attached to a free surface interface or fluid-fluid interface, and is also attached to a fluid-structure interface or rigid-wall interface, Solution 601 proceeds as follows. The node has a free normal direction (determined from the free surface interface or fluid-fluid interface), a structural normal direction (determined from the fluid-structure interface or rigid-wall interface), and, in 3D, another direction orthogonal to the free normal and structural normal directions, which may be a zero stiffness direction or another structural normal direction. The free normal direction is modified to be orthogonal to the structural normal directions. The free normal, structural normal and zero stiffness directions are identified, and, if they are not aligned with the global directions, a skew system is generated that is aligned with the free normal, structural normal and zero stiffness directions.

The intent of step 2 is to identify the zero stiffness and free normal directions of the nodes.

3) Solution 601 loops over all nodes attached to interface elements. If the node is attached to a structural element, the node is skipped. If the node (node A) is attached to a fluid-structure interface element and is close to a structural node B, node A is constrained to node B as follows. Each displacement degree of

freedom for node A is constrained to the corresponding degrees of freedom for node B, accounting for differences in skew systems between A and B, accounting for B possibly being a slave node in a constraint equation (but not accounting for B possibly being a slave in a rigid link), accounting for B possibly being fixed. But a displacement degree of freedom for node A is not constrained if the degree of freedom is a free normal direction (see 2 above).

The intent of step 3) is to connect the fluid mesh with the structural mesh, when different nodes are used for the fluid and structure. The connection still allows the fluid nodes to slip relative to the structural nodes on intersections between free surfaces and the structure.

4) In static analysis, when there are no body force loads, Solution 601 loops over all nodes on a free surface or fluid-fluid interface. If the node is attached to a structural element, the node is skipped. Otherwise, constraint equations are defined for all nodes so that the displacements in the direction of the free normal are equal.

The intent is to remove the zero pivots in the stiffness matrix that are otherwise present (see the discussion after equation (2.8-10)).

5) When there are body force loads, Solution 601 loops over all nodes on a fluid-fluid interface. If the node is attached to a structural element, the node is skipped. Otherwise constraint equations are defined for all pairs of nodes, so that the displacements in the direction of the free normal are compatible.

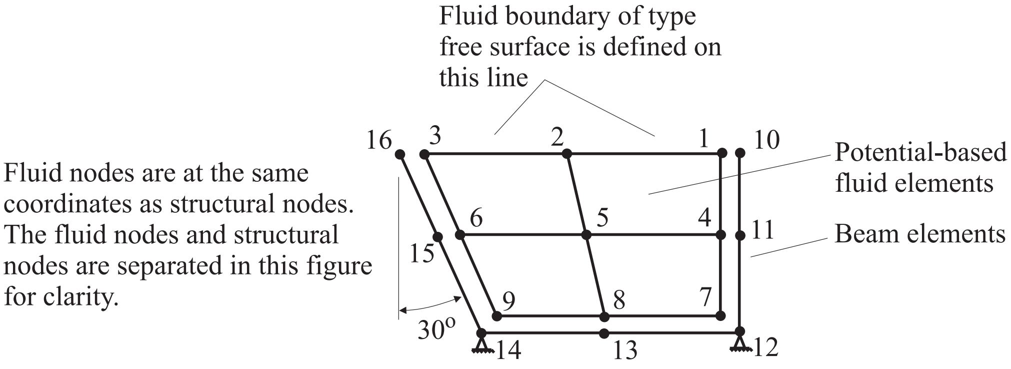
The intent is to enforce displacement compatibility between the fluids.

6) Solution 601 then loops over all nodes with zero stiffness degrees of freedom and defines fixities for each zero stiffness degree of freedom.

7) Solution 601 constrains all of the potential degrees of freedom for fluid elements with the same PID together.

Example 1: We now present a detailed example for a 2-D fluid filled basin with flexible walls.

Fig. 2.8-3(a) shows the model before phi model completion. The model is defined with separate nodes for the fluid and the structure. A fluid boundary of type free-surface is defined on the horizontal line, as shown.

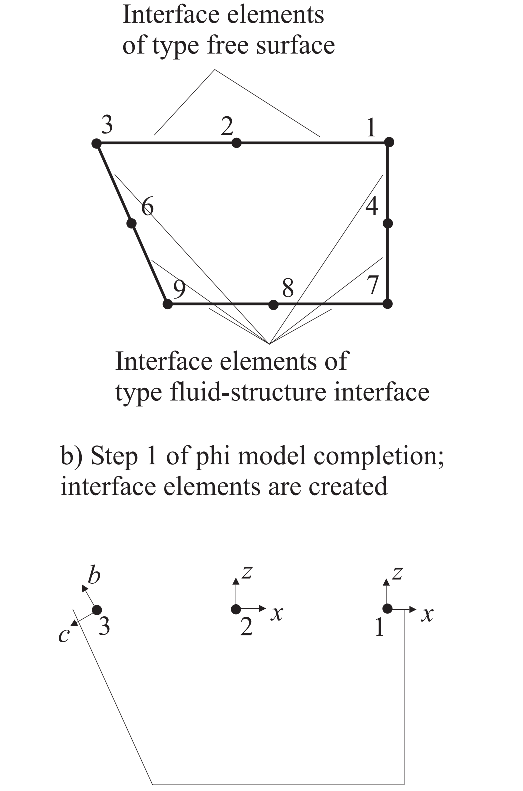

Fluid boundary of type
free surface is defined on
this line
Fluid nodes are at the same coordinates as structural nodes. The fluid nodes and structural nodes are separated in this figure for clarity.
Potential-based fluid elements
Beam elements
16
3
2
1
10
6
5
4
11
15
9
8
7
30°
14
13
12

a) Finite element model before phi model completion

Fig. 2.8-3: Example of phi model completion

In step 1 of phi model completion, Solution 601 generates fluid-structure interface elements where the fluid is adjacent to the structure, and free surface interface elements corresponding to the fluid boundary (Fig. 2.8-3(b)).

In step 2 of phi model completion, Solution 601 classifies the displacement directions on the free surface (Fig. 2.8-3(c)). Notice that the free normal for node 2 is taken from the free surface, but the free normal for node 3 is modified by the presence of the adjacent structure. A skew system is defined for node 3 because the free normal and structural normal are not aligned with the global coordinate directions. The zero stiffness direction of node 2 will be fixed in step 6 below.



b) Step 1 of phi model completion;
interface elements are created

Node 1: *x* = structural normal direction,
z = free normal direction

Node 2: *x* = zero stiffness direction,
z = free normal direction

Node 3: *b* = free normal direction,
c = structural normal direction

c) Step 2 of phi model completion: classification
of displacement directions on free surface

Fig. 2.8-3: (continued)

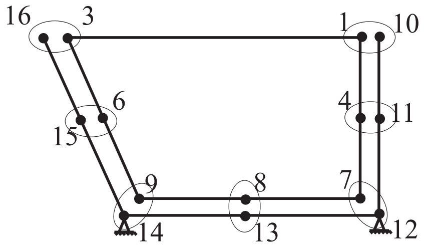
In step 3 of phi model completion, Solution 601 constrains the fluid displacement directions to the structure (Fig. 2.8-3(d)). At node 1, only the *x* displacement direction is constrained; the *z* displacement is left free so that the free surface can slip along the wall. At node 3, only the *c* displacement direction is constrained;

the *b* displacement is left free so that the free surface can slip along the wall.

Notice that at node 4, both the *x* and *z* displacements are constrained to the structure. The fluid still slips in the *z* direction because only the normal displacement (the *x* displacement in this case) is used by the fluid equations. Similar statements hold for nodes 6 and 8.

Nodes 7 and 9 are fixed because corresponding nodes 12 and 14 are fixed.

If the analysis is static without body forces, then Solution 601 performs step 4 of phi model completion (Fig. 2.8-3(e)). The free surface can only translate vertically as a rigid body.



Node 1: $u_x = u_x^{10}$

Node 3: $u_c = -\cos 30^o u_x^{16} - \sin 30^o u_z^{16}$

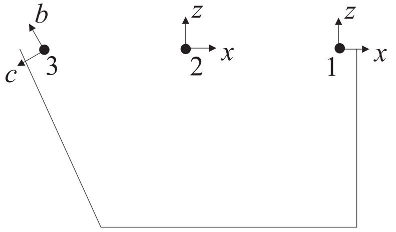
Node 4: $u_x = u_x^{11}, u_z = u_z^{11}$ Node 6: $u_x = u_x^{15}, u_z = u_z^{15}$

Node 7: $u_x = u_z =$ fixed Node 8: $u_x = u_x^{13}, u_z = u_z^{13}$

Node 9: $u_x = u_z =$ fixed

d) Step 3: Creation of constraint equations and fixities

Fig. 2.8-3: (continued)



Node 2: $u_z = u_z^1$

Node 3: $u_b = u_z^1 \; / \cos 30^{\mathrm{o}}$

Step 4 is only performed in static analysis when there are no body forces.

e) Step 4 of phi model completion: defining constraint equations to set normal displacements equal on free surface

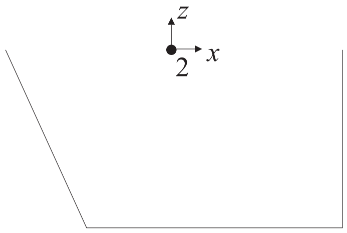
Fig. 2.8-3: (continued)

This motion affects the total volume of the fluid region, so that there is no zero pivot in the system matrices.

If there are body forces, then step 4 is not necessary because all boundary motions are given stiffness by the matrix $\left(\mathbf{K}_{\mathrm{UU}}\right)_{\mathrm{S}}$.

Step 5 of phi model completion is skipped because there are no fluid-fluid boundaries.

In step 6 of phi model completion, the zero stiffness direction at node 2 is fixed (Fig. 2.8-3(f)). Vertical motions of the nodes attached only to free surface interface elements are allowed, but horizontal motions of these nodes are not allowed (because the fluid does not provide stiffness, damping or mass to horizontal motions).

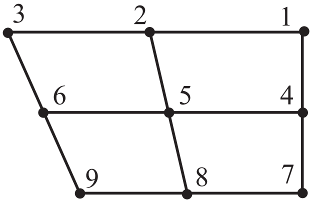


Node 2: u_x=fixed

f) Step 6 of phi model completion: defining fixities to eliminate zero stiffness degrees of freedom

Fig. 2.8-3: (continued)

Then Solution 601 performs step 7 of phi model completion (Fig. 2.8-3g). Only constant (in space) potentials are allowed in static analysis.

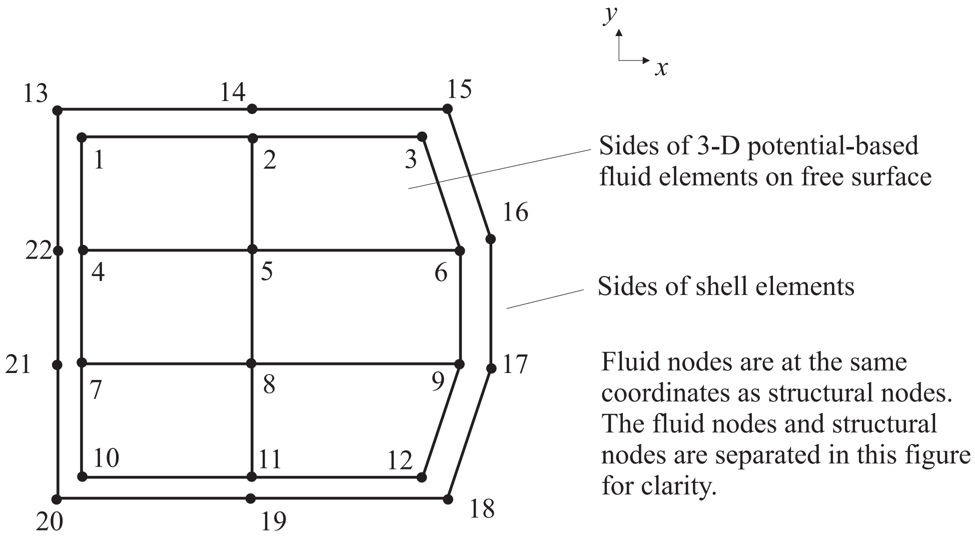


$\phi^2=\phi^1, ..., \phi^9=\phi^1$

g) Step 7: defining constraint equations to set all potential degrees of freedom equal

Fig. 2.8-3: (continued)

Example 2: In 3-D analysis of a fluid-filled basin, there is one additional consideration. Consider the model shown in Fig. 2.8-4, in which only the free surface and the adjacent structural nodes are shown.

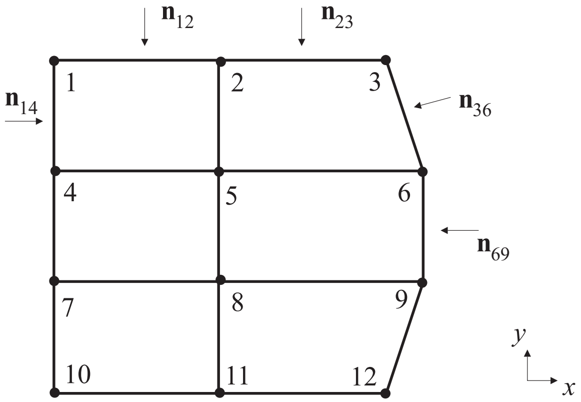

y
x
13
14
15
16
17
18
19
20
21
22
1
2
3
4
5
6
7
8
9
10
11
12
Sides of 3-D potential-based fluid elements on free surface
Sides of shell elements
Fluid nodes are at the same coordinates as structural nodes. The fluid nodes and structural nodes are separated in this figure for clarity.

a) Top view of finite element model

Fig. 2.8-4: Example 2 of phi model completion

During step 2 of phi model completion, Solution 601 determines the structural normal direction(s), zero stiffness direction(s) and free normal direction for the nodes of the free surface (Fig. 2.8-4(b)). Notice that node 3 has two structural normals, but node 6 has only one structural normal. That is because the angle between the two structural normals for node 3 is greater than 30 degrees, but the angle between the two structural normals for node 6 is less than 30 degrees.

Nodes 3, 6, 9 and 12 are assigned skew systems because the structural normal directions are not aligned with the global system.



Node 1: Structural normal 1 = $\mathbf{n}_{14}$
Structural normal 2 = $\mathbf{n}_{12}$
Free normal = $\mathbf{z}$

Node 2: Structural normal = $\mathbf{n}_{12}$ = $\mathbf{n}_{23}$
Zero stiffness direction = $\mathbf{x}$
Free normal = $\mathbf{z}$

Node 3: Structural normal 1 = $\mathbf{n}_{23}$
Structural normal 2 = $\mathbf{n}_{36}$
Free normal = $\mathbf{z}$

Node 5: Zero stiffness direction 1 = $\mathbf{x}$
Zero stiffness direction 2 = $\mathbf{y}$
Free normal = $\mathbf{z}$

Node 6: Structural normal = average of $\mathbf{n}_{36}$ and $\mathbf{n}_{69}$
Free normal = $\mathbf{z}$
Zero stiffness direction = remaining orthogonal direction

b) Classification of structural normals and zero stiffness directions for some nodes on the free surface

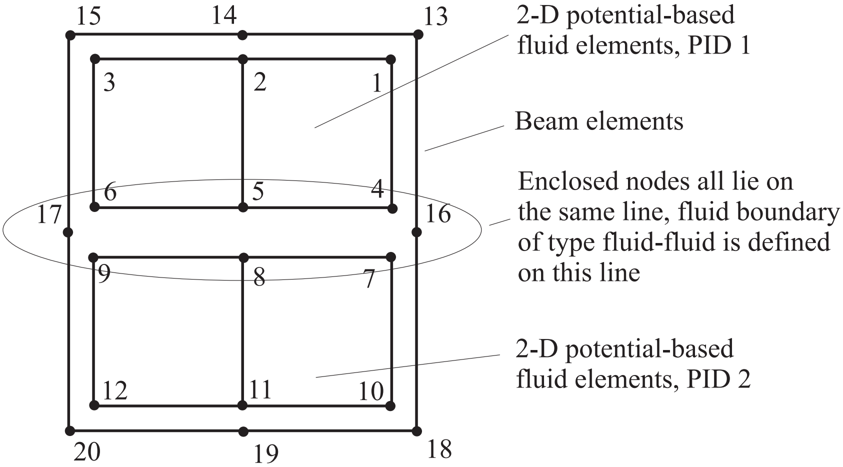
Fig. 2.8-4: (continued)

During step 3 of phi model completion, Solution 601 creates constraint equations for the fluid nodes adjacent to the structural nodes. For example, node 1 is constrained in both the x and y directions to node 13, because both directions are structural normal directions. Node 2 is also constrained in both the x and y directions to node 14, here because the y direction is a structural normal direction and the x direction is a zero stiffness direction. (It is assumed that the structure provides stiffness in the x direction.)

During step 6 of phi model completion, Solution 601 fixes the x and y directions for nodes 5 and 8, because these directions are zero stiffness directions, and there is no adjacent structure.

Example 3: We now present some of the steps for phi model completion of an enclosure with two distinct fluid regions. Fig. 2.8-5(a) shows the model before phi model completion.

Fluid nodes are at the same coordinates as structural nodes. The fluid nodes and structural nodes are separated in this figure for clarity.


15
14
13
2-D potential-based fluid elements, PID 1
3
2
1
Beam elements
6
5
4
17
16
Enclosed nodes all lie on the same line, fluid boundary of type fluid-fluid is defined on this line
9
8
7
2-D potential-based fluid elements, PID 2
12
11
10
20
19
18

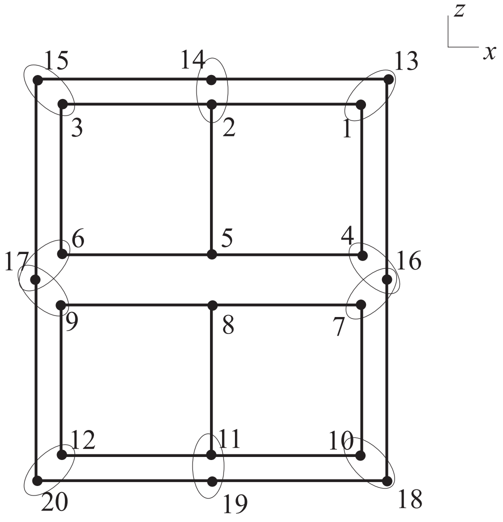
a) Finite element model before phi model completion

Fig. 2.8-5: Example 3 of phi model completion

In step 1 of phi model completion, Solution 601 generates fluid-structure interface elements where the fluid is adjacent to the structure, and fluid-fluid interface elements corresponding to the fluid boundary. Four fluid-fluid interface elements are generated, two for each shared element side.

In step 2 of phi model completion, Solution 601 classifies the displacement directions on the fluid-fluid interface. Here the free normal is always in the *z* direction and the zero stiffness directions for nodes 5 and 8 are in the *x* direction.

In step 3 of phi model completion, the fluid nodes are constrained to the adjacent structural nodes (Fig. 2.8-5(b)). Notice that the nodes on the fluid-fluid interface are allowed to slip relative to the structure.



Node 1: $u_x = u_x^{13}, u_z = u_z^{13}$ Node 2: $u_x = u_x^{14}, u_z = u_z^{14}$

Node 3: $u_x = u_x^{15}, u_z = u_z^{15}$ Node 4: $u_x = u_x^{16}$

Node 6: $u_x = u_x^{17}$ Node 7: $u_x = u_x^{16}$

Node 9: $u_x = u_x^{17}$ Node 10: $u_x = u_x^{18}, u_z = u_z^{18}$

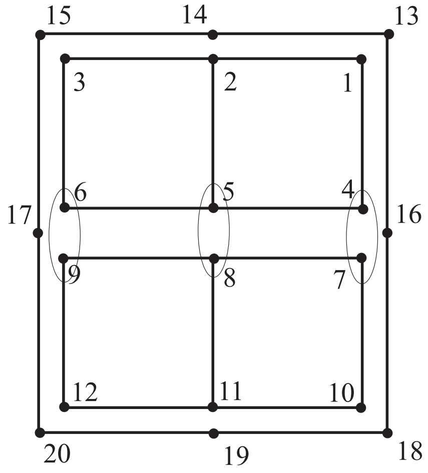
Node 11: $u_x = u_x^{19}, u_z = u_z^{19}$ Node 12: $u_x = u_x^{20}, u_z = u_z^{20}$

b) Step 3: constraining fluid nodes to adjacent structural nodes

Fig. 2.8-5: (continued)

If the analysis is static without body forces, then Solution 601 performs step 4 of phi model completion. In this case, the z displacements of nodes 5 to 9 are constrained to be equal to the z displacement of node 4. The free surface can only translate vertically as a rigid body.

In step 5 of phi model completion, the fluid nodes on the fluid-fluid interface are constrained to each other (Fig. 2.8-5(c)). Step 5 is not performed if step 4 was performed. Notice that the potential degrees of freedom are not constrained.



Node 4: $u_z = u_z^7$ Node 5: $u_z = u_z^8$

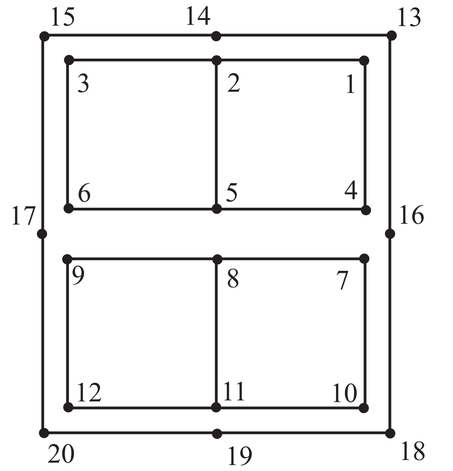
Node 6: $u_z = u_z^9$

c) Step 5: constraining adjacent nodes of the two fluid regions together

Fig. 2.8-5: (continued)

In step 6 of phi model completion, the zero stiffness direction of nodes 5 and 8 are fixed. Vertical motions of the nodes attached only to fluid-fluid interface elements are allowed, but horizontal motions are not allowed (because the fluid does not provide stiffness, damping or mass to horizontal motions).

Then Solution 601 performs step 7 of phi model completion (Fig. 2.8-5(d)). Only constant (in space) potentials are allowed in static analysis, but the potential can be different for the regions with different PIDs.



$\phi^2 = \phi^1, \ldots, \phi^6 = \phi^1 \qquad \phi^8 = \phi^7, \ldots, \phi^{12} = \phi^7$

d) Step 7: defining constraint equations to set the potential degrees of freedom of each fluid region equal

Fig. 2.8-5: (continued)

2.9 Other element types

2.9.1 Gap element

- The gap element is used in Advanced Nonlinear Solution to connect two nodes as shown in Fig. 2.9-1. Gap elements are defined using the CGAP element connectivity entry.

- The initial gap opening is U_0. When the gap is closed the element has a stiffness of K_A (should be stiff), and when it is open the stiffness is K_B (should be soft).

- The tangential behavior of the gap element represented by KT, MUI and MUZ is not supported.

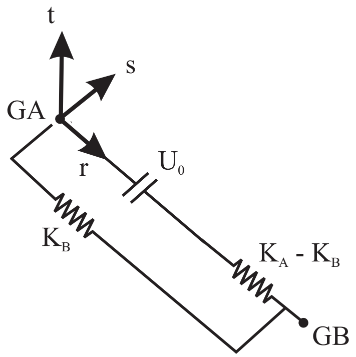


Fig. 2.9-1: CGAP element coordinate system

2.9.2 Concentrated mass element

- Advanced Nonlinear Solution supports the CONM1 and CONM2 entries for defining concentrated masses.

- For CONM1, only the diagonal mass terms are supported, and the resulting mass matrix is given by:

$$\mathbf{M} = \begin{bmatrix} M_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & M_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & M_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & M_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & M_{66} \end{bmatrix}$$

- For CONM2, the off-diagonal mass moments of inertia terms are neglected, and the resulting mass matrix is

$$\mathbf{M} = \begin{bmatrix} M & 0 & 0 & 0 & 0 & 0 \\ 0 & M & 0 & 0 & 0 & 0 \\ 0 & 0 & M & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{11} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{22} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{33} \end{bmatrix}$$

2.9.3 Bushing element

- The one-dimensional bushing element CBUSH1D is used in Advanced Nonlinear Solution to provide an axial stiffness and damping between two nodes as shown in Fig. 2.9-2.

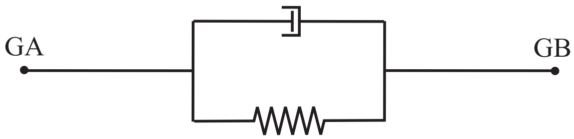


Fig. 2.9-2: BUSH1D element

- The stiffness and damping act along the axis of the element, which is the line connecting its two nodes. In large displacement analysis the element axis is updated with deformation. A fixed element axis can be specified via the CID parameter in the CBUSH1D entry.

- The element can have a constant or a nonlinear stiffness defined via a lookup table.

- Any mass assigned to the BUSH1D element is lumped at the grid points.

3. Material models and formulations

The objective of this chapter is to summarize the theoretical basis and practical use of the material models and formulations available in Advanced Nonlinear Solution.

The stress and strain measures used by different materials and formulations are first summarized in Section 3.1.

The table below lists the material models available in Advanced Nonlinear Solution, and how they can be obtained from the material entry cards. Note that Tables 2-2 and 2-3 list the acceptable combinations of elements and material properties for Solutions 601 and 701.

Table 3.1: Material models available in Advanced Nonlinear Solution

Material Entries	Advanced Nonlinear Solution material	Sol 701 availability[1]
MAT1	Elastic isotropic	✓
MAT1, CREEP or MAT1, MATCRP	Elastic-creep	
MAT1, CREEP, MATTC or MAT1, MATCRP	Thermal elastic-creep	
MAT1, MATG	Gasket	
MAT1, MATS1[2]	Elastic isotropic nonlinear[3]	✓
MAT1, MATS1[4], MATSR	Elasto-plastic	✓
MAT1, MATS1[6]	Thermal elasto-plastic	✓
MAT1, MATS1, MATT1[6]	Thermal elasto-plastic, temperature-dependent elastic properties	✓
MAT1, MATS1[5]	Thermal elasto-plastic, temperature-dependent plastic properties	✓
MAT1, MATS1, MATT1[5]	Thermal elasto-plastic, temperature-dependent elastic and plastic properties	✓

Table 3.1: Material models available in Advanced Nonlinear Solution (continued)

Material Entries	**Advanced Nonlinear Solution material**	**Sol 701 availability[1]**
MAT1, MATS1, CREEP[6]	Plastic-creep	
MAT1, MATS1, CREEP, MATT1[6]	Plastic-creep with temperature-dependent properties	
MAT1, MATS1, CREEP[5]	Thermal plastic-creep, temperature-dependent plastic properties	
MAT1, MATS1, MATT1, CREEP[5]	Thermal plastic-creep, temperature-dependent elastic and plastic properties	
MAT1, MATS1, CREEP, MATTC[6]	Thermal plastic-creep, temperature-dependent creep properties	
MAT1, MATS1, MATT1, CREEP, MATTC[6]	Thermal plastic-creep, temperature-dependent elastic and creep properties	
MAT1, MATS1, CREEP, MATTC[5]	Thermal plastic-creep, temperature-dependent plastic and creep properties	
MAT1, MATS1, MATT1, CREEP, MATTC[5]	Thermal plastic-creep, temperature-dependent elastic, plastic and creep properties	
MAT2	Elastic orthotropic (surface elements)	✓
MAT2, MATT2	Thermal elastic orthotropic (surface elements)	✓
MAT3	Elastic orthotropic (2D elements)	
MATT3, MAT3	Thermal elastic orthotropic (2D elements)	
MAT4	Isotropic heat transfer	

Table 3.1: Material models available in Advanced Nonlinear Solution (continued)

Material Entries	Advanced Nonlinear Solution material	Sol 701 availability[1]
MAT4, MATT4	Temperature dependent isotropic heat transfer	
MAT5	Orthotropic heat transfer	
MAT5, MATT5	Temperature dependent orthotropic heat transfer	
MAT8	Elastic orthotropic (surface elements)	✓
MAT8, MATT8	Thermal elastic orthotropic (surface elements)	✓
MAT9	Elastic orthotropic (solid elements)	✓
MAT9, MATT9	Thermal elastic orthotropic (solid elements)	✓
MAT10	Fluid	
MAT11	Elastic orthotropic (solid elements)	✓
MAT11, MATT11	Thermal elastic orthotropic (solid elements)	✓
MATHE, MATHEM, MATHEV	Hyperelastic (Mooney-Rivlin, Ogden, Arruda-Boyce, Sussman-Bathe and Hyperfoam)	✓[7]
MATHP	Hyperelastic (Mooney-Rivlin only)	✓
MATPLCY, PLCYISO, PLCYKIN, PLCYRUP	Plastic-cyclic	✓

Table 3.1: Material models available in Advanced Nonlinear Solution (continued)

Material Entries	Advanced Nonlinear Solution material	Sol 701 availability[1]
MATSMA	Shape memory alloy (SMA)	
MATVE	Viscoelastic	

Notes:

1. Temperature interpolation at the start of the analysis only in Solution 701.
2. With MATS1 TYPE=NELAST.
3. Cannot be used with beam element for SOL 601. Can only be used with rod element for SOL 701.
4. With MATS1 TYPE=PLASTIC.
5. With MATS1 TYPE=PLASTIC and TID pointing to a TABLEST entry.
6. With MATS1 TYPE=PLASTIC and TID pointing to a TABLES1 entry.
7. Only Mooney-Rivlin, Ogden and Sussman-Bathe hyperelastic materials are available in Solution 701.

3.1 Stress and strain measures

- It is important to recognize which stress and strain measures are employed in each material model: this is necessary in the preparation of the input data and the interpretation of the analysis results.

- This section summarizes the stress and strain measures in Advanced Nonlinear Solution and how they are used with the different element types and nonlinear features. More details on stress/strain measures are provided in ref. KJB, Section 6.2.

3.1.1 Kinematic formulations

Small displacement/small strain kinematics

Input of material parameters: All elements and material models use the engineering stress-engineering strain relationship.

Output: All elements and material models output engineering stresses and engineering strains. Note that, as long as the displacements and strains are small, Cauchy stresses and engineering stresses are nealy equal.

- Using a linear material model with small displacement/small strain kinematics results in a linear finite element formulation.

- Using a nonlinear material model with small displacement/small strain kinematics results in a materially-nonlinear only (MNO) formulation.

Large displacement/small strain kinematics

Input of material parameters: 2nd Piola-Kirchhoff stresses and Green-Lagrange strains. Note that under small strain conditions, 2nd Piola-Kirchhoff stresses are nearly equal to engineering stresses, and Green-Lagrange strains are nearly equal to engineering strains. Strains should be less than 2%.

Output: The output depends on the element type. Note that as long as the strains are small, Green-Lagrange strains are practically the same as engineering strains in the element coordinate system. Similarly, 2nd Piola-Kirchhoff stresses are practically the same as Cauchy stresses in the element coordinate system.

(1) 2-D, 3-D solid elements: all supported material models output Cauchy stresses and Green-Lagrange strains.

(2) Shell elements: all supported material models output 2nd Piola-Kirchhoff stresses and Green-Lagrange strains.

(3) Rods and beams: all supported material models output Cauchy stresses and engineering strains in the element coordinate system.

Large displacement/large strain kinematics

This kind of formulation can only be used with 2-D and 3-D solid elements and with shell elements.

For 2-D and 3-D solid elements

(1) Both the updated Lagrangian Hencky formulation and the updated Lagrangian Jaumann formulation can be used with elastic-plastic materials (including thermal and creep effects). In this case,

Input of material parameters: Cauchy (true) stresses and logarithmic (true) strains. For the multilinear stress-strain curves, it is also possible to enter engineering stress-strain data along with the input NXSTRAT CVSSVAL=1, see detailed description of the CVSSVAL feature in Section 3.4.1.

Output:

ULH formulation: Cauchy stresses and logarithmic strains in the element coordinate system.

ULJ formulation: Cauchy stresses and Jaumann strains.

(2) For hyperelastic materials a total Lagrangian formulation is used. In this case,

Input of material parameters: Hyperelastic material constants.

Output: Cauchy stresses and Green-Lagrange strains in the element coordinate system.

For shell elements

Both the updated Lagrangian Jaumann (ULJ) formulation and the updated Lagrangian Hencky (ULH) formulation can be used. For more details on how these formulations apply to shell elements, see Section 2.3.

Input of material parameters: Cauchy (true) stresses and logarithmic (true) strains. For the multilinear stress-strain curves, it is also possible to enter engineering stress-strain data along with the input NXSTRAT CVSSVAL=1, see detailed description of the CVSSVAL feature in Section 3.4.1.

When the ULJ formulation is used:

Output: Cauchy stresses and Jaumann strains.

When the ULH formulation is used:

Output: Kirchhoff stresses and left Hencky strains (practically equivalent to Cauchy stresses and logarithmic strains).

3.1.2 Strain measures

The strain measures used in Advanced Nonlinear Solution are illustrated here in the simplified case of a rod under uniaxial tension (see Fig. 3.1-1).

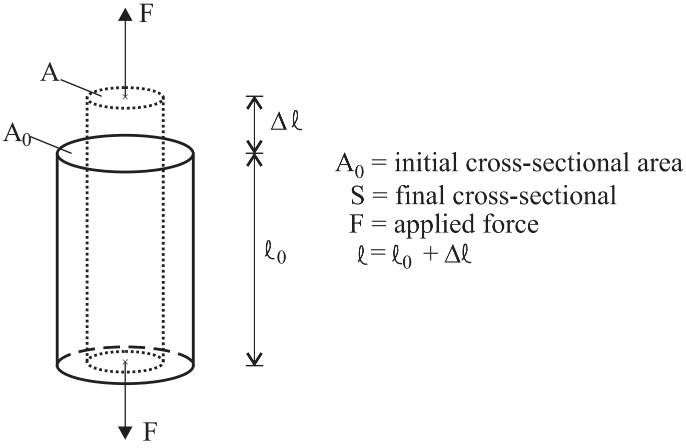


Fig. 3.1-1: Rod under uniaxial tension

Engineering strain: $$e_0 = \frac{\ell - \ell_0}{\ell_0}$$

Green-Lagrange strain: $$\varepsilon = \frac{1}{2}\frac{\ell^2 - \ell_0^2}{\ell_0^2}$$

Logarithmic strain, Hencky strain, Jaumann strain:

$$e = \ln\left(\frac{\ell}{\ell_0}\right)\left[= \int_{\ell_0}^{\ell} \frac{d\ell}{\ell}\right]$$

Stretch: $$\lambda = \frac{\ell}{\ell_0}$$

- Note that for the small strains assumption to be valid, the strains should be less than about 2%.

ref. KJB Sec. 6.2.2

- Green-Lagrange strains are used in the large displacement/small strain formulations. This is because this strain measure is invariant with respect to rigid-body rotations. Therefore, for small strains, Green-Lagrange strains and the rotated engineering strains are equivalent.

- Engineering strains are also called nominal strains in the literature.

- Logarithmic strains are also known as true strains.

3.1.3 Stress measures

The stress measures used in Advanced Nonlinear Solution include engineering stresses, 2nd Piola-Kirchhoff stresses, Kirchhoff stresses, and Cauchy stresses (see ref. KJB). These stress measures are illustrated here in the simplified case of a rod under uniaxial tension (see Fig. 3.1-1).

Engineering stress: $\sigma = \dfrac{F}{A_0}$

Cauchy stress: $\tau = \dfrac{F}{A} = \dfrac{\sigma A_0}{A}$

2nd Piola-Kirchhoff stress: $S = \dfrac{F\ell_0}{A_0\ell} = \dfrac{\sigma\ell_0}{\ell}$

Kirchhoff stress: $J\tau = \dfrac{F\ell}{A_0\ell_0} = \dfrac{\sigma\ell}{\ell_0}$

- Cauchy stresses are also called true stresses in the literature.

- For the case in which the material is incompressible, $\tau = J\tau = \dfrac{\sigma\ell}{\ell_0}$ can be used to compute the Cauchy stress and the Kirchhoff stress from the engineering stress.

- When the strains are small, the 2nd Piola-Kirchhoff stresses are nearly equal to the Cauchy stresses from which the rigid body rotations of the material have been removed.

- When the volume change of the material is small, the Kirchhoff stresses are nearly equal to the Cauchy stresses.

- Since Kirchhoff stresses are input/output only for large strain analysis with materials that are nearly incompressible, practically speaking, the differences between Kirchhoff and Cauchy stresses are negligible.

3.1.4 Large strain thermo-plasticity analysis with the ULH formulation

- This section discusses the ULH formulation for large strain analysis. ULH stands for updated Lagrangian Hencky.

- The following is a quick summary of the theory of large strain inelastic analysis with the ULH formulation. For further information, see ref KJB, Section 6.6.4 and also the following references:

ref. F.J. Montáns and K.J. Bathe, "Computational issues in large strain elasto-plasticity: an algorithm for mixed hardening and plastic spin", *Int. J. Numer. Meth. Engng,* 2005; 63;159-196.

ref. M. Kojić and K.J. Bathe, *Inelastic Analysis of Solids and Structures*, Springer-Verlag, 2003.

Total deformation gradient tensor: Let **X** be the total deformation gradient tensor at time *t* with respect to an initial configuration taken at time 0. For ease of writing, we do not include the usual left superscripts and subscripts.

Polar decomposition into rotation and right stretch tensor: The total deformation gradient tensor **X** can be decomposed into a material rigid-body rotation tensor **R** and a symmetric positive-definite (right) stretch tensor **U** (polar decomposition):

$$\mathbf{X} = \mathbf{R}\,\mathbf{U} \qquad (3.1\text{-}1)$$

Principal directions of right stretch tensor: The right stretch tensor **U** can be represented in its principal directions by a diagonal tensor $\mathbf{\Lambda}$, such that

$$\mathbf{U} = \mathbf{R}_L \mathbf{\Lambda}\, \mathbf{R}_L^T \qquad (3.1\text{-}2)$$

where $\mathbf{R}_L$ is a rotation tensor with respect to the fixed global axes (see Fig. 3.1-2).

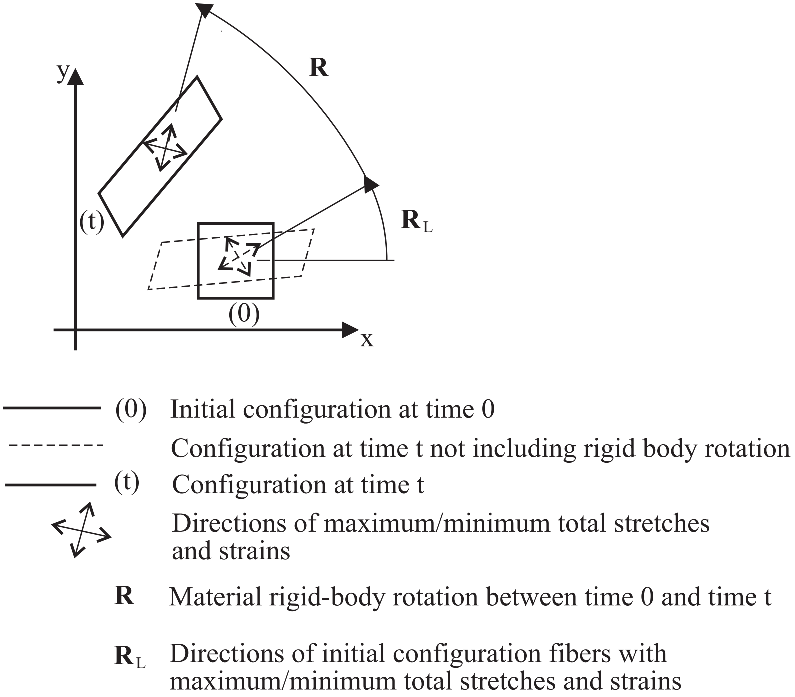

y
R
(t)
$\mathbf{R}_L$
(0)
x
(0) Initial configuration at time 0
Configuration at time t not including rigid body rotation
(t) Configuration at time t
Directions of maximum/minimum total stretches and strains
R Material rigid-body rotation between time 0 and time t
$\mathbf{R}_L$ Directions of initial configuration fibers with maximum/minimum total stretches and strains

Fig. 3.1-2: Directions of maximum/minimum total stretches and strains

(Note that the rotation $\mathbf{R}_L$ does not correspond to a material rigid-body rotation, but to a rotation of the coordinate system: **U** and $\mathbf{\Lambda}$ are two representations of the same deformed state, respectively in the basic coordinate system and in the **U** principal directions coordinate system.)

Right Hencky strain tensor: The Hencky strain tensor (computed in the right basis) is given by

$$\mathbf{E}^R = \ln \mathbf{U} = \mathbf{R}_L \ln \mathbf{\Lambda}\, \mathbf{R}_L^T \qquad (3.1\text{-}3)$$

The superscript "R" symbolizes the right basis.

Polar decomposition into rotation and left stretch tensor: The total deformation gradient tensor **X** can also be decomposed into a material rigid-body rotation **R** and a symmetric positive-definite (left) stretch tensor **V** (polar decomposition):

$$\mathbf{X} = \mathbf{V}\,\mathbf{R} \qquad (3.1\text{-}4)$$

R in (3.1-4) is the same as **R** in (3.1-1).

Principal directions of left stretch tensor: The left stretch tensor **V** can be represented in its principal directions by a diagonal tensor $\mathbf{\Lambda}$, such that

$$\mathbf{V} = \mathbf{R}_E \mathbf{\Lambda}\, \mathbf{R}_E^T \qquad (3.1\text{-}5)$$

where $\mathbf{R}_E$ is a rotation tensor with respect to the fixed global axes. Note that $\mathbf{R}_E = \mathbf{R}\,\mathbf{R}_L$.

Left Hencky strain tensor: The Hencky strain tensor (computed in the left basis) is given by

$$\mathbf{E}^L = \ln \mathbf{V} = \mathbf{R}_E \ln \mathbf{\Lambda}\, \mathbf{R}_E^T \qquad (3.1\text{-}6)$$

The superscript “L” symbolizes the left basis.

Comparison of left and right Hencky strain tensors: The principal values of the left and right Hencky strain tensors are identical, and equal to the logarithms of the principal stretches. Hence both of these strain tensors can be considered to be logarithmic strain tensors. However, the principal directions of the left and right Hencky strain tensors are different. The principal directions of the right Hencky strain tensor do not contain the rigid body rotations of the material, but the principal directions of the left Hencky strain tensor contain the rigid body rotations of the material.

Therefore, for a material undergoing rigid body rotations, the principal directions of the right Hencky strain tensor do not rotate, however the principal directions of the left Hencky strain tensor rotate with the material. Hence, the left Hencky strain tensor is preferred for output and visualization of the strain state.

Multiplicative decomposition of deformation gradient in inelastic analysis: In inelastic analysis, the following multiplicative decomposition of the total deformation gradient into an elastic deformation gradient $\mathbf{X}^E$ and an inelastic deformation gradient $\mathbf{X}^P$ is assumed:

$$\mathbf{X} = \mathbf{X}^E \, \mathbf{X}^P \tag{3.1-7}$$

To understand (3.1-7), consider a small region of material under a given stress state with deformation gradient **X**. If this region of material is separated from the rest of the model and subjected to the same stress state, the deformation gradient is still **X**. Now if the stress state is removed, (3.1-7) implies that the deformation gradient of the unloaded material is $\mathbf{X}^P$. The stresses are due entirely to the strains associated with the elastic deformation gradient $\mathbf{X}^E$.

It can be shown (see Montáns and Bathe), that (3.1-7) is equivalent to the additive decomposition of the displacements into elastic displacements and plastic displacements.

For the materials considered here, $\det \mathbf{X}^P = 1$.

Polar decomposition of elastic deformation gradient: The elastic deformation gradient can be decomposed into an elastic rotation tensor $\mathbf{R}^E$ and elastic right and left stretch tensors $\mathbf{U}^E$, $\mathbf{V}^E$:

$$\mathbf{X}^E = \mathbf{R}^E \, \mathbf{U}^E = \mathbf{V}^E \, \mathbf{R}^E \tag{3.18-a,b}$$

Elastic Hencky strain tensors: The elastic Hencky strain tensors in the right and left bases are given by

$$\mathbf{E}^{ER} = \ln \mathbf{U}^E, \quad \mathbf{E}^{EL} = \ln \mathbf{V}^E \tag{3.1-9a,b}$$

Stress-strain relationships: The stresses are computed from the elastic Hencky strain tensors using the usual stress-strain law of isotropic elasticity. However, the stress measures used depend upon the strain measures used. When the right Hencky strain measure is used, the stress measure used is the rotated Kirchhoff stress

$$\overline{\boldsymbol{\tau}} = \left(\mathbf{R}^E\right)^T J\boldsymbol{\tau} \, \mathbf{R}^E \tag{3.1-10}$$

and when the left Hencky strain measure is used, the stress measure is the (unrotated) Kirchhoff stress $J\boldsymbol{\tau}$. $J = \det \mathbf{X}$ is the volume change of the material, and, using $\det \mathbf{X}^P = 1$, $J = \det \mathbf{X}^E$.

With these choices of stress and strain measures, the stresses and strains are work-conjugate.

The choice of right Hencky strain and rotated Kirchhoff stresses gives the same numerical results as the choice of left Hencky strain and (unrotated) Kirchhoff stresses.

Implementation notes: For 2-D and 3-D solid elements, the difference between the Cauchy and Kirchhoff stresses is neglected. The stress measure used with the right Hencky strains is $\overline{\boldsymbol{\tau}} = \left(\mathbf{R}^E\right)^T \boldsymbol{\tau}\, \mathbf{R}^E$. The input of material properties is assumed to be in terms of Cauchy stresses, and the output of stresses is in terms of Cauchy stresses.

For shell elements, Kirchhoff stresses are used throughout. The input of material properties is assumed to be in terms of Kirchhoff stresses, and the output of stresses is in terms of Kirchhoff stresses.

These assumptions are justified because they are used with material models in which the plastic deformations are incompressible and the plastic deformations are generally much larger than the elastic deformations.

3.1.5 Large strain thermo-plasticity analysis with the ULJ formulation

- This section discusses the ULJ formulation for large strain inelastic analysis (ULJ formulation). ULJ stands for updated Lagrangian Jaumann.

- The following is a quick summary of the theory of large strain inelastic analysis with the ULJ formulation:

For further information, see ref KJB, Section 6.2.2 and also the following reference:

ref. M. Kojić and K.J. Bathe, *Inelastic Analysis of Solids and Structures*, Springer-Verlag, 2003.

Velocity gradient tensor: The velocity gradient tensor is defined as

$$\mathbf{L} = \left[\frac{\partial\, {}^t\dot{u}_i}{\partial\, {}^t x_j} \right] = \dot{\mathbf{X}}\mathbf{X}^{-1} \qquad (3.1\text{-}11)$$

Notice that the derivative is taken with respect to the current coordinates.

Rate of deformation tensor, spin tensor: The rate of deformation tensor is defined as

$$\mathbf{D} = \frac{1}{2}\left(\mathbf{L} + \mathbf{L}^T\right) \qquad (3.1\text{-}12)$$

and the spin tensor is defined as

$$\mathbf{W} = \frac{1}{2}\left(\mathbf{L} - \mathbf{L}^T\right) \qquad (3.1\text{-}13)$$

D is the symmetric part of **L** and **W** is the skew-symmetric part of **L**.

Rate of change of Jaumann strain tensor: The rate of change of the Jaumann strain is defined as

$$\dot{\boldsymbol{\varepsilon}}^J = \mathbf{D} + \mathbf{W}\boldsymbol{\varepsilon}^J - \boldsymbol{\varepsilon}^J\mathbf{W} \qquad (3.1\text{-}14)$$

The quantity $\boldsymbol{\varepsilon}^J$ is termed the Jaumann strain in analogy with the more often-used Jaumann stress. But we do not use the Jaumann stress in the ULJ formulation.

Jaumann strain tensor: In practice, increments are used in computing the Jaumann strain tensor, i.e.

$${}^{t+\Delta t}\boldsymbol{\varepsilon}^J = {}^t\boldsymbol{\varepsilon}^J + \left(\mathbf{D}\Delta t\right) + \left(\mathbf{W}\Delta t\right){}^t\boldsymbol{\varepsilon}^J - {}^t\boldsymbol{\varepsilon}^J\left(\mathbf{W}\Delta t\right) \qquad (3.1\text{-}15)$$

Comparison of Jaumann strain with left Hencky strain: When the rate of change of the principal directions of the left stretch tensor **V** is zero, the rate of change of the left Hencky strain is the same as the rate of change of the Jaumann strain. Hence the Jaumann strain can be used as an approximate replacement for the left Hencky strain. The Jaumann strain can be computed more efficiently than the left Hencky strain, because it is not necessary to take the square root or logarithm of a tensor when computing the Jaumann strain. On the other hand, the time step size affects the Jaumann strain, so that finite time step sizes lead to an error in the calculation of the Jaumann strain.

For a uniaxial deformation, the Jaumann strain approaches the logarithmic strain as the step size is reduced. For a rigid-body rotation, the Jaumann strain also rotates, with the rotation of the Jaumann strain approaching the expected rotation as the step size is reduced.

It can also be shown that the Jaumann strain is path-dependent in general, so that a deformation history in which the final deformations equal the initial deformations can produce (non-physical) non-zero Jaumann strains, even in the limit of infinitesimally small time steps.

Stress-strain relationships: In elasto-plasticity, the same algorithms are used as in small-strain elasto-plasticity. The mechanical strains are computed as the total strains minus the plastic strains (and also any thermal strains), in which the total strains are the Jaumann strains.

As in the ULH formulation, the stresses are Cauchy stresses for 2-D / 3-D elements, and are Kirchhoff stresses for shell elements.

3.1.6 Thermal strains

- Calculation of thermal strains is needed for temperature-dependent material models (thermo-elastic isotropic, thermo-elastic orthotropic, thermo-plastic), as well as temperature-invariant material models with non-zero thermal expansion coefficients.

- The current temperature ${}^{t}\theta$ and the initial temperature ${}^{0}\theta$ (corresponding to zero thermal strains) are both needed for the calculation of thermal strains. The current temperature field is set via the TEMPERATURE(LOAD) case control entry, while the initial temperature is set via the TEMPERATURE(INITIAL) case control entry. See Section 5.6 for more details.

- The temperature at an integration point is evaluated based on the nodal temperatures and the element shape functions, and then used to calculate the thermal strains.

- For isotropic temperature independent materials, the following expression is used for thermal expansion.

$$ {}^{t}e_{ij}^{TH} = \alpha \left({}^{t}\theta - {}^{0}\theta \right) \delta_{ij} \tag{3.1-16} $$

where δ_{ij} is the Kronecker delta ($\delta_{ij} = 1$ for $i = j$ and $\delta_{ij} = 0$ for $i \neq j$).

- If the thermal expansion is temperature dependent and isotropic, the thermal strains are calculated as follows:

$$ {}^{t}e_{ij}^{TH} = {}^{t}\overline{\alpha} \left({}^{t}\theta - {}^{0}\theta \right) \delta_{ij} \tag{3.1-17} $$

where

$$ {}^{t}\overline{\alpha} = \frac{1}{\left({}^{t}\theta - {}^{0}\theta \right)} \left(\alpha\left({}^{t}\theta \right)\left({}^{t}\theta - \theta_{REF} \right) - \alpha\left({}^{0}\theta \right)\left({}^{0}\theta - \theta_{REF} \right) \right) \tag{3.1-18} $$

and θ_{REF} is the material reference temperature.

- For temperature independent orthotropic materials Eq. (3.1-16) is replaced by a thermal expansion coefficient vector,

$$ {}^{t}e_{ij}^{TH} = \alpha_i \left({}^{t}\theta - {}^{0}\theta \right) \delta_{ij} \quad \text{(no summation over } i\text{)} \tag{3.1-19} $$

- For temperature dependent orthotropic materials Eq. (3.1-17) and Eq. (3.1-18) are modified for each direction similar to Eq. (3.1-19).

- Equations (3.1-17) and (3.1-18) are derived as follows: Suppose that, from experimental data, the dependence of the length of a bar as a function of temperature is obtained, as shown in Fig. 3.1-3.

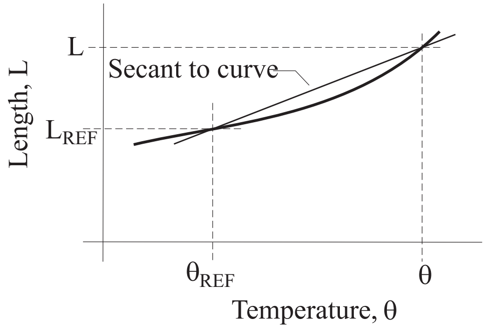


Fig. 3.1-3: Length of bar vs. temperature

The thermal strain with respect to the reference length may be calculated as

$$\varepsilon^{TH} = \frac{L - L_{REF}}{L_{REF}}$$

Then we define the mean coefficient of thermal expansion for a given temperature as follows:

$$\alpha\left(\theta\right) = \frac{\varepsilon^{TH}\left(\theta\right)}{\theta - \theta_{REF}}$$

With this definition, the secant slope in Fig. 3.1-3 is $L_{REF}\,\alpha\left(\theta\right)$.

Now, in Solution 601, we assume that the thermal strains are initially zero. To do this, we subtract the thermal strain corresponding to ${}^{0}\theta$ to obtain

$${}^{t}\varepsilon^{TH} = \alpha\left({}^{t}\theta\right)\left({}^{t}\theta - \theta_{REF}\right) - \alpha\left({}^{0}\theta\right)\left({}^{0}\theta - \theta_{REF}\right)$$

which leads to Equations (3.1-17) and (3.1-18).

Notice that if the mean coefficient of thermal expansion is constant, θ_{REF} no longer enters into the definition of ${}^{t}\overline{\alpha}$ and the equations reduce to Eq. 3.1-16. In general, when the mean coefficient of thermal expansion is not constant, θ_{REF} must be chosen based on knowledge of the experiment used to determine $\alpha\left(\theta\right)$ since for the same material curve, different choices of θ_{REF} yield different values of $\alpha\left(\theta\right)$.

3.2 Linear elastic material models

- The following material models are discussed in this section:

  Elastic-isotropic: isotropic linear elastic non-thermal dependent material model obtained with MAT1

  Elastic-orthotropic: orthotropic linear elastic non-thermal dependent material model obtained with MAT2 and MAT8 for surface elements and MAT9 and MAT11 for 3-D solid elements

- These models can be employed using **small displacement/ small strain** or **large displacement/small strain** kinematics. The strains are always assumed to be small.

- Thermal strains are supported for the elastic isotropic materials and the elastic-orthotropic materials.

- When the elastic-isotropic and elastic-orthotropic materials are used with the small displacement formulation, the formulation is linear.

- In the small displacement formulation, the stress-strain relationship is

$$ {}_{0}^{t}\boldsymbol{\sigma} = \mathbf{C}\ {}_{0}^{t}\mathbf{e} $$

in which ${}_{0}^{t}\boldsymbol{\sigma}$ = engineering stresses and ${}_{0}^{t}\mathbf{e}$ = engineering strains.

- In the large displacement formulation used by 2-D solid, 3D-solid and shell elements, the stress-strain relationship is

ref. KJB Section 6.6.1

$${}_0^t\mathbf{S} = \mathbf{C}\;{}_0^t\boldsymbol{\varepsilon}$$

in which ${}_0^t\mathbf{S}$ = second Piola-Kirchhoff stresses and ${}_0^t\boldsymbol{\varepsilon}$ = Green-Lagrange strains.

- In the large displacement formulation used by rod and beam elements, the stress-strain relationship is

$${}^t\boldsymbol{\tau} = \mathbf{C}\;{}_t^t\boldsymbol{\varepsilon}^*$$

in which ${}^t\boldsymbol{\tau}$ = Cauchy stresses and ${}_t^t\boldsymbol{\varepsilon}^*$ = rotated engineering strain.

- In the presence of thermal strains the following stress-strain relationship is used instead in small displacement analysis:

$${}_0^t\boldsymbol{\sigma} = \mathbf{C}\left({}_0^t\mathbf{e} - {}_0^t\mathbf{e}^{TH}\right)$$

where ${}_0^t\mathbf{e}^{TH}$ are the thermal strains. A similar ${}_0^t\boldsymbol{\varepsilon}^{TH}$ or ${}_t^t\boldsymbol{\varepsilon}^{*TH}$ term is added for the large displacement formulations. The calculation of thermal strain is detailed in Section 3.1.6.

- The same matrix **C** is employed in all of these formulations. As long as the strains remain small, the difference in the responses is negligible.

- However, if the strains are large, the difference in the response predictions is very significant (see ref. KJB, pp 589-590). If the strains are large, it is recommended that these linear elastic material models not be used.

3.2.1 Elastic-isotropic material model

- This material model is available for the **rod**, **2-D solid**, **3-D solid**, **beam**, and **shell** type elements.

- The two material constants used to define the constitutive relation (the matrix **C**) are

E = Young's modulus, ν = Poisson's ratio

- The thermal expansion coefficient α is also used if thermal strains are present.

3.2.2 Elastic-orthotropic material model

- The elastic-orthotropic material model is available for the **2-D solid**, **3-D solid** and **shell** elements.

- The thermal expansion coefficient α is also used if thermal strains are present.

3-D solid elements: The orthotropic 3-D material is defined either using the MAT9 entry or the MAT11 entry. When the MAT9 entry is used, the following assumptions are made:

$C_{14} = C_{15} = C_{16} = C_{24} = C_{25} = C_{26} = C_{34} = C_{35} = C_{36} = 0$

and

$C_{45} = C_{46} = C_{56} = 0.$

resulting in

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \tau_{12} \\ \tau_{23} \\ \tau_{31} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ & C_{22} & C_{23} & 0 & 0 & 0 \\ & & C_{33} & 0 & 0 & 0 \\ & & & C_{44} & 0 & 0 \\ \text{symmetric} & & & & C_{55} & 0 \\ & & & & & C_{66} \end{bmatrix} \begin{bmatrix} e_{11} \\ e_{22} \\ e_{33} \\ \gamma_{12} \\ \gamma_{23} \\ \gamma_{31} \end{bmatrix}$$

When the MAT11 entry is used, the following inverse stress-strain relationship is used:

$$\begin{bmatrix} e_{11} \\ e_{22} \\ e_{33} \\ \gamma_{12} \\ \gamma_{23} \\ \gamma_{31} \end{bmatrix} = \begin{bmatrix} 1/E_1 & -\nu_{12}/E_1 & -\nu_{13}/E_1 & 0 & 0 & 0 \\ -\nu_{21}/E_2 & 1/E_2 & -\nu_{23}/E_2 & 0 & 0 & 0 \\ -\nu_{31}/E_3 & -\nu_{32}/E_3 & 1/E_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/G_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/G_{23} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/G_{13} \end{bmatrix} \begin{bmatrix} \tau_{11} \\ \tau_{22} \\ \tau_{33} \\ \tau_{12} \\ \tau_{23} \\ \tau_{31} \end{bmatrix}$$

The MATCID entry can be used to define the material coordinate system (when using either the MAT9 or MAT11 entries).

Shell elements: The orthotropic shell material is preferably defined using the MAT8 entry, which leads to the following inverse stress-strain relationship defined in the shell material coordinate system (1,2,3):

$$\begin{bmatrix} e_{11} \\ e_{22} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{bmatrix} = \begin{bmatrix} 1/E_1 & -\nu_{12}/E_1 & 0 & 0 & 0 \\ -\nu_{21}/E_2 & 1/E_2 & 0 & 0 & 0 \\ 0 & 0 & 1/G_{12} & 0 & 0 \\ 0 & 0 & 0 & 1/G_{1z} & 0 \\ 0 & 0 & 0 & 0 & 1/G_{2z} \end{bmatrix} \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{bmatrix}$$

The MAT2 entry can also be used to define a shell material with only in-plane orthotropy:

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \tau_{12} \\ \tau_{13} \\ \tau_{23} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{22} & 0 & 0 & 0 \\ 0 & 0 & C_{33} & 0 & 0 \\ 0 & 0 & 0 & C_{33} & 0 \\ 0 & 0 & 0 & 0 & C_{33} \end{bmatrix} \begin{bmatrix} e_{11} \\ e_{22} \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{bmatrix}$$

2-D solid elements: The orthotropic 2-D material is defined using the MAT3 command.

For the axisymmetric element, the stress-strain relationship defined in the (x,θ,z) plane is

$$\begin{bmatrix} e_x \\ e_\theta \\ e_z \\ \gamma_{zx} \end{bmatrix} = \begin{bmatrix} \frac{1}{E_x} & -\frac{\nu_{\theta x}}{E_\theta} & -\frac{\nu_{zx}}{E_z} & 0 \\ -\frac{\nu_{x\theta}}{E_x} & \frac{1}{E_\theta} & -\frac{\nu_{z\theta}}{E_z} & 0 \\ -\frac{\nu_{xz}}{E_x} & -\frac{\nu_{\theta z}}{E_\theta} & \frac{1}{E_z} & 0 \\ 0 & 0 & 0 & \frac{1}{G_{zx}} \end{bmatrix} \begin{bmatrix} \sigma_x \\ \sigma_\theta \\ \sigma_z \\ \sigma_{zx} \end{bmatrix}$$

For the plane stress element, the stress-strain relationship defined in the (x,y,z) plane, with the y direction as the transverse direction, is:

$$\begin{bmatrix} e_x \\ e_z \\ \gamma_{zx} \end{bmatrix} = \begin{bmatrix} \frac{1}{E_x} & -\frac{\nu_{zx}}{E_z} & 0 \\ -\frac{\nu_{xz}}{E_x} & \frac{1}{E_z} & 0 \\ 0 & 0 & \frac{1}{G_{zx}} \end{bmatrix} \begin{bmatrix} \sigma_x \\ \sigma_z \\ \sigma_{zx} \end{bmatrix}$$

For the plane strain element, the stress-strain relationship defined in the (x,y,z) plane, with the y direction as the transverse direction, is:

$$\begin{bmatrix} e_x \\ e_z \\ \gamma_{zx} \end{bmatrix} = \begin{bmatrix} \frac{1-\nu_{yx}\nu_{xy}}{E_x} & -\frac{\nu_{zx}+\nu_{yx}\nu_{zy}}{E_z} & 0 \\ -\frac{\nu_{xz}+\nu_{yx}\nu_{xy}}{E_x} & \frac{1-\nu_{yz}\nu_{zy}}{E_z} & 0 \\ 0 & 0 & \frac{1}{G_{zx}} \end{bmatrix} \begin{bmatrix} \sigma_x \\ \sigma_z \\ \sigma_{zx} \end{bmatrix}$$

All elements when Young's moduli, Poisson's ratios and shear moduli are specified: The material constants must be defined so that the stress-strain constitutive matrix is positive-definite. The conditions are illustrated for the MAT11 entry. In this case, the following conditions must be satisfied:

$$\left|\nu_{ji}\right| < \left(\frac{E_j}{E_i}\right)^{\frac{1}{2}}, \quad i,j = 1,2,3$$

$$\nu_{21}\nu_{32}\nu_{13} < 0.5\left(1-\nu_{21}^2\frac{E_1}{E_2}-\nu_{32}^2\frac{E_2}{E_3}-\nu_{13}^2\frac{E_3}{E_1}\right) \le 0.5$$

Based on the input values for ν_{ij}, the remaining constants ν_{ji} are calculated so as to have a symmetric constitutive matrix; i.e.,

$$\frac{\nu_{ji}}{E_j} = \frac{\nu_{ij}}{E_i}$$

3.3 Nonlinear elastic material model

- Advanced Nonlinear Solution supports the nonlinear elastic material for the rod, 2-D solid, 3-D solid and shell elements. The nonlinear effect is obtained with a MATS1 entry which has TYPE = ‘NELAST’. The formulations used for the rod element are slightly different (and simpler) and are detailed in Section 3.3.1.

- This material uses a nonlinear elastic uniaxial stress-strain data input in tabular form and shown in Fig. 3.3-1. This material is not based on the classical theory of finite elasticity, and is not intended for large strain analysis. However, it is a useful material model when used appropriately, and with awareness of its limitations.

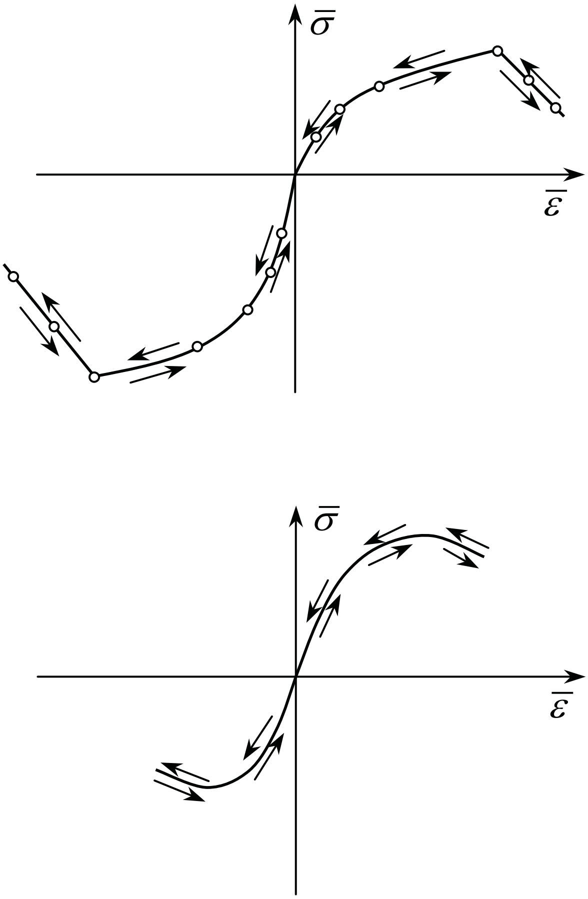


Fig. 3.3-1: Stress-strain behavior of nonlinear elastic material model

- Note that the material unloads along the same curve, so that no permanent inelastic strains are introduced.

- The material can have different stress-strain curves in tension and compression. Under predominantly uniaxial tension or compression, the material response will follow the input curve exactly. Under shear dominated loading, the stress is interpolated from both tension and compression parts of the material stress-strain curve.

- In order to use the unaxial stress-stain data $\bar{\sigma}(\bar{\varepsilon})$ of Fig. 3.3-1, the effective stress and strain ($\bar{\sigma}$ and $\bar{\varepsilon}$) must be calculated based on the 2-D or 3-D total stress and strain tensors ($\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$). The von-Mises stress is used as the effective stress, while the effective strain is based on

$$\int \bar{\sigma} d\bar{\varepsilon} = \int \boldsymbol{\sigma}^T d\boldsymbol{\varepsilon} \tag{3.3-1}$$

which equates the deformation work per unit volume in unaxial loading to the multi-dimensional state. This results in a unique equation for $\bar{\varepsilon}$ as a function of $\boldsymbol{\varepsilon}$, ν and the stress-strain state that depends on the element type.

- The effective strain, $\bar{\varepsilon}$, is defined by

$$\frac{1}{2} E_0 \bar{\varepsilon}^2 = \frac{1}{2} \boldsymbol{\varepsilon}^T \mathbf{C}_0 \boldsymbol{\varepsilon} \tag{3.3-2}$$

where E_0 is Young’s modulus which is determined by the most stiff region of the input stress-strain curve, $\mathbf{C}_0$ is the elastic stress-strain matrix obtained using E_0 and ν. (E_0 cancels out from both sides of Eq. (3.3-2))

Differentiating Eq. (3.3-2) with respect to the total strain, we have

$$d\bar{\varepsilon} = \frac{1}{E_0 \bar{\varepsilon}} \boldsymbol{\varepsilon}^T \mathbf{C}_0 d\boldsymbol{\varepsilon} \tag{3.3-3}$$

Substituting Eq. (3.3-3) into Eq. (3.3-1), the stresses can be expressed in terms of total strains, i.e.,

$$\boldsymbol{\sigma} = \frac{\bar{\sigma}}{E_0 \bar{\varepsilon}} \mathbf{C}_0 \boldsymbol{\varepsilon} \tag{3.3-4}$$

or

$$\boldsymbol{\sigma} = \frac{\bar{\sigma}}{E_0 \bar{\varepsilon}} \boldsymbol{\sigma}_e \qquad (3.3\text{-}5)$$

where $\boldsymbol{\sigma}_e = \mathbf{C}_0 \boldsymbol{\varepsilon}$ which is the elastic trial stress.

- The effective stress $\bar{\sigma}$ is taken from the tensile part of the stress-strain curve for predominantly tensile loading, from the compression part of the stress-strain curve for predominantly compression loading and is interpolated between the two curves otherwise.

- The consistent tangent stress-strain matrix is obtained by differentiating Eq. (3.3-4) or (3.3-5) with respect to the total strain tensor. The stress-strain matrix is symmetric in predominant tensile or compression loading (when only one of the two material curves is used), and is non-symmetric otherwise (when interpolation between the curves is required). The constitutive matrix is symmetrized and in most cases reasonable convergence is obtained.

- Note that discontinuities are not allowed in the user-supplied stress-strain curve. The table look-up is performed using linear interpolation within the table and linear extrapolation outside the table using the two starting or ending points.

Stress update algorithm

For an iteration i, given ${}^{t}\boldsymbol{\sigma}$, ${}^{t}\boldsymbol{\varepsilon}$, ${}^{t+\Delta t}\mathbf{u}^{(i)}$, E_0, ν, update ${}^{t+\Delta t}\boldsymbol{\sigma}^{(i)}$, ${}^{t+\Delta t}\boldsymbol{\varepsilon}^{(i)}$

Step 1. Calculate the new total strain state ${}^{t+\Delta t}\boldsymbol{\varepsilon}^{(i)}$ based on displacements ${}^{t+\Delta t}\mathbf{u}^{(i)}$

Step 2. Calculate the elastic trial stress,

$${}^{t+\Delta t}\boldsymbol{\sigma}_e = \mathbf{C}_0 \, {}^{t+\Delta t}\boldsymbol{\varepsilon} \qquad (3.3\text{-}6)$$

Step 3. Compute the magnitude of the effective strain, $\bar{\varepsilon}$.

Step 4. Calculate the ratio

$$r = C \frac{I_{1_e}}{{}^{t+\Delta t}\overline{\sigma}_e} \tag{3.3-7}$$

where C is a constant that biases the general stress state towards the pure tension or compression curves and is internally set to 3/2, I_{1_e} is the first elastic stress invariant, and ${}^{t+\Delta t}\overline{\sigma}_e$ is the effective elastic stress which is calculated as follows,

$${}^{t+\Delta t}\overline{\sigma}_e = E_0 \, {}^{t+\Delta t}\overline{\varepsilon} \tag{3.3-8}$$

Restrict r to be between -1 and 1.

Step 5. Calculate the effective stress in tension $\overline{\sigma}_t$ and in compression $\overline{\sigma}_c$, based on the user-supplied stress-strain curve and $\overline{\varepsilon}$, as follows:

$${}^{t+\Delta t}\overline{\sigma}_t = {}^{t+\Delta t}\sigma({}^{t+\Delta t}\overline{\varepsilon}) \tag{3.3-9}$$

$${}^{t+\Delta t}\overline{\sigma}_c = -{}^{t+\Delta t}\overline{\sigma}(-{}^{t+\Delta t}\overline{\varepsilon}) \tag{3.3-10}$$

Step 6. Calculate the actual effective stress, ${}^{t+\Delta t}\overline{\sigma}$, as

$${}^{t+\Delta t}\overline{\sigma} = \frac{1+r}{2}\,{}^{t+\Delta t}\overline{\sigma}_t + \frac{1-r}{2}\,{}^{t+\Delta t}\overline{\sigma}_c \tag{3.3-11}$$

Step 7. Evaluate the new stress state by

$${}^{t+\Delta t}\boldsymbol{\sigma} = \frac{{}^{t+\Delta t}\overline{\sigma}}{E_0\,{}^{t+\Delta t}\overline{\varepsilon}}\,{}^{t+\Delta t}\boldsymbol{\sigma}_e \tag{3.3-12}$$

Step 8. Evaluate the tangential stress-strain matrix and symmetrize it.

3.3.1 Nonlinear elastic material for rod element

- For the rod element, the stress-strain relationship is defined as a piecewise linear function, as shown in Fig. 3.3-2.

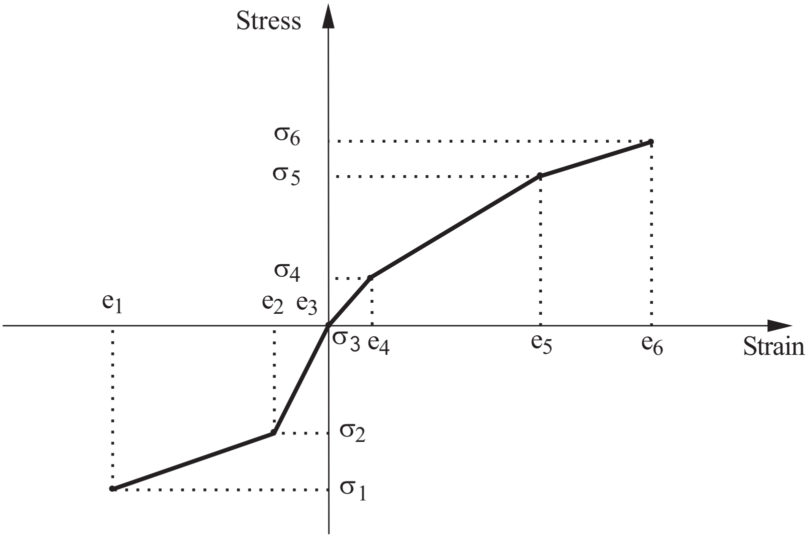


Fig. 3.3-2: Nonlinear elastic material for rod elements

Note that the stress is uniquely defined as a function of the strain only; hence for a specific strain ${}^t e$, reached in loading or unloading, a unique stress is obtained from the curve in Fig. 3.3-2.

- A sufficient range (in terms of the strain) must be used in the definition of the stress-strain relation so that the element strain evaluated in the solution lies within that range; i.e., referring to Fig. 3.3-2, we must have $e_1 \le {}^t e \le e_6$ for all *t*.

- The stress-strain curve does not necessarily have to pass through the origin.

- A typical example of the nonlinear elastic model for rod elements is shown in Fig. 3.3-3. This example corresponds to a cable-like behavior in which the rod supports tensile but no compressive loading.

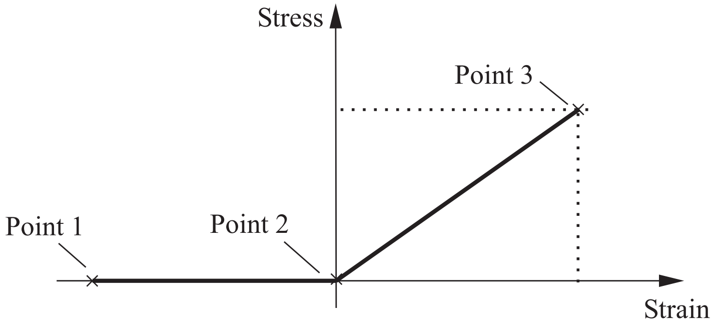


Fig. 3.3-3: Nonlinear elastic material model corresponding to a tension-only cable

- The rod element with this nonlinear elastic material model is particularly useful in modeling gaps between structures. This modeling feature is illustrated in Fig. 3.3-4. Note that to use this element to simulate a contact gap, it is necessary to know which node of one body will come into contact with which node of the other body, and then to connect these two nodes with a rod element.

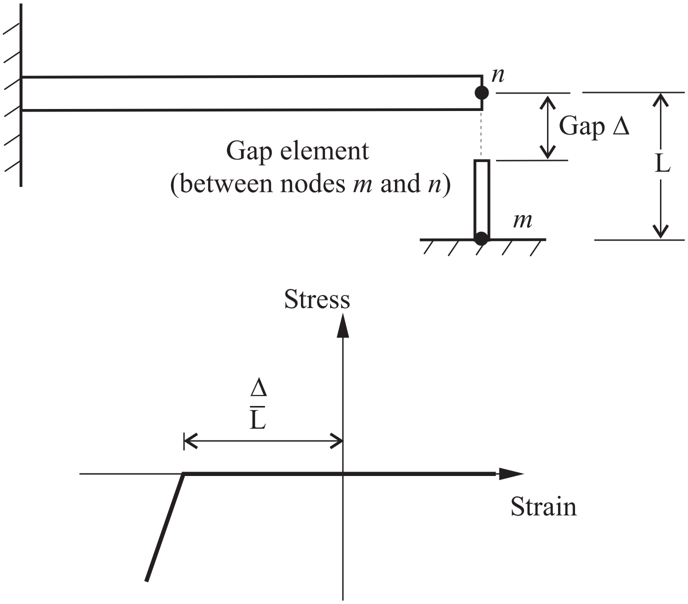


Fig. 3.3-4: Modeling of gaps

3.4 Isothermal plastic material models

- This section describes the following material models:

  Plastic bilinear, plastic multilinear: von Mises model with isotropic, kinematic hardening or mixed hardening

  Plastic-cyclic: von Mises model with hardening rules suitable for modeling cyclic plasticity.

- All elasto-plasticity models use the flow theory to describe the elastic-plastic response; the basic formulations for the von Mises models are summarized on pp. 596-604, ref. KJB.

3.4.1 Plastic-bilinear and plastic-multilinear material models

- Elasto-plastic materials with bilinear or multilinear hardening are defined using the MATS1 material entry with TYPE = ‘PLASTIC’.

- These material models are based on

  - The von Mises yield condition (see p. 597, ref. KJB)
  - An associated flow rule using the von Mises yield function
  - Isotropic, kinematic, or mixed hardening
  - Bilinear or multilinear stress-strain curves (based on H and TID fields in MATS1)

- Figs. 3.4-1 to 3.4-3 summarize some important features of these material models. Point A marks the first onset of plasticity, point B marks unloading into elasticity, point C marks reverse loading to plasticity and point D marks unloading into elasticity.

- These models can be used with the **rod**, **2-D solid**, **3-D solid**, **beam** (plastic-bilinear only), and **shell** elements.

- All elastic and plastic material constants are thermally invariant. However, thermal strains can be present when there is a temperature load and a coefficient of thermal expansion.

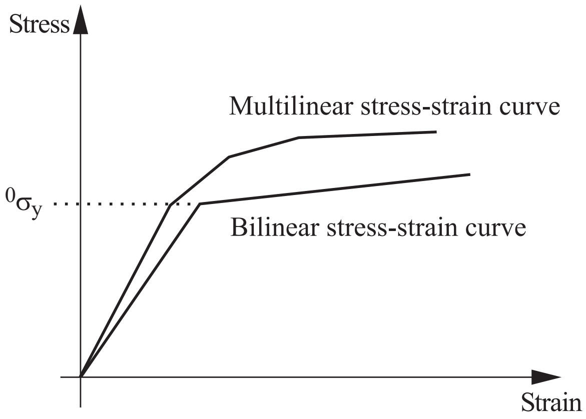


Fig. 3.4-1: von Mises model

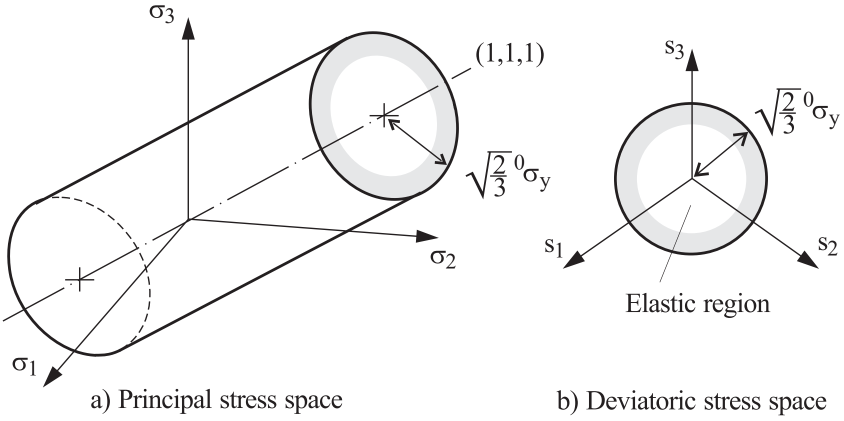
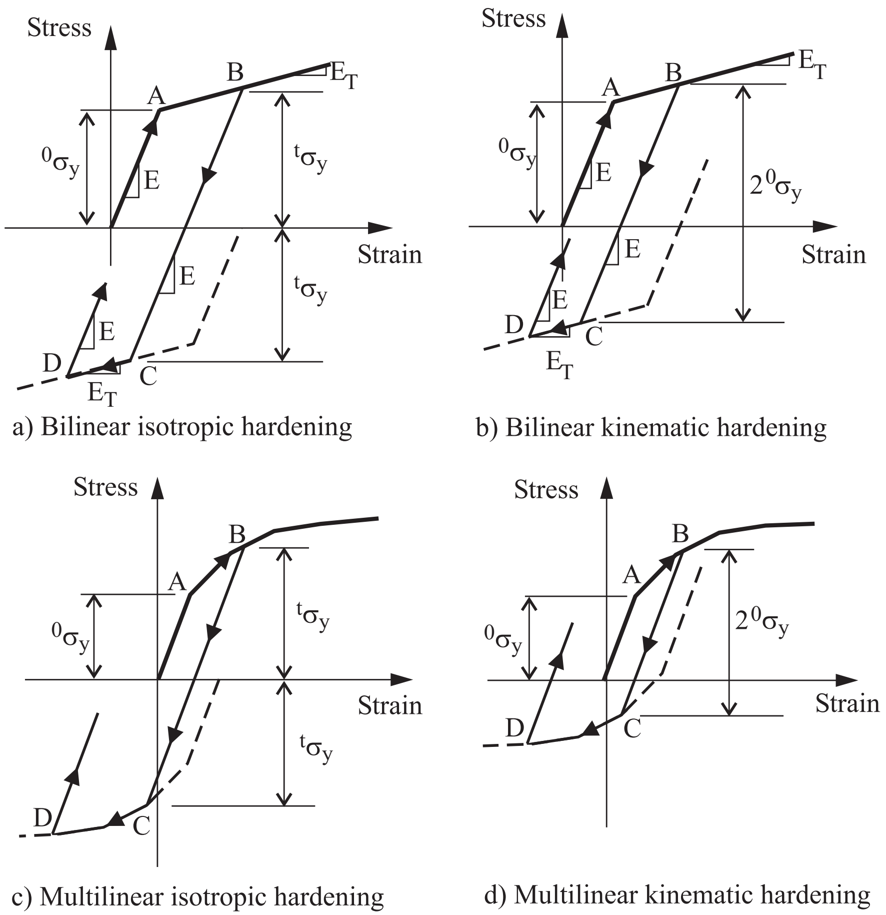

σ_3
(1,1,1)
$\sqrt{\frac{2}{3}}\,{}^0\sigma_y$
σ_2
σ_1
a) Principal stress space
s_3
$\sqrt{\frac{2}{3}}\,{}^0\sigma_y$
s_1
s_2
Elastic region
b) Deviatoric stress space

Fig. 3.4-2: von Mises yield surface



a) Bilinear isotropic hardening

b) Bilinear kinematic hardening

c) Multilinear isotropic hardening

d) Multilinear kinematic hardening

Fig. 3.4-3: Isotropic and kinematic hardening

- These models can be used with **small displacement/small strain**, **large displacement/small strain** and **large displacement/large strain** kinematics.

When used with small displacements/small strain kinematics, a materially-nonlinear-only formulation is employed.

When used with large displacements/small strain kinematics, either a TL or a UL formulation is employed (depending on element type).

When used with large displacement/large strain kinematics, a ULH formulation or a ULJ formulation can be employed. Large displacement/large strain kinematics can only be used with the 2-D solid, 3-D solid and shell elements (only single layer shell elements).

If geometrically nonlinear effects are to be included, the large displacements/large strain kinematics are preferred to the large displacement/small strain kinematics, even when the strains are numerically small. The large displacement/small strain kinematics should be used only when the large displacement/large strain kinematics are not supported by the element.

- For multilinear plasticity, there is no restriction on the number of stress-strain points in the stress-strain curve.

- Mixed hardening is available only for bilinear plasticity.

- Plane strain, axisymmetric or 3-D solid elements that reference these material models should preferably employ the mixed displacement-pressure (u/p) element formulation. This is done by setting UPFORM = 1 in the NXSTRAT command.

- In the von Mises model with isotropic hardening, the following yield surface equation is used:

$$ {}^{t}f_y = \frac{1}{2}\left({}^{t}\mathbf{s} \cdot {}^{t}\mathbf{s}\right) - \frac{1}{3}\,{}^{t}\sigma_y^2 = 0 $$

where ${}^{t}\mathbf{s}$ is the deviatoric stress tensor and ${}^{0}\sigma_y^2$ the updated yield stress at time *t*.

In the von Mises model with kinematic hardening, the following yield surface equation is used:

$$ {}^{t}f_y = \frac{1}{2}\left({}^{t}\mathbf{s} - {}^{t}\boldsymbol{\alpha}\right) \cdot \left({}^{t}\mathbf{s} - {}^{t}\boldsymbol{\alpha}\right) - \frac{1}{3}\,{}^{0}\sigma_y^2 = 0 $$

where ${}^{t}\boldsymbol{\alpha}$ is the shift of the center of the yield surface (back stress tensor) and ${}^{0}\sigma_y^2$ is the virgin, or initial, yield stress.

In the von Mises model with mixed hardening, the following yield surface equation is used:

$$ {}^{t}f_y = \frac{1}{2}\left({}^{t}\mathbf{s} - {}^{t}\boldsymbol{\alpha}\right)\cdot\left({}^{t}\mathbf{s} - {}^{t}\boldsymbol{\alpha}\right) - \frac{1}{3}\,{}^{t}\sigma_y^2 = 0 $$

where

$$ {}^{t}\sigma_y = {}^{0}\sigma_y + ME_p\mathbf{e}^p $$

The back stress ${}^{t}\boldsymbol{\alpha}$ is evolved by

$$ d\boldsymbol{\alpha} = C_p\left(1 - M\right)d\mathbf{e}^p $$

C_p is Prager's hardening parameter, related to the plastic modulus E_p and M is the factor used in general mixed hardening ($0 < M < 1$) which is currently restricted to 0.5.

The formulation for the von Mises model with mixed hardening is given in the following reference:

ref K.J. Bathe and F.J. Montáns, "On Modeling Mixed Hardening in Computational Plasticity", *Computers and Structures*, Vol. 82, No. 6, pp. 535–539, 2004.

The yield stress is a function of the effective plastic strain, which defines the hardening of the material. The effective plastic strain is defined as

$$ {}^{t}\overline{e}^{P} = \int_0^t \sqrt{\frac{2}{3}d\mathbf{e}^p \cdot d\mathbf{e}^p} $$

in which $d\mathbf{e}^p$ is the tensor of differential plastic strain increments and in which $d\mathbf{e}^p \cdot d\mathbf{e}^p$ is calculated as $de_{ij}^p\, de_{ij}^p$ (see ref. KJB, p. 599). In finite element analysis, ${}^{t}\overline{e}^{P}$ is approximated as the sum of all of the plastic strain increments up to the current solution time:

$$ {}^{t}\overline{e}^{P} = \sum_{\textit{all solution steps}} \Delta\mathbf{e}^p $$

where $\Delta \overset{-}{e}^{P} = \sqrt{\frac{2}{3}\Delta \mathbf{e}^{p} \cdot \Delta \mathbf{e}^{p}}$ and $\Delta \mathbf{e}^{p}$ is the tensor of plastic strain increments in a solution step. Because of the summation over the solution steps, the calculated value of ${}^{t}\overset{-}{e}^{P}$ is referred to as the accumulated effective plastic strain.

- If a thermal load is applied to the structure, the thermal strains are taken into account but the material characteristics are considered to be temperature independent.

- Notice that for the multilinear material models, the plastic tangent modulus at point C in Fig. 3.4-3 is smaller than the plastic tangent modulus at point A. The plastic tangent modulus decreases for each successive cycle. Hence these multilinear material models are not well suited for modeling cyclic behavior.

If cyclic behavior is to be modeled, either the bilinear kinematic model can be used (because the plastic tangent modulus is constant), or the plastic-cyclic material model can be used.

Stress-strain input data for multilinear plasticity in large strain analysis

In large strain analysis, Advanced Nonlinear Solution works internally with true (Cauchy) stresses and true (logarithmic) strains. However, typical tension tests used to determine experimental data return forces and displacements. These forces and displacements are used to compute engineering stresses (force per unit original area) and engineering strain (change in length per unit length). Therefore it is necessary to convert engineering stress-strain data to true stress-strain data. This conversion is either done by the user, or is done automatically by Advanced Nonlinear Solution (using the CVSSVAL=1 entry on the NXSTRAT card). We now discuss this conversion process in detail.

Consider an experiment in which a fully incompressible material is put into uniaxial tension. The force-displacement curve is determined, and the following information extracted from the force-displacement curve:

The engineering stress is computed as the force / original area. The engineering strain is computed as the displacement / original length.

This data is determined for a number of points on the force-displacement curve, starting at point 2, which is considered to be the elastic limit (point 1 is at the origin). And the Young's modulus is also determined as the ratio of engineering stress / engineering strain at the elastic limit.

Now consider duplicating the experimental results with a finite element model that uses a large strain formulation. The tension test finite element model should return the correct force for each displacement point on the force-displacement curve.

Because the stress and strain measures used in the large strain finite element formulation are true stress and true strain, it is necessary to convert the engineering stress / engineering strain data to true stress / true strain data.

The conversion can be done using an algorithm similar to

```
{
    For (each stress-strain point i, i=1, 2, ... ) {
        $e_i =$ engineering strain
        $\sigma_i =$ engineering stress
        $\tau_i = \sigma_i(1+e_i)$ = true stress
        $\varepsilon_i = \ln(1+e_i)$ = true strain
    }
    $E = \tau_2 / \varepsilon_2$
}
```

Notice that E also needs to be converted. The reason is as follows. If E is not converted, Advanced Nonlinear Solution computes $\varepsilon_2 = \tau_2 / E$ but now ε_2 is no longer the true strain at the elastic limit.

Here is a numerical example:

$e_2 = 0.1$, $\sigma_2 = 30$ MPa , $E = \sigma_2 / e_2 = 300$ MPa

The conversion given above produces

$$\varepsilon_2 = 0.09531,\ \tau_2 = 33 \text{ MPa}$$

This point is assumed to be at the elastic limit. If E is kept at 300, then Advanced Nonlinear Solution computes

$$\varepsilon_2 = 33/300 = 0.11$$

and this ε_2 is no longer at the elastic limit. Therefore E must be recalculated as part of the conversion process:

$$E = 33/0.09531 = 346.2 \text{ MPa}$$

Assuming that the above conversion is performed, then the tension test finite element model will return the correct force for each of the displacements corresponding to the points on the original force-displacement curve.

Assumptions in the conversion formulas

There are a number of assumptions in the above formulas, as follows:

Elastic response

It is assumed that the stress-strain behavior is linearly elastic up to the elastic limit. However, since Advanced Nonlinear Solution uses true stress / true strain data internally, the linear elastic behavior is also based on true stress / true strain data. In the above example, the linear elastic response computed by the tension test finite element model is based on a Young’s modulus of 346.2 MPa. This response is, of course, quite different than a linear elastic response based on a Young’s modulus of 300 MPa.

The reason that the Young’s modulus is so different is because the strain at the elastic limit is “large”. When the strain at the elastic limit is small, then the change in Young’s modulus is also small. The change in Young’s modulus caused by the conversion is approximately equal to $\Delta E = \frac{3}{2} e_2 E$.

Compressible elastic material

The above conversion assumes that the material response is fully incompressible, under both elastic and plastic conditions. However, in most cases, the material is compressible under elastic conditions.

Nevertheless, the above formulas are frequently used anyway. The error thus incurred will be largest for point 2, and will diminish for larger values of strain.

Response in uniaxial compression

The above formulas use tension data to convert from engineering to true values. However, it should be recognized that Advanced Nonlinear Solution uses the same stress-strain curve in both tension and compression. Therefore, if a large strain finite element model is put into uniaxial compression, the observed (compressive) force will not be equal in magnitude to the expected (tensile) force, at a given (compressive) displacement level.

Here is an example. Suppose that in a tensile test, the following data is obtained:

Point	Engineering strain	Engineering stress (MPa)	True strain	True stress
...				
3	0.05	50	0.04879	52.5
4	0.05263	51	0.05129	53.7
...				

First consider a materially-nonlinear-only analysis. Enter the engineering stress and strain values for the two points as part of the stress-strain input data. When the model is run in compression to an engineering strain of -0.05, the engineering stress is -50 MPa. The force-displacement response is symmetric in tension and compression.

Next consider a large strain analysis. Enter the true stress and strain values for the two points as part of the stress-strain input data. Now run the model in compression to an engineering strain of

-0.05. An engineering strain of -0.05 corresponds to a true strain of -0.05129, therefore the true stress is -53.7 and the engineering stress is -56.5. The force-displacement response is not symmetric in tension and compression.

Homogeneous deformation

The above conversion assumes that the specimen is uniform and that the specimen deforms homogeneously under load. Therefore the conversion is only valid up to the ultimate tensile strength of the material, because beyond that point the deformation of the specimen might be no longer homogeneous due to localized necking.

NXSTRAT CVSSVAL feature

When CVSSVAL=0 on the NXSTRAT card (the default), Advanced Nonlinear Solution does not perform any conversion. True stress-strain data should be input, and the user could compute this true stress-strain data using the formulas given above.

When CVSSVAL=1, then Advanced Nonlinear Solution performs this conversion using the formulas given above. Therefore engineering stress-strain data should be input.

The CVSSVAL=1 feature should only be used when all of the elements that use multilinear plastic materials also use large strain formulations.

The CVSSVAL=1 feature does not perform any conversions for the bilinear plastic material models, or for the plastic-cyclic material models.

Material behavior beyond the last point of the stress-strain curve in multilinear plasticity

The material behavior beyond the last point of the stress-strain curve in multilinear plasticity can be considered ruptured, or the curve can be extended indefinitely with the slope of its final segment. This depends on a global setting of the XTCURVE parameter in NXSTRAT with indefinite extension as the default.

Modeling of rupture: Rupture conditions can also be modeled for the multilinear stress-strain curve. The rupture plastic strain corresponds to the effective plastic strain at the last point input for the stress-strain curve. No rupture strain exists for the bilinear case.

When rupture is reached at a given element integration point, the corresponding element is removed from the model (see Section 10.4).

Rate-dependent plasticity: The rate-dependent model is used to simulate the increase in the yield stress with an increase in strain rate.

The rate-dependent model only applies to the isotropic plasticity models with isotropic hardening (bilinear or multilinear).

The rate-dependent model is implemented for 2-D solid, 3-D solid, rod and shell elements.

The effective yield stress including strain rate effects is

$$\sigma_y = \sigma_y^0 \left[1 + b \ln \left(1 + \frac{\dot{\varepsilon}^P}{\dot{\varepsilon}_0} \right) \right]$$

where σ_y^0 is the static yield stress, $\dot{\varepsilon}_0$ is the transition strain rate and b is the strain hardening parameter. Here $\dot{\varepsilon}^P$ is calculated using $\dot{\varepsilon}^P = \dfrac{\Delta \bar{\varepsilon}^P}{\Delta t}$.

For more information on this formula, see the following reference

ref W.H. Drysdale and A.R. Zak, “Mechanics of Materials and Structural Theories – A Theory for Rate Dependent Plasticity”, *Computers and Structures*, Vol. 20, pp. 259-264, 1985.

Rate-dependent plasticity is specified using the MATSR entry, in which the MID number refers to the MATS1 entry for the material. The MATSR entry contains BVALUE (corresponds to b) and TSRATE (corresponds to $\dot{\varepsilon}_0$). MADSR also contains TID, the identification number of a TABLEST entry. If TID is specified, and the referenced TABLEST entry contains stress-strain curves at one or more strain rates, then Solution 601 calculates BVALUE by curve-fitting, overwriting any specified BVALUE. If TID is specified, and the referenced TABLEST entry contains stress-strain curves at two or more strain rates, then Solution 601 calculates BVALUE and TSRATE by curve-fitting, overwriting any specified BVALUE and TSRATE.

In the TABLEST entry, the Ti are interpreted as plastic strain-rates $\dot{\varepsilon}^P$, not as temperatures. In the TABLES1 entries referred to by the TIDi parameters, (xi, yi) are interpreted as (strain, stress).

The curve-fitting is done as follows: For each plastic strain rate $\dot{\varepsilon}^P$, the average overstress ratio $\dfrac{\sigma_y}{\sigma_y^0}$ is determined from the associated user-input stress-strain curve (in TABLES1). Then the curve-fitting is performed using the plastic strain rates and overstress ratios.

3.4.2 Plastic-cyclic material model

- This material model is based on

  - The von Mises yield condition (see p. 597, ref. KJB)
  - A flow rule using the von Mises yield function
  - An isotropic and/or kinematic hardening rule. The isotropic and kinematic hardening rules used in the plastic-cyclic model are suitable for use in modeling cyclic plasticity.

- The plastic-cyclic material model differs from the plastic-bilinear and plastic-multilinear material models because the isotropic and kinematic hardening rules are different. However, the plastic-cyclic material model can reproduce the plastic-bilinear and plastic-multilinear models when suitable material constants are chosen.

- This material model can be used with the **rod, 2-D solid, 3-D solid, Hermitian beam** and **shell** elements. However this material model cannot be used with the CQUAD8 shell element, or with the composite shell element.

- This model can be used with the **small displacement/small strain**, **large displacement/small strain** and **large displacement/large strain** formulations . Large displacement/large strain kinematics can only be used with the 2-D solid, 3-D solid and shell elements.

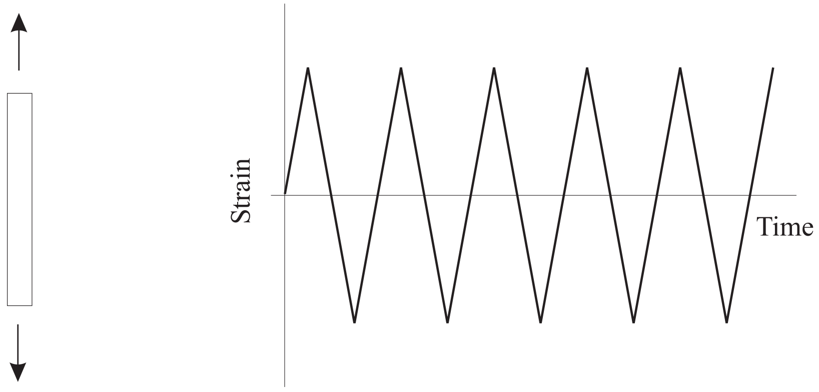
3.4.2.1 Fundamental concepts

- Many of the ideas used in the plastic-cyclic material model are given in the following reference:

ref J. Lemaitre and J.-L. Chaboche, *Mechanics of Solid Materials*, Cambridge University Press, 1990.

We abbreviate this reference as ref LC in the discussion below. As an aid to understanding the model, whenever our notation differs from the notation used in ref LC, we give the equivalent Lemaitre and Chaboche notation.

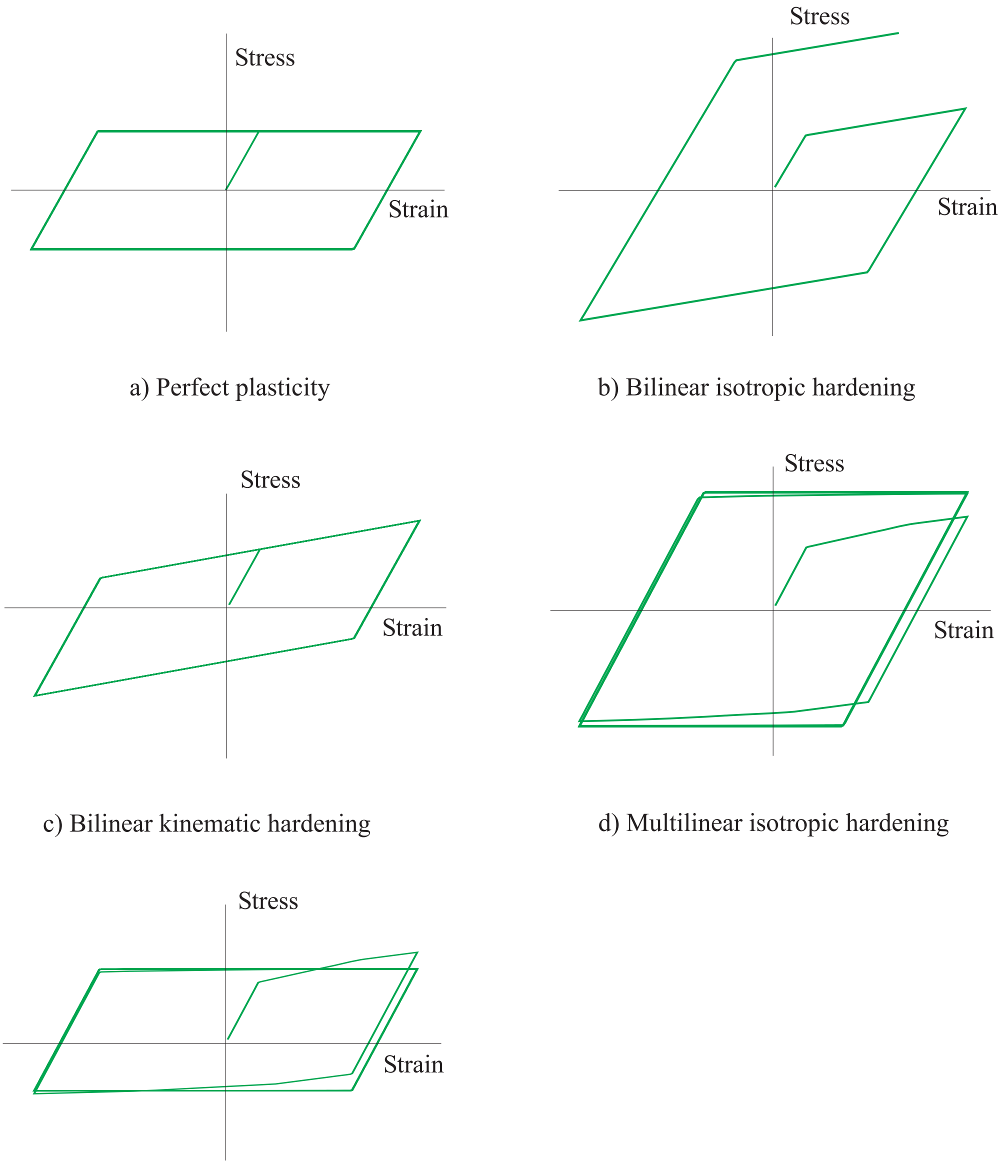
- The motivation for the plastic-cyclic material model is illustrated in Figs. 3.4-4 and 3.4-5. Fig. 3.4-4 shows a bar subjected to uniaxial cycling, with the strain prescribed. Fig. 3.4-5 shows response predictions from the plastic-bilinear and plastic-multilinear material models of Section 3.4.1. On repeated cyclic loading, perfect plasticity and multilinear hardening plasticity produce stabilized plastic cycles with no additional hardening. Bilinear isotropic hardening does not produce a stable plastic cycle and bilinear kinematic hardening produces a very rough approximation to a stable plastic cycle.



a) Uniaxial cycling of a bar

b) Prescribed strain time history

Fig. 3.4-4: Uniaxial cycling example



a) Perfect plasticity

b) Bilinear isotropic hardening

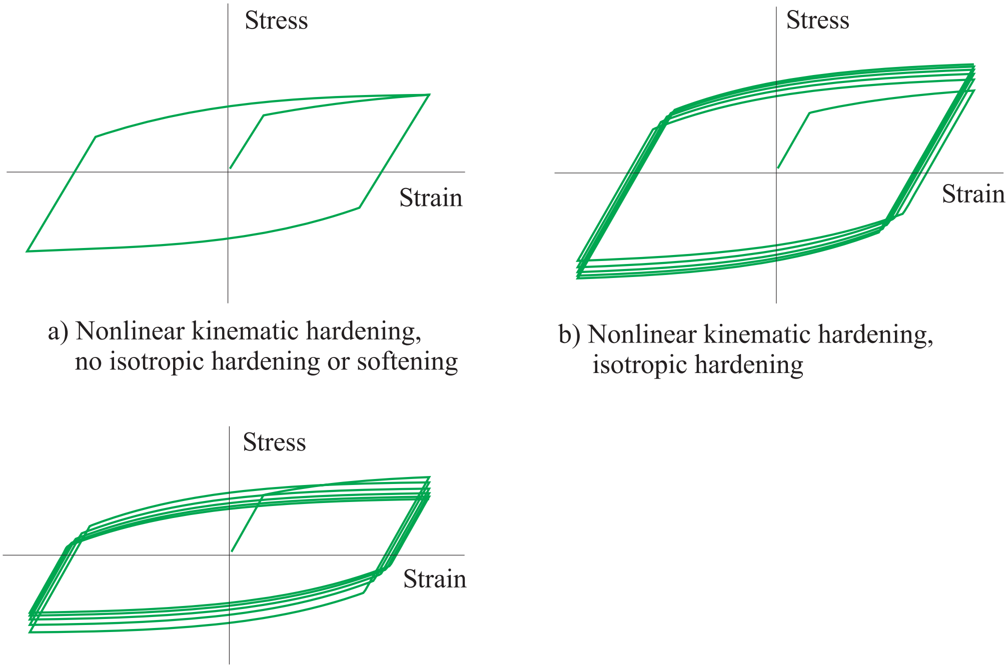
c) Bilinear kinematic hardening

d) Multilinear isotropic hardening

e) Multilinear kinematic hardening

Fig. 3.4-5: Response predictions using the plastic-bilinear and plastic-multilinear models

- Response predictions from the plastic-cyclic material model are illustrated in Fig. 3.4-6. When nonlinear kinematic hardening is used without isotropic hardening, a stable plastic cycle is reached after one cycle. In this stable plastic cycle, the transition from elastic to plastic conditions occurs more gradually than the corresponding transition from bilinear kinematic hardening. Cyclic hardening and cyclic softening are obtained by combining the nonlinear kinematic hardening with isotropic hardening or softening.



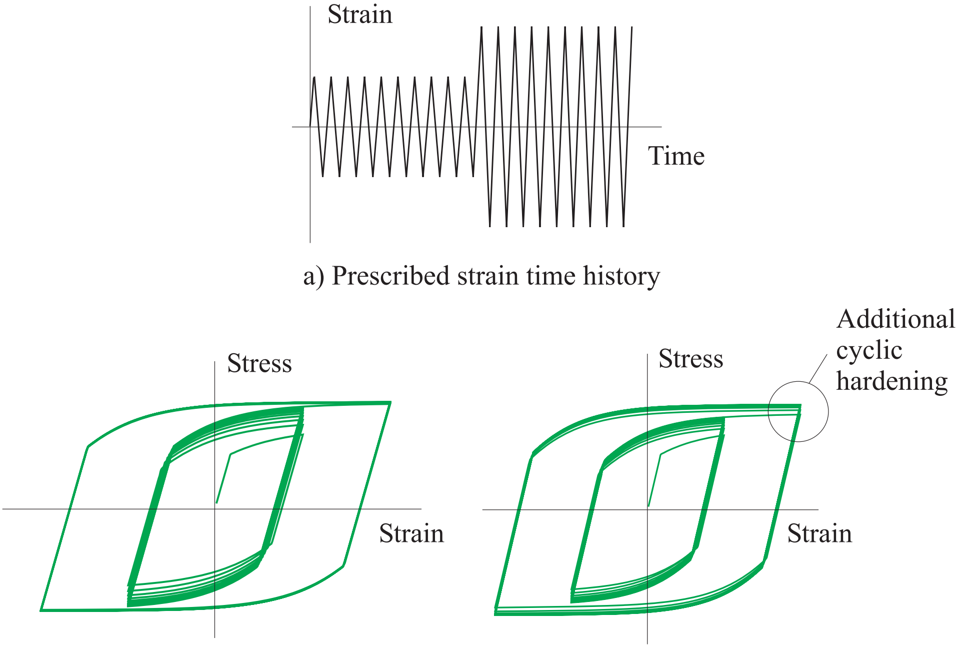
a) Nonlinear kinematic hardening, no isotropic hardening or softening

b) Nonlinear kinematic hardening, isotropic hardening

c) Nonlinear kinematic hardening, isotropic softening

Fig. 3.4-6: Response predictions using the plastic-cyclic material model

- The plastic-cyclic material model can optionally contain a strain memory surface. The motivation for using the strain memory surface is shown in Fig. 3.4-7. Considering an increase in the prescribed strain amplitude, if no strain memory surface is used, then no additional cyclic hardening takes place, whereas if a strain memory surface is used, additional cyclic hardening takes place.



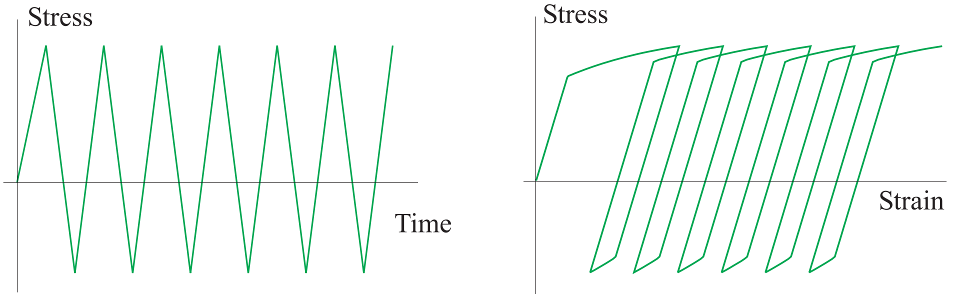
a) Prescribed strain time history

b) Stress-strain curve without strain memory surface

c) Stress-strain curve with strain memory surface

Fig. 3.4-7: Response predictions with and without strain memory surface

- Ratchetting occurs when prescribed stresses with non-zero mean stress are applied to the bar, as shown in Fig. 3.4-8.



a) Prescribed stress time history

b) Stress-strain curve

Fig. 3.4-8: Ratchetting using the plastic-cyclic material model

Stresses, strains, stress-strain law:

$$^{t+\Delta t}\boldsymbol{\tau} = {}^{t+\Delta t}\mathbf{s} + {}^{t+\Delta t}\tau_m \mathbf{I}$$

where $^{t+\Delta t}\boldsymbol{\tau}$ is the stress tensor ($\boldsymbol{\sigma}$ in ref LC), $^{t+\Delta t}\mathbf{s}$ is the deviatoric stress tensor ($\boldsymbol{\sigma}'$ in ref LC) and $^{t+\Delta t}\tau_m$ is the mean stress (hydrostatic stress σ_H in ref LC).

$$^{t+\Delta t}\mathbf{e} = {}^{t+\Delta t}\mathbf{e}' + {}^{t+\Delta t}e_m \mathbf{I}$$

where $^{t+\Delta t}\mathbf{e}$ is the strain tensor ($\boldsymbol{\varepsilon}$ in ref LC), $^{t+\Delta t}\mathbf{e}'$ is the deviatoric strain tensor ($\boldsymbol{\varepsilon}'$ in ref LC) and $^{t+\Delta t}e_m$ is the mean strain (hydrostatic strain ε_H in ref LC).

$$^{t+\Delta t}\tau_m = 3\kappa\, {}^{t+\Delta t}e_m$$
$$^{t+\Delta t}\mathbf{s} = 2G({}^{t+\Delta t}\mathbf{e}' - {}^{t+\Delta t}\mathbf{e}^P)$$

where κ and G are the bulk modulus and shear modulus, and $^{t+\Delta t}\mathbf{e}^P$ is the plastic strain ($\boldsymbol{\varepsilon}^P$ in ref LC). Thermal strains are not included in the above equations, but are included in the program when there are thermal effects.

Yield condition: The von Mises yield condition is

$$ {}^{t+\Delta t}f_y = \frac{1}{2}\left\| {}^{t+\Delta t}\mathbf{s} - {}^{t+\Delta t}\boldsymbol{\alpha} \right\|^2 - \frac{1}{3}\,{}^{t+\Delta t}\sigma_y^2 = 0 $$

where ${}^{t+\Delta t}\boldsymbol{\alpha}$ is the back stress tensor ($\mathbf{X}$ and $\mathbf{X}'$ in ref LC, note that back stress is always deviatoric) and ${}^{t+\Delta t}\sigma_y$ is the yield stress ($\sigma_Y + R$ or k in ref LC). The norm of a symmetric tensor $\mathbf{a}$ is defined as $\left\|\mathbf{a}\right\| = \sqrt{\mathbf{a}:\mathbf{a}}$. The yield condition is always evaluated at time $t+\Delta t$.

Fig. 3.4-9 shows the yield condition.

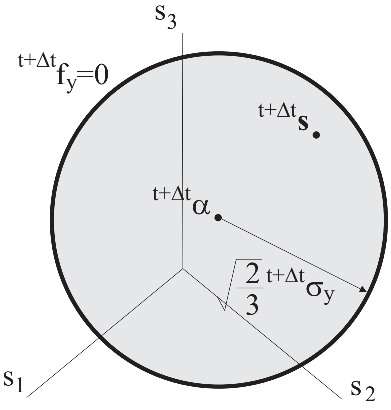


Fig. 3.4-9: von Mises yield surface in principal deviatoric stress space

Flow rule: The flow rule states that the direction of plastic strain increments is normal to the yield surface.

$$ \text{directions of } \boldsymbol{\Delta}\mathbf{e}^{\mathbf{P}} = \text{directions of } \left({}^{t+\beta\Delta t}\mathbf{s} - {}^{t+\beta\Delta t}\boldsymbol{\alpha} \right) $$

where $\boldsymbol{\Delta}\mathbf{e}^{\mathbf{P}}$ is the increment in plastic strain, and β is a constant used to choose the yield surface configuration ($\beta = 0$ corresponds to the configuration at time t, $\beta = 1$ corresponds to the configuration at time $t + \Delta t$, other values of β correspond to intermediate configurations). (The concept is similar to that used in alpha-integration; we would have used α instead of β, except that α is used for the back stresses.) The directions of a symmetric tensor $\mathbf{a}$ are defined as $\text{directions of } \mathbf{a} = \frac{\mathbf{a}}{\|\mathbf{a}\|}$.

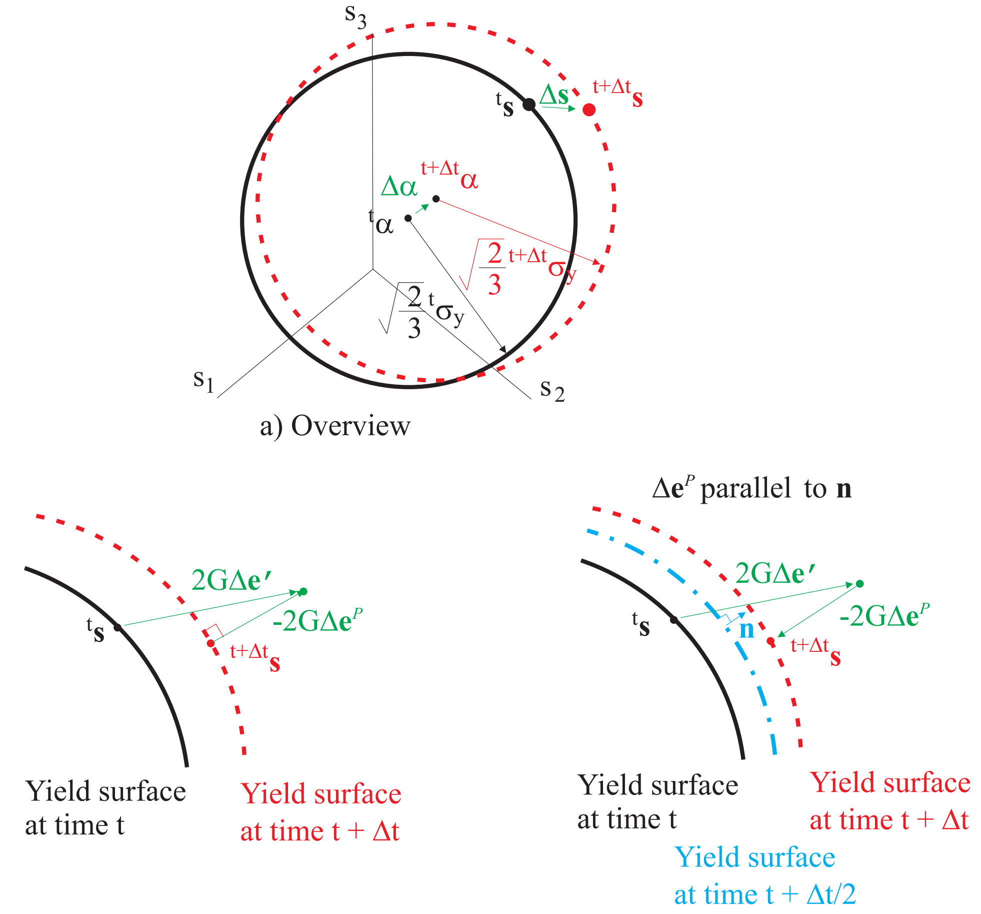
Fig. 3.4-10 shows the evolution of the yield surface using the stress-strain law, yield condition and flow rule.

Strain memory surface: The memory-exponential isotropic hardening rule uses the concept of a strain memory surface with additional internal variables ${}^{t+\Delta t}\boldsymbol{\xi}$ and ${}^{t+\Delta t}q$. The strain memory surface is now briefly described.

The strain memory surface is defined in the space of plastic strains as the surface of a sphere centered at position ${}^{t+\Delta t}\boldsymbol{\xi}$ with radius $\sqrt{\frac{3}{2}}\,{}^{t+\Delta t}q$. (In ref LC, ${}^{t+\Delta t}\boldsymbol{\xi}$ is written $\boldsymbol{\zeta}$). This surface can be written as

$$f_m = \frac{2}{3}\left\| {}^{t+\Delta t}\mathbf{e}^{\mathbf{P}} - {}^{t+\Delta t}\boldsymbol{\xi} \right\|^2 - {}^{t+\Delta t}q^2 = 0$$

Fig. 3.4-11 shows the strain memory surface.

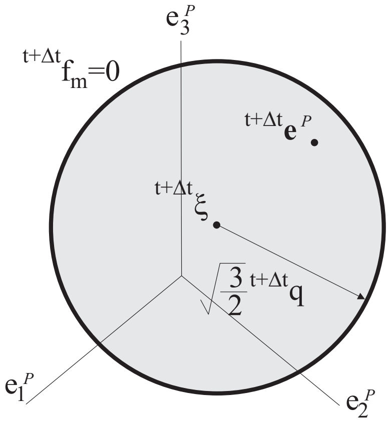

s_3
${}^{t}\mathbf{s}$
$\Delta\mathbf{s}$
${}^{t+\Delta t}\mathbf{s}$
$\Delta\alpha$
${}^{t+\Delta t}\alpha$
${}^{t}\alpha$
$\sqrt{\frac{2}{3}}\,{}^{t+\Delta t}\sigma_y$
$\sqrt{\frac{2}{3}}\,{}^{t}\sigma_y$
s_1
s_2
$2G\Delta\mathbf{e}'$
${}^{t}\mathbf{s}$
$-2G\Delta\mathbf{e}^P$
${}^{t+\Delta t}\mathbf{s}$
Yield surface at time t
Yield surface at time t + Δt
$\Delta\mathbf{e}^P$ parallel to $\mathbf{n}$
$2G\Delta\mathbf{e}'$
${}^{t}\mathbf{s}$
$\mathbf{n}$
$-2G\Delta\mathbf{e}^P$
${}^{t+\Delta t}\mathbf{s}$
Yield surface at time t
Yield surface at time t + Δt
Yield surface at time t + Δt/2

a) Overview

b) $\beta=1$, Δe^P normal to yield surface at time t + Δt

c) $\beta=1/2$, Δe^P normal to yield surface at time t + Δt/2

Fig. 3.4-10: Incremental update of yield surface and plastic strains



e_3^P
${}^{t+\Delta t}f_m=0$
${}^{t+\Delta t}\mathbf{e}^P$
${}^{t+\Delta t}\xi$
$\sqrt{\frac{3}{2}}\,{}^{t+\Delta t}q$
e_1^P
e_2^P

Fig. 3.4-11: Strain memory surface in principal plastic strain space

The differential rules used for the evolution of the strain memory surface are

$$dq = \eta \left\langle (\text{directions of } \mathbf{de^P}) : (\text{directions of } ({}^t\mathbf{e^P} - {}^t\boldsymbol{\xi})) \right\rangle d\overline{e}^P$$

and

$$\text{directions of } \mathbf{d}\boldsymbol{\xi} = \text{directions of } ({}^t\mathbf{e^P} - {}^t\boldsymbol{\xi})$$

in which the symbol <> means that <*u*>=0 when *u*<0, and <*u*>=*u* when *u*>0. The incremental versions of these rules are

$$\Delta q = \eta \left\langle (\text{directions of } \boldsymbol{\Delta}\mathbf{e^P}) : (\text{directions of } ({}^{t+\beta\Delta t}\mathbf{e^P} - {}^{t+\beta\Delta t}\boldsymbol{\xi})) \right\rangle \Delta\overline{e}^P$$

and

$$\text{directions of } \boldsymbol{\Delta\xi} = \text{directions of } ({}^{t+\beta\Delta t}\mathbf{e^P} - {}^{t+\beta\Delta t}\boldsymbol{\xi})$$

Again, the β notation is used to denote a configuration between t and $t+\Delta t$. These concepts are illustrated in Fig. 3.4-12.

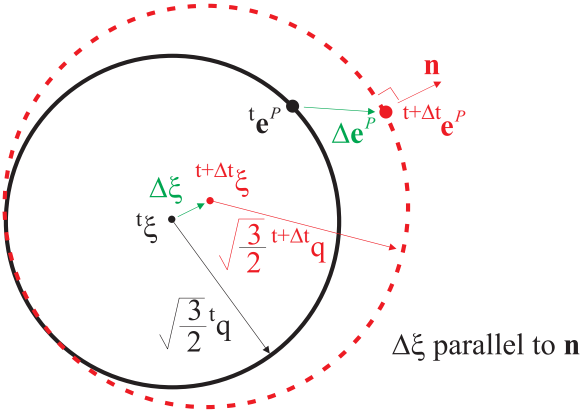

n
$^{t}\mathbf{e}^{P}$
$\Delta\mathbf{e}^{P}$
$^{t+\Delta t}\mathbf{e}^{P}$
$\Delta\xi$
$^{t+\Delta t}\xi$
$^{t}\xi$
$\sqrt{\frac{3}{2}}\,{}^{t+\Delta t}q$
$\sqrt{\frac{3}{2}}\,{}^{t}q$
$\Delta\xi$ parallel to **n**

Fig. 3.4-12: Incremental update of strain memory surface, for β=1

η is a material constant that must be between 0 and $\frac{1}{2}$. Typically $\eta = \frac{1}{2}$.

These rules ensure that the strain memory surface at time $t + \Delta t$ encloses the strain memory surface for all preceding times. The evolution of the strain memory surface in 1D uniaxial straining is shown in Fig. 3.4-13.

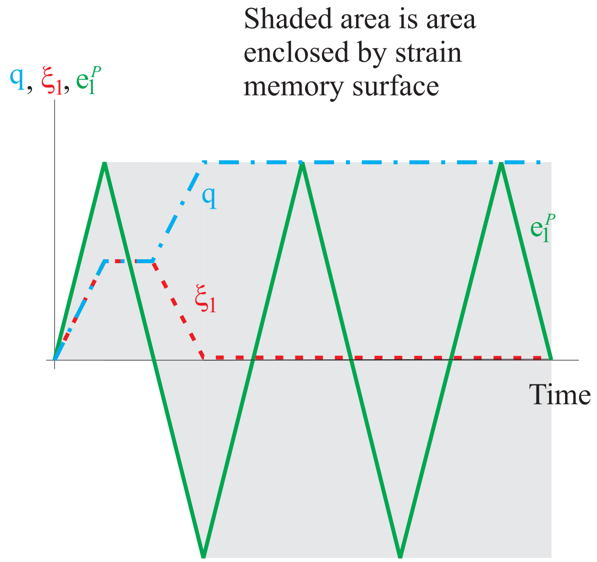


Fig. 3.4-13: Evolution of strain memory surface in uniaxial cyclic straining

Isotropic hardening rules: The isotropic hardening rules are shown in Fig. 3.4-14.

Bilinear hardening:

$$^{t+\Delta t}\sigma_y = {}^{0}\sigma_y + E_P \, {}^{t+\Delta t}\overline{e}^P$$

where ${}^{0}\sigma_y$ and E_P are material constants, and ${}^{t+\Delta t}\overline{e}^P$ is the accumulated effective plastic strain (p in ref LC). ${}^{t+\Delta t}\overline{e}^P$ is calculated using ${}^{t+\Delta t}\overline{e}^P = {}^{t}\overline{e}^P + \Delta\overline{e}^P$, where $\Delta\overline{e}^P = \sqrt{\frac{2}{3}\mathbf{\Delta e^P} : \mathbf{\Delta e^P}}$.

For bilinear hardening, $E_P = \dfrac{EE_T}{E - E_T}$ where E_T is the slope of the stress-strain curve during plasticity (this formula assumes no kinematic hardening).

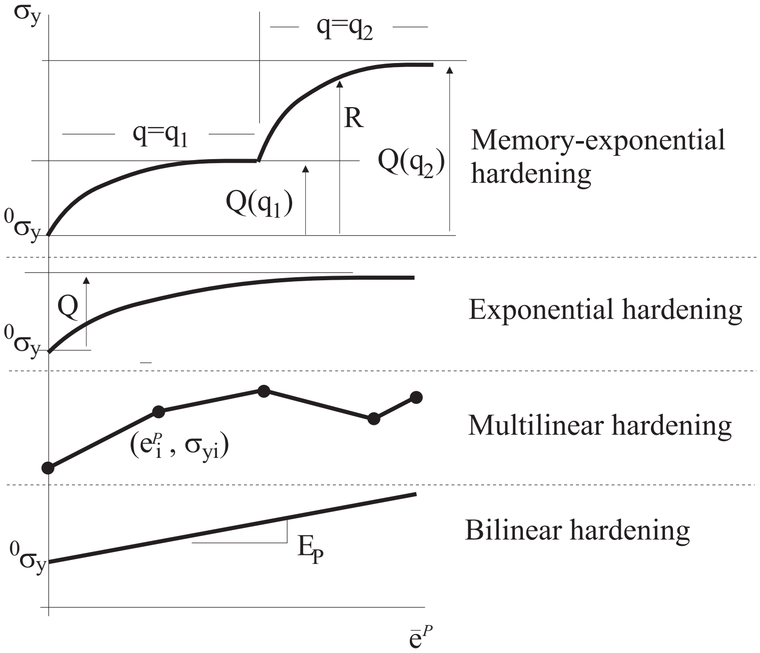


Fig. 3.4-14: Dependence of stress radius on accumulated effective plastic strain

Multilinear hardening:

Pairs of $(\overline{e}^P, \sigma_y)$ are given. It is allowed for σ_y to either increase or decrease as $\overline{e}^P$ increases.

If the multilinear hardening curve is given in terms of the uniaxial response (e, σ) where σ is the tensile stress corresponding to the unaxial strain e, then $\left(\overline{e}^P = e - \dfrac{\sigma}{E}, \sigma_y = \sigma\right)$ are the corresponding points for the points (e, σ). This formula assumes no kinematic hardening.

Exponential hardening:

$$ {}^{t+\Delta t}\sigma_y = {}^{0}\sigma_y + Q(1 - \exp(-b\, {}^{t+\Delta t}\overline{e}^P)) $$

where Q and b are material constants. Q can be positive to model cyclic hardening, and Q can be negative to model cyclic softening.

Memory-exponential hardening (exponential hardening with strain memory surface):

In this model, the yield surface size ${}^{t+\Delta t}\sigma_y$ depends on the strain memory size ${}^{t+\Delta t}q$ (see above for a description of the strain memory surface).

The yield surface size is

$${}^{t+\Delta t}\sigma_y = {}^{0}\sigma_y + {}^{t+\Delta t}R$$

where ${}^{t+\Delta t}R$ is the change in the size of the yield surface. ${}^{t+\Delta t}R$ is defined using the differential equation

$$dR = b({}^{t}Q - {}^{t}R)d\overline{e}^{P}$$

and is calculated using ${}^{t+\Delta t}R = {}^{t}R + \Delta R$, $\Delta R = b({}^{t+\Delta t}Q - {}^{t+\Delta t}R)\Delta\overline{e}^{P}$. ${}^{t+\Delta t}Q$ is the asymptotic change in yield surface size, and is calculated as

$${}^{t+\Delta t}Q = Q({}^{t+\Delta t}q) = Q_M - (Q_M - Q_0)\exp(-2\mu\,{}^{t+\Delta t}q)$$

where ${}^{t+\Delta t}q$ is the size of the strain memory surface.

The material constants for the memory-exponential material model are ${}^{0}\sigma_y$, b, Q_0, Q_M, b, μ and the strain memory surface parameter η.

Kinematic hardening rule: The kinematic hardening rule is:

Armstrong-Fredrick nonlinear kinematic hardening:

The back stress is expressed as a sum of partial back stresses

$$^{t+\Delta t}\boldsymbol{\alpha} = \sum {}^{t+\Delta t}\boldsymbol{\alpha}^{(m)}$$

where $^{t+\Delta t}\boldsymbol{\alpha}^{(m)}$ is partial back stress number *m*. All of the partial back stresses are independent of each other.

Each partial back stress evolves according to the differential rule

$$\mathbf{d}\boldsymbol{\alpha}^{(m)} = \frac{2}{3} h^{(m)} \mathbf{de}^{\mathbf{P}} - \zeta^{(m)}\, {}^{t}\boldsymbol{\alpha}^{(m)} d\overline{e}^{P}$$

where $h^{(m)}$ and $\zeta^{(m)}$ are material constants (these material constants are C_l and γ_l for partial back stress number *l* in ref LC). Assuming that the directions of plastic strain increments are constant during a time step, this can be integrated to obtain

$$^{t+\Delta t}\boldsymbol{\alpha}^{(m)} = \mathbf{A}^{(m)} - (\mathbf{A}^{(m)} - {}^{t}\boldsymbol{\alpha}^{(m)}) \exp(-\Delta b^{(m)})$$

where $\mathbf{A}^{(m)} = \sqrt{\frac{2}{3}} \frac{h^{(m)}}{\zeta^{(m)}} (\text{directions of } \boldsymbol{\Delta}\mathbf{e}^{\mathbf{P}})$, $\Delta b^{(m)} = \zeta^{(m)} \Delta\overline{e}^{P}$.

It is allowed to use one partial back stress, with $\zeta = 0$. Then linear kinematic hardening is recovered, with $E_p = h$ (this formula assumes no isotropic hardening).

Stress $-$plastic strain curve for uniaxial cycling

In uniaxial cycling, a typical stress – plastic strain curve is shown in Fig. 3.4-15. Here only one partial back stress is used with material constants h and ζ. Since a stable cycle is considered here, σ_y is taken from the isotropic hardening rule assuming a very large value of $\overline{e}^{P}$. The hardening modulus is given by

$E_P = h - \zeta\left(\sigma - \sigma_y\right)$ on segment A-B. The size of the plastic strain cycle can be related to the material constants using the relation

$$\frac{\Delta\sigma}{2} = \frac{h}{\zeta}\tanh\left(\zeta\frac{\Delta e^P}{2}\right) + \sigma_y$$

Thus, given several cycles for different cyclic strains, the material constants can be estimated using the above formula.

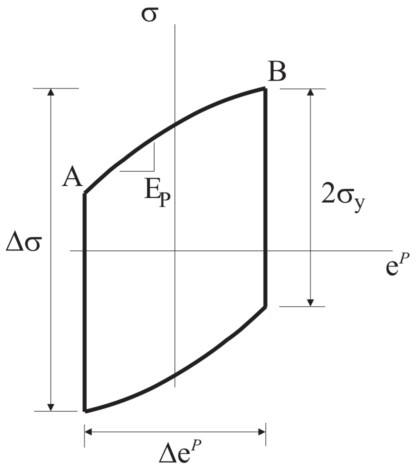


Fig. 3.4-15: Stable uniaxial plastic cycle using one term in Armstrong-Fredrick nonlinear kinematic hardening rule

Stress − plastic strain curve for initial loading

During initial loading, it can be shown that

$$\sigma = \sigma_y + \frac{h}{\zeta}\left(1 - \exp\left(\zeta e^P\right)\right)$$

The corresponding stress – plastic strain curve is shown in Fig. 3.4-16. Here only one partial back stress is used with material constants h and ζ. In this formula it is assumed that there is no isotropic hardening, and that the yield stress is σ_y. Hence the hardening observed during initial loading can be modeled using nonlinear kinematic hardening, without the use of isotropic hardening.

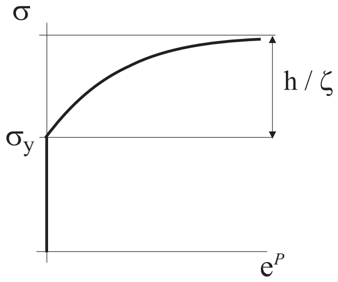


Fig 3.4-16: Stress - plastic strain curve during initial loading, using nonlinear kinematic hardening

Combination of isotropic and kinematic hardening rules: By combining the Armstrong-Fredrick nonlinear kinematic hardening rule with the isotropic hardening rules, a wide variety of cyclic phenomena can be simulated, such as cyclic hardening and softening, shakedown and ratcheting. See ref LC for details.

Stress-strain integration: When plasticity is detected, the incremental plastic strains are solved for using an iterative solution procedure. The maximum number of iterations can be specified, and the tolerance used in assessing convergence can be specified.

Constitutive tensor: The constitutive tensor (stress-strain matrix) is constructed to be tangent. However, in general, the tangent constitutive tensor is non-symmetric, therefore the constitutive tensor is symmetrized.

Formulations: When used with the small displacement/small strain formulation, a materially-nonlinear-only formulation is employed, when used with the large displacement/small strain formulation, a TL formulation is employed, and when used with the large displacement/large strain formulation, either a ULH or ULJ formulation is employed.

If geometrically nonlinear effects are to be included, the large displacements/large strain kinematics are preferred to the large displacement/small strain kinematics, even when the strains are numerically small. The large displacement/small strain kinematics should be used only when the large displacement/large strain kinematics are not supported by the element.

Mixed displacement-pressure formulation: Plane strain, axisymmetric or 3-D solid elements that reference this material model should use the mixed displacement-pressure (u/p) element formulation. This is because the plastic strains are incompressible. This is done by setting UPFORM = 1 in the NXSTRAT command.

Thermal strains: If a thermal load is applied to the structure, the thermal strains are taken into account but the material characteristics are considered to be temperature independent.

Modeling of rupture: Rupture conditions can also be included with this model.

The maximum accumulated effective plastic strain can be specified for the rupture condition. When rupture is reached at a given element integration point, the corresponding element is removed from the model (see Section 10.4).

3.4.2.2 Specification of input

MATPLCY entry

Parameters ISO, KIN, RUP in this entry reference data defined in entries PLCYISO, PLCYKIN, PLCYRUP respectively. These entries provide the input for the isotropic, kinematic and rupture parts of the plastic-cyclic model, see below for a detailed description of these commands.

It is allowed to not specify a value for KIN or RUP. Then these effects are not included in the model.

BETA is the integration factor β described above. When BETA is blank (the default), the program chooses the value of β as follows: β=1 for static or implicit time integration (Solution 601); β=0 for explicit time integration (Solution 701). MAXITE is the maximum number of iterations used to solve for the incremental plastic strains. RTOL is a tolerance used to assess convergence of the iterations. RTOL can be thought of as a reference incremental plastic strain.

PLCYISO entry

Parameter TYPE=BLIN, MTLIN, EXP, MEMEXP specifies the isotropic hardening type as bilinear, multilinear, exponential or memory-exponential respectively. The remaining parameters of this entry specify the material constants for the isotropic hardening.

PLCYKIN entry

The parameters of this entry are used to specify the constants for the kinematic part of the plastic-cyclic model.

PLCYRUP entry

The parametrs of this entry are used to specify the constants for the rupture part of the plastic-cyclic model. Rupture is based on the accumulated effective plastic strain.

Conversion formulas

Plastic-bilinear to plastic-cyclic:

In general, given a plastic-bilinear material with given yield stress and tangent modulus ET, this material can be converted into an equivalent plastic-cyclic material with bilinear isotropic hardening, the same yield stress, and plastic modulus $\text{EP} = \dfrac{(\text{E})(\text{ET})}{\text{E - ET}}$.

Plastic-multilinear to plastic-cyclic:

In general, given a plastic-multilinear material with Young's modulus E and stress-strain points

strain(i) stress(i)

this material can be converted into an equivalent plastic-cyclic material with multilinear isotropic hardening and multilinear points aeps(i) sr(i)

using the formulas

$$\text{aeps(i)} = \text{strain(i)} - (\text{stress(i)}/\text{E}), \quad \text{sr(i)} = \text{stress(i)}$$

Note that aeps at the elastic limit will always be zero.

3.5 Temperature-dependent elastic material models

- The thermal isotropic and thermal orthotropic material models are discussed in this section.

  The thermal isotropic material is obtained with the MAT1 and MATT1 material entries.

  The thermal orthotropic material is obtained for surface elements with the MAT2 and MATT2 material entries, or MAT8 and MATT8 material entries; for 2-D elements with MAT3 and MATT3 material entries; and for solid elements with the MAT9 and MATT9 material entries, or MAT11 and MATT11 material entries.

  These commands allow the different elastic material constants to vary with temperature. Thermal strains are taken into account in these materials.

- The thermal isotropic model is available for the **rod**, **2-D solid**, **3-D solid** and **shell** elements.

- The thermal orthotropic model is available for the **2-D solid**, **3-D solid** and **shell** elements.

- Both models can be used with **small displacement/small strain** and **large displacement/small strain** kinematics. The strains are always assumed to be small.

  When used with small displacement/small strain kinematics, a materially-nonlinear-only formulation is employed.

  When used with large displacement/small strain kinematics, either the TL or UL formulation is employed. 2-D, 3-D solids and shells use the TL formulations, and rods use a UL formulation.

- In the data input for the analysis, the nodal point temperatures must be defined for all time steps. See Section 5.6.

- For these models, the elastic moduli, the shear moduli, the Poisson's ratios and the coefficients of thermal expansion defined in Section 3.2 are input as piecewise linear functions of the temperature, as illustrated in Fig. 3.5-1. Linear interpolation is used to calculate the material properties between input points.

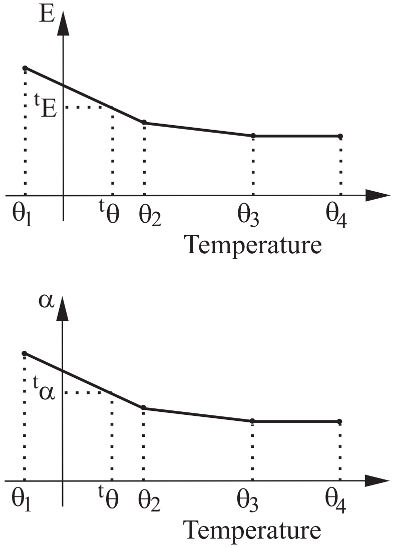


Fig. 3.5-1: Variation of material properties for thermo-elastic model

- The calculation of thermal strains is described in Section 3.1.6.

- Note that if the material constants are all temperature independent, and the material is isotropic, then thermal strains could alternatively have been modeled using the elastic isotropic material (non-thermal) of Section 3.2.

- For the evaluation of the temperatures ${}^{t}\theta$ and ${}^{0}\theta$ at the integration point considered, the isoparametric interpolation functions h_i are used; e.g., in two-dimensional analysis we have

$$\theta = \sum_{i=1}^{q} h_i \theta_i$$

where θ_i is the temperature at element nodal point i (see Fig.

3.5-2). Note that when higher-order elements are used the temperatures at the integration point can be significantly different from the values at the nodal point (for example negative although all nodal point temperatures are greater than or equal to zero).

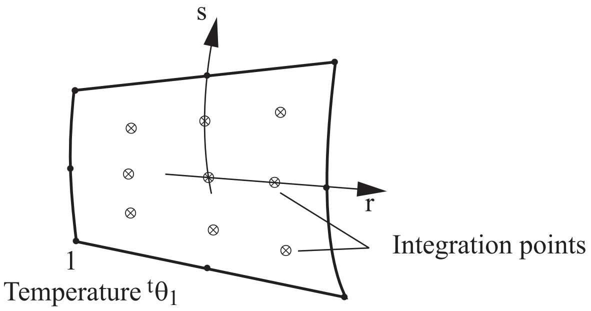


Fig. 3.5-2: Interpolation of temperature at integration points

- For shell elements, the temperature at an integration point is established using the temperatures at the midsurface nodes and the temperature gradients. Note that the temperature gradient at a shell midsurface node is defined in the direction of the mid-surface director vector at that node, see Section 2.3.3.

3.6 Thermal elasto-plastic and creep material models

- This section groups together thermal elasto-plastic materials and creep materials, since a unified general solution can be applied to these material types. The computational procedure is based on the effective stress function algorithm, detailed in Section 3.6.4.

ref. KJB Section 6.6.3

- The thermal elasto-plastic and creep models include the effects of

  - Isotropic elastic strains, via the MAT1 entry
  - Thermal strains, ${}^t e_{rs}^{TH}$, via the MATT1 or the MAT1 entries.
  - Time-independent plastic strains, ${}^t e_{rs}^{p}$, via the MATS1 entry
  - Time-dependent creep strains, ${}^t e_{rs}^{C}$, via the CREEP entry, the CREEP and MATTC entries, or the MATCRP entry.

- The constitutive relation used is

$$ {}^{t}\sigma_{ij} = {}^{t}C^{E}_{ijrs} \left({}^{t}e_{rs} - {}^{t}e^{P}_{rs} - {}^{t}e^{C}_{rs} - {}^{t}e^{TH}_{rs} \right) \qquad (3.6\text{-}1) $$

where ${}^{t}\sigma_{ij}$ is the stress tensor at time *t* and ${}^{t}C^{E}_{ijrs}$ is the elasticity tensor at the temperature corresponding to time *t*. The tensor ${}^{t}C^{E}_{ijrs}$ can be expressed in terms of Young's modulus ${}^{t}E$ and Poisson's ratio ${}^{t}\nu$ both of which may be temperature-dependent.

- Note that the thermal, plastic and creep parts of these material models are optional. If, however, the omitted strain components result in one of the material models detailed in one of the previous sections, then the program will select that material model.

- The formulations provided in this section are very general, and can describe any material combining elastic, plastic, thermal and creep strains. The combinations given in Table 3.6-1 are allowed.

- These material models can be used with the **rod**, **2-D solid**, **3-D solid**, and **shell** elements.

- These models can be used with **small displacement/small strain**, **large displacement/small strain** and **large displacement/large strain** kinematics.

When used with small displacement/small strain kinematics, a materially-nonlinear-only formulation is employed.

When used with large displacement/small strain kinematics, either a TL or a UL formulation is employed (TL for 2-D and 3-D solids and shells, and UL for rods).

When used with large displacement/large strain kinematics, the ULH (updated Lagrangian Hencky) formulation is employed. This is only supported for 2-D solid and 3-D solid elements.

- If geometrically nonlinear effects are to be included, the large displacements/large strain kinematics are preferred to the large displacement/small strain kinematics, even when the strains are numerically small. The large displacement/small strain kinematics should be used only when the large displacement/large strain kinematics are not supported by the element.

Table 3.6-1: Combinations of elastic, plastic and creep strains

Description	Elastic	Plastic	Creep	Bulk data entries
Elastic creep	Yes	No	Yes	MAT1, CREEP, or MAT1, MATCRP
Thermal elastic creep	Yes	No	Temp-dep	MAT1, CREEP, MATTC, or MAT1, MATCRP
Thermal elasto-plastic	Temp-dep	Yes	No	MAT1, MATT1, MATS1, with TID in MATS1 pointing to a TABLES1 entry
	Yes	Temp-dep	No	MAT1, MATS1, with TID in MATS1 pointing to a TABELST entry
	Temp-dep	Temp-dep	No	MAT1, MATT1, MATS1, with TID in MATS1 pointing to a TABELST entry
Plastic-creep	Yes	Yes	Yes	MAT1, MATS1, CREEP, with TID in MATS1 pointing to a TABLES1 entry
	Temp-dep	Yes	Yes	MAT1, MATT1, MATS1, CREEP, with TID in MATS1 pointing to a TABLES1 entry
Thermal plastic-creep	Yes	Temp-dep	Yes	MAT1, MATS1, CREEP, with TID in MATS1 pointing to a TABLEST entry
	Temp-dep	Temp-dep	Yes	MAT1, MATT1, MATS1, CREEP, with TID in MATS1 pointing to a TABLEST entry
	Yes	Yes	Temp-dep	MAT1, MATS1, CREEP, MATTC, with TID in MATS1 pointing to a TABLES1 entry
	Temp-dep	Yes	Temp-dep	MAT1, MATT1, MATS1, CREEP, MATTC, with TID in MATS1 pointing to a TABLES1 entry
	Yes	Temp-dep	Temp-dep	MAT1, MATS1, CREEP, MATTC, with TID in MATS1 pointing to a TABLEST entry
	Temp-dep	Temp-dep	Temp-dep	MAT1, MATT1, MATS1, CREEP, MATTC, with TID in MATS1 pointing to a TABLEST entry

Notes:

1. "No" means that this strain is not included in the material. "Yes" means that this strain is included in the material description, and that the material constants for this strain are temperature-independent. "Temp-dep" means that this strain is included in the material description, and that the material constants for this strain are temperature-dependent.

2. Instead of using TID in MATS1 pointing to a TABLES1 entry, it is also allowed to specify H and LIMIT1 in MATS1. The resulting material uses bilinear hardening, however the yield stress and hardening modulus are temperature-independent.

- Plane strain, axisymmetric or 3-D solid elements that reference these material models should preferably employ the mixed u/p element formulation. This is done by setting UPFORM=1 in the NXSTRAT entry.

- Note that the constitutive relations for the thermal, plastic and creep strains are independent of each other; hence the only interaction between the strains comes from the fact that all strains affect the stresses according to Eq. 3.6-1. Fig. 3.6-1 summarizes the constitutive description for a one-dimensional stress situation and a bilinear stress-strain curve.

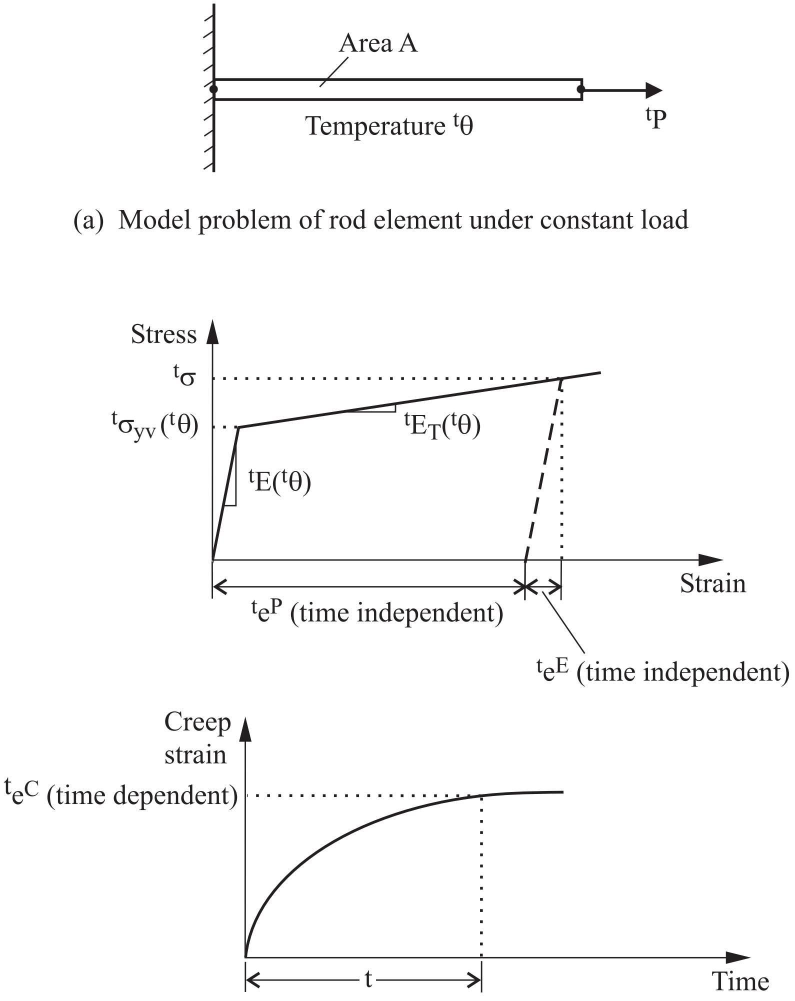
- Since there is no direct coupling in the evaluation of the different strain components, we can discuss the calculation of each strain component independently.

ref. M.D. Snyder and K.J. Bathe, "A Solution Procedure for Thermo-Elastic-Plastic and Creep Problems," *J. Nuclear Eng. and Design,* Vol. 64, pp. 49-80, 1981.

ref. M. Kojić and K.J. Bathe, "The Effective-Stress-Function Algorithm for Thermo-Elasto-Plasticity and Creep," *Int. J. Numer. Meth. Engng.*, Vol. 24, No. 8, pp. 1509-1532, 1987.

- In multilinear plasticity, the rupture plastic strain corresponds to the effective plastic strain at the last point input for the stress-strain curve.

- When rupture is reached at a given element integration point, the corresponding element is removed from the model (see Section 10.4).



(a) Model problem of rod element under constant load

(b) Strains considered in the model

Fig. 3.6-1: Thermo-elasto-plasticity and creep constitutive description in one-dimensional analysis

3.6.1 Evaluation of thermal strains

- The thermal strains are calculated as described in Section 3.1.6.

3.6.2 Evaluation of plastic strains

- Plasticity effects are included in the thermal elasto-plastic material model and is based on the von Mises yield criterion, an associated flow rule, isotropic or kinematic hardening (no mixed hardening), and bilinear or multilinear stress-strain curves (based on the H, LIMIT1 and TID fields in MATS1).

- The plastic strains are calculated using the von Mises plasticity model (see Section 3.4) with temperature-dependent material parameters (Young's modulus, Poisson's ratio, stress-strain curves, ...).

- The yield function is, for the case of isotropic hardening

$$ {}^{t}f_{y} = \frac{1}{2}\,{}^{t}\mathbf{s}\cdot{}^{t}\mathbf{s} - \frac{1}{3}\,{}^{t}\sigma_{y\mathrm{v}}^{2} $$

and for the case of kinematic hardening

$$ {}^{t}f_{y} = \frac{1}{2}\left({}^{t}\mathbf{s} - {}^{t}\boldsymbol{\alpha}\right)\cdot\left({}^{t}\mathbf{s} - {}^{t}\boldsymbol{\alpha}\right) - \frac{1}{3}\,{}^{t}\sigma_{y\mathrm{v}}^{2} $$

where ${}^{t}\mathbf{s}$ is the deviatoric stress tensor, ${}^{t}\sigma_{y\mathrm{v}}$ is the virgin yield stress corresponding to temperature ${}^{t}\theta$ and ${}^{t}\boldsymbol{\alpha}$ is the shift of the stress tensor due to kinematic hardening.

- The expressions for plastic strain increments resulting from the flow theory are $de_{ij}^{P} = d\lambda\,{}^{t}s_{ij}$ for isotropic hardening and $de_{ij}^{P} = d\lambda\left({}^{t}s_{ij} - {}^{t}\alpha_{ij}\right)$ for kinematic hardening, in which $d\lambda$ is the plastic multiplier (positive scalar) which can be determined from the yield condition ${}^{t}f_{y} = 0$. In the case of kinematic hardening, we express the change of the yield surface position in the form

$$d\alpha_{ij} = {}^{t}C de_{ij}^{P}$$

where ${}^{t}C$ is the modulus

$${}^{t}C = \frac{2}{3}\frac{{}^{t}E\,{}^{t}E_{T}}{{}^{t}E - {}^{t}E_{T}} = \frac{2}{3}\,{}^{t}E_{P}$$

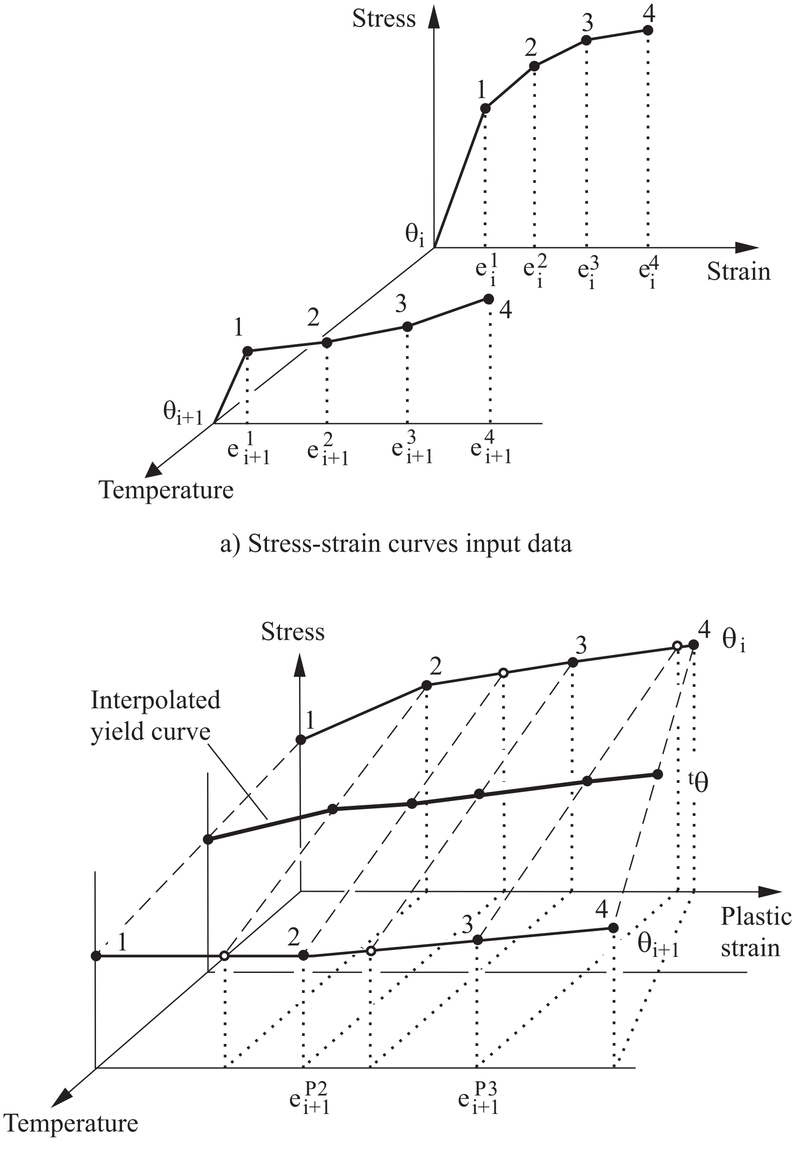
In the case of multilinear yield curves, ${}^{t}E_{T}$ represents the tangent modulus of the segment on the stress - total strain yield curve corresponding to the accumulated effective plastic strain ${}^{t}\overline{e}^{P}$, and ${}^{t}E_{P}$ represents the tangent modulus of the segment on the stress - plastic strain yield curve corresponding to the accumulated effective plastic strain ${}^{t}\overline{e}^{P}$.

- When H and LIMIT1 are specified in MATS1, bilinear hardening plasticity is assumed, in which only the elastic material parameters can be temperature dependent (Young's modulus, Poisson's ratio and coefficient of thermal expansion). The yield stress and hardening modulus are temperature independent.

- When TID in MATS1 points to a TABLES1 entry, multilinear hardening plasticity is assumed, in which only the elastic material parameters can be temperature dependent. The yield curve is temperature independent.

- When TID in MATS1 points to a TABLEST entry, multilinear hardening plasticity is assumed, in which the elastic material parameters, and also the yield curves, can be temperature dependent

  The yield curves are interpolated as shown in Fig. 3.6-2.



a) Stress-strain curves input data

b) Yield curves

Fig. 3.6-2: Interpolation of multilinear yield curves with temperature

- The formula $d\alpha_{ij} = {}^{t}Cde_{ij}^{P}$ used in kinematic hardening requires modification to avoid nonphysical effects when ${}^{t}E_{P}$, and hence ${}^{t}C$, is a function of temperature.

- Here is an example showing a nonphysical effect that occurs when the formula $d\alpha_{ij} = {}^{t}Cde_{ij}^{P}$ is used without modification. Consider the following material description, in which kinematic hardening is used:

Temperature	E	ν	σ_{yv}	E_T	E_P
0	1E6	0	100	1996.008	2000
200	1E6	0	100	999.001	1000

The yield curves for this material are shown in Fig. 3.6-3. Take a uniaxial specimen, set the temperature to 0 and load with prescribed force until the uniaxial stress is 130. The corresponding plastic strain is $e^{P} = 0.015$ (since $100 + 2000e^{P} = 130$). This point is labeled A in Fig. 3.6.3.

Now, without changing the prescribed force, change the temperature to 200. For the plastic strain $e^{P} = 0.015$, the yield stress is $100 + 1000e^{P} = 115$, which is less than the stress of 130. So we would expect that the material would plastically deform further until the (plastic strain, stress) point is on the yield curve corresponding to temperature 200 (point B in Fig. 3.6-4). However the plastic strain does not change and the current stress remains above the yield curve for temperature 200 (point A).

The reason for this unexpected behavior is as follows: The yield condition involves the initial yield stress, which is unchanged, the current stress, which is unchanged and the back stress, which is also unchanged (if the back stress had changed, then the yield condition would not be satisfied). Since the change in plastic strain is related to the change in back stress through $d\alpha_{ij} = {}^{t}Cde_{ij}^{P}$, because the back stress does not change, the plastic strain does not change either.

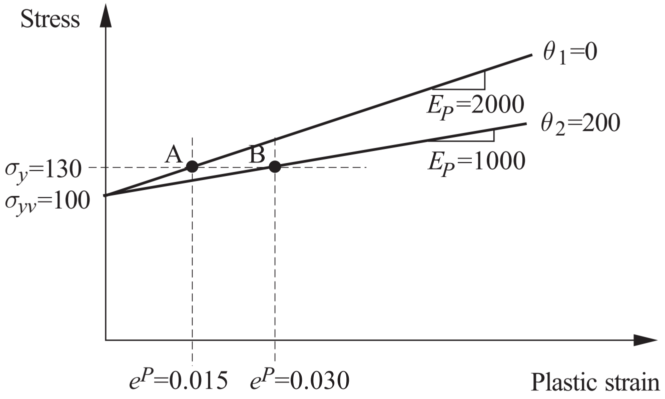


Fig. 3.6-3: Example showing nonphysical results in kinematic hardening when E_P is a function of temperature

- Chaboche has discussed the formulation of kinematic hardening under varying temperature conditions, and has pointed out other nonphysical effects when the above back stress evolution rule is used in varying temperature conditions, see Section 4.4 of the following reference:

ref. J.L. Chaboche, “Cyclic viscoplastic constitutive equations, part I: A thermodynamically consistent formulation”, *J. Appl. Mech*, December 1993, Vol 60, pp 813-821.

When the yield curve is a straight line for each temperature (that is, when ${}^tC$ is a function of temperature but not of plastic strain), Chaboche suggests a rule equivalent to

$${}^t\alpha_{ij} = {}^tC\,{}^te_{ij}^P$$

which in differential form becomes

$$d\alpha_{ij} = {}^tC\,de_{ij}^P + {}^te_{ij}^P dC$$

It is seen that the first term on the right-hand-side corresponds to the formulation presented above. The second term on the right-hand side corresponds to a “back stress temperature correction”, since this term is nonzero only if the temperature is varying.

When the yield curve is not a straight line for each temperature (that is, when ${}^{t}C$ is a function of temperature and of plastic strain), we again include a temperature correction corresponding to the second term on the right-hand side, but in which dC in the second term is evaluated only due to changes in temperature.

When the temperature correction is included in the above example, the plastic strain in the above example increases to point B when the temperature is increased to 200, so that the current stress remains on the yield curve. This response is the expected behavior.

- The back stress temperature correction can be selected using the BSTC parameter of the NXSTRAT entry. BSTC=0 means to not use the back stress temperature correction (the default). BSTC=1 means to use the back stress temperature correction.

3.6.3 Evaluation of creep strains

- Two creep laws are currently available in Solution 601. The first called the Power creep law is obtained by setting TYPE = 300 in the CREEP material entry, or by setting TYPE = 301 in the MATCRP material entry. The second creep law called the Exponential creep law is obtained by setting TYPE = 222 in the CREEP material entry. The Power creep law is currently supported for the elastic-creep, thermal elastic-creep, plastic-creep and thermal plastic-creep material models. The Exponential creep law is currently supported only for the elastic-creep and plastic-creep material models.

- The effective creep strain is calculated as follows:

Power creep law (creep law 1) :

$${}^{t}\overline{e}^{C} = a \cdot {}^{t}\sigma^{b} \cdot t^{d}$$

in which σ is the effective stress, t is the time, and a, b, d are material constants from the CREEP or MATCRP material entries. These three constants can be set to be temperature dependent via the MATTC entry.

Exponential creep law (creep law 2) :

$$ {}^{t}\overline{e}^{C} = F(\sigma)\cdot\left(1-e^{-R(\sigma)\cdot t}\right)+G(\sigma)\cdot t $$

with

$$ F(\sigma)=a\cdot\left(e^{b\cdot{}^{t}\sigma}\right);\quad R(\sigma)=c\cdot\left({}^{t}\sigma\right)^{d};\quad G(\sigma)=e\cdot\left(e^{f\cdot{}^{t}\sigma}\right) $$

in which *a* through *f* are material constants from the CREEP material entry.

- The creep strains are evaluated using the strain hardening procedure for load and temperature variations, and the O.R.N.L. rules for cyclic loading conditions.

ref. C.E. Pugh, J.M. Corum, K.C. Liu and W.L. Greenstreet, "Currently Recommended Constitutive Equations for Inelastic Design of FFTF Components," *Report No. TM-3602, Oak Ridge National Laboratory*, Oak Ridge, Tennessee, 1972.

The procedure used to evaluate the incremental creep strains is summarized in the following: Given the total creep strains ${}^{t}e_{ij}^{C}$ and the deviatoric stresses ${}^{t+\Delta t}s_{ij}$,

1) Calculate the effective stress

$$ {}^{t+\Delta t}\overline{\sigma} = \left[\frac{3}{2}\,{}^{t+\Delta t}s_{ij}\;{}^{t+\Delta t}s_{ij}\right]^{\frac{1}{2}} $$

2) Calculate the pseudo-effective creep strain

$$ {}^{t}\overline{e}^{C} = \left[\frac{2}{3}\left({}^{t}\overline{e}_{ij}^{C}-e_{ij}^{orig}\right)\left({}^{t}\overline{e}_{ij}^{C}-e_{ij}^{orig}\right)\right]^{\frac{1}{2}} $$

3a) For power creep with temperature-independent material constants, calculate the effective creep strain and effective creep strain rate at time $t+\Delta t$ using

$$\left({}^{t+\Delta t}\overline{e}^{C}\right)^{1/a_2} = \left({}^{t}\overline{e}^{C}\right)^{1/a_2} + \left(a_0\,{}^{t+\Delta t}\overline{\sigma}^{a_1}\right)^{1/a_2}\Delta t$$

$${}^{t+\Delta t}\dot{\overline{e}}^{C} = \frac{{}^{t+\Delta t}\overline{e}^{C} - {}^{t}\,\overline{e}^{C}}{\Delta t}$$

3b) For other creep laws (including power creep with temperature dependent constants), calculate the pseudo-time $\overline{t}$ satisfying

$$\overline{e}^{C}\left({}^{t+\Delta t}\overline{\sigma}, {}^{t+\Delta t}\theta, \overline{t}\right) = {}^{t}\,\overline{e}^{C} + \dot{\overline{e}}^{C}\left({}^{t+\Delta t}\overline{\sigma}, {}^{t+\Delta t}\theta, \overline{t}\right)\Delta t$$

where $\overline{e}^{C}\left({}^{t+\Delta t}\overline{\sigma}, {}^{t+\Delta t}\theta, \overline{t}\right)$ is the generalized uniaxial creep law and $\dot{\overline{e}}^{C}\left({}^{t+\Delta t}\overline{\sigma}, {}^{t+\Delta t}\theta, \overline{t}\right) = \dfrac{d\overline{e}^{C}\left({}^{t+\Delta t}\overline{\sigma}, {}^{t+\Delta t}\theta, \overline{t}\right)}{d\,\overline{t}}$. Then calculate the effective creep strain and effective creep strain rate at time $t+\Delta t$ using

$${}^{t+\Delta t}\overline{e}^{C} = \overline{e}^{C}\left({}^{t+\Delta t}\overline{\sigma}, {}^{t+\Delta t}\theta, \overline{t}\right), \quad {}^{t+\Delta t}\dot{\overline{e}}^{C} = \dot{\overline{e}}^{C}\left({}^{t+\Delta t}\overline{\sigma}, {}^{t+\Delta t}\theta, \overline{t}\right).$$

4) Calculate ${}^{t+\Delta t}\gamma$ using

$${}^{t+\Delta t}\gamma = \frac{3}{2}\frac{{}^{t+\Delta t}\dot{\overline{e}}^{C}}{{}^{t+\Delta t}\overline{\sigma}}$$

5) Calculate the incremental creep strains using

$$\Delta\overline{e}_{ij}^{C} = \Delta t\;{}^{t+\Delta t}\gamma\;{}^{t+\Delta t}s_{ij}$$

The use of the pseudo-time in step 3b corresponds to a strain hardening procedure. See ref. KJB, pp 607-608 for a discussion of strain hardening for calculation of creep strains.

3.6.4 Computational procedures

ref. KJB Section 6.6.3

- The stresses and strains at the integration points are evaluated using the effective-stress-function algorithm.

ref. M. Kojić and K.J. Bathe, "The Effective-Stress-Function Algorithm for Thermo-Elasto-Plasticity and Creep," *Int. J. Numer. Meth. Engng.*, Vol. 24, No. 8, pp. 1509-1532, 1987.

Briefly, the procedure used consists of the following calculations. The general constitutive equation

$$ {}^{t+\Delta t}\sigma^{(i)} = {}^{t+\Delta t}\mathbf{C}^{E}\left({}^{t+\Delta t}\mathbf{e}^{(i)} - {}^{t+\Delta t}\mathbf{e}^{P(i)} - {}^{t+\Delta t}\mathbf{e}^{C(i)} - {}^{t+\Delta t}\mathbf{e}^{TH} \right) \quad (3.6\text{-}1)$$

is solved separately for the mean stress and for the deviatoric stresses. In this equation the index (i) denotes the iteration counter in the iteration for nodal point equilibrium. For easier writing this index will be dropped in the discussion to follow. The mean stress is calculated as

$$ {}^{t+\Delta t}\sigma_m = \frac{{}^{t+\Delta t}E}{1-2\,{}^{t+\Delta t}\nu}\left({}^{t+\Delta t}e_m - {}^{t+\Delta t}e^{TH} \right) \quad (3.6\text{-}2)$$

The deviatoric stresses ${}^{t+\Delta t}\mathbf{s}$ depend on the inelastic strains and they can be expressed as

$$ {}^{t+\Delta t}\mathbf{s} = \frac{1}{{}^{t+\Delta t}a_E + \alpha\,\Delta t\,{}^{\tau}\gamma + \Delta\lambda}\left[{}^{t+\Delta t}\mathbf{e}'' - \left(1-\alpha\right)\Delta t\,{}^{\tau}\gamma\,{}^{t}\mathbf{s} \right] \quad (3.6\text{-}3)$$

where ${}^{t+\Delta t}a_E = \dfrac{{}^{t+\Delta t}E}{1+{}^{t+\Delta t}\nu}$, ${}^{t}\mathbf{s}$ = deviatoric stress at the start of the time step and α is the integration parameter used for stress evaluation $\left(0 \le \alpha < 1\right)$. The creep and plastic multipliers ${}^{\tau}\gamma$ and $\Delta\lambda$ are functions of the effective stress ${}^{t+\Delta t}\bar{\sigma}$ only, and they account for creep and plasticity; also

$$^{t+\Delta t}\mathbf{e}'' = {}^{t+\Delta t}\mathbf{e}' - {}^{t}\mathbf{e}^{P} - {}^{t}\mathbf{e}^{C}$$

is known since the deviatoric strains ${}^{t+\Delta t}\mathbf{e}'$, plastic strains ${}^{t}\mathbf{e}^{P}$ and creep strains ${}^{t}\mathbf{e}^{C}$ are known from the current displacements and the stress/strain state at the start of the current time step.

The following scalar function $f\left({}^{t+\Delta t}\bar{\sigma}\right)$ is obtained from Eq. (3.6-3)

$$f\left({}^{t+\Delta t}\bar{\sigma}\right) = a^{2}\,{}^{t+\Delta t}\bar{\sigma}^{2} + b\,{}^{\tau}\gamma - c^{2}\,{}^{\tau}\gamma^{2} - d^{2} = 0 \qquad (3.6\text{-}4)$$

The zero of Eq. (3.6-4) provides the solution for the effective stress ${}^{t+\Delta t}\bar{\sigma}$, where

$$\begin{aligned}
a &= {}^{t+\Delta t}a_{E} + \alpha\,\Delta t\,{}^{\tau}\gamma + \Delta\lambda \\
b &= 3\left(1-\alpha\right)\Delta t\,{}^{t+\Delta t}e'_{ij}\,{}^{t}s_{ij} \\
c &= \left(1-\alpha\right)\Delta t\,{}^{t}\bar{\sigma} \\
d^{2} &= \frac{3}{2}\,{}^{t+\Delta t}e''_{ij}\,{}^{t+\Delta t}e''_{ij}
\end{aligned}$$

with summation on the indices *i, j.*

Once the solution for ${}^{t+\Delta t}\bar{\sigma}$ has been determined from Eq. (3.6-4), simultaneously with the scalars ${}^{\tau}\gamma$ and $\Delta\lambda$ from the creep and plasticity conditions, the deviatoric stress ${}^{t+\Delta t}\mathbf{s}$ is calculated from Eq. (3.6-3), and the plastic and creep strains at the end of the time step are obtained as

$$\begin{aligned}
{}^{t+\Delta t}\mathbf{e}^{P} &= {}^{t}\mathbf{e}^{P} + \Delta\lambda\,{}^{t+\Delta t}\mathbf{s} \\
{}^{t+\Delta t}\mathbf{e}^{C} &= {}^{t}\mathbf{e}^{C} + \left[(1-\alpha)\,{}^{t}\mathbf{s} + \alpha\,{}^{t+\Delta t}\mathbf{s}\right]\Delta t\,{}^{\tau}\gamma
\end{aligned}$$

The above equations correspond to isotropic hardening conditions and a general 3-D analysis. The solution details for kinematic hardening conditions and for special problems (for the plane stress and shell elements) are given in the above cited references, and also in the following reference:

ref. M. Kojić and K.J. Bathe, "Thermo-Elastic-Plastic and Creep Analysis of Shell Structures", *Computers & Structures,* Vol. 26, No 1/2, pp. 135-143, 1987.

3.7 Hyperelastic material models

- The hyperelastic material models available in Advanced Nonlinear Solution are the Mooney-Rivlin, Ogden, Arruda-Boyce, Hyperfoam, and Sussman-Bathe material models. They are all defined using the MATHE command. In addition MATHP can be used to define a hyperelastic Mooney-Rivlin material.

- This material model can be employed with the **2-D solid, 3-D solid** and **3D-shell** elements.

- This material model uses **large displacement/large strain** kinematics. A total Lagrangian (TL) formulation is employed. The same formulation is used if a **large displacement/small strain** kinematics is selected.

- Viscoelastic effects and Mullins effects can be included using the MATHEV and MATHEM entries.

- Thermal strains can be included via a constant thermal expansion coefficient. Section 3.7.6 shows how thermal strains are computed for hyperelastic materials.

- In Solution 701 only the Mooney-Rivlin, Ogden and Sussman-Bathe material models can be used, and only for 3-D solid elements.

- The isotropic hyperelastic effects are mathematically described by specifying the dependence of the strain energy density (per unit original volume) W on the Green-Lagrange strain tensor ε_{ij} .

- We now give a brief summary of the quantities and concepts used. For more information, refer to ref KJB, section 6.6.2. Here and below, we omit the usual left superscripts and subscripts for ease of writing. Unless otherwise stated, all quantities are evaluated at time t and referred to reference time 0.

- Useful quantities are the Cauchy-Green deformation tensor C_{ij}, given by

$$C_{ij} = 2\varepsilon_{ij} + \delta_{ij} \tag{3.7-1}$$

where δ_{ij} is the Kronecker delta; the principal invariants of the Cauchy-Green deformation tensor,

$$I_1 = C_{kk}\,, \quad I_2 = \frac{1}{2}\left(I_1^2 - C_{ij}C_{ij}\right), \quad I_3 = \det \mathbf{C} \tag{3.7-2a,b,c}$$

the reduced invariants:

$$\overline{I}_1 = I_1 I_3^{-\frac{1}{3}},\ \overline{I}_2 = I_2 I_3^{-\frac{2}{3}},\ J = I_3^{\frac{1}{2}}, \tag{3.7-3a,b,c}$$

the stretches λ_i where the λ_i's are the square roots of the principal stretches of the Cauchy-Green deformation tensor; and the reduced stretches:

$$\overline{\lambda_i} = \lambda_i \left(\lambda_1 \lambda_2 \lambda_3\right)^{-\frac{1}{3}} \tag{3.7-4}$$

Note that

$$J = \lambda_1 \lambda_2 \lambda_3 \tag{3.7-5}$$

is the volume ratio (ratio of the deformed volume to the undeformed volume).

- The strain energy density W is written either in terms of the invariants or in terms of the stretches. In many cases, the strain energy density is conveniently written as the sum of the deviatoric strain energy density W_D and the volumetric strain energy density W_V.

- With knowledge of how the strain energy density W depends on the Green-Lagrange strain tensor (through the invariants or stretches), the 2nd Piola-Kirchhoff stress tensor is evaluated using

$$S_{ij} = \frac{1}{2}\left(\frac{\partial W}{\partial \varepsilon_{ij}} + \frac{\partial W}{\partial \varepsilon_{ji}} \right) \tag{3.7-6}$$

and the incremental material tensor is evaluated using

$$C_{ijrs} = \frac{1}{2}\left(\frac{\partial S_{ij}}{\partial \varepsilon_{rs}} + \frac{\partial S_{ij}}{\partial \varepsilon_{sr}} \right) \tag{3.7-7}$$

3.7.1 Mooney-Rivlin material model

- The Mooney-Rivlin material model is obtained by setting Model=Mooney in the MATHE material entry. It can also be obtained using the MATHP material entry. It is based on the following expression of the strain energy density:

$$\begin{aligned} W_D = {} & C_{10}\left(I_1-3\right) + C_{01}\left(I_2-3\right) + C_{20}\left(I_1-3\right)^2 + C_{11}\left(I_1-3\right)\left(I_2-3\right) + \\ & C_{02}\left(I_2-3\right)^2 + C_{30}\left(I_1-3\right)^3 + C_{21}\left(I_1-3\right)^2\left(I_2-3\right) + \\ & C_{12}\left(I_1-3\right)\left(I_2-3\right)^2 + C_{03}\left(I_2-3\right)^3 \end{aligned} \tag{3.7-8}$$

where C_{ij} are material constants , and I_1 and I_2 are the first and second strain invariants at time t, referring to the original configuration (see ref. KJB, Section 6.6.2 for the definitions of the strain invariants).

Note that constants A_{ij} used in the MATHP material entry are identical to C_{ij} constants used in MATHE and in the equation above.

ref. KJB Section 6.6.2

- This strain energy density expression assumes a totally incompressible material $\left(I_3 = 1\right)$. It is modified as explained below for plane strain, axisymmetric or 3-D analysis.

Plane stress analysis: In plane stress analysis, the material is assumed to be totally incompressible. Therefore W_V is zero and $W = W_D$. A displacement-based finite element formulation is used, in which the incompressibility condition of the material is imposed by calculating the appropriate thickness of the material.

ref. KJB Section 6.6.2

Plane strain, axisymmetric and 3-D analysis: In plane strain, axisymmetric and 3-D analysis, the material is modeled as compressible (that is, the bulk modulus is not infinite), but the bulk modulus can be set high so that the material is "almost incompressible".

The Mooney-Rivlin strain energy density equation is modified by:

1) substituting for the invariants I_1, I_2 the reduced invariants $\overline{I}_1, \overline{I}_2$,
2) removing the condition $I_3 = 1$, and
3) adding the volumetric strain energy density

$$W_V = \frac{1}{2}\kappa\left(J-1\right)^2 \tag{3.7-9}$$

where κ is the bulk modulus given by K in the MATHE material entry (or two times D_1 in the MATHP material entry). This expression for the volumetric strain energy density yields the following relationship between the pressure and the volume ratio:

$$p = -\kappa\left(J-1\right) \tag{3.7-10}$$

The mixed u/p formulation (u/p formulation) is always used for these elements, to avoid volumetric locking. The material stress-strain descriptions are obtained by differentiation of W to obtain stresses due to the element displacements and then taking into account the effect of the separately interpolated pressure.

ref. T. Sussman and K.J. Bathe, "A Finite Element Formulation for Nonlinear Incompressible Elastic and Inelastic Analysis," *J. Computers and Structures,* Vol.

26, No. 1/2, pp. 357-409, 1987.

Selection of material constants: The Mooney-Rivlin material description used here has 9 C_{ij} constants and the bulk modulus κ. Strictly speaking, this material law is termed a higher-order or generalized Mooney-Rivlin material law. Choosing only $C_{10} \neq 0$ yields the neo-Hookean material law, and choosing only $C_{10} \neq 0$, $C_{01} \neq 0$ yields the standard two-term Mooney-Rivlin material law.

The small strain shear modulus and small strain Young's modulus can be written in terms of these constants as (assuming $\kappa = \infty$)

$$G = 2\left(C_{10} + C_{01}\right) \tag{3.7-11}$$

$$E = 6\left(C_{10} + C_{01}\right) \tag{3.7-12}$$

These moduli must be greater than zero.

- The bulk modulus κ is used to model the compressibility of the material for plane strain, axisymmetric and 3-D analysis.

- Solution 601 assumes a default for the bulk modulus based on small strain near-incompressibility, i.e.,

$$\kappa = \frac{E}{3\left(1-2\nu\right)} \quad \text{with } \nu = 0.499 \tag{3.7-13}$$

where E is the small strain Young's modulus or, in terms of the small strain shear modulus G,

$$\kappa = \frac{2G\left(1+\nu\right)}{3\left(1-2\nu\right)} = 500G \quad \text{for } \nu = 0.499 \tag{3.7-14}$$

This rule of thumb can be used to estimate the bulk modulus in the absence of experimental data. However, lower values of the bulk modulus can be used to model compressible materials.

- Solution 701 assumes the same bulk modulus based on small strain near-incompressibility. However, this can significantly reduce the stable time step. In such cases, is better to use a bulk modulus that results in ν=0.49.

- When automatic time step calculation is used for a Mooney-Rivlin material in Solution 701, the critical time step is governed by the dilatational wave speed. This is most frequently an acceptable assumption since the material is almost incompressible.

- As the material deforms, the bulk to shear modulus ratio may change, because the instantaneous shear modulus is dependent on the amount of deformation. A value of the bulk modulus that corresponds to near incompressibility for small strains may not be large enough to correspond to near incompressibility for large strains.

3.7.2 Ogden material model

- The Ogden material model is obtained by setting Model=Ogden in the MATHE material entry. It is based on the following expression:

$$W_D = \sum_{n=1}^{9}\left(\frac{\mu_n}{\alpha_n}\left[\lambda_1^{\alpha_n} + \lambda_2^{\alpha_n} + \lambda_3^{\alpha_n} - 3\right]\right) \qquad (3.7\text{-}15)$$

where μ_n and α_n are Ogden material constants.

ref. KJB Section 6.6.2

- This strain energy density expression assumes a totally incompressible material $\left(I_3 = 1\right)$. As in the Mooney-Rivlin material, the strain energy density expression of the Ogden material is used unmodified for plane stress analysis, and is modified for plane strain, axisymmetric and 3-D analysis. The modification is made by:

1) substituting for the stretches $\lambda_1, \lambda_2, \lambda_3$ the reduced stretches $\bar{\lambda}_1, \bar{\lambda}_2, \bar{\lambda}_3$,
2) removing the condition $\lambda_1\lambda_2\lambda_3 = 1$, and
3) adding the volumetric strain energy density

$$W_V = \frac{1}{2}\kappa\left(\lambda_1\lambda_2\lambda_3 - 1\right)^2 = \frac{1}{2}\kappa\left(J-1\right)^2 \qquad (3.7\text{-}16)$$

where κ is the bulk modulus. The relationship between the pressure and the volumetric ratio is the same as for the Mooney-Rivlin material.

The u/p formulation is always used for plane strain, axisymmetric and 3-D elements. For comments about the u/p formulation, see the corresponding comments in the Mooney-Rivlin material description.

Selection of material constants: The Ogden material description used here has 19 constants: $\mu_n, \alpha_n,\ n = 1,...,9$ and the bulk modulus. Choosing only $\mu_n, \alpha_n \neq 0,\ n = 1,2,3$ the standard three-term Ogden material description is recovered.

The small strain shear modulus and small strain Young's modulus can be written as (assuming $\kappa = \infty$)

$$G = \frac{1}{2}\sum_{n=1}^{9}\mu_n\alpha_n \qquad (3.7\text{-}17)$$

$$E = \frac{3}{2}\sum_{n=1}^{9}\mu_n\alpha_n \qquad (3.7\text{-}18)$$

These moduli must be greater than zero.

- When automatic time step calculation is used for an Ogden material in Solution 701, the critical time step is governed by the dilatational wave speed. This is most frequently an acceptable assumption since the material is almost incompressible.

- For comments about the bulk modulus, see the corresponding comments about the bulk modulus in the Mooney-Rivlin material description.

3.7.3 Arruda-Boyce material model

- The Arruda-Boyce model is obtained by setting MODEL = ABOYCE in the MATHE material entry. It is based on the following expression:

$$
\begin{aligned}
W_D = N_{KT}[\frac{1}{2}\left(I_1-3\right)+\frac{1}{20N}\left(I_1^2-9\right)+\frac{11}{1050N^2}\left(I_1^3-27\right) \\
+\frac{19}{7000N^3}\left(I_1^4-81\right)+\frac{519}{673750N^4}\left(I_1^5-243\right)]
\end{aligned}
\qquad (3.7\text{-}19)
$$

where N_{KT} is a material constant and N is a material parameter representing the number of statistical links of the material chain.

- The Arruda-Boyce material model is described in the following reference:

ref. E. M. Arruda and M. C. Boyce, “A three-dimensional constitutive model for the large stretch behavior of rubber elastic materials”, *J. Mech. Phys. Solids*, Vol,. 41 (2), pp 389-412 (1993).

ref. KJB
Section 6.6.2

- This strain energy density expression assumes a totally incompressible material $\left(I_3=1\right)$. As in the Mooney-Rivlin material, the strain energy density expression of the Arruda-Boyce material is used unmodified for plane stress analysis, and is modified for plane strain, axisymmetric and 3-D analysis. The modification is made by:

1) substituting for the strain invariant I_1 the reduced strain invariant $\overline{I}_1$,
2) removing the condition $I_3=1$, and
3) adding the volumetric energy term

$$W_V=\kappa\left(J\ln J-(J-1)\right) \qquad (3.7\text{-}20)$$

where κ is the small-strain bulk modulus. The relationship between the pressure and the volume ratio is

$$p=-\kappa\ln J \qquad (3.7\text{-}21)$$

The u/p formulation is always used for plane strain, axisymmetric and 3-D elements. For comments about the u/p formulation, see the corresponding comments in the Mooney-Rivlin material description.

- When plane strain, axisymmetric or 3-D elements are used, there should be at least one solution unknown. This is because the constraint equation used in the u/p formulation is nonlinear in the unknown pressures. Therefore equilibrium iterations are required for convergence, even when all of the displacements in the model are prescribed.

- Material constant N_{KT} is only an approximation to the initial shear modulus G. The relationship between N_{KT} and G is

$$G = N_{KT}\left(1+\frac{3}{5N}+\frac{99}{175N^2}+\frac{513}{875N^3}+\frac{42039}{67375N^4}\right)$$

3.7.4 Hyperfoam material model

- The hyperfoam material model is obtained by setting Model=Foam in the MATHE material entry. It is based on the following expression:

$$W=\sum_{n=1}^{N}\frac{\mu_n}{\alpha_n}\left[\lambda_1^{\alpha_n}+\lambda_2^{\alpha_n}+\lambda_3^{\alpha_n}-3+\frac{1}{\beta_n}\left(J^{-\alpha_n\beta_n}-1\right)\right] \qquad (3.7\text{-}22)$$

in which there are the material constants $\mu_n, \alpha_n, \beta_{n,}\ n=1,...,N$. The maximum value of *N* is 9.

- A material model similar to the hyperfoam material model is described in the following reference:

ref. B. Storåkers, “On material representation and constitutive branching in finite compressible elasticity”, *J. Mech. Phys. Solids*, Vol,. 34(2), pp 125-145 (1986).

In this reference, β_n is the same for all values of *n*.

- The strain energy density can be split into deviatoric and volumetric parts

$$W_D = \sum_{n=1}^{N} \frac{\mu_n}{\alpha_n} \left[\lambda_1^{\alpha_n} + \lambda_2^{\alpha_n} + \lambda_3^{\alpha_n} - 3J_3^{\alpha_n/3} \right] \tag{3.7-23}$$

$$W_V = \sum_{n=1}^{N} \frac{\mu_n}{\alpha_n} \left[3\left(J_3^{\alpha_n/3} - 1 \right) + \frac{1}{\beta_n}\left(J_3^{-\alpha_n\beta_n} - 1 \right) \right] \tag{3.7-24}$$

Notice that $W = W_D + W_V$. This decomposition of the strain energy density has the advantage that the stresses obtained from the deviatoric and volumetric parts separately are zero when there are no deformations:

$$S_{ij}^D \Big|_{\varepsilon_{ij}=0} = \frac{1}{2}\left(\frac{\partial W_D}{\partial \varepsilon_{ij}} + \frac{\partial W_D}{\partial \varepsilon_{ji}} \right)\Bigg|_{\varepsilon_{ij}=0} = 0\,,$$

$$S_{ij}^V \Big|_{\varepsilon_{ij}=0} = \frac{1}{2}\left(\frac{\partial W_V}{\partial \varepsilon_{ij}} + \frac{\partial W_V}{\partial \varepsilon_{ji}} \right)\Bigg|_{\varepsilon_{ij}=0} = 0$$

Notice that W_D contains the volumetric part of the motion through the term $3J_3^{\alpha_n/3}$. Therefore W_D is not entirely deviatoric.

- The material is *not* assumed to be totally incompressible. Because both W_D and W_V contain the volumetric part of the motion, the mixed u/p formulation cannot be used with the hyperfoam material. A displacement-based formulation is used.

Selection of material constants: The hyperfoam material description used here has 27 constants: $\mu_n, \alpha_n, \beta_n,\ n = 1,...,9$.

The small strain shear modulus and small strain bulk modulus can be written as

$$G = \frac{1}{2}\sum_{n=1}^{9} \mu_n \alpha_n \tag{3.7-25}$$

$$\kappa = \sum_{n=1}^{9} \left(\beta_n + \frac{1}{3} \right) \mu_n \alpha_n \qquad (3.7\text{-}26)$$

These moduli must be greater than zero, hence we note that β_n should be greater than –1/3.

When all of the β_n are equal to each other $= \beta$, then the Poisson's ratio is related to β using

$$\beta = \frac{\nu}{1-2\nu} \qquad (3.7\text{-}27)$$

- The hyperfoam material model is generally used for highly compressible elastomers.

If the ratio of the bulk modulus to shear modulus is high (greater than about 10), the material is almost incompressible and we recommend that one of the other hyperelastic materials be used.

3.7.5 Sussman-Bathe material model

- The Sussman-Bathe model is obtained by setting MODEL = SUSSBAT in the MATHE material entry. It is based on the following equation:

$$W_D = w(e_1) + w(e_2) + w(e_3) \qquad (3.7\text{-}28)$$

where $w(e)$ is a function of the principal logarithmic strain (Hencky strain) and e_1, e_2, and e_3 are the principal logarithmic strains.

- The primary goal of the model is to fit given uniaxial tension/compression data very well. This goal is accomplished by using a spline to fit the derivative of $w(e)$, as described in detail below.

Of course, when uniaxial tension/compression data is known, a curve fitting approach can, in theory, be used to determine the constants for the other hyperelastic models, e.g. the Ogden material model. But this curve fitting in practice does not provide good fits to the data under many circumstances.

- This strain energy density expression assumes a totally incompressible material ($I_3 = 1$) and is modified as explained below for plane strain, axisymmetric or 3-D analysis.

- The Sussman-Bathe model is given in the following reference

ref. T. Sussman and K.J. Bathe, “A model of incompressible isotropic hyperelastic material behavior using spline interpolations of tension-compression test data”, *Commun. Numer. Meth. Engng,* Vol. 25, Issue 1, pp. 53-63, January 2009.

- The following gives a quick summary of the Sussman-Bathe model. In this summary, we assume that the material is totally incompressible. Differences due to slight compressibility are small.

Theoretical background:

1) The Cauchy stress τ_i corresponding to the principal strain e_i is

$$\tau_i = \frac{\partial W_D}{\partial e_i} + p = w'(e_i) + p \qquad (3.7\text{-}29)$$

where $w'(e_i) \equiv dw/de$.

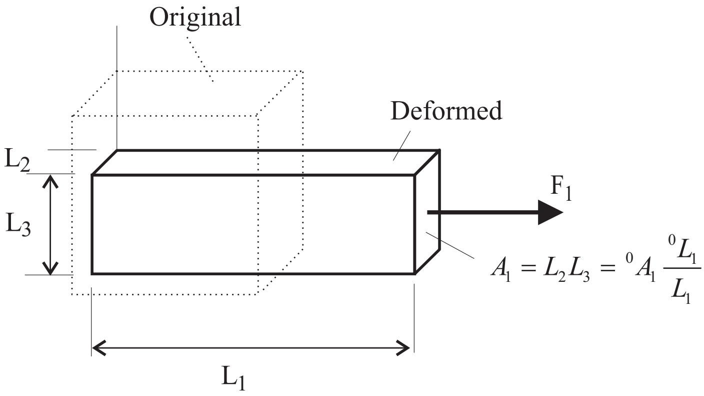
2) In uniaxial tension/compression (Fig. 3.7-1), $e_1 = e$, $e_2 = e_3 = -\frac{1}{2}e$ so

$$\tau = w'(e) - w'(-\tfrac{1}{2}e) \qquad (3.7\text{-}30)$$

(3.7-30) can be inverted to obtain

$$w'(e) = \sum_{k=0}^{\infty} \tau\left(\left(-\tfrac{1}{2}\right)^k e\right) \qquad (3.7\text{-}31)$$

The series converges when $\tau(e) \to 0$ as $e \to 0$.



$$e = e_1 = \ln \frac{L_1}{{}^0L_1} \qquad e_2 = e_3 = -\frac{1}{2} e_1 \qquad \tau_1 = \frac{F_1}{A_1} \qquad \tau_2 = \tau_3 = 0$$

Fig. 3.7-1: Uniaxial tension/compression test

3) The asymptotic conditions for *w* are $w'(e) \to -\infty$ as $e \to -\infty$; $w'(e) \to \infty$ as $e \to \infty$. These asymptotic conditions correspond to the asymptotic conditions of infinite stresses for infinite strains.

4) For a stable material, it is necessary (but not sufficient) that $w''(e) > 0$ for all *e*. Not all materials for which $\tau'(e) > 0$ have $w''(e) > 0$. For example, the material with

$$\begin{aligned} \tau(e) &= E_T e, \ \ e > 0 \\ &= E_C e, \ \ e < 0 \end{aligned}$$

where E_T and E_C are constants greater than zero, has $w''(e) > 0$ only if

$$\tfrac{1}{2} E_T < E_C < 2E_T$$

5) Given only simple tension data for $\tau(e)$, there are multiple $w'(e)$ that exactly correspond to $\tau(e)$, for positive *e* only. Two such $w'(e)$ are

$$\begin{aligned} w'(e) &= 0, e < 0 \\ &= \tau(e), e > 0 \end{aligned}$$

and

$$\begin{aligned} w'(e) &= -\tau(-2e), e < 0 \\ &= 0, e > 0 \end{aligned}$$

Hence the material is not uniquely described given only uniaxial tension (or uniaxial compression) data. Both uniaxial tension and uniaxial compression data must be provided to uniquely describe the material.

6) The small-strain Young's modulus *E* is found by differentiating the uniaxial stress-strain curve, and evaluating at *e* = 0, and, since the material is almost incompressible, the small-strain shear modulus *G* is $G = \frac{1}{3}E$. The results are

$$E = \frac{3}{2}w''(0), \; G = \frac{1}{2}w''(0) \qquad (3.7\text{-}32a, b)$$

8) The Ogden material model can be considered a special case of (3.7-28), since the Ogden material model can be written in terms of $w'(e)$:

$$w'(e) = \sum_n \mu_n \left(\exp(\alpha_n e) - 1 \right) \qquad (3.7\text{-}33)$$

Spline representation of $w'(e)$:

In the Sussman-Bathe model, we choose $w'(e)$ to fit given uniaxial tension/compression data very well, as follows.

The uniaxial tension-compression data is in the form of user-specified data points $\left(e_i, \tau_i\right)$. From these data points, we build a non-uniform cubic spline representation of the uniaxial tension/compression stress-strain data $\tau = \tau(e)$, as shown in Fig. 3.7-2. For the non-uniform cubic spline representation of $\tau(e)$,

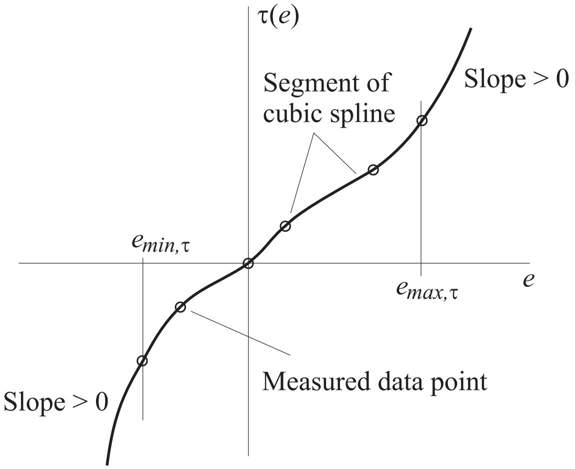


Fig. 3.7-2: Uniaxial tension/compression stress-strain spline

1) A spline segment is placed between two successive user-input data points. The user-input data points need not be equally spaced.

2) The range of the cubic spline is between the first and last user-input data points.

3) Outside the range of the cubic spline, the slope of $\tau(e)$ is greater than zero. This ensures that the asymptotic conditions of $\tau(-\infty) = -\infty$, $\tau(\infty) = \infty$ are met.

Using the non-uniform cubic spline representation of $\tau(e)$ and (3.7-31), we build a uniform cubic spline for $w'(e)$ as shown in Fig. 3.7-3. For the uniform cubic spline representation of $w'(e)$,

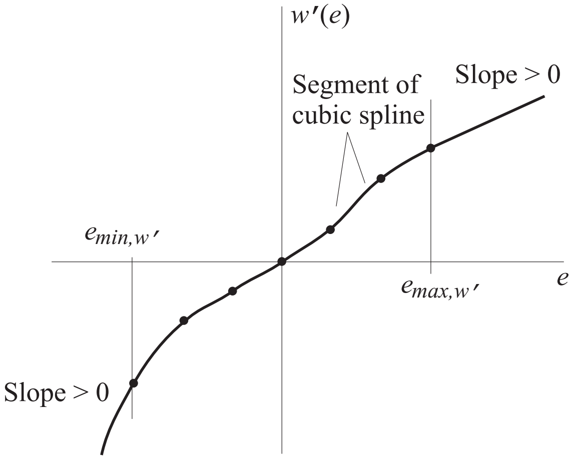


Fig. 3.7-3: $w'(e)$ spline

1) The same number of spline segments is used in tension and in compression.

2) The range of the cubic spline is the same in tension and in compression. This range includes the range of the user-input data points.

3) Outside the range of the cubic spline, the slope of $w'(e)$ is greater than zero, whenever possible.

In order to measure the accuracy of the spline representation of $w'(e)$, we define the relative interpolation error

$$r = \max_e \left| \frac{\overline{\tau}(e) - \tilde{\tau}(e)}{\tilde{\tau}(e)} \right| \qquad (3.7\text{-}34)$$

in which $\overline{\tau}(e)$ is the stress evaluated from the spline representation of $w'(e)$ (using 3.7-30), and $\tilde{\tau}(e)$ is the stress evaluated from the spline representation of $\tau(e)$.

The number of spline segments is automatically chosen to make the interpolation error *r* smaller than a user-specified value. Typically only a few spline segments need be used for $w'(e)$ in order to reduce the interpolation error to a value smaller than experimental error.

Each cubic spline segment in $w'(e)$ can be written

$$w'(e) = A_{i+1}z + B_{i+1}z^3 + A_i(1-z) + B_i(1-z)^3 \tag{3.7-35}$$

for the segment $e_i \le e \le e_{i+1}$ where $z = (e-e_i)/(e_{i+1}-e_i)$. With this definition, $w(e)$ can be written

$$w(e) = C_i + (e_{i+1}-e_i)\left(A_{i+1}\frac{z^2}{2} + B_{i+1}\frac{z^4}{4} + A_i\frac{1-(1-z)^2}{2} + B_i\frac{1-(1-z)^4}{4} \right) \tag{3.7-36}$$

The program determines the constants A_i, B_i, C_i from uniaxial stress-strain data, as described above.

Plane stress analysis: The material is assumed to be totally incompressible. Therefore W_V is zero and $W = W_D$. A displacement-based finite element formulation is used, exactly as for the Mooney-Rivlin material model described above.

Plane strain, axisymmetric and 3-D analysis: The material is modeled as compressible (that is, the bulk modulus is not infinite), but the bulk modulus can be set high so that the material is “almost incompressible”.

Equation (3.7-28) is modified by 1) substituting the deviatoric principal strains for the corresponding principal strains, 2) removing the condition $e_1 + e_2 + e_3 = 0$, and 3) adding the volumetric strain energy density

$$W_V = \kappa\left(J \ln J - (J-1)\right) \tag{3.7-37}$$

where κ is the bulk modulus. The relationship between the pressure and the volume ratio is

$$p = -\kappa \ln J = -\kappa\left(e_1 + e_2 + e_3\right) \tag{3.7-38}$$

which is a generalization of the small-strain pressure-strain relationship. The u/p formulation is always used. For comments about the u/p formulation, see the corresponding comments in the Mooney-Rivlin material description.

Data input considerations:

1) Data input is in the form of a set of stress-strain data points, with positive stresses/strains corresponding to uniaxial tension and negative stresses/strains corresponding to uniaxial compression. Compression and tension data are entered together in the same data set.

2) The data set should contain both tension and compression data (compression data is possibly converted from equibiaxial tension data, see below). If the data set contains only tension data, the program will assume that the true stress / true strain curve in compression is a straight line, which is most likely not a good assumption.

3) The stresses and strains in the set of stress-strain data points can be either

a) True stresses and logarithmic strains (SSTYPE=True in MATHE)
b) Engineering stresses and engineering strains (SSTYPE=Eng in MATHE)
c) Engineering stresses and stretches (SSTYPE=Stretch in MATHE)

4) Data points from equibiaxial tension experiments can be converted into equivalent uniaxial compression data. The conversion formulas are:

$$e_u = -2e_b \,,\ \lambda_u = \lambda_b^{-2} \,,\ {}_0e_u = \left(1 + {}_0e_b\right)^{-2} - 1 \qquad (3.7\text{-}39)$$
$$\tau_u = -\tau_b \,,\ {}_0\sigma_u = -{}_0\sigma_b \lambda_b^3$$

where e_u is the is the equivalent uniaxial logarithmic strain (< 0), e_b is the equibiaxial logarithmic strain (> 0), λ_u is the equivalent uniaxial stretch, λ_b is the equibiaxial stretch, ${}_0e_u$ is the equivalent uniaxial engineering strain, ${}_0e_b$ is the equibiaxial engineering strain, τ_u is the equivalent uniaxial true (Cauchy) stress, τ_b is the equibiaxial true (Cauchy) stress, ${}_0\sigma_u$ is the equivalent uniaxial engineering stress, ${}_0\sigma_b$ is the equibiaxial engineering stress. All of these conversion formulas assume that the material is incompressible.

5) The Sussman-Bathe model fits the data so closely that roughness and waviness in the data causes roughness and waviness in the $w'(e)$ splines. The program does not smooth the data in order to eliminate roughness and waviness. If the original data set contains roughness and waviness that should not be present in the analysis, the data set should be smoothed before entering the data into the program.

6) If the data set corresponds to a stable material, then the Sussman-Bathe model is stable, otherwise the Sussman-Bathe model may not be stable.

7) The strain range of the data set should contain the range of strains anticipated during the analysis.

8) Do not confuse uniaxial compression with hydrostatic compression. These two test cases are very different.

3.7.6 Thermal strain effect

- When the material is temperature-dependent, a coefficient of thermal expansion can be included. The coefficient of thermal expansion is constant. The thermal strain is calculated as

$$\varepsilon_{th} = \bar{\alpha}\left(\theta - \theta_0\right) \qquad (3.7\text{-}40)$$

where θ_0 is the initial temperature, and the thermal strain is assumed to be isotropic. This is similar to the formula as is used for the other thermo-elastic materials, see Section 3.1.6 assuming a constant thermal expansion coefficient.

- When the thermal strain is non-zero, the deformation gradient $\mathbf{X}$ is assumed to be decomposed into a thermal deformation gradient $\mathbf{X}_{th}$ and a mechanical deformation gradient $\mathbf{X}_m$, using

$$\mathbf{X} = \mathbf{X}_m \mathbf{X}_{th} \qquad (3.7\text{-}41)$$

The thermal deformation gradient is

$$\mathbf{X}_{th} = (1 + e_{th})\mathbf{I} \qquad (3.7\text{-}42)$$

therefore the mechanical deformation gradient is

$$\mathbf{X}_m = (1 + e_{th})^{-1}\mathbf{X} \qquad (3.7\text{-}43)$$

the mechanical Cauchy-Green deformation tensor is

$$\mathbf{C}_m = (1 + e_{th})^{-2}\mathbf{C} \qquad (3.7\text{-}44)$$

and the mechanical Green-Lagrange strain tensor is

$$\boldsymbol{\varepsilon}_m = (1 + e_{th})^{-2}\boldsymbol{\varepsilon} - \frac{1}{2}\left(1 - (1 + e_{th})^{-2}\right)\mathbf{I} \qquad (3.7\text{-}45)$$

For small thermal strains, the last equation reduces to $\boldsymbol{\varepsilon}_m \approx \boldsymbol{\varepsilon} - e_{th}\mathbf{I}$, so that the strains are nearly the sum of the mechanical and thermal strains, as in small strain analysis. However, we do not assume that the thermal strains are small.

- The strain energy densities are computed using the mechanical deformations. This is done by computing all invariants and stretches using the mechanical deformations, e.g. the mechanical Cauchy-Green deformation tensor.

The 2nd Piola-Kirchhoff stresses are obtained by differentiating the strain energy density with respect to the total strains. Since the strain energy density is a function of the mechanical strains, we obtain

$$\begin{aligned}
S_{ij} &= \frac{1}{2}\left(\frac{\partial W}{\partial(\varepsilon)_{ij}} + \frac{\partial W}{\partial(\epsilon)_{ji}} \right) \\
&= \frac{1}{2}\left(\frac{\partial W}{\partial(\varepsilon_m)_{ab}} \frac{\partial(\varepsilon_m)_{ab}}{\partial(\varepsilon)_{ij}} + \frac{\partial W}{\partial(\epsilon_m)_{ba}} \frac{\partial(\varepsilon_m)_{ba}}{\partial(\varepsilon)_{ji}} \right) \qquad (3.7\text{-}46) \\
&= \left(\left(1+e_{th}\right)^{-2} \right) \frac{1}{2}\left(\frac{\partial W}{\partial(\varepsilon_m)_{ij}} + \frac{\partial W}{\partial(\varepsilon_m)_{ji}} \right)
\end{aligned}$$

With this definition, the 2nd Piola-Kirchhoff stresses are conjugate to the Green-Lagrange strains.

3.7.7 Viscoelastic effects (Solution 601 only)

- Viscoelastic effects can be included in the Mooney-Rivlin, Ogden, Arruda-Boyce, hyperfoam and Sussman-Bathe material models.

The viscoelastic model used is due to Holzapfel, see the following references:

ref. G. A. Holzapfel, “On large strain viscoelasticity: continuum formulation and finite element applications to elastomeric structures”, *Int. J. Num. Meth. Engng.*, Vol. 39, pp 3903-3926, 1996.

ref. G. A. Holzapfel, *Nonlinear solid mechanics. A continuum approach for engineering*. John Wiley & Sons, Chichester, pp 278-295, 2000.

ref. G. A. Holzapfel, “Biomechanics of soft tissue”, in Lemaitre (ed.), *Handbook of Materials Behavior Models: Nonlinear Models and Properties*, Academic Press, 2001, pp 1057-1071.

In the following, we give a brief discussion of the Holzapfel model for finite strain viscoelasticity.

Equivalent 1D model: The equivalent 1D model is shown in Fig 3.7-4. It is the same as a generalized Maxwell model with many chains. A generic chain is denoted with superscript α , as shown in the figure.

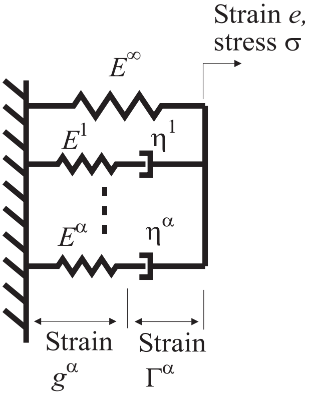


Fig. 3.7-4: Equivalent 1D model for viscoelastic effects

The spring E^∞ is equivalent to the elastic stiffness of the model. Each chain contains a spring with stiffness E^α and dashpot with viscosity η^α . (Note that the superscripts ∞ and α do not denote exponentiation.) The strain in each chain is the sum of the strain in the spring g^α and the strain in the dashpot Γ^α . The observed stress is

$$\sigma = \sigma^\infty + \sum_\alpha q^\alpha \tag{3.7-47}$$

where $\sigma^\infty = E^\infty e$ is the elastic stress and $q^\alpha = E^\alpha g^\alpha = \eta^\alpha \dot{\Gamma}^\alpha$ is the stress in chain α . Using the definition $\tau^\alpha = \dfrac{\eta^\alpha}{E^\alpha}$ and the assumption $E^\alpha = \beta^\alpha E^\infty$, the following expression is obtained:

$$\dot{q}^\alpha + \frac{1}{\tau^\alpha} q^\alpha = \beta^\alpha \dot{\sigma}^\infty \tag{3.7-48}$$

Assuming that ${}^{0}q^{\alpha} = 0$, (3.7-48) can be written in convolution form as

$$q^{\alpha} = \int_{0}^{t} \exp\left(-\frac{t-t'}{\tau^{\alpha}}\right) \beta^{\alpha}\, {}^{t'}\dot{\sigma}^{\infty}\, dt' \qquad (3.7\text{-}49)$$

from which the total stress is

$$\sigma = \int_{0}^{t} E^{\infty} \left[1 + \sum_{\alpha} \beta^{\alpha} \exp\left(-\frac{t-t'}{\tau^{\alpha}}\right)\right] {}^{t'}\dot{e}\, dt' \qquad (3.7\text{-}50)$$

Evidently the relaxation modulus is $E(t) = E^{\infty}\left[1 + \sum_{\alpha} \beta^{\alpha} \exp\left(-\frac{t}{\tau^{\alpha}}\right)\right]$ which is a Prony series expression.

The dissipation in dashpot α is

$$D^{\alpha} = q^{\alpha}\dot{\Gamma}^{\alpha} = q^{\alpha}\left(\dot{e} - \dot{g}^{\alpha}\right) = q^{\alpha}\left(\dot{e} - \frac{1}{\beta^{\alpha}}\frac{\dot{q}^{\alpha}}{E^{\infty}}\right) \qquad (3.7\text{-}51)$$

and the total dissipation is $D = \sum_{\alpha} D^{\alpha}$. In the above, the viscoelastic material constants for each chain are τ^{α} and β^{α}.

Potential-based 1D model: The 1D model can be written in terms of a potential as follows:

$$\Psi = \Psi^{\infty}(e) + \sum_{\alpha} \Psi^{\alpha}(g^{\alpha}) \qquad (3.7\text{-}52)$$

where $\Psi^{\infty}(e) = \frac{1}{2}E^{\infty}e^{2}$ is the strain energy of the elastic chain and $\Psi^{\alpha}(g^{\alpha}) = \frac{1}{2}E^{\alpha}\left(g^{\alpha}\right)^{2}$ is the strain energy in the spring of chain α. In terms of β^α, $\Psi^\alpha\left(g^\alpha\right)=\beta^\alpha\Psi^\infty\left(g^\alpha\right)$. The 1D model is recovered by defining

$$\sigma=\left.\frac{\partial\Psi}{\partial e}\right|_{\Gamma^\alpha\text{ fixed}},\ q^\alpha=-\left.\frac{\partial\Psi}{\partial\Gamma^\alpha}\right|_{e\text{ fixed}} \qquad (3.7\text{-}53\text{a,b})$$

Notice that (3.7-47) and (3.7-53a) imply

$$q^\alpha=\left.\frac{\partial\Psi^\alpha}{\partial e}\right|_{\Gamma^\alpha\text{ fixed}}=\frac{\partial\Psi^\alpha}{\partial g^\alpha}.$$

Finite strain model: The finite strain model is derived from the potential-based 1D model as follows. The elastic potential is defined as

$$\Psi^\infty\left(\boldsymbol{\varepsilon}\right)=W\left(\boldsymbol{\varepsilon}\right) \qquad (3.7\text{-}54)$$

where $W\left(\varepsilon_{ij}\right)$ is the strain energy density from the elastic part of the material model. The potential of each chain α is defined as

$$\begin{aligned}\Psi^\alpha\left(\varepsilon_{ij},\Gamma_{ij}^\alpha\right)&=\beta^\alpha W\left(G_{ij}^\alpha\right),\text{ usage=combined}\\&=\beta^\alpha W_D\left(G_{ij}^\alpha\right),\text{ usage=deviatoric}\\&=\beta^\alpha W_V\left(G_{ij}^\alpha\right),\text{ usage=volumetric}\end{aligned} \qquad (3.7\text{-}55\text{a,b,c})$$

in which the usage flag (which is a user-input flag) determines whether the entire elastic strain energy density, deviatoric strain energy density or volumetric strain energy density is taken for chain α. Here G_{ij}^α is analogous to the strain in the 1D spring g^α, and we assume $G_{ij}^\alpha=\varepsilon_{ij}-\Gamma_{ij}^\alpha$. Note that with this definition of G_{ij}^α, we have

$$S_{ij} = \left.\frac{\partial \Psi}{\partial \varepsilon_{ij}}\right|_{\Gamma_{ij}^{\alpha} \text{ fixed}}, \; Q_{ij}^{\alpha} = \left.\frac{\partial \Psi^{\alpha}}{\partial \varepsilon_{ij}}\right|_{\Gamma_{ij}^{\alpha} \text{ fixed}} = -\left.\frac{\partial \Psi}{\partial \Gamma_{ij}^{\alpha}}\right|_{\varepsilon_{ij} \text{ fixed}} = \frac{\partial \Psi}{\partial G_{ij}^{\alpha}}$$

(3.7-56a,b)

where S_{ij} are the 2nd Piola-Kirchhoff stresses and Q_{ij}^{α} is analogous to the stress q^{α}.

Following exactly the same arguments as in the 1D case, we obtain

$$\dot{Q}_{ij}^{\alpha} + \frac{1}{\tau^{\alpha}} Q_{ij}^{\alpha} = \beta^{\alpha} \dot{S}_{ij}^{\infty} \qquad (3.7\text{-}57)$$

Assuming that ${}^{0}Q_{ij}^{\alpha} = 0$, (3.7-57) can be written in convolution form as

$$Q_{ij}^{\alpha} = \int_{0}^{t} \exp\left(-\frac{t - t'}{\tau^{\alpha}} \right) \beta^{\alpha} \, {}^{t'}\dot{S}_{ij}^{\infty} \, dt' \qquad (3.7\text{-}58)$$

and (3.7-58) can be numerically approximated using

$${}^{t+\Delta t}Q_{ij}^{\alpha} = \exp\left(-\frac{\Delta t}{\tau^{\alpha}} \right) {}^{t}Q_{ij}^{\alpha} + \beta^{\alpha} \frac{1 - \exp\left(-\frac{\Delta t}{\tau^{\alpha}} \right)}{\frac{\Delta t}{\tau^{\alpha}}} \left({}^{t+\Delta t}S_{ij}^{\infty} - {}^{t}S_{ij}^{\infty} \right)$$

(3.7-59)

(3.7-59) is exact when S_{ij}^{α} does not change during the time step, and is more accurate than the formula given by Holzapfel:

$${}^{t+\Delta t}Q_{ij}^{\alpha} = \exp\left(-\frac{\Delta t}{\tau^{\alpha}} \right) {}^{t}Q_{ij}^{\alpha} + \beta^{\alpha} \exp\left(-\frac{\Delta t}{2\tau^{\alpha}} \right) \left({}^{t+\Delta t}S_{ij}^{\infty} - {}^{t}S_{ij}^{\infty} \right)$$

(3.7-60)

especially when $\dfrac{\Delta t}{\tau^{\alpha}}$ is large.

Dissipation calculations: If the dissipation is required, it is calculated using

$$D^{\alpha} = Q_{ij}^{\alpha}\,\dot{\Gamma}_{ij}^{\alpha} = Q_{ij}^{\alpha}\left(\dot{\varepsilon}_{ij} - \dot{G}_{ij}^{\alpha}\right) \qquad (3.7\text{-}61)$$

where

$$\frac{\partial^2 \Psi^{\alpha}}{\partial G_{ij}^{\alpha}\partial G_{rs}^{\alpha}}\dot{G}_{rs}^{\alpha} = \dot{Q}_{ij}^{\alpha} \qquad (3.7\text{-}62)$$

is used to compute the unknown $\dot{G}_{rs}^{\alpha}$ from the known $\dot{Q}_{ij}^{\alpha}$.

If usage=combined,

$$\frac{\partial^2 \Psi^{\alpha}}{\partial G_{ij}^{\alpha}\partial G_{rs}^{\alpha}} = \beta^{\alpha}\,\frac{\partial^2 W}{\partial \varepsilon_{ij}\partial \varepsilon_{rs}} = \beta^{\alpha} C_{ijrs}\,. \qquad (3.7\text{-}63)$$

where the tensor C_{ijrs} is evaluated at the strain state G_{ij}^{α} . The dissipation calculation requires the solution of a set of simultaneous linear equations of at most order 6 (in the 3D case) at each integration point.

If usage=deviatoric,

$$\frac{\partial^2 \Psi^{\alpha}}{\partial G_{ij}^{\alpha}\partial G_{rs}^{\alpha}} = \beta^{\alpha}\,\frac{\partial^2 W_D}{\partial \varepsilon_{ij}\partial \varepsilon_{rs}} = \beta^{\alpha}\left(C_D\right)_{ijrs} \qquad (3.7\text{-}64)$$

where the tensor $\left(C_D\right)_{ijrs}$ is evaluated at the strain state G_{ij}^{α} . Here, the dissipation calculation requires a singular value decomposition of $\left(C_D\right)_{ijrs}$, since $\left(C_D\right)_{ijrs}$ has a zero eigenvalue. A similar situation applies when usage=volumetric, except that the corresponding material tensor has only one nonzero eigenvalue.

The procedure given in (3.7-61) to (3.7-64) is only approximate, since the fundamental assumption $G_{ij}^{\alpha} = \varepsilon_{ij} - \Gamma_{ij}^{\alpha}$ strictly speaking only holds for small strain analysis.

Restrictions and recommendations: The allowed values of the usage flag depend upon the material model and finite element type, as shown in Table 3.7-1.

In view of the restrictions, we recommend that usage="combined" be used in conjunction with the hyperfoam material model, and that usage="deviatoric" be used in conjunction with the Mooney-Rivlin, Ogden, Arruda-Boyce and Sussman-Bathe material models.

Table 3.7-1: Allowed values of the usage flag

	Mooney-Rivlin, Ogden, Arruda-Boyce, Sussman-Bathe		hyperfoam	
	Plane stress[1]	Plane strain, axisymmetric, 3D[2]	Plane stress[3]	Plane strain, axisymmetric, 3D
usage=combined (usage(i)=2)	yes	no	yes	yes
usage=deviatoric (usage(i)=0)	yes	yes	no	yes
usage=volumetric (usage(i)=1)	no	no	no	yes

1. Usage cannot be equal to "volumetric". This is because the material is assumed to be fully incompressible, hence the volumetric strain energy density is zero.
2. When the u/p formulation is used, the usage flag cannot be "combined" or "volumetric". This is because the modification to the volumetric stresses caused when the usage flag is "combined" or "volumetric" is not taken into account in the u/p formulation.
3. The only allowable value of the usage flag is "combined". This is because the out-of-plane stress component S_{xx} must be zero, and in the Holzapfel finite strain viscoelastic model, the only way that this can happen is if S_{xx}^{∞} is zero.

Time-temperature superposition: The preceding derivation assumes that the viscoelastic response is not temperature-dependent. One method of including the effects of temperature is the method of time-temperature superposition.

In time-temperature superposition, the actual time t is replaced by the reduced time ζ. The relationship between the actual time and reduced time is given by

$$\frac{d\zeta}{dt} = \frac{1}{a_T({}^t\theta)} \tag{3.7-65}$$

where ${}^t\theta$ is the temperature and $a_T({}^t\theta)$ is the shift function. Evidently

$${}^t\zeta = \int_0^t \frac{1}{a_T({}^{t'}\theta)} dt' \tag{3.7-66}$$

The shift function used here is either the WLF shift function,

$$\log_{10} a_T({}^t\theta) = -\frac{C_1({}^t\theta - \theta_{ref})}{C_2 + {}^t\theta - \theta_{ref}} \tag{3.7-67a}$$

or the Arrhenius shift function

$$\begin{aligned} \log_{10} a_T({}^t\theta) &= C_1\left(\frac{1}{{}^t\theta} - \frac{1}{\theta_{ref}}\right), \ {}^t\theta \ge \theta_{ref} \\ &= C_2\left(\frac{1}{{}^t\theta} - \frac{1}{\theta_{ref}}\right), \ {}^t\theta < \theta_{ref} \end{aligned} \tag{3.7-67b}$$

where θ_{ref} is the reference temperature and C_1, C_2 are material constants. Notice that as ${}^t\theta$ increases, $a_T({}^t\theta)$ decreases and $\frac{d\zeta}{dt}$ increases.

When using the Arrhenius shift function, the temperature unit must be absolute (Kelvin or Rankine).

For the viscoelastic model used here, the differential equation of the 1D model (3.7-48) becomes

$$\frac{dq^\alpha}{d\zeta}+\frac{1}{\tau^\alpha}q^\alpha=\beta^\alpha\frac{d\sigma^\infty}{d\zeta} \tag{3.7-68}$$

and using (3.7-65), (3.7-68) can be written as

$$\dot{q}^\alpha+\frac{1}{a_T(\theta)\tau^\alpha}q^\alpha=\beta^\alpha\dot{\sigma}^\infty \tag{3.7-69}$$

It is seen that the effect of temperature is to modify the time constants. As the temperature increases, the modified time constants become smaller, that is, the material relaxes more quickly.

The convolution equation of the finite strain model becomes

$$Q_{ij}^\alpha=\int_0^{{}^t\zeta}\exp\left(-\frac{\zeta-\zeta'}{\tau^\alpha}\right)\beta^\alpha\frac{d\,{}^{t'}S_{ij}^\infty}{d\zeta}d\zeta' \tag{3.7-70}$$

and (3.7-70) is numerically approximated by

$${}^{t+\Delta t}Q_{ij}^\alpha=\exp\left(-\frac{\Delta\zeta}{\tau^\alpha}\right){}^tQ_{ij}^\alpha+\beta^\alpha\frac{1-\exp\left(-\dfrac{\Delta\zeta}{\tau^\alpha}\right)}{\dfrac{\Delta\zeta}{\tau^\alpha}}\left({}^{t+\Delta t}S_{ij}^\infty-{}^tS_{ij}^\infty\right) \tag{3.7-71}$$

The only additional consideration is to calculate $\Delta\zeta$, and this is done using

$$\Delta\zeta=\int_t^{t+\Delta t}\frac{1}{a_T({}^{t'}\theta)}dt' \tag{3.7-72}$$

This integration is performed numerically assuming that $\ln a_T(\theta)$ varies linearly over the time step.

Heat generation: A user-specified fraction of the energy dissipated by the viscoelastic model can be considered as heat generation. This heat generation can cause heating in a TMC (thermo-mechanical-coupling) analysis.

Specification of input: Viscoelastic effects are added to rubber-like materials using the MATHEV bulk data entry. The MATHEV bulk data entry includes:

- SHIFT: Indicates the shift function (none, WLF or Arrhenius).
- C1, C2: The shift function material constants C_1, C_2
- A table with one row for each chain. Each row in the table contains beta(i)= β^α , tau(i)= τ^α , hgen(i)= the heat generation factor (fraction of dissipation considered as heat generation, default value is 0.0), and usage(i)=usage flag (default value is deviatoric). There is no restriction on the number of chains permitted. The usage flag can be different for each chain.

- It is seen that the dissipation calculation can be quite expensive. Furthermore the dissipation is not required for the stress solution. Therefore it is the default to not perform the dissipation calculation. The dissipation is only calculated for the chain α when the heat generation factor is non-zero.

3.7.8 Mullins effect (Solution 601 only)

When rubber is loaded to a given strain state, unloaded, then reloaded to the same strain state, the stress required for the reloading is less than the stress required for the initial loading. This phenomenon is referred to as the Mullins effect.

The Mullins effect can be included in the rubber-like materials. The material model used is the one described in the following reference:

ref. R.W. Ogden and D. G. Roxburgh, “A pseudo-elastic model for the Mullins effect in filled rubber”, *Proc. R. Soc. Lond. A (1999)* 455, 2861-2877.

We briefly summarize the main concepts below.

Fig 3.7-5 shows the Mullins effect in simple tension. On initial loading to force F_c, the specimen follows the force-deflection curve a-b-c. When the load is removed, the specimen follows the unloading curve c-d-a. On reloading to force F_c, the specimen follows the reloading curve a-d-c, and on further loading to force F_f, the specimen follows the loading curve c-e-f. When the load is removed, the specimen follows the unloading curve f-g-a, and, on reloading to force F_g, the specimen follows the reloading curve a-g-f.

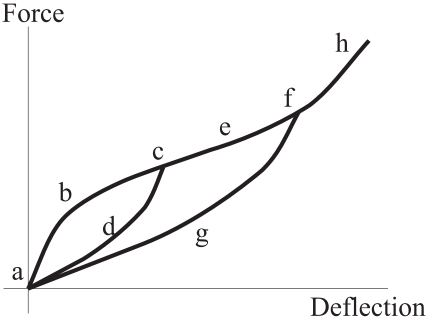


Fig 3.7-5: Mullins effect loading-unloading-reloading curves

Note that any permanent set associated with the Mullins effect is not included in the Ogden-Roxburgh model used here.

The Ogden-Roxburgh model, as implemented in Advanced Nonlinear Solution, uses the following strain energy density expression:

$$\begin{aligned}\tilde{W} &= \eta W_D(\varepsilon_{ij}) + \phi(\eta), \text{ all except hyper-foam} \\ &= \eta W(\varepsilon_{ij}) + \phi(\eta), \text{hyper-foam}\end{aligned} \qquad (3.7\text{-}73a,b)$$

where $W(\varepsilon_{ij})$ is the total elastic strain energy density, $W_D(\varepsilon_{ij})$ is the deviatoric elastic strain energy density, η is an additional solution variable describing the amount of unloading and $\phi(\eta)$ is the damage function. $\tilde{W}$ is referred to as the pseudo-energy function. In our implementation, the deviatoric strain energy density is used for the (almost) incompressible materials and the total strain energy density is used for compressible materials. For ease of writing, we discuss only the case of compressible materials; for incompressible materials, replace W by W_D in the equations below.

η is computed as

$$\eta = 1 - \frac{1}{r}\operatorname{erf}\left[\frac{1}{m}\left(W_m - W\right)\right] \qquad (3.7\text{-}74)$$

where $\operatorname{erf}(x)$ is the error function

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x \exp\left(-u^2\right)du \qquad (3.7\text{-}75)$$

W_m is the maximum value of W encountered during the deformation history and m and r are material constants.

$\phi(\eta)$ is defined by

$$\frac{d\phi(\eta)}{d\eta} = -W \qquad (3.7\text{-}76)$$

and is computed by numerical integration of $\dot{\phi}(\eta) = -W\dot{\eta}$. For a given value of W_m, there is a minimum value of η computed as

$$\eta_m = 1 - \frac{1}{r}\operatorname{erf}\left[\frac{W_m}{m}\right] \qquad (3.7\text{-}77)$$

The value of $\phi(\eta)$ at $\eta = \eta_m$ is written $\phi(\eta_m)$. (Note: the subscript *m* in the term W_m means "maximum", but the subscript *m* in the term η_m means "minimum".) The time rate of change of $\phi(\eta_m)$ can be written as

$$\dot{\phi}(\eta_m) = (1-\eta_m)\dot{W}_m = \frac{1}{r}\operatorname{erf}\left[\frac{W_m}{m}\right]\dot{W}_m \qquad (3.7\text{-}78)$$

Physically, $\phi(\eta_m)$ is interpreted as the dissipation.

During loading, $W_m = W$, $\dot{W}_m > 0$ and $\eta = 1$. Therefore $\dot{\phi}(\eta) = 0$ and $\dot{\phi}(\eta_m) > 0$ during loading.

During unloading or reloading, $W_m > W$, $\dot{W}_m = 0$ and $\eta_m \le \eta < 1$. Therefore $\dot{\phi}(\eta) \ne 0$ and $\dot{\phi}(\eta_m) = 0$ during unloading or reloading.

Material constants m and r do not have any direct physical significance. However Fig 3.7-6 shows the dependence of an unloading-reloading curve in simple tension on these parameters. It is seen that, for an unloading-reloading loop in which $W_m >> m$, the initial slope of the reloading curve is reduced by the factor $1-\frac{1}{r}$. r must therefore be greater than 1.

It can also be shown that the dissipation of a loading-unloading cycle, as shown in Fig 3.7-7, can be written as

$$\phi = \int_A^E {}_0\sigma\, de = \frac{m}{r}\left[\frac{W_m}{m}\operatorname{erf}\left(\frac{W_m}{m}\right) - \frac{1}{\sqrt{\pi}}\left(1-\exp\left(-\left(\frac{W_m}{m}\right)^2\right)\right)\right]$$
(3.7-79)

where

$$W_m = \int_A^C {}_0\sigma\, de \qquad (3.7\text{-}80)$$

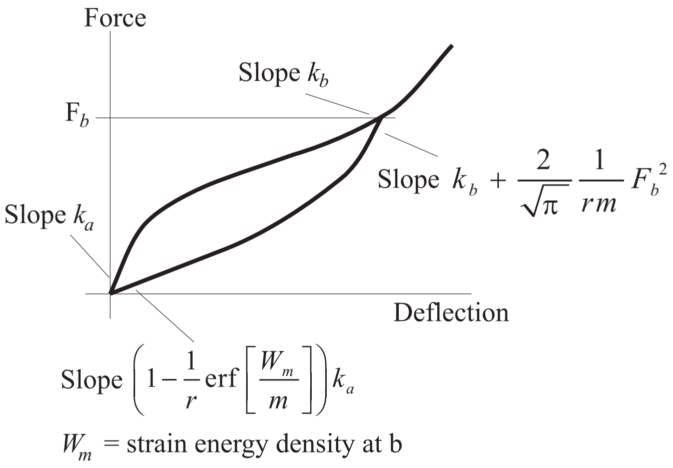

Force
Slope k_b
F_b
Slope $k_b + \frac{2}{\sqrt{\pi}} \frac{1}{rm} F_b^2$
Slope k_a
Deflection
Slope $\left(1 - \frac{1}{r}\mathrm{erf}\left[\frac{W_m}{m}\right]\right) k_a$
W_m = strain energy density at b

Fig. 3.7-6: Dependence of reloading curve on Mullins effect material constants

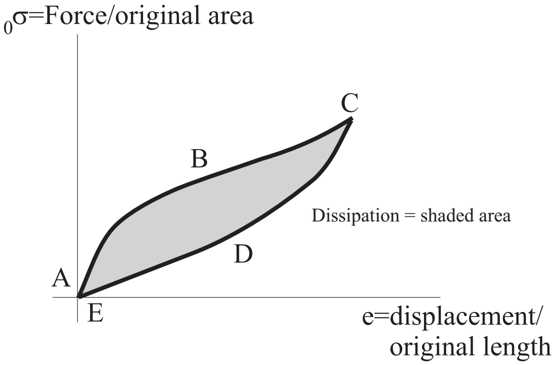


Fig. 3.7-7: Dissipation in Mullins effect in a loading-unloading cycle

$_0\sigma$ is the engineering stress, *e* is the engineering strain. Therefore, given ϕ and W_m from two loading-unloading cycles of different amplitude, m and r can be computed.

Heat generation: A user-specified fraction of the energy dissipated by the Mullins effect model can be considered as heat generation. This heat generation can cause heating in a TMC (thermo-mechanical-coupling) analysis.

Specification of input: Mullins effects are added to the rubber-like material model using the MATHEM bulk data entry.

The rubber-Mullins data set includes:

- R, M: The material constants r and m.
- HGEN: The heat generation factor (fraction of dissipation considered as heat generation). The default value is 0.

3.8 Gasket material model (Solution 601 only)

- Gaskets are relatively thin components placed between two bodies/surfaces to create a sealing effect and prevent fluid leakage (see Fig. 3.8-1). While most gaskets are flat, any arbitrary gasket geometry can be modeled in Solution 601.

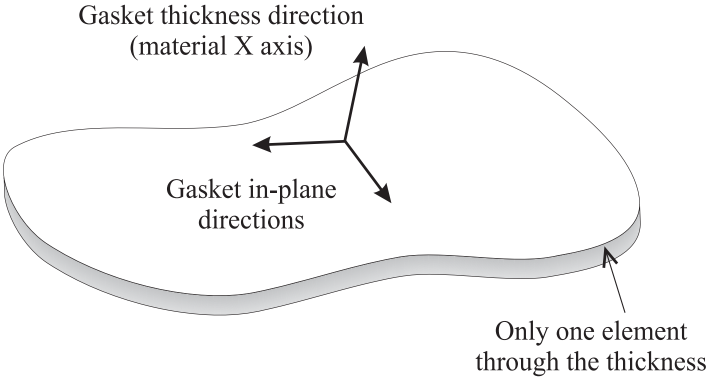


Fig. 3.8-1: Schematic of gasket

- The sealing effect is created when the compressive load, applied in the direction of the gasket thickness, exceeds the initial yield stress of the gasket. The sealing effect is maintained as long as the compressive stress does not drop beyond a specified threshold value. The gasket ruptures if the compressive stress exceeds the gasket's ultimate stress. Unlike rupture, if a gasket leaks it still maintains its load-deflection characteristics.

- The gasket model can be used with 3-D solid elements. It can also be used with **small displacement/small strain**, **large displacement/small strain** formulations.

- The gasket behaves as a nonlinear elasto-plastic material when compressed in the thickness or gasket direction. Its load-deformation characteristics are typically represented by pressure-closure curves. Tensile stiffness can be assumed to be constant or zero. Hardening is assumed to be isotropic.

- Since the gasket material has different properties in different directions, it is considered to be an orthotropic material. The gasket thickness direction is defined as the x direction of the material coordinate system. The program attempts to determine this direction automatically, see below.

- The closure strain is always defined as the change in gasket thickness divided by the original gasket thickness. It is positive in compression. The gasket pressure has units of stress, and it is also positive in compression.

- The gasket's uni-directional plasticity model speeds up computations, and allows more flexibility in defining the shape of the loading and unloading curves.

User input

The gasket material is obtained using the MATG material entry to define the transverse and through-thickness gasket properties together with an elastic isotropic MAT1 material entry to define the in-plane gasket properties.

EPL is the Young's modulus for tensile loading in the thickness direction ($E_{tensile}$).It is allowed to set EPL to a very small number, such as 1E-5, to model a gasket with very small tensile stiffness.

GPL is the transverse shear modulus.

YPRESS specifies the initial yield pressure. This value must correspond to a point on the TABLD curve, see below.

IDMEM is the ID of a MAT1 entry that provides the material constants for the membrane (in-plane) behavior.

TABLD and the TABLUi are used to enter the pressure - closure strain relationship. These curves are defined using the TABLES1 entry.

The leakage pressure is automatically set to 1% of the initial yield pressure.

Pressure - closure strain input

We now describe the pressure - closure strain input in more detail. Fig. 3.8-2 shows a typical pressure - closure strain relationship. The main loading curve is specified by parameter TABLD. The corresponding TABLES1 entry can contain any number of points, and successive points are connected by straight lines.

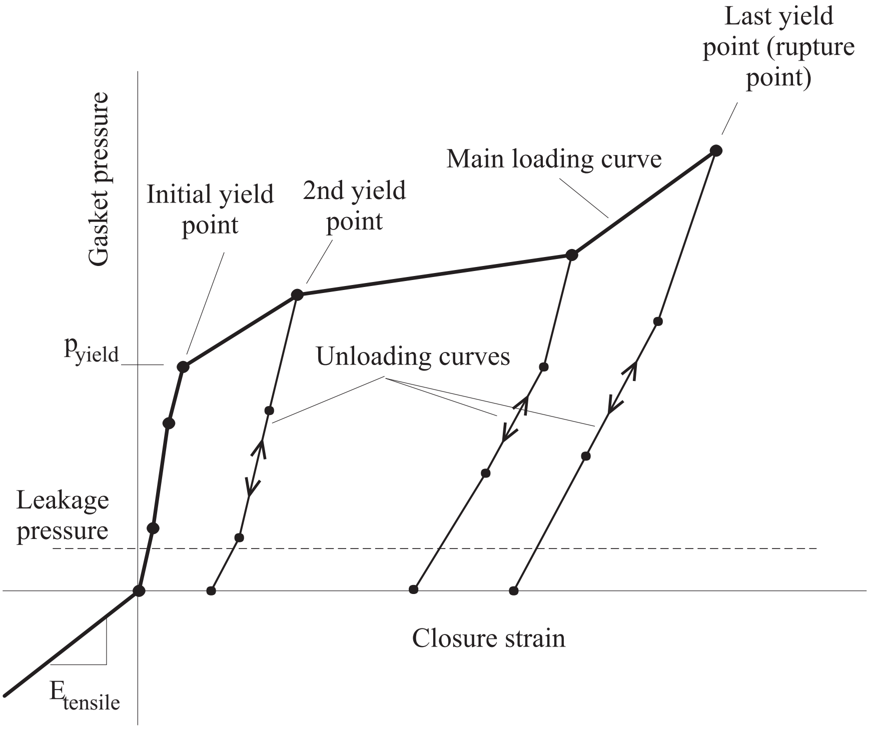


Fig. 3.8-2: Pressure-closure strain input for a gasket material

The point number of the first yield point on the yield curve is given by the value of p_{yield} (parameter YPRESS). p_{yield} must match the third, fourth or higher point on the yield curve. For future reference, the point number corresponding to p_{yield} is denoted NPOINTS. NPOINTS must be greater than or equal to 3.

For each point on the main loading curve higher than the first yield point, there is an associated unloading curve, as shown. Notice that each unloading curve must have the same number of points, and this number of points is equal to NPOINTS. There must be at least three unloading curves, and therefore the total number of points on the yield curve must be at least NPOINTS + 2.

There are two options for specification of each unloading curve:

1) The unloading curve can be specified using one of the TABLUi. The corresponding TABLES1 entry must contain NPOINTS points, the first point must have zero pressure, and the last point must correspond to the (closure strain, pressure) of the main loading curve yield point.
2) The program can automatically construct the unloading curve by interpolation of the user-input unloading curves.

It is necessary to specify at least one of the unloading curves using the TABLUi.

User input for nonlinear elastic gasket

We now describe how to input material data for a nonlinear elastic gasket.

Fig. 3.8-3 shows an example for input of pressure - closure strain data for a nonlinear elastic gasket. In this example, we assume that points 2 to 4 correspond to experimental data. Points 5 to 7 are on a line extended from segment 3-4.

NPOINTS for this example is 5.

Points NPOINTS to NPOINTS+2 should correspond to very high closure strain, so that the program never reaches the closure strain corresponding to point NPOINTS.

The gasket material will be nonlinear elastic as long as the closure strain is less than the closure strain at point NPOINTS, and the relationship between closure strain and gasket pressure is given by the segments connecting points 1 to NPOINTS.

Two unloading curves must be entered, corresponding to points NPOINTS+1 and NPOINTS+2 on the loading curve, as shown. Each unloading curve has NPOINTS equally spaced points, with unloading point 1 for pressure =0 and unloading point NPOINTS for the point coincident with the loading curve.

The tensile behavior of the gasket is governed by constant EPL ($E_{tensile}$).

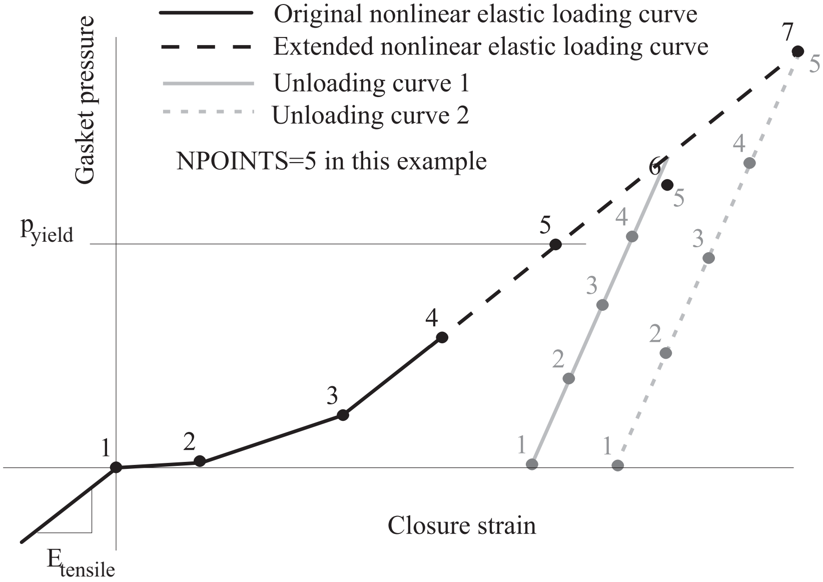


Fig. 3.8-3: Example input for nonlinear elastic gasket

Pressure - closure strain response

Fig. 3.8-4 shows a typical pressure - closure strain response. The gasket is loaded to point B, then unloaded to point C. When the closure strain becomes negative, the response is governed by $E_{tensile}$. Then the gasket is loaded to point D. Up to this time, the gasket behaves as a nonlinear elastic material. Then the gasket is loaded to point E and unloaded to point F. The unloading to point F is done on an unloading curve based at point E; this unloading curve is interpolated from the input unloading curves. The gasket is loaded to point G and unloaded to point H. The gasket follows an unloading curve based at point G and interpolated from the input unloading data. During the loading to point I, the gasket follows the unloading curve to point G, then follows the loading curve to point I. The gasket is then unloaded to points J and K. When the pressure becomes negative, the response is again governed by $E_{tensile}$. During reloading to point L, the gasket follows the same unloading curve as used for points J and K. The gasket is then loaded to point M, at which time the gasket ruptures. The pressure drops to zero and remains zero thereafter (point N).

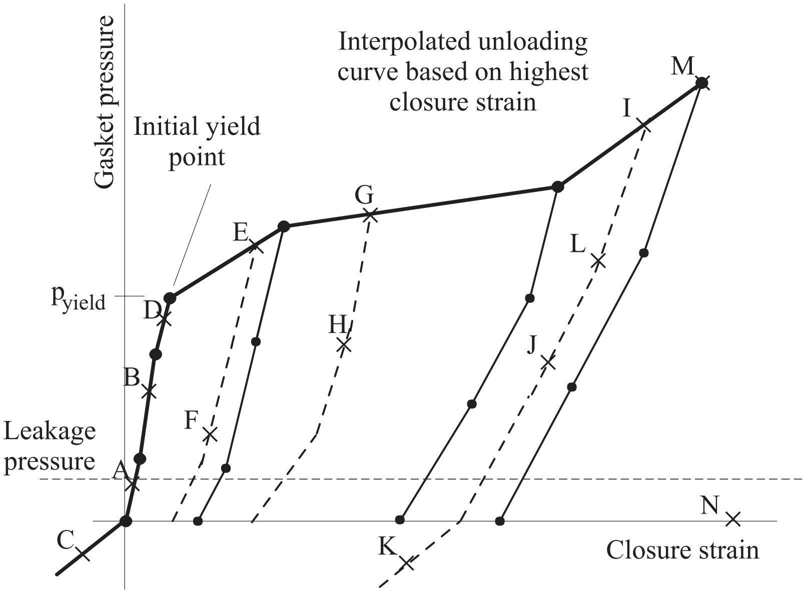


Fig. 3.8-4: Typical pressure - closure strain history

- Each gasket can have one of the following five states:

Open: The gasket pressure is less than the leakage pressure and the gasket has never undergone plastic deformation .

Closed: The gasket pressure is higher than the leakage pressure and the gasket has never undergone plastic deformation.

Sealed: There has been plastic gasket deformation, and thereafter the pressure has always remained above the gasket leakage pressure.

Leaked: There has been plastic gasket deformation, and the gasket pressure has dropped below the gasket leakage pressure for at least one solution step after the plastic deformation occurred.

Crushed: Gasket closure strain has exceeded the rupture value for at least one solution step.

Modeling issues

- The gasket must be modeled as a single layer of 3-D elements. Only linear elements are possible (6-node wedge and 8-node brick elements).

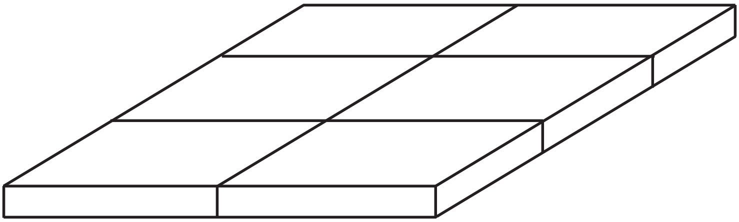
- If the user does not explicitly set the material axes coordinate system in gasket elements (using CORDM of the PSOLID entry), the program attempts to automatically determine the gasket material axes directions. This automatic determination is made based on the element layout. Fig 3.8-5 shows element layouts in which the program can and cannot automatically determine the gasket material directions. If the program can automatically determine the gasket material directions, then the program constructs a material coordinate system in which the x direction corresponds to the gasket thickness direction.

- Warning, if the program cannot automatically determine the gasket material directions, and if the material axes for the gasket are not explicitly defined by the user, then the basic coordinate system is used, and the x direction of the basic coordinate system corresponds to the gasket thickness direction. If the gasket thickness direction is in fact not the x direction of the basic coordinate system, the results will be incorrect.

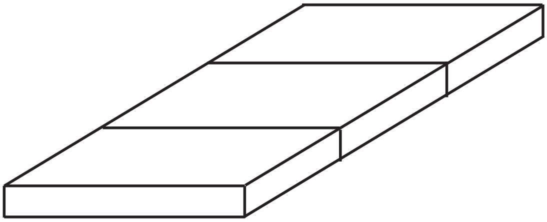
- The top and bottom surfaces of a gasket can be separate from those of the mating surfaces. In this case, they should be connected via contact. The gasket can also share a common surface with the intended mating surface. In this case, contact is not needed, however, the gasket cannot separate from its target. A gasket surface can also be attached to its mating surface via tied contact, mesh glueing, constraint conditions, or rigid elements.

Output variables: The following gasket output variables are available: Gasket pressure, Gasket closure strain, Gasket yield stress, Gasket plastic closure strain, Gasket status.

- Note that all these output variables are scalar quantities.



(a) Each element is surrounded by neighboring elements on adjacent sides. Gasket direction can be automatically determined.



(b) At least one element is surrounded by neighboring elements on opposite sides only. Gasket direction cannot be automatically determined.

Fig. 3.8-5: Automatic determination of gasket material directions

3.9. Shape memory alloy (Solution 601 only)

- The Shape Memory Alloy (SMA) material model is intended to model the superelastic effect (SE) and the shape memory effect (SME) of shape-memory alloys. It is defined using the MATSMA material entry.

- The SMA material model can be used with **rod**, **2-D solid**, **3-D solid** and **shell** elements. It is available only for implicit analysis (Solution 601).

- Shape memory alloy materials can undergo solid-to-solid phase transformations induced by stress or temperature. The high temperature phase is called austenite (A) with a body-centered cubic structure and the low-temperature phase is called martensite (M) with a monoclinic crystal structure in several variants.

Fig.3.9-1 shows a schematic SMA stress-temperature diagram. The martensite phase in two generalized variants and the austenite phase are shown, as well as stress- and temperature-dependent transformation conditions. The martensite phase is favored at low temperatures or high stresses. Upon heating from low temperature the material begins transforming from martensite to austenite at temperature A_s. The transformation is 100% complete at temperature A_f. If the material is then cooled again, the austenite starts transforming back to martensite at temperature M_s. This transformation is 100% complete at temperature M_f. These four temperatures are also stress dependent with high stresses favoring the martensite phase. This stress dependence is assumed linear with slope C_M and C_A for the martensite and austenite temperatures, respectively. A typical variation of volume fraction of martensite in the SMA material with temperature is shown in Fig. 3.9-2.

- A typical uniaxial isothermal stress-strain curve is shown in Fig. 3.9-3.

- The superelastic effect is evident when the material is deformed at temperature $\theta > A_f$ and is displayed in Fig. 3.9-3(a). The stress cycle application induces transformations from A→M and then from M→A to exhibit the hysteresis loop. The shape memory effect is evident when the material is deformed at temperature $\theta < A_s$ and is displayed in Fig. 3.9-3(b). A residual transformation strain remains after unloading; however heating the material to temperature above A_f leads to thermally induced M→A transformation and the recovery of transformation strain.

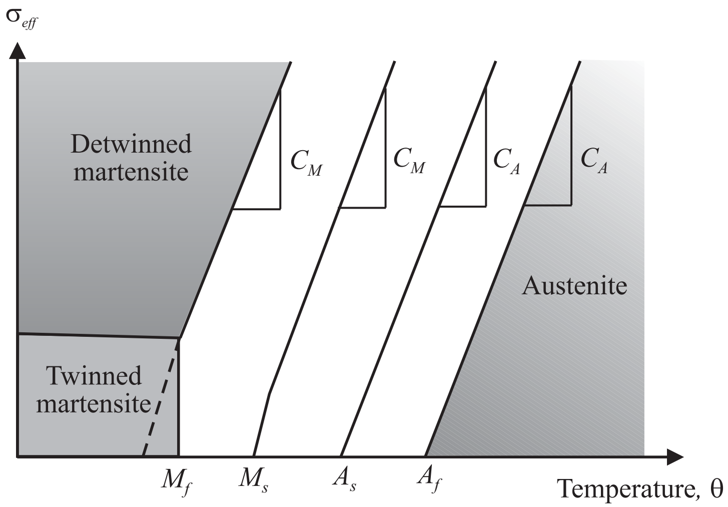


Fig. 3.9-1: SMA stress-temperature phase diagram

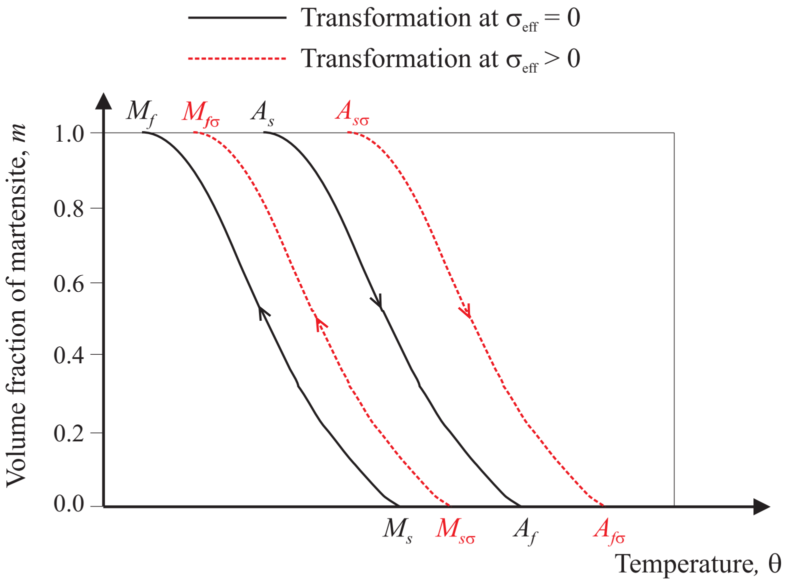
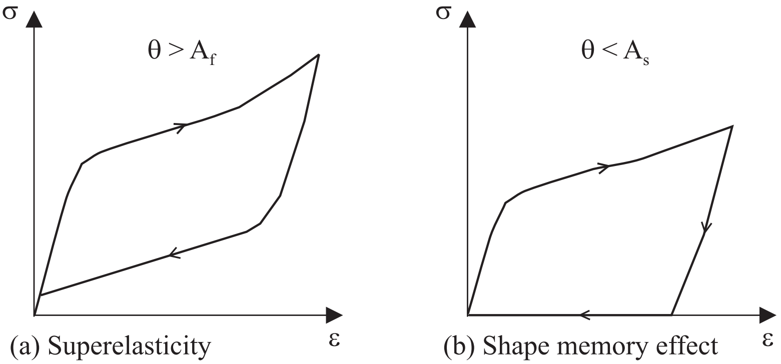


Fig. 3.9-2: Volume fraction of martensite vs. temperature



(a) Superelasticity

(b) Shape memory effect

Fig. 3.9-3: Schematic of stress-strain curves for shape-memory alloys

- Both shape memory effects due to transformation from martensite to austenite and due to re-orientation of the martensite are captured by modeling the twinned and detwinned martensites as different phases.

- The SMA material model is based on the following equations:

  ▸ The total strain,

$$\varepsilon = \varepsilon^e + \varepsilon^t + \varepsilon^\theta$$

where

$\varepsilon^e =$ elastic strain
$\varepsilon^\theta =$ thermal strain
$\varepsilon^t =$ transformation strain; to be evaluated

  ▸ The one-dimensional macro-scale model,

$$\xi = \xi_s + \xi_t; \qquad 0 \leq \xi \leq 1$$

$$\xi + \xi_A = 1$$

$$\varepsilon^t = \varepsilon^t_{max}\, \xi_s$$

$$\sigma = ((1-\xi)E_A + \xi E_M)(\varepsilon - \varepsilon^{\theta} - \varepsilon^{t}{}_{\max}\xi_s)$$

ξ_t = twinned martensite volume fraction
ξ_s = detwinned martensite volume fraction
ξ_A = austenite volume fraction
$\varepsilon^t{}_{max}$ = maximum recoverable residual strain; a material property usually obtained from a simple tension test when the material is fully detwinned martensite (ξ_s = 1)

- The flow rule of three-dimensional constitutive model,

$$\Delta\varepsilon_{ij}^{t} = \Delta\xi_s \varepsilon_{\max}^{t} n_{ij}^{t}$$

$$n_{ij}^{t} = \sqrt{\frac{3}{2}}\left(\frac{s_{ij}}{\bar{\sigma}}\right)\text{; for the martensitic transformation}$$

$$n_{ij}^{t} = \sqrt{\frac{3}{2}}\left(\frac{\varepsilon_{ij}^{t}}{\bar{\varepsilon}^{t}}\right)\text{; for the reverse transformation}$$

where

s_{ij} = deviatoric stresses

$$\bar{\sigma} = \sqrt{\frac{3}{2} s_{ij} s_{ij}} \quad \text{is the effective von Mises stress}$$

$$\bar{\varepsilon}^{t} = \sqrt{\frac{3}{2} \varepsilon_{ij}^{t} \varepsilon_{ij}^{t}} \quad \text{is the effective transformation strain}$$

This results in the following equation for deviatoric stress calculation:

$${}^{t+\Delta t}s_{ij} = \frac{{}^{t+\Delta t}E(\xi)}{1+{}^{t+\Delta t}\nu(\xi)}\left({}^{t+\Delta t}\varepsilon_{ij}'' - \Delta\varepsilon_{ij}^{t}\right)$$

where

$$^{t+\Delta t}\varepsilon''_{ij} = {}^{t+\Delta t}\varepsilon'_{ij} - {}^{t}\varepsilon^{t}_{ij}$$

- Four phase transformation conditions,

1. Starting condition for the martensitic transformation

$$f_{M_S} = \sqrt{3J_2} - C_M(\theta - M_S)$$

2. Ending condition for the martensitic transformation

$$f_{M_f} = \sqrt{3J_2} - C_M(\theta - M_f)$$

3. Starting condition for the reverse transformation

$$f_{A_S} = \sqrt{3J_2} - C_A(\theta - A_S)$$

4. Ending condition for the reverse transformation

$$f_{A_f} = \sqrt{3J_2} - C_A(\theta - A_f)$$

- The phase transformation rate using linear kinetic rule (Auricchio and E. Sacco, 1999),

$$\Delta\xi = \frac{R_\xi}{f_f}\Delta f_f$$

$$\Delta f = \frac{3}{2}\frac{^{t+\Delta t}J_2 - {}^{t}J_2}{^{t+\Delta t}\overline{\sigma}} - c(^{t+\Delta t}\theta - {}^{t}\theta)$$

where, for the austenite to martensite transformation,

f_f = -f_{M_f}, c = C_M and R_ξ = 1 - ξ

and for the reverse martensite to austenite transformation,

$$f_f = f_{A_f}, \quad c = C_A \text{ and } R_\xi = \xi$$

- ▸ Evolution of single-variant detwinned martensite:

Martensite re-orientation is based on the following condition

$$f_R = \sqrt{3J_2} - C_R\theta - \sigma_R$$

where

σ_R = material yield property at $\theta = 0$

C_R = slope of yield function temperature variation

Austenite to martensite transformation leads to

$$\dot{\xi}_s = \dot{\xi}$$

Martensite to austenite transformation leads to proportional transformation of the twinned and detwinned phases:

$$\dot{\xi}_s = \frac{\xi_s}{\xi}\dot{\xi}$$

$$\dot{\xi}_t = \frac{\xi_t}{\xi}\dot{\xi}$$

- The computational steps for the stress-integration of the SMA model are as follows (Kojić and Bathe, 2005):

1. Calculate the trial deviatoric stresses, assuming no additional phase transformation or re-orientation,

$$^{t+\Delta t}s_{ij}^{TR} = \frac{E(^t\xi)}{1+\nu(^t\xi)}(^{t+\Delta t}\varepsilon''_{ij})$$

2. Check for martensitic re-orientation,

$$f_R > 0 \text{ and } {}^t\xi_s < {}^t\xi \text{ and } {}^t\xi < 1$$

Check for austenite to martensite transformation,

$$f_{M_f} f_{M_s} < 0 \text{ and } {}^t\xi < 1 \text{ and } \Delta f > 0$$

Check for martensite to austenite transformation,

$$f_{A_f} f_{A_s} < 0 \text{ and } {}^t\xi > 0 \text{ and } \Delta f < 0$$

3. In case of martensitic re-orientation solve the following governing equation:

$$g_2(\Delta\xi_s) = \sqrt{\frac{3}{2}\,{}^{t+\Delta t}s_{ij}\,{}^{t+\Delta t}s_{ij}} - C_R\,{}^{t+\Delta t}\theta - \sigma_R = 0$$

$$(3.9\text{-}1)$$

$${}^{t+\Delta t}s_{ij} = \frac{{}^{t+\Delta t}E(\Delta\xi)}{1+{}^{t+\Delta t}\nu(\Delta\xi)}({}^{t+\Delta t}\varepsilon''_{ij} - \Delta\xi_s \cdot \varepsilon^t_{\max}\,{}^{t+\Delta t}n^t_{ij})$$

The martensite reorientation calculation step is optional; it is activated when $\sigma_R > 0$ is input.

4. In case of austenite to martensite transformation, solve the following governing equation:

$$g(\Delta\xi) = \Delta\xi - \frac{{}^{t+\Delta t}R_\xi}{{}^{t+\Delta t}f_f}\left[\frac{3}{2}\frac{{}^{t+\Delta t}J_2(\Delta\xi) - {}^tJ_2}{{}^{t+\Delta t}\overline{\sigma}} - c({}^{t+\Delta t}\theta - {}^t\theta)\right] = 0$$

$$(3.9\text{-}2)$$

where

$${}^{t+\Delta t}s_{ij} = \frac{{}^{t+\Delta t}E(\Delta\xi)}{1+{}^{t+\Delta t}\nu(\Delta\xi)}({}^{t+\Delta t}\varepsilon''_{ij} - \Delta\xi_s(\Delta\xi) \cdot \varepsilon^t_{\max}\,{}^{t+\Delta t}n^t_{ij})$$

$R_{\xi} = 1 - \xi$, $f_f = -f_{M_f}$ and $c = C_M$

5. In case of martensite to austenite transformation, solve the governing equation (3.9-2) with

$R_{\xi} = \xi$, $f_f = f_{A_f}$ and $c = C_A$

6. Update history-dependent variables for this time step/iteration step.

7. Calculate the consistent tangent constitutive matrix.

ref. M. Kojić and K.J. Bathe, *Inelastic Analysis of Solids and Structures*, Springer, 2005

ref. F. Auricchio and E. Sacco, “A Temperature-Dependent Beam for Shape-Memory Alloys: Constitutive Modelling, Finite-Element Implementation and Numerical Simulation”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 174, pp. 171-190 (1999)

3.10 Viscoelastic material model (Solution 601 only)

- The viscoelastic model can be used with the **rod, 2-D solid**, **3-D solid** and **shell** elements.

- The viscoelastic model can be used with the **small displacement/small strain**, **large displacement/small strain** and **large displacement/large strain** kinematics (2-D solid and 3-D solid elements only).

When used with the small displacement/small strain kinematics, a materially-nonlinear-only formulation is employed, when used with the large displacement/small strain kinematics, a TL formulation is employed and when used with the large displacement/large strain kinematics, the ULH formulation is employed.

- The mechanical behavior for an isotropic and linear viscoelastic material may be expressed in tensor notation as

$$s_{ij}(t) = 2G(0)e_{ij}(t) + 2\int_{0^+}^{t} e_{ij}(t-\tau)\frac{dG(\tau)}{d\tau}d\tau \qquad (3.10\text{-}1)$$

$$\sigma_{kk}(t) = 3K(0)\varepsilon_{kk}(t) + 3\int_{0^+}^{t} \varepsilon_{kk}(t-\tau)\frac{dK(\tau)}{d\tau}d\tau \qquad (3.10\text{-}2)$$

where t is the time, $s_{ij} = \sigma_{ij} - \frac{1}{3}\delta_{ij}\sigma_{kk}$ is the deviatoric stress, δ_{ij} is the Kronecker delta, σ_{ij} is the stress, $e_{ij} = \varepsilon_{ij} - \frac{1}{3}\delta_{ij}\varepsilon_{kk}$ is the deviatoric strain, ε_{ij} is the strain, $G(t)$ is the shear modulus and $K(t)$ is the bulk modulus.

In the presence of a temperature variation $\theta(t)$ the stresses for an isotropic and thermorheologically linear viscoelastic material may be written as

$$s_{ij}(t) = 2G(0)e_{ij}(t) + 2\int_{0^+}^{\xi} e_{ij}(\xi-\zeta)\frac{dG(\zeta)}{d\zeta}d\zeta \qquad (3.10\text{-}3)$$

$$\sigma_{kk}(t) = 3K(0)\left(\varepsilon_{kk}(t) - \varepsilon_{kk}^{th}(t)\right) + 3\int_{0^+}^{\xi}\left[\varepsilon_{kk}(\xi-\zeta)\right]\frac{dK(\zeta)}{d\zeta}d\zeta - 3\varepsilon_{kk}^{th}(t)\int_{0^+}^{\xi}\frac{dK(\zeta)}{d\zeta}d\zeta \qquad (3.10\text{-}4)$$

where

$$\zeta = \int_{0^+}^{\tau}\psi\left[\theta(\eta)\right]d\eta\,, \quad \xi = \int_{0^+}^{t}\psi\left[\theta(\tau)\right]d\tau \qquad (3.10\text{-}5)$$

and the thermal strain is given by

$$\varepsilon_{kk}^{th}(t) = 3\alpha\left(\theta(t)\right)\left[\theta(t) - \theta_{TALPHA}\right] - 3\alpha\left(\theta_0\right)\left[\theta_0 - \theta_{TALPHA}\right] \qquad (3.10\text{-}6)$$

$\alpha\left(\theta(t)\right)$ is the temperature-dependent coefficient of thermal expansion and $\psi(t)$ is the shift function, which obeys

$$\psi(\theta_{TALPHA}) = 1,\ \ \psi(\theta) > 0,\ \ \frac{d\psi}{d\theta} > 0 \tag{3.10-7}$$

Note that θ_{TALPHA} is the reference temperature used for thermal strain calculation.

In equations (3.10-3) and (3.10-4) it is assumed that the mechanical and thermal responses are uncoupled. Furthermore if the temperature is constant, equations (3.10-3) and (3.10-4) reduce to equations (3.10-1) and (3.10-2).

- We assume the following thermo-material properties:

$$G(t) = G_\infty + \sum_{i=1}^{\eta_G} G_i e^{-\beta_i t} \tag{3.10-8}$$

$$K(t) = K_\infty + \sum_{i=1}^{\eta_K} K_i e^{-\gamma_i t} \tag{3.10-9}$$

$$\theta(t) \neq 0 \tag{3.10-10}$$

$$\alpha = \alpha\left(\theta(t)\right) \tag{3.10-11}$$

where G_∞ and K_∞ are the long-time shear modulus and bulk modulus respectively, β_i and γ_i are the decay constants for the shear modulus and bulk modulus respectively and η_G and η_K are the number of time-dependent terms for the shear modulus and bulk modulus respectively. Equations (3.10-8) and (3.10-9) are referred to in the literature as Prony or Dirichlet series. η_G and η_K are limited to a maximum value of 15.

The shift function used is either the Williams-Landell-Ferry (WLF) equation, written as follows

$$\log_{10}\psi(\theta) = -\frac{C_1\left(\theta-\theta_0\right)}{C_2+\left(\theta-\theta_0\right)} \tag{3.10-12}$$

or the Arrhenius shift function

$$\begin{aligned}\log_{10}\psi(\theta) &= C_1\left(\frac{1}{\theta}-\frac{1}{\theta_0}\right), \theta \geq \theta_0 \\ &= C_2\left(\frac{1}{\theta}-\frac{1}{\theta_0}\right), \theta < \theta_0\end{aligned} \qquad (3.10\text{-}13)$$

in which C_1 and C_2 are material constants, and θ_0 is defined as the initial temperature of the model, which must be the same as the reference temperature of the viscoelastic material model.

- The viscoelastic material is specified using the MATVE bulk data entry. The MATVE bulk data entry uses TABVE bulk data entries for the input of the shear and bulk modulus relaxation functions.

- The nodal point temperatures are input as discussed in Section 5.6.

- For more information, see the following references:

ref. W.N. Findley, J.S. Lai and K. Onaran, *Creep and relaxation of nonlinear viscoelastic materials*, Dover Publications, 1976.

ref. R.L. Frutiger and T.C. Woo, "A thermoviscoelastic analysis for circular plates of thermorheologically simple material", *Journal of Thermal Stresses*, 2:45-60, 1979.

3.11 Heat transfer materials (Solution 601 only)

- Heat transfer materials are available for heat transfer analyses and coupled structural heat transfer analyses (SOL 601,153 and SOL 601,159)

- The isotropic materials in this section (MAT4 and MATT4) are available for rod, beam, 2-D solid, 3-D solid, and shell elements. The orthotropic materials (MAT5 and MATT5) are only available for 3-D solid and shell elements.

- The convection heat transfer coefficient and heat generation capacity are input via the MAT4/MAT5 entries. However, in this manual they are considered as loads and boundary conditions and are therefore addressed in Chapter 5.

3.11.1 Constant isotropic material properties

- This material model is obtained with a MAT4 material entry. The thermal conductivity and heat capacity are independent of temperature and time and do not exhibit any directional dependence due to the material.

3.11.2 Constant orthotropic conductivity

- This material model is obtained with a MAT5 material entry. The thermal conductivity is orthotropic, that is, the model exhibits a directional dependency. Three constants k_1, k_2, k_3 give the thermal conductivity along material axes (1,2,3), respectively.

- The heat capacity is isotropic for this model.

3.11.3 Temperature dependent thermal properties

- Both the constant isotropic and the constant orthotropic material models can be made temperature dependent by adding MATT4 or MATT5 material entries.

- Both thermal conductivity and heat capacity can be made temperature dependent. In this case they are defined using piecewise linear input curves.

4. Contact conditions

- Contact conditions can be specified in Advanced Nonlinear Solution to model contact involving 3-D solid elements, shell elements, 2-D solid elements, and rigid surfaces.

- Very general contact conditions are assumed:

  - ▸ The points of contact are assumed not known a priori.
  - ▸ Friction can be modeled according to various friction laws (only standard Coulomb friction for Solution 701).
  - ▸ Both sticking and sliding can be modeled.
  - ▸ Repeated contact and separation between multiple bodies is permitted in any sequence.
  - ▸ Self-contact and double-sided contact are permitted.
  - ▸ Tied contact can be modeled (Solution 601 only).
  - ▸ A small displacement contact feature is available.

Some of the contact algorithms used in Advanced Nonlinear Solution are described in the following references:

ref. KJB Section 6.7

ref. Bathe, K.J. and Chaudhary, A., "A Solution Method for Planar and Axisymmetric Contact Problems," *Int. J. Num. Meth. in Eng.,* Vol. 21, pp. 65-88, 1985.

ref. Eterovic, A. and Bathe, K.J., "On the Treatment of Inequality Constraints Arising From Contact Conditions in Finite Element Analysis," *J. Computers & Structures*, Vol. 40, No. 2, pp. 203-209, July 1991.

ref. Pantuso, D., Bathe, K.J. and Bouzinov, P.A."A Finite Element Procedure for the Analysis of Thermo-mechanical Solids in Contact," *J. Computers & Structures*, Vol. 75, No. 6, pp. 551-573, May 2000.

- Contact in Advanced Nonlinear Solution is modeled using contact sets, contact surfaces (regions), contact segments and contact pairs, as explained in much greater detail below.

- Table 4-1 lists the case control commands related to contact, Table 4-2 lists the bulk data entries related to contact surface definition, and Table 4-3 lists the bulk data entries related to contact set definition.

- Most of the features and tolerances needed for contact sets are provided in the BCTPARA entry. An explanation of this entry is provided in the NX Nastran Quick Reference Guide. Some contact settings however apply to all contact sets (such as contact convergence tolerances, suppression of contact oscillations). These settings are provided in the NXSTRAT entry.

Table 4-1: Case Control commands related to contact

Contact Case Control Command	**Description**
BCSET	Selects which contact set to use
BCRESULTS	Selects which contact results to output

Table 4-2: Bulk Data entries related to contact surface definition

Contact Surface Bulk Data Entry	**Description**
BSURFS	Define contact surface on 3-D solid elements (by element and nodes)
BSURF	Define contact surface on shell elements (by element number)
BCPROP	Define contact surface on shell elements (by property ID)
BCPROPS	Define contact surface on free faces of 3-D solid elements (by property ID)
BEDGE	Define contact surface on 2-D axisymmetric, plane strain and plane stress solid elements (by element edges)
BLSEG	Define contact surface on 2-D axisymmetric, plane strain and plane stress solid elements (by node numbers)
BCRPARA	Set parameters for contact surface

Table 4-3: Bulk Data entries related to contact set definition

Contact Set Bulk Data Entry	**Description**
BCTSET	Define contact sets
BCTADD	Define union of contact sets
BCTPARA	Set parameters for contact sets

4.1 Overview

- Contact sets (and their contact surfaces) in Advanced Nonlinear Solution can be either 2-D or 3-D. The contact surfaces should be defined as regions that are initially in contact or that are anticipated to come into contact during the solution.

  - 2-D contact surfaces are either axisymmetric or planar and must lie in the global XZ plane, with all Y coordinates equal to zero. Typical two-dimensional contact surfaces are shown in Fig. 4.1-1.
  - A 3-D contact surface is made up of a group of 3-D contact segments (faces) either on solid elements, shell elements or attached to rigid nodes. See Fig. 4.1-2 for an illustration.

- A contact pair consists of the two contact surfaces that may come into contact during the solution. One of the contact surfaces in the pair is selected to be the contactor surface and the other contact surface to be the target surface. In the case of self-contact, the same surface is selected to be both contactor and target. (Self-contact is when a contact surface is expected to come into contact with itself during the solution.)

- Within a contact pair, the nodes of the contactor surface are prevented from penetrating the segments of the target surface, and not vice versa.

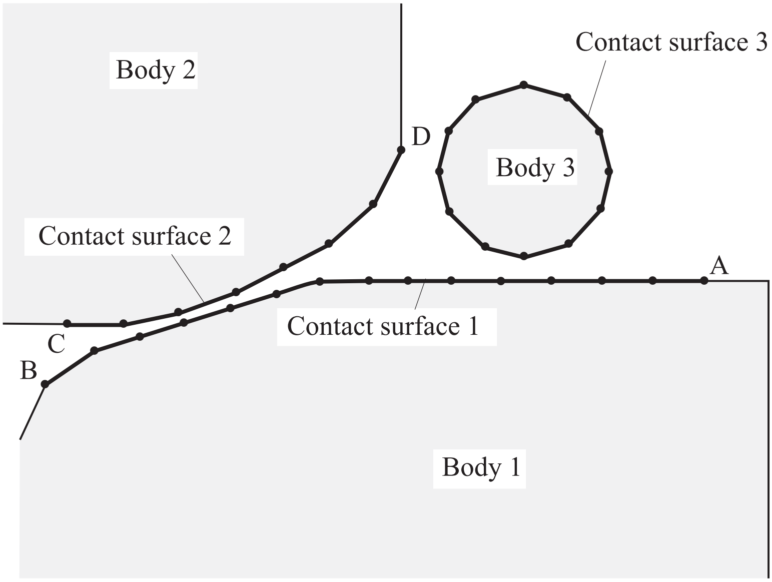


Fig. 4.1-1: Typical 2-D contact surfaces

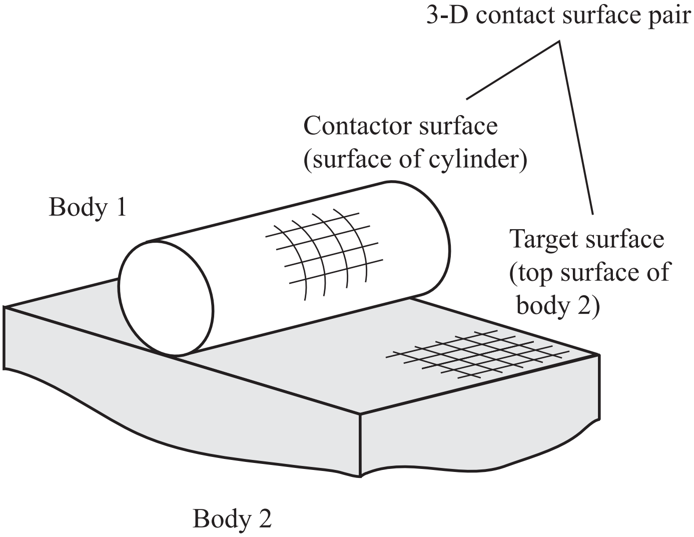


Fig. 4.1-2: Typical contact surfaces and contact pair

- Fig. 4.1-3 shows the effect of contactor and target selection on the different contact configurations.

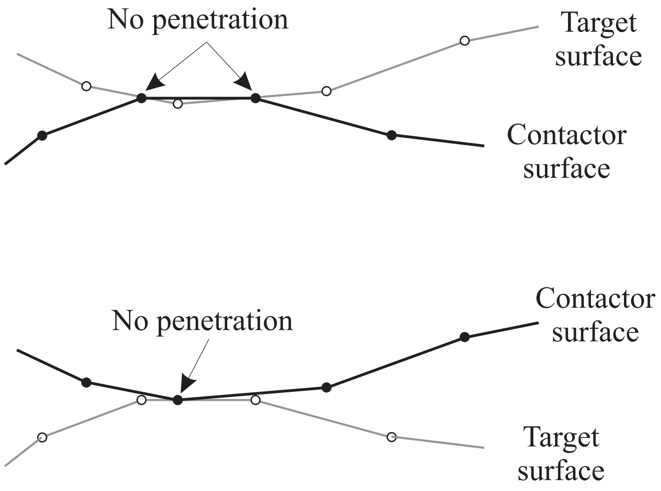


Fig. 4.1-3: Contactor and target selection

- In Solution 601 at least one of the two contact surfaces in a contact pair must not be rigid. If one surface is rigid, this surface should, in most cases, be the target surface.

- In Solution 701 both contactor and target surfaces can be rigid if the penalty algorithm is used. Otherwise, the same restriction mentioned above for Solution 601 applies.

- Rigid surfaces have no underlying elements and therefore no flexibility apart from rigid body motions. All their nodal degrees of freedom must be either fixed, have enforced displacement, or be rigidly linked to a master node which is defined on the BCRPARA entry.

- Symmetric contact pairs can be defined, where in one contact pair surface A can be the contactor and surface B the target, and in another contact pair surface B is the contactor and surface A is the target.

In Solution 601, a non-zero contact surface compliance should always be used with symmetric contact pairs. (Contact surface compliance is discussed in Section 4.4.) Without compliance, the system of contact equations can become overconstrained, that is, the same contact equation can be present more than once in the system of contact equations. When this happens, the global system of equations becomes singular and the solution diverges.

- **Basic concepts**

ref. KJB Section 6.7.2

- The normal contact conditions can ideally be expressed as

$$g \geq 0; \quad \lambda \geq 0; \quad g\lambda = 0 \tag{4.1-1}$$

where g is a gap, and λ is the normal contact force. Different algorithms may vary in the way they impose this condition.

- For friction, a nondimensional friction variable τ can be defined as

$$\tau = \frac{F_T}{\mu\lambda} \tag{4.1-2}$$

where F_T is the tangential force and λ is the normal contact force.

- The standard Coulomb friction condition can be expressed as

$$\begin{aligned} &|\tau| \leq 1 \\ \text{and} \quad &|\tau| < 1 \quad \text{implies } \dot{u} = 0 \\ \text{while} \quad &|\tau| = 1 \quad \text{implies } \operatorname{sign}(\dot{u}) = \operatorname{sign}(\tau) \end{aligned} \tag{4.1-3}$$

where $\dot{u}$ is the sliding velocity.

- In static analysis, the sliding velocity is calculated by dividing the incremental sliding displacement by the time increment. Hence, time is not a dummy variable in static frictional contact problems.

- When (Coulomb) friction is used, the friction coefficient can be constant or calculated from one of several predefined friction laws.

- The possible states of the contactor nodes and/or segments are

  No contact: the gap between the contactor node and target segment is open.

  Sliding: the gap between the contactor node and the target segment is closed; a compression force is acting onto the contactor node and the node kinematically slides along the target segments (either due to frictionless contact or to a frictional restrictive force less than the limit Coulomb force.

  Sticking: as long as the tangential force on the contactor node that initiates sliding is less than the frictional capacity (equal to the normal force times the Coulomb friction coefficient), the contactor node sticks to the target segment.

- **Old and new contact surface representations**

Two types of contact surface representation are supported in Advanced Nonlinear Simulation, an old and a new contact surface representation (set via the CSTYPE parameter in the NXSTRAT entry). The new contact surface representation is the default. The main differences between the two representations are:

- ▸ In the old representation, contact segments are linear (2 nodes for 2D contact; 3- or 4 nodes for 3D contact). In the new representation, contact segments can be linear or quadratic (up to 3 nodes for 2D contact, up to 9 nodes for 3D contact).
- ▸ In the new representation, contact is based on the actual faces of the contact segments which results in more accurate contact constraints.
- ▸ The new representation uses a more accurate contact traction calculation algorithm.
- ▸ The new representation generates more accurate contact constraints for 3-D contact segments resulting from 10, 11 node tet elements and 20, 21 node brick elements. These elements generate zero (10, 11 node tets) or negative (20, 21 node bricks) contact forces at their corner nodes when subjected to a uniform contact pressure.
- ▸ Tractions are reported as nodal quantities in the new surface representation.

The new contact surfaces cannot be used with the following features:

- Segment method algorithm
- Rigid target algorithms

- **Single-sided contact**

For single-sided contact, which is defined using NSIDE=1 parameter on the BCTPARA card (see Fig. 4.1-4), one side of the contact surface is assumed to be internal and the other side to be external. Any contactor node within the internal side of a target surface is assumed to be penetrating and will be moved back to the surface. This single-sided option is ideal for contact surfaces on the faces of solid elements since in that case it is clear that one side is internal to the solid while the other is external. In this case, the external side can usually be predicted from the geometry. This option is also useful for shells when it is known that contact will definitely occur from one direction. In this case, however, the program cannot intuitively predict the internal side of the contact surface.

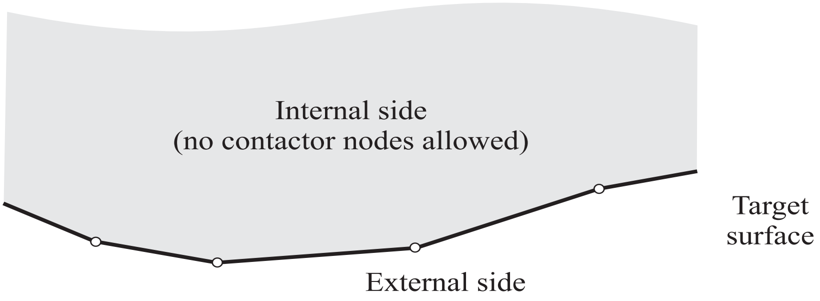


Fig. 4.1-4: Single-sided contact surface

- **Double-sided contact**

In double-sided contact, which is defined using NSIDE=2 parameter on the BCTPARA card (see Fig. 4.1-5), there are no internal or external sides. The contactor surface nodes in this case are prevented from crossing from one side of the target contact surface to the other during solution. This option is more common for shell-based contact surfaces. If a contactor node is on one side of the target surface at time t, it will remain on the same side at time $t + \Delta t$. Note that double sided contact is only supported in 3-D.

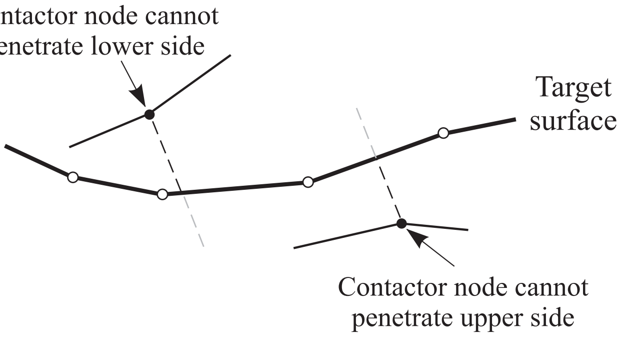

Contactor node cannot penetrate lower side
Target surface
Contactor node cannot penetrate upper side

Fig. 4.1-5: Double-sided contact

- **Tied contact**

When the tied contact feature is selected for a contact set (TIED parameter in the BCTPARA card), Solution 601 performs an initial contact check at the start of the analysis. All contactor nodes that are found to be in contact or overlapping are permanently attached to their respective target segments. Contactor nodes that are not in contact are also set to be tied if the contact gap is less than a user-specified contact tolerance (TIEDTOL parameter in the BCTPARA card). This tolerance is useful when the contact gap is due to non-matching finite element discretizations of the contacting surfaces.

The tied contact feature is conceptually similar to using Rigid elements or multipoint constraints to attach the node to the target surface. The main difference is that the coefficients for the rigid elements are automatically determined by the program and they are only applied for the nodes that are initially in contact. The basic idea is illustrated in Fig. 4.1-6.

Tied contact is not "real" contact because there can be tension between tied contact surfaces. Also no sliding can occur between tied contact surfaces.

The tied contact option can be used to connect two incompatible meshes. However, the mesh glueing feature described in Section 5.9 produces more accurate results.

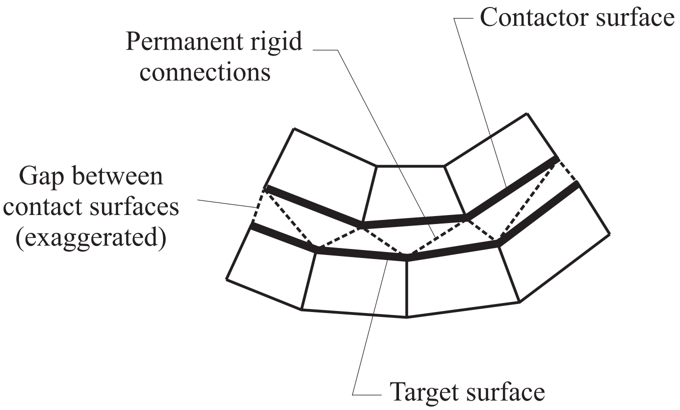

Contactor surface
Permanent rigid connections
Gap between contact surfaces (exaggerated)
Target surface

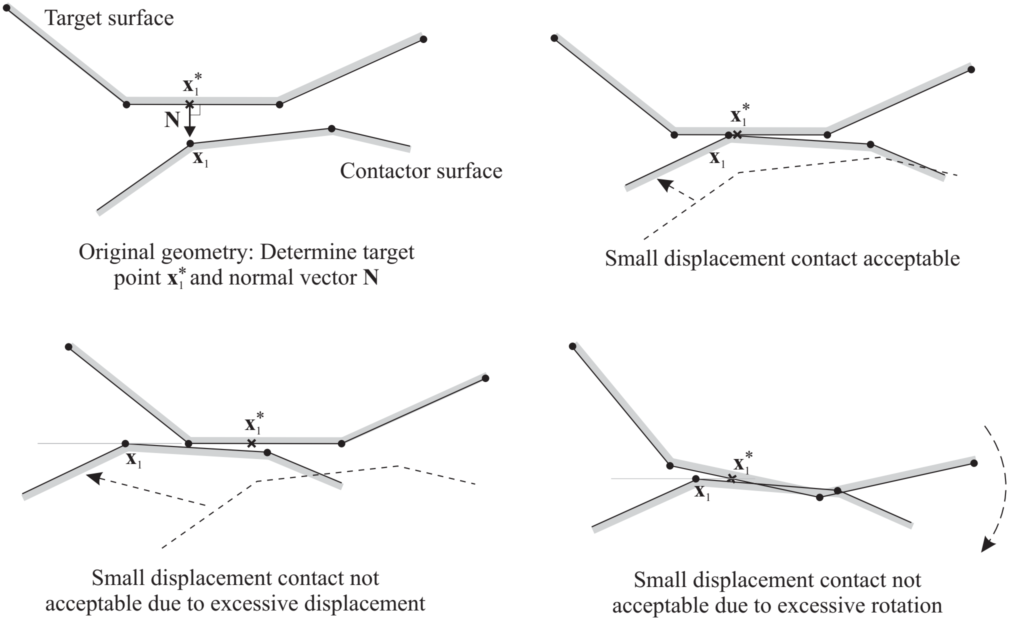
Fig. 4.1-6: Tied contact option

If the contact surfaces initially overlap, they are not pushed back to eliminate the overlap. Similarly, if there is an initial gap it is not eliminated.

The tied contact constraint equations are computed based on the initial nodal positions only. The constraints generated in tied contact are not updated during the analysis. Hence, the constraints will be inaccurate if the bodies experience large rotations.

- **Small displacement contact**

If the small displacement contact feature is used (CTDISP = 1 in the NXSTRAT entry or DISP =1 in the BCTPARA entry), the contact constraints are generated once in the beginning of the analysis and are kept constant, as shown in Fig. 4.1-7. A target location is identified for each contactor node if possible, and its gap and normal direction are determined. The local coordinates of the target point and the normal direction are then kept constant for the remainder of the analysis. This is in contrast to the standard large displacement contact, where the contact constraints are updated every iteration, and the contactor nodes can undergo any amount of sliding.



Original geometry: Determine target point $\mathbf{x}_1^*$ and normal vector **N**

Small displacement contact acceptable

Small displacement contact not acceptable due to excessive displacement

Small displacement contact not acceptable due to excessive rotation

Fig. 4.1-7: Small displacement contact feature

This feature is useful when there is very little relative deformation around the contact region. For such problems, it is much more computationally efficient to perform only one detailed contact search at the beginning of the analysis, rather than repeating the search every iteration. Also, in some cases, convergence can also be slow or unachievable with the general algorithm, for example as nodes oscillate between one target segment and another equally valid neighboring target segment.

- Contact result output is controlled by the BCRESULTS Case Control command. The user can request output of nodal contact forces, nodal contact tractions and/or contact gap distances. Tractions and contact gap distances are only generated on contactor surfaces.

4.2 Contact algorithms for Solution 601

- Solution 601 offers three contact solution algorithms (set via the TYPE flag in the BCTPARA entry):

  ▸ Constraint-function method,
  ▸ Segment (Lagrange multiplier) method, or
  ▸ Rigid target method

- Each contact set must belong to one of these three contact algorithms. However, different contact sets can use different algorithms.

- All 3 contact algorithms can be used with or without friction.

4.2.1 Constraint-function method

- In this algorithm (selected using TYPE=0 on BCTPARA card), constraint functions are used to enforce the no-penetration and the frictional contact conditions.

The inequality constraints of Eq. (4.1-1) are replaced by the following normal constraint function:

$$w(g,\lambda)=\frac{g+\lambda}{2}-\sqrt{\left(\frac{g-\lambda}{2}\right)^2+\varepsilon_N}$$

where ε_N is a small user-defined parameter. The function is shown in Fig. 4.2-1. It involves no inequalities, and is smooth and differentiable. The parameter ε_N is set via the EPSN variable in the BCTPARA entry. The default value of 1.0x10^{-12} is suitable for most applications and should rarely be modified.

It is possible to set EPSN=0.0. In this case Solution 601 automatically determines EPSN. However, this determination may not result in correct results for some problems. Hence EPSN=0.0 should not be used in general.

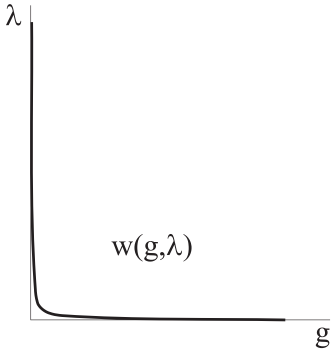


Fig. 4.2-1: Constraint function for normal contact

- The constraint function method is also used to approximate the rigid non-differentiable stick-slip transition of Eq. (4.1-3). This results in a smooth transition from stick to slip and vice versa, and it also results in a differentiable friction law that is less likely to cause convergence difficulties.

Two friction regularization algorithms are available in Advanced Nonlinear Solution. Both constraint functions take the form $\mathrm{v}\left(\dot{u},\tau\right)=0$.

The newer default algorithm involves a more accurate linearization of the frictional constraints and, in general, converges much faster than its predecessor. The v function is defined implicitly as a multilinear function as shown in Fig. 4.2-2. Here ε_T is a small parameter (EPST parameter in the BCTPARA entry) which has the physical meaning of the "sticking velocity", that is, the maximum velocity corresponding to sticking conditions. The default value of EPST is 0.001.

In the old friction algorithm, the v function is defined implicitly via

$$\tau+\mathrm{v}-\frac{2}{\pi}\arctan\left(\frac{\dot{u}-\mathrm{v}}{\varepsilon_T}\right)=0$$

Here ε_T is a small parameter (EPST parameter in BCTPARA entry) which provides some elastic slip to the Coulomb friction law as shown in Fig. 4.2-3. Guidelines for selecting ε_T are provided in section 4.7.3.

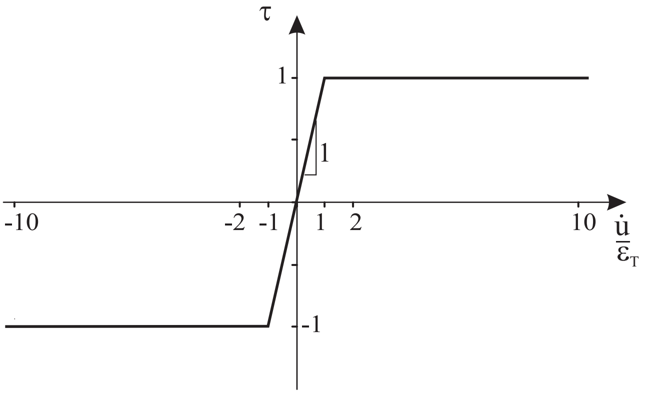


Fig. 4.2-2: Frictional contact constraint function for new friction algorithm

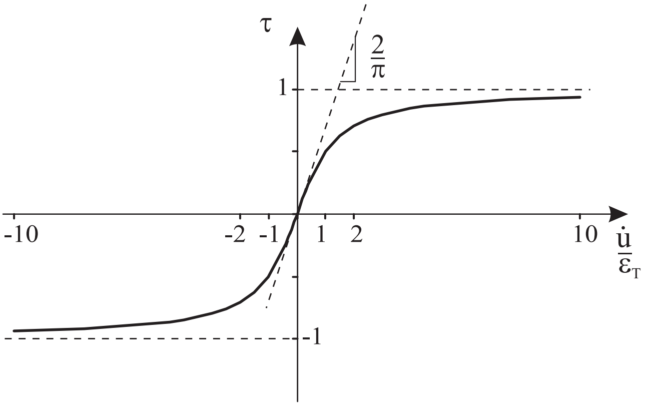


Fig. 4.2-3: Frictional contact constraint function for old friction algorithm

The old friction algorithm can still be accessed via the FRICALG parameter in the NXSTRAT entry.

4.2.2 Segment (Lagrange multiplier) method

- In this method (selected using TYPE=1 on BCTPARA card), Lagrange multipliers are used to enforce the contact conditions of Eq. (4.1-1). The kinematic conditions are enforced at the contactor nodes, and the frictional conditions are enforced over the contact segments.

- This method involves distinct sticking and sliding states. It also calculates this state for each contactor node based on the contact forces on the target segment.

4.2.3 Rigid target method

- This is a simplified contact algorithm (selected using TYPE=2 on BCTPARA card). The algorithm is fully described in Section 4.8.

4.2.4 Selection of contact algorithm

- Our experience is that in most frictionless contact problems the constraint function method is more effective than the segment method. The constraint function method is the default.

- For problems involving rigid targets, either the constraint function or the rigid target algorithm can be employed.

- Note that the target surface can be rigid in all three contact algorithms. The presence of a rigid target does not mean that the rigid target algorithm must be used.

4.3 Contact algorithms for Solution 701

- Solution 701 offers three contact solution algorithms (set via the XTYPE flag in the BCTPARA entry):

  - ▸ Kinematic constraint method,
  - ▸ Penalty method, or
  - ▸ Rigid target method

- Each contact set must belong to one of these three contact algorithms. However, different contact sets can use different algorithms.

- All Solution 701 contact algorithms can be used with or without friction.

4.3.1 Kinematic constraint method

- This algorithm is selected by setting XTYPE=0 on BCTPARA card). It is the default explicit contact algorithm for Solution 701.

- A predictor step is first done without applying contact constraints or forces. Then displacements are evaluated and penetration is detected and corrected. The exact correction of displacements requires the solution of a non-diagonal system of equations. Instead, a good approximation is done. In this case, for each penetrating contactor node, a penetration force

$$\mathbf{F}_C^N = M_C \mathbf{a}_C^{N*} = M_C \frac{\delta_N}{\Delta t^2} \mathbf{N}$$

is calculated. This is the force required to remove the penetration at the contactor node. However, not all the penetration will be removed by moving the contactor. The target will get some motion depending on its mass relative to the contactor and how many contactor nodes are touching it. So, the $\mathbf{F}_C^N$ force above is projected to the target segment nodes:

$$\mathbf{F}_{T_i}^N = N_i \mathbf{F}_C^N$$

where N_i is the shape function relating the contactor displacement to that of each target node. Similarly, the mass of the contactor node is projected to the target in the same way:

$$M_{T_i} = N_i M_C$$

and this mass is added to that of the target node itself. Then the acceleration of the target node is determined as

$$\mathbf{a}_T^N (M_T + \sum M_{T_i}) = \sum \mathbf{F}_{T_i}^N$$

This correction is then used to update the target displacements. The contactor acceleration is

$$\mathbf{a}_C^N = \mathbf{a}_C^{N*} - \sum \mathbf{a}_T^N N_i$$

- For friction, a similar approach is used. A correction force is calculated

$$\mathbf{F}_C^{T*} = M_C \frac{\mathbf{v}_T}{\Delta t}$$

where $\mathbf{v}_T$ is the tangential sliding velocity. However, this force cannot exceed the limit force based on the normal force and the coefficient of friction

$$\mathbf{F}_C^T = \min\left(\mu \mathbf{F}_C^N, \mathbf{F}_C^{T*}\right)$$

The rest of the procedure is very similar to the case of normal contact. The form of the equations is different if there is damping, and is also different if the previous and current time steps are not the same.

- A modification is also required for rigid targets, which are common in contact. The form of the equations in this case depends on whether the rigid target has natural or essential boundary conditions.

- The kinematic constraint algorithm should not be used when the target surface degrees of freedom are fixed.

4.3.2 Penalty method

- In this algorithm (selected using XTYPE=1 on BCTPARA card), contact conditions are imposed by penalizing the interpenetration between contacting surfaces. When a penetration is detected, a normal force of

$$\mathbf{F} = A\ \ \mathbf{P} = A\ \ \mathbf{N}\ \ (K_N\delta_N + K_D\dot{\delta}_N)$$

is applied to the contactor node, where K_N is the normal stiffness, K_D is a normal rate stiffness, δ_N is the penetration, $\dot{\delta}_N$ is the penetration rate, **N** is the normal vector pointing towards the contactor, A is the contact area and **P** is the normal contact traction. An opposing force is distributed to the target nodes.

- Similarly, in the presence of friction, the relative sliding velocity between the two bodies is penalized as follows:

$$\mathbf{F}_T = \min\left(A\,K_T\mathbf{x}_T, \mu\left\|\mathbf{F}_N\right\|\frac{\mathbf{x}_T}{\left\|\mathbf{x}_T\right\|}\right)$$

where $\mathbf{x}_T$ is the relative tangential sliding displacement.

- The normal and tangential penalty stiffnesses K_N and K_T can be selected by the user, or determined automatically by the program based on the following BCTPARA parameters: XKN, XKNCRIT, XKT, XKTCRIT. The penalty rate stiffness K_D can be explicitly selected by the user, or determined by the program as a ratio of critical damping for the contact node (using the XDAMP and XNDAMP parameters).

- When penalty stiffnesses are automatically determined they are chosen based on the masses of the contactor nodes and the time step. They are selected such that they have a minimal effect on the existing time step.

  Note that unduly small penalty stiffnesses will lead to excessive penetrations, and unduly large penalty stiffnesses will lead to excessive oscillations or unstable explicit time integration.

4.3.3 Rigid target method

- This algorithm is similar to the rigid target method used in Solution 601. It is selected using XTYPE=3 on the BCTPARA card. The algorithm is fully described in Section 4.8.

4.3.4 Selection of contact algorithm

- The kinematic constraint method is the default in Solution 701.

- The penalty method is the simplest and fastest of the explicit contact algorithms. It can also handle rigid contactor and target surfaces. It also allows a contactor node to be in contact with multiple targets simultaneously.

- The main disadvantage of the penalty method is that contact conditions are not exactly satisfied and it usually shows oscillations in contact forces. These oscillations can usually be removed by using the penalty rate stiffness factor K_D. The penalty method is also sensitive to the choice of the penalty stiffness. If that stiffness is too large it leads to instability and oscillations, and if it is too small it leads to excessive penetrations.

- The default penalty stiffness selected by Solution 701 is, in most cases, a suitable compromise.

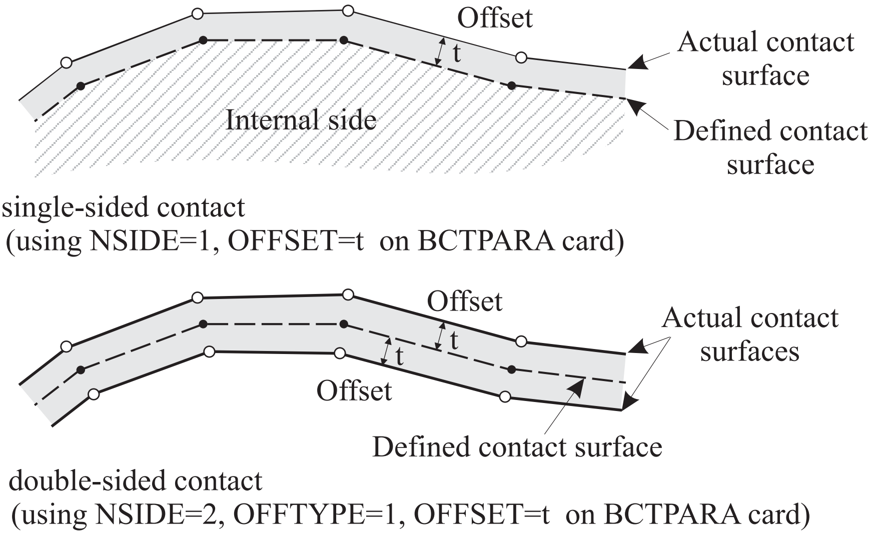
4.4 Contact set properties

This section describes the main options available for contact sets.

- **Contact surface offsets**

Penetration of a contact surface occurs when the plane or line defined by the contact segment nodes is penetrated. However, an offset distance can be specified which causes the actual contact surface to be offset from the plane defined by the contact surface nodes. In the case of double-sided contact, the offset creates two separate surfaces above and below the reference surface. Fig. 4.4-1 shows the possibilities for single and double-sided contact. Note that the offset distance should be small compared to the contact surface length.

Offsets for a whole contact set are specified via the OFFSET parameter in the BCTPARA entry, while offsets for a specific contact surface are set via the OFFSET parameter in the BCRPARA entry. If one of the contact surfaces has a defined offset, it will overwrite the contact set offset.



(a) single-sided contact
(using NSIDE=1, OFFSET=t on BCTPARA card)

(b) double-sided contact
(using NSIDE=2, OFFTYPE=1, OFFSET=t on BCTPARA card)

Fig. 4.4-1: Contact surface offsets

If the contact surface is on a shell then half the true shell thickness can automatically be used as the offset (OFFTYPE=2 in the BCTPARA entry). In this case the shell thickness at a node is obtained as the average of all of the shell thicknesses at that node.

The use of contact surface offsets in double-sided contact is not recommended.

- **Continuous normals (Solution 601 only)**

The normal direction to a contact segment will in general not be continuous between segments as illustrated in Fig. 4.4-2. This sometimes causes convergence difficulties due to the non-unique normals at nodes and segment edges. The continuous normals feature first calculates nodal normals as averages of all the normals from the attached segments, and then interpolates these nodal normals across the segment. This leads to a uniformly varying normal direction.

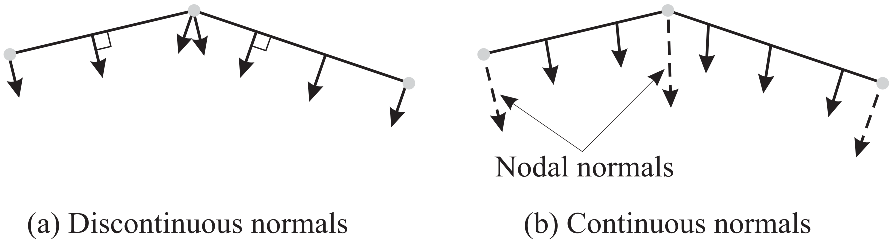


Fig. 4.4-2: Contact surface normals

The SEGNORM parameter in the BCTPARA card determines the setting for continuous normals. Continuous normals (SEGNORM=1) is the default.

In modeling target surfaces with sharp corners, either use discontinuous normal vectors, or use small segments near the corners, in order that the normal vectors for segments near the corners be computed correctly. See Section 4.7.2 for modeling tips related to this feature.

Continuous normals give poor results with explicit time integration. Therefore, they are blocked from Solution 701.

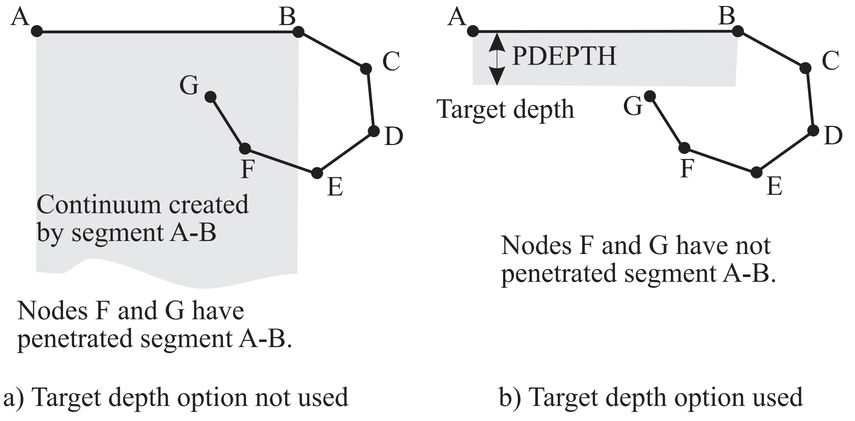
- **Contact surface depth**

By default, the contact region extends for an infinite distance below the contact surface (for single-sided contact). However, a contact surface depth can be defined (by setting the PDEPTH parameter in the BCTPARA card), below which the contact surface is no longer active. The default PDEPTH=0.0 results in an infinite contact depth extension. Fig. 4.4-3 shows some of the possibilities.

- **Initial penetration**

The treatment of initial penetrations in Solution 601 is governed by the INIPENE parameter in the BCTPARA entry. By default, if there is initial overlap (penetration) between a contact node and a target segment in the first solution step, the program attempts to eliminate the overlap. Advanced Nonlinear Solution can eliminate the overlap at the first step or over a user-specified time using the TZPENE parameter in BCTPARA. This feature is useful if the initial penetrations are too large to be eliminated in a single step.

Contact surface is used in self-contact.



Nodes F and G have penetrated segment A-B.

Nodes F and G have not penetrated segment A-B.

a) Target depth option not used

b) Target depth option used

Fig. 4.4-3: Contact surface depth.

The program can also calculate initial penetrations at the start of solution and ignore them in future steps. In this case the program does not detect penetration for a contactor node if the amount of penetration is less than or equal to the recorded amount. Fig. 4.4-4 shows some of the possibilities. See Section 4.7.2 for modeling tips related to this feature.

This feature can be interpreted as a redefinition of the contact gap, as follows:

- *Eliminate penetration option (INIPENE=0):* The contact gap is equal to the geometric gap.
- *Ignore penetration option (INIPENE=2):*
  — Initial geometric gap is positive or zero: The contact gap is equal to the geometric gap.
  — Initial geometric gap is negative: The contact gap is equal to the geometric gap minus the initial geometric gap.

Initial penetrations can also be set to gap override (see below).

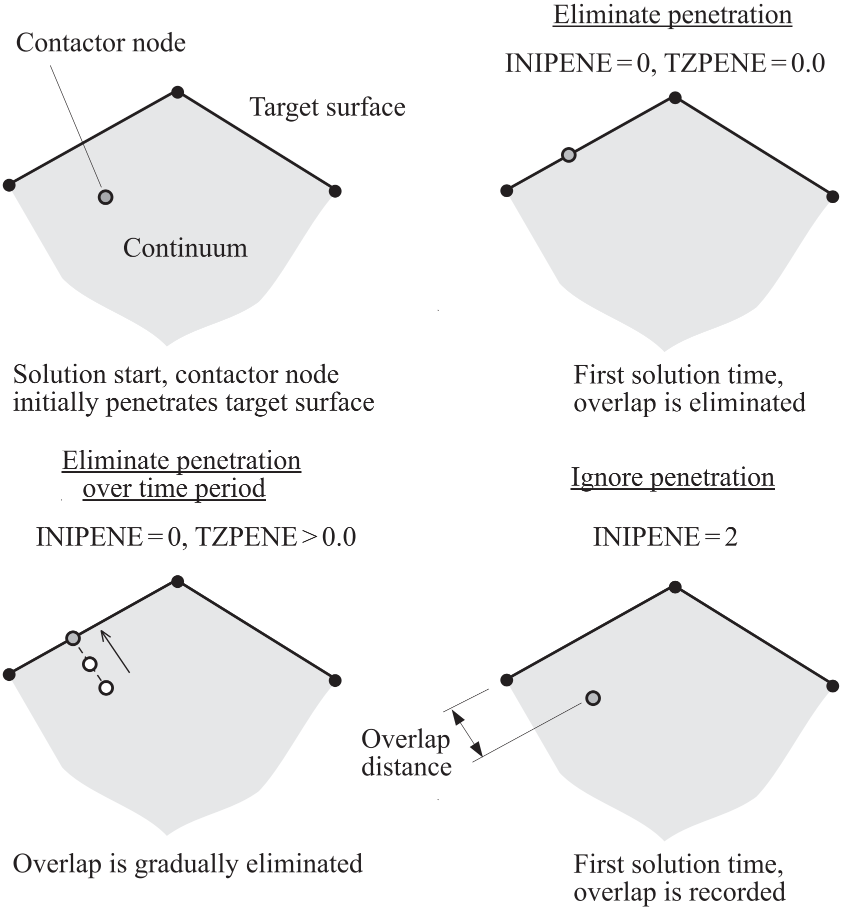

Contactor node
Target surface
Continuum
Solution start, contactor node initially penetrates target surface
Eliminate penetration
INIPENE = 0, TZPENE = 0.0
First solution time, overlap is eliminated
Eliminate penetration over time period
INIPENE = 0, TZPENE > 0.0
Overlap is gradually eliminated
Ignore penetration
INIPENE = 2
Overlap distance
First solution time, overlap is recorded

Fig. 4.4-4: Initial penetration options

- **Gap override**
In the gap override feature (set via the initial penetration flag INIPENE=3 in BCTPARA), the gaps and penetrations calculated from the finite element mesh are replaced by a fixed user-specified value (GAPVAL parameter in the BCTPARA entry). A positive value represents an initial gap, zero means that the contact is touching the target, and a negative value represents an initial penetration (which can be removed either immediately or over a user-specified time as explained in the Section on initial penetration above).

This feature can also be interpreted as a redefinition of the contact gap, as follows:

- *Gap-override option (INIPENE=3):* The contact gap is equal to the geometric gap minus the initial geometric gap plus the gap override GAPVAL. This formula is applied regardless of whether the initial geometric gap is positive, zero or negative.

This feature is useful for problems involving curved meshes in close proximity, such as the shrink fit example shown in Fig. 4.4-5. The gaps and penetrations measured from the discretized finite element mesh are sometimes inaccurate for such problems (unless matching meshes are used). In some problems, such as that shown in the figure, a constant geometry based overlap should be applied to all nodes, which corresponds to a gap override value of –δ.

Note that mesh refinement and quadratic elements reduce the error in the measured overlaps but frequently a very high mesh density would have to be used if gap override is not used.

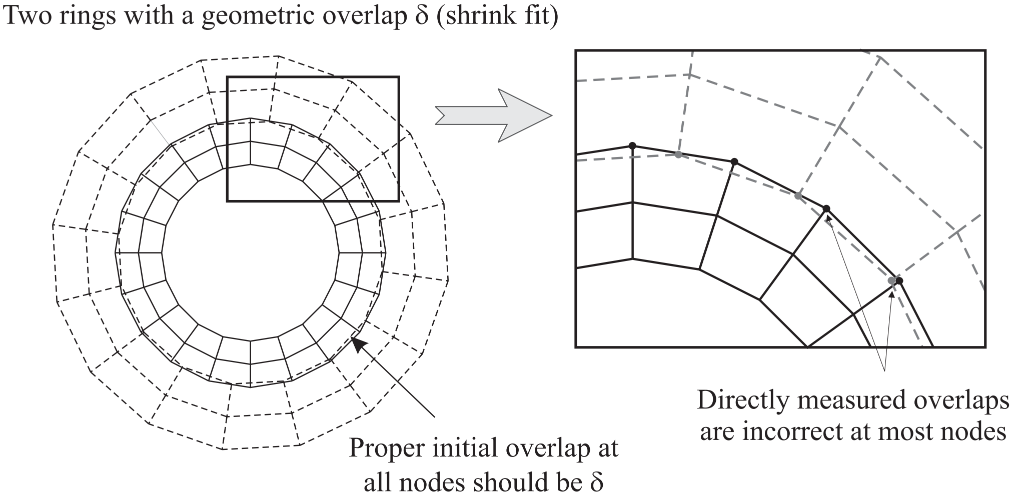

Two rings with a geometric overlap δ (shrink fit)
Proper initial overlap at all nodes should be δ
Directly measured overlaps are incorrect at most nodes

Fig. 4.4-5: Significance of gap override for curved non-matched geometries

Note also that the error in mesh based gaps and penetrations for curved surfaces can be more significant when low precision numbers are used for the node coordinates (such as when short input file format is used). Gap override is also useful for such cases.

- **Contact surface extension**

The target surface can be enlarged beyond its geometric bounds, so that contactor nodes that slip outside the target can still be considered in contact (via the EXTFAC parameter on the BCTPARA card). This feature is useful where the edge of the contactor and target surfaces coincide, as shown in Fig. 4.4-6. Each target segment is enlarged by an amount equal to the contact surface extension factor multiplied by the length of the segment.

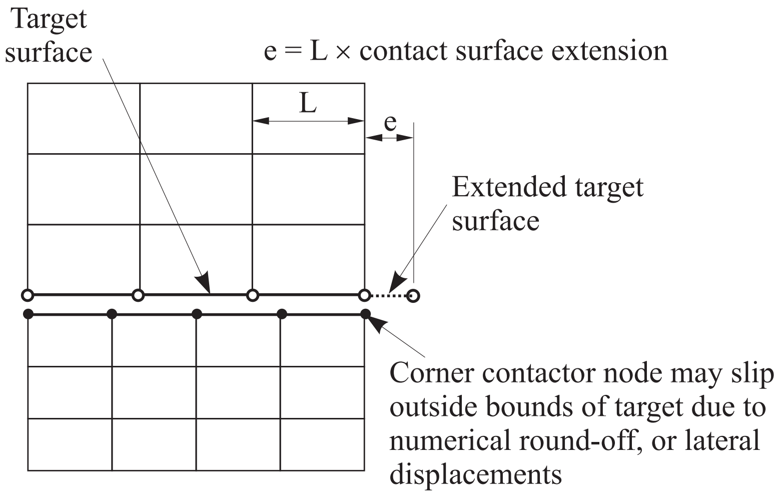

Target surface
e = L × contact surface extension
L
e
Extended target surface
Corner contactor node may slip outside bounds of target due to numerical round-off, or lateral displacements

Fig. 4.4-6: Contact requiring contact surface extension

- **Contact surface compliance (Solution 601 only)**

Contact surface compliance is set via the CFACTOR1 parameter on the BCTPARA card and is only available with the constraint function algorithm in Solution 601. Contact surfaces are commonly assumed to be rigid meaning that no interpenetration is allowed. This situation corresponds to a contact surface compliance of 0.0. However, the contact surface compliance feature can be used to simulate soft or compliant surfaces. The amount of allowed interpenetration between the contacting surfaces in this case is

$$\text{penetration} = \varepsilon_p \times \text{normal contact pressure} \qquad (4.4\text{-}1)$$

where

normal contact pressure = normal contact force / contact area

The constraint function in the presence of a compliance factor is modified as shown in Fig. 4.4-7. *A* is the contact area of the contactor node.

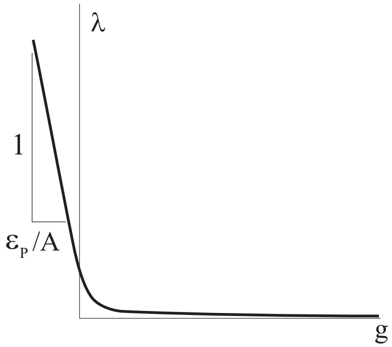


Fig. 4.4-7: Constraint function for compliant contact

The benefits of contact compliance are that it:

Improves the convergence rate. By adding compliance to the contact surface, more contactor nodes come into contact with the target surface. It is better for convergence to have many nodes in contact with small forces at each contactor node, than to have few nodes in contact with large forces at each contactor node.

Improves the contact tractions. Without contact compliance, the discretization error of the mesh often cause some contactor nodes on the contact area to not be in contact which results in spotty contact (or patchy) contact tractions. An example of spotty contact is shown in Fig. 4.4-8.

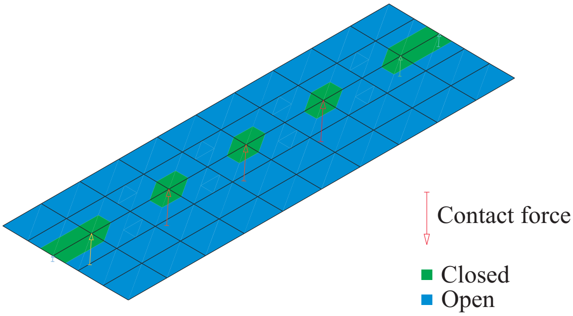


Fig. 4.4-8: Spotty contact on a plate

Reduces numerical oscillations that sometimes result from dynamic time integration. Numerical oscillations, such as chatter or ringing, can be triggered by contact reversal. A node that is out of contact at time $t + \Delta t$, in contact at time t and out of contact at time $t + \Delta t$ is said to have had contact reversal. Contact surface compliance can be used to prevent these oscillations.

However, the drawbacks of contact compliance are:

Nodal contact overlap. The contactor nodes penetrate the target segment when contact compliance is used. The compliance factor must be selected such that the contact pressures do not cause excessive nodal overlap.

Additional (spurious) energy is stored in the contact pair. Elastic energy is stored in the compliant contact surface. This energy is released when the contact pair opens which can have an effect in certain dynamic problems.

Can reduce the resultant contact force and stresses. If a prescribed displacement is applied, the contact surface compliance feature will reduce the stress in the body for a given displacement. The stresses are not reduced if the body is subjected to an applied load.

By default, the compliance factor is set to zero ($\varepsilon_p = 0$). However, in general, some amount of contact compliance should always be used. It is not the default because the appropriate compliance factor is model dependent.

The value of the contact compliance factor is usually set by trial and error. A good starting value is 10^{-5} . The objective is to find the smallest compliance factor that results in convergence with good tractions (i.e., not spotty contact), but does not cause excessive penetration (nodal contact overlap).

After contact compliance is used, the amount of penetration should always be checked to make sure the maximum overlap is acceptable.

When deciding on what penetration is acceptably small we must consider the mesh used. For example, if the target surface is curved, there will be a geometric error associated with using a coarse contractor surface. There is no advantage if the maximum penetration is less than the geometric error. So, if the target surface is coarse, a large maximum penetration can be used.

- **Consistent contact stiffness (Solution 601 only)**

The consistent contact stiffness feature is set via the parameter CSTIFF on the BCTPARA card. Changes in the direction of the contact normal provide an additional contribution to the stiffness matrix that is proportional to the value of the contact force and the change in the normal direction. Therefore, higher convergence rates (closer to quadratic) can sometimes be obtained by selecting the consistent contact stiffness option which accounts for these additional stiffness contributions. This results, however, in an increase in the size of the stiffness matrix which is detrimental for large problems. This option is more beneficial when discontinuous contact normals are selected, because the derivation assumes that the contact normals are discontinuous.

The consistent contact stiffness feature is not used when the target surface is rigid.

Consistent contact stiffness is not used in dynamic analysis.

The default is CSTIFF=0 (consistent contact stiffness is not used).

- **Contact birth/death**

The contact birth feature activates a contact set at a specific time, while the contact death feature disables a contact set at a specific time. They are set via the TBIRTH and TDEATH parameters on the BCTPARA card. A 0.0 birth time means that the contact set starts active at the beginning of the analysis, and a death time less than or equal to the birth time means that the contact set does not die.

- **Friction delay (Solution 601 only)**

When the friction delay feature is activated (FRICDLY parameter in the BCTPARA entry), frictional conditions are applied to a contactor node one time step after contact is established. This feature can be useful in many problems, since it delays the nonlinearity associated with friction until contact is established.

Note that the relative sliding velocity cannot be uniquely determined when a node was not in contact at time *t*, and is in contact at time $t+\Delta t$. That velocity depends on the exact time at which contact started, which is somewhere between times *t* and $t+\Delta t$ (see Fig. 4.4-9). Delaying friction is equivalent to assuming that contact was established close to time $t+\Delta t$, and hence the sliding velocity is zero and so is the frictional force.

Frictoin delay is off by default (FRICDLY=0).

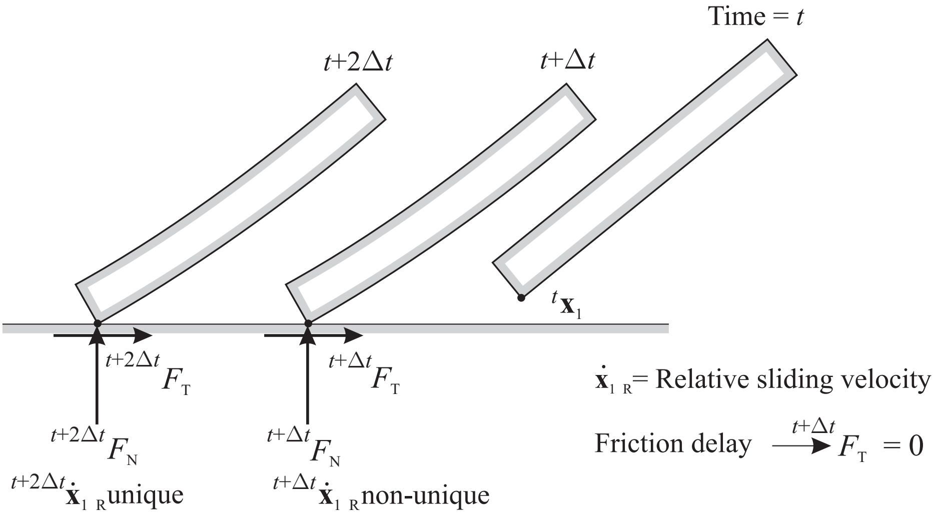

Time = t
$t+2\Delta t$
$t+\Delta t$
${}^{t}\mathbf{x}_1$
${}^{t+2\Delta t}F_T$
${}^{t+\Delta t}F_T$
$\dot{\mathbf{x}}_{1\ R}$= Relative sliding velocity
${}^{t+2\Delta t}F_N$
${}^{t+\Delta t}F_N$
Friction delay $\xrightarrow{}$ ${}^{t+\Delta t}F_T = 0$
${}^{t+2\Delta t}\dot{\mathbf{x}}_{1\ R}$unique
${}^{t+\Delta t}\dot{\mathbf{x}}_{1\ R}$non-unique

Fig. 4.4-9: Friction delay feature

4.5 Friction

Advanced Nonlinear Solution has a general Coulomb type friction model, where the coefficient of friction μ can be a constant or calculated based on several pre-defined friction laws. Solution 701 however, only supports standard Coulomb friction.

- In friction, the slip velocity is used to determine whether sticking or slipping conditions are occuring.

The slip velocity at a contactor node is defined as the velocity of the contactor node relative to the contacting point of the target surface, projected onto the contact plane. This definition is opposite to the definition used in ref KJB equation (6.310) (in which the slip velocity is defined as the velocity of the contacting point of the target surface relative to the contactor node, projected onto the contact plane).

The slip velocity is evaluated as the change in slip displacement divided by the change in solution time. In this way the same evaluation is used in static and dynamic analysis.

4.5.1 Basic friction model

By default, a constant coefficient friction is used. It is specified for each contact pair via the BCTSET entry.

4.5.2 Pre-defined friction models (Solution 601 only)

One of the following predefined friction laws can be used instead of constant Coulomb friction. The friction law and its input parameters are set via the BCTPARA entry. The following variables are used in the friction laws: the magnitude of the relative sliding velocity $\dot{u}$, the contact traction T_n, the consistent contact force F_n, the current nodal coordinates **x**, the direction of sliding **v**, and the time *t*. The setting for the FRICMOD parameter required for each friction law is given in parentheses. The A_1 through A_5 constants used in the predefined friction laws are set up via FPARA1 through FPARA5 parameters in BCTPARA.

- Constant coefficient of friction (FRICMOD = 1)

$$\mu = A_1$$

- Different static and dynamic friction coefficients (FRICMOD = 4)

$$\mu = \begin{cases} A_1 & \text{if } \dot{u} \le A_3 \\ A_2 & \text{if } \dot{u} > A_3 \end{cases}$$

- Friction coefficient varying with sliding velocity (FRICMOD = 5)

$$\mu = \begin{cases} A_1 + \dfrac{\dot{u}}{A_2}(A_3 - A_1) & \text{if } \dot{u} < A_2 \\ A_3 & \text{if } \dot{u} > A_2 \end{cases}$$

- Anisotropic friction model (FRICMOD = 6)

$$\mu = \begin{cases} \sqrt{(A_1 \mathbf{v}_{(1)})^2 + (A_2 \mathbf{v}_{(2)})^2 + (A_3 \mathbf{v}_{(3)})^2} & \text{if } \dot{u} > A_5 \\ A_4 & \text{if } \dot{u} \le A_5 \end{cases}$$

where $\mathbf{v}_{(1)}$, $\mathbf{v}_{(2)}$ and $\mathbf{v}_{(3)}$ are the x, y and z components of the sliding direction.

- Friction coefficient varying with consistent contact force (FRICMOD = 7)

$$\mu = A_1 + A_2 F_n,\ 0 \le \mu \le 1$$

- Time varying friction model (FRICMOD = 8)

$$\mu = \begin{cases} A_1 + \dfrac{t}{A_2}(A_3 - A_1) & \text{if } t < A_2 \\ A_3 & \text{if } t > A_2 \end{cases}$$

- Coordinate-dependent friction model (FRICMOD = 9)

$$\mu = \begin{cases} A_1 + A_3\mathbf{x}_{(1)} + A_4\mathbf{x}_{(2)} & \text{in 2D} \\ A_1 + A_3\mathbf{x}_{(1)} + A_4\mathbf{x}_{(2)} + A_5\mathbf{x}_{(3)} & \text{in 3D} \end{cases},\quad 0 \le \mu \le A_2$$

- Friction model 1a (FRICMOD = 2)

$$\mu = \frac{1 - \exp(-A_2 T_n)}{T_n / A_1}$$

- Friction model 1b (FRICMOD = 12)

$$\mu = \frac{1 - \exp(-A_2 F_n)}{F_n / A_1}$$

- Friction model 2a (FRICMOD =3)

$$\mu = A_2 + (A_2 - A_1)\exp(-A_3 T_n)$$

- Friction model 2b (FRICMOD = 13)

$$\mu = A_2 + (A_2 - A_1)\exp(-A_3 F_n)$$

4.5.3 Frictional heat generation

The heat generation resulting from frictional contact can be accounted for in a coupled TMC analysis. The user selects the fractions of the generated heat going into the contactor and target surfaces via the TMCFC and TMCFT parameters in the BCTPARA entry. If these two fractions do not add up to 1.0 the remaining portion is assumed to be lost.

4.6 Contact analysis features

4.6.1 Dynamic contact/impact

For Solution 601

- Oscillations in velocities and accelerations can sometimes be present in implicit dynamic contact analysis especially for high speed impact problems. These oscillations can be reduced by

  - Applying post-impact corrections,
  - Setting the Newmark parameter $\alpha = 0.5$,
  - Adding compliance to the contact surfaces,
  - Using the Bathe time integration method.

- In post-impact corrections, the velocities and accelerations of the contactor and target can be forced to be compatible during contact (only in the normal contact direction). This feature is activated by setting IMPACT = 1 in the NXSTRAT entry. This is achieved by modifying the velocities and accelerations of the contact nodes once convergence is reached such that they satisfy conservation of linear and angular momentum.

The post-impact correction option requires additional memory and computations.

The post-impact correction feature should not be used together with compliant contact surfaces, since the velocities and accelerations of the contactor and target surfaces are no longer expected to be identical.

If post-impact correction is activated, all target nodes, except those with all degrees of freedom fixed or enforced displacements, must have a positive non-zero mass. The contactor nodes can have zero mass.

- Setting the Newmark $\alpha = 0.5$ instead of the default $\alpha = 0.25$ (trapezoidal rule — see Section 6.3) results in an accurate solution of rigid body impact problems, and frequently has a positive effect on reducing numerical oscillations in flexible body contact. This feature can be activated by setting IMPACT = 2 in the NXSTRAT entry, or by changing ALPHA to 0.5 also in the NXSTRAT entry. It is, however, recommended that the Bathe method be used instead, whenever possible.

- Adding compliance to the contact surface can also significantly reduce the numerical oscillations that result from dynamic time integration. This is done by setting a non-zero CFACTOR1 in the BCTPARA entry. In this case, the compliance factor must be selected based on Eq. (4.4-1) such that the contact pressures do not cause excessive penetration. Allowing penetration of the order of 1% of the element size usually eliminates numerical oscillations.

- The Bathe time integration method provides some numerical damping to the high frequency content of the solution, which includes the contact oscillations.

For Solution 701

- Oscillations in velocities and accelerations can sometimes be present in explicit dynamic contact analysis especially for high speed impact problems. These oscillations are more common with the penalty contact algorithm. In that case, they can be reduced by

  ▸ Reducing the normal penalty stiffness,
  ▸ Adding penalty contact damping.

See Section 4.3.2 for details on the explicit penalty contact algorithm.

In addition, other sources of damping such as Rayleigh damping can reduce contact oscillations by damping the high frequency modes that generate them.

- Oscillations in results can also occur when using the kinematic constraint algorithm. These oscillations can be due to a mismatch in the masses of the two contacting surfaces. See Section 4.7.3 for more details.

4.6.2 Contact detection

- As explained earlier in this chapter, the contact conditions prevent the contactor nodes from penetrating the target segments. During each equilibrium iteration, the most current geometry of the contactor and target surfaces is used to determine and eliminate the overlap at the contactor nodes.

- For single-sided contact, the calculation of overlap at a contactor node *k* consists of a contact search, followed by a penetration calculation. The contact search starts by identifying all possible target surfaces where node *k* can come into contact. For each of these target surfaces:

  - ▸ Find the closest target node *n* to node *k*.
  - ▸ Find all the target segments attached to node *n*.
  - ▸ Determine if node *k* is in contact with any of these segments.
  - ▸ If node *k* is in contact, update the information.
  - ▸ If no appropriate target segment is detected, the contact search is expanded beyond the target segments attached to node *n*.

- For double-sided contact, the contact search algorithm uses time tracking and checks whether the contactor node penetrated a target segment between times *t* and $t + \Delta t$.

4.6.3 Suppression of contact oscillations (Solution 601)

- In some problems contactor nodes may oscillate during equilibrium iterations between several (usually two) neighboring target segments. Frequently, both solutions are acceptable. A special procedure can be used to prevent such oscillations. This is done by selecting a non-zero NSUPP parameter in the NXSTRAT entry. In this case, the program records the pairing target segment for each contactor node in the previous NSUPP iterations. Once this array is full, and the contactor node is still in contact, and the pairing target segment is one of those recorded in previous iterations, the suppression feature is activated. The contactor node from this iteration onwards is associated with only that target segment. It may remain in contact with the segment, or in contact with an infinite plane passing through the segment, or it can separate from contact completely. The node is released from its restrictions once iteration ceases, either because convergence is reached, or due to non-convergence.

  The default is not to use oscillation suppression (NSUPP=0).

- If this oscillation suppression feature is used, it is recommended that NSUPP be set greater or equal to 5 and at least 5 less than the maximum number of iterations.

- Note that there is memory overhead associated with this feature, where an integer array of size NSUPP is defined for all contactor nodes.

4.6.4 Restart with contact

- Changes in contact parameters are allowed between restarts, with some exceptions. Some restrictions exist, such as no restart from friction to frictionless and vice versa.

- The contact algorithm itself for a certain contact set can also change in a restart. For this purpose, the contact algorithms can be divided into two categories. The first category includes the constraint function (implicit), Lagrange multiplier segment (implicit), kinematic constraint (explicit), and penalty (explicit). Restarts are possible between different algorithms in this category. The second category includes the implicit and explicit Rigid Target algorithms. Restarts are possible between these algorithms. However, restarts are not allowed between the two categories.

4.6.5 Contact damping

- The contact damping feature allows the user to add normal and tangential grounded viscous dampers to all contactor nodes in the model. The damping is activated via the CTDAMP parameter in NXSTRAT, and the normal and tangential damping coefficients are CTDAMPN and CTDAMPT. Using the same value for both normal and tangential direction results in isotropic viscous damping. The damping force on each node is

$$\mathbf{F}_{Damp} = C_N \dot{\mathbf{u}}_N + C_T \dot{\mathbf{u}}_T$$

- This damping can be useful in static problems for stabilizing the model especially when there are insufficient boundary conditions to remove rigid body modes. It can also be useful in dynamic analysis to dampen out high frequency numerical oscillations. The damping can be set to act only at the initial time step (CTDAMP=1), or to be constant throughout the analysis (CTDAMP=2).

- Using the initial damping option, the damping will be active at the beginning of the first time step, and will be reduced gradually (between iterations) until it fully dies out by the end of the first time step. Thus the final solution at the first time step will be free of any damping. Note that if contact is not established and nothing else stabilizes the model, the program will not converge and will give an appropriate warning message.

- Constant damping remains active throughout the analysis. In this case, the program outputs the sum of all damping forces in the output file, and the user must check that these forces are significantly smaller than the sum of the reaction forces (also written to the output file).

See Section 4.7.6 for modeling hints on using contact damping to handle improperly supported structures and how to choose the damping constants.

4.7 Modeling considerations

4.7.1 Contactor and target selection

- For some contact problems, the contactor and target surfaces in a contact pair can be interchanged without much effect on the solution. However, for many cases, one of the two alternatives is better.

- If it is more important for the nodes of one surface not to penetrate the other, then that surface should be the contactor. This factor is usually important when one surface has a much coarser mesh than the other as shown in Fig. 4.7-1. The coarse surface should preferably be the target in this case. A related condition occurs around corners or edges as shown in Fig. 4.7-2. The upper surface should preferably be the contactor in this case.

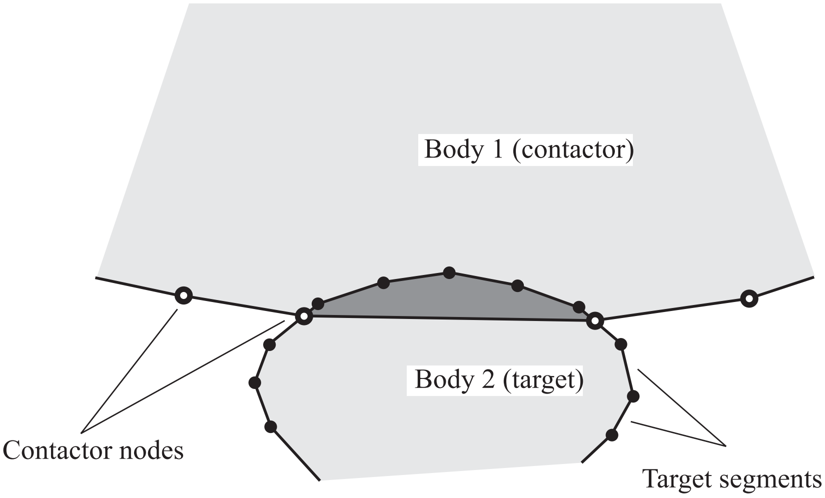


Fig. 4.7-1: Effect of incorrect contactor-target selection due to mesh density

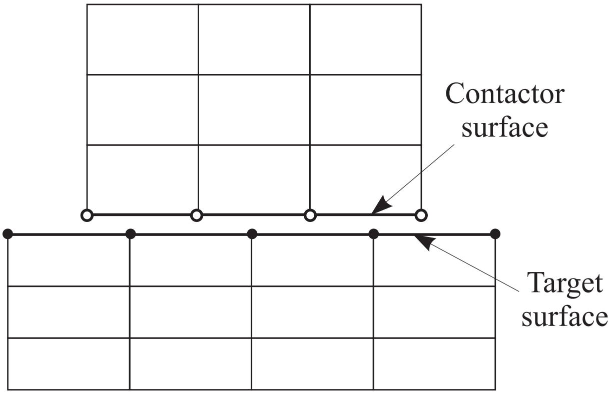


Fig. 4.7-2: Target selection for surfaces of different sizes

- If one of the surfaces has mostly dependent degrees of freedom, it should be the target. This dependency can be due to boundary conditions, constraints or rigid elements. The surface can also be rigid if its nodes are not attached to any elements. In that case too it has to be the target (except in the explicit penalty algorithm where this is permitted).

- If one surface is significantly stiffer than the other, it should preferably be the target, unless one of the two conditions above also exist.

4.7.2 General modeling hints

- Advanced Nonlinear Solution automatically defines the direction of the contact surfaces on the faces of solid elements (defined using the BSURFS or BCPROPS entries in 3-D or the BEDGE or BLSEG entries in 2-D). For target contact surfaces defined on shells (using the BSURF or BCPROP entries) the user has to ensure that the correct direction is defined using the BCRPARA entry (except when double-sided contact is used).

- In some cases, even though the contact surface is on the faces of 3D solid elements, it is more convenient to define the surface using shell elements. In this case, fictitious shell elements should be defined and referenced in the BSURF or BCPROP entries, and TYPE should be set to COATING in BCRPARA. The program will automatically transfer the contact surface to the underlying solid elements and delete the fictitious shell elements.

- Rigid target surfaces can be modeled using nodes with no degrees of freedom or nodes with enforced displacements for all active degrees of freedom. As a result, a fine discretization of a complex rigid surface geometry is possible with only a small increase in the solution cost.

- The commands for 3-D contact surface definition all require the contact surface nodes to be connected with 3-D solid or shell elements. Therefore, to model a rigid target, dummy shell elements should be used to define the surface. These shell elements are removed from the model if they are found to be attached to a rigid contact surface. A contact surface is deemed rigid if:

  - It is the target of a contact pair in a contact set using the rigid target algorithm, or
  - The TYPE flag in the BCRPARA entry is set to RIGID.

- If the contact surface is rigid the MGP parameter in the BCRPARA command can also be used to define a master node that will control the motion of the rigid surface. Internally, rigid links are created between the master node and all the nodes on the rigid target.

- It is acceptable for the nodes on the contactor and target surfaces to be coincident (have identical coordinates). In this case, it is important to ensure that the two surfaces do not share the same nodes.

- In general it is recommended that the lengths of segments on the contactor and target surfaces be approximately equal. This is particularly important if multiple contact surface pairs are considered in the analysis or if the contact surface geometries are complex.

- If required, a contactor surface can be modeled as almost rigid by choosing a reasonably high Young's modulus for the finite elements modeling the contactor surface. However, the stiffness of the surface elements should not be excessively high and make the model ill-conditioned.

- If the degrees of freedom of a node on a contactor surface are used in constraint equations or attached to a rigid element (see Section 5.8), the contactor node degrees of freedom should preferably be independent.

- If the contact surfaces are smooth (i.e., the coefficient of friction is small), the frictionless model is recommended as it is less costly to use. It is also recommended that prior to any contact analysis involving friction, the frictionless solution is first obtained, whenever possible.

- It is not recommended that contact pairs with friction coexist with contact pairs without friction in the same contact set.

- A contactor node should preferably not belong to more than one contact surface in a contact set, otherwise the contactor node may be over-constrained. If it is necessary for a contactor node to belong to more than one contact surface, then contact compliance should be used.

- For problems in which the contactor and target surfaces are initially relatively close to each other and no significant sliding between these surfaces is expected throughout the analysis, the small displacement contact feature may be used. The analysis will be faster in this case, since the relatively time consuming contact search is only performed once, and convergence difficulties associated with a contactor node oscillating between one target to another are eliminated. It is the user's responsibility however, to make sure that the problem is suitable for small displacement contact.

- The friction delay feature can sometimes lead to better convergence since friction will only act once a converged contact solution is established. This feature is also very useful for many problems involving initial penetrations. In this case, the first time step during which these initial penetrations are removed will be frictionless.

- Restarting from frictionless contact to contact with friction and vice versa is not possible. However, it can be done if the frictionless analysis is replaced by a frictional analysis with a very small friction coefficient.

- Ignoring initial penetrations is a useful option when these penetrations are just a product of the finite element discretization, meaning that they do not exist in the physical model. Fig. 4.7-3 illustrates one such case involving contact between concentric cylinders. In this situation, if initial penetrations are eliminated, the contact algorithm will try to push the penetrating contactor nodes to the target surface segments in the first step, creating initial prestressing. These initial penetrations and any prestressing that they might cause are unrealistic. Ignoring them is useful in this case. Note however, that if either cylinder is significantly rotated the initial penetrations calculated at each contactor node (in the initial configuration) will no longer be valid. In this case, the best alternative would be to use a much finer mesh.

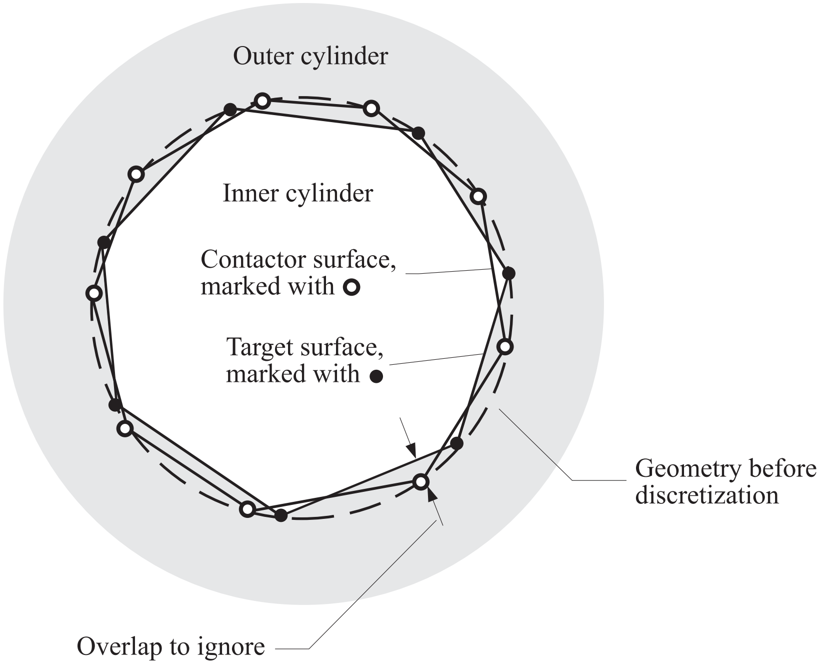

Outer cylinder
Inner cylinder
Contactor surface, marked with ○
Target surface, marked with ●
Geometry before discretization
Overlap to ignore

Fig. 4.7-3: Analysis of contact between concentric cylinders, initial penetration is ignored

- When higher order elements are used in contact, specifically the 10-node tetrahedral and the 20-node brick elements, tensile consistent nodal contact forces can develop even when the underlying contact tractions are compressive. The program can accept such tensile forces as if they are compressive. This is done via the TNSLCF flag in NXSTRAT. Accepting these tensile forces gives more uniform results for problems involving the above mentioned elements. However, it may slow down or even prevent convergence in other problems. It is off by default.

- The option NXSTRAT DIAGSOL=2 outputs much useful information. In Solution 601, this information is output every equilibrium iteration. This information includes the number of nodes going into and out of contact, the number of nodes transisioning between stick and slip (in frictional contact), the number of nodes exceeding the boundaries of target surfaces, etc. Because this information is so useful, it is recommended that

NXSTRAT DIAGSOL=2 be used when there are convergence difficulties (not the default).

4.7.3 Modeling hints specific to Solution 601

- It is recommended that the ATS method be used in contact analysis (see Section 6.2.4). It can also be effective to use the low-speed dynamics option of the ATS method (ATSLOWS parameter of the NXSTRAT entry).

- Line search can sometimes be beneficial for contact problems.

- Frictional contact problems using the constraint function algorithm can be sensitive to the choice of frictional regularization constant (EPST parameter in BCTPARA entry). For most problems, this parameter should be one or two orders of magnitude smaller than the expected sliding velocity. Using an excessively large value leads to a smoother friction law, which generally converges faster but results in smaller frictional forces or more sliding. Using an excessively small value enforces the Coulomb law more accurately but is more likely to experience convergence difficulties.

- Friction is not regularized or smoothed in the Lagrange multiplier segment algorithm. This results in accurate enforcement of stick and slip, but is more likely to experience convergence difficulties.

- Geometric and material nonlinearities can highly depend on the sequence of load application. Thus, for problems involving such features, the load steps should be small. The time step Δt should also be small in dynamic analysis and when time dependent material constitutive relations (e.g., creep) are used.

- If rigid elements are connected to contact surface nodes, the flexible option can be used. In this case, the rigid element does not create any dependent degrees of freedom. This feature is activated via the EQRBAR and EQRBE2 flags in the NXSTRAT entry.

- If a contact surface with corners or edges is modeled with continuous contact normals, the normal vectors may be inaccurate as shown in Fig. 4.7-4(a). In this case, switch to discontinuous normals, use different contact surfaces for each smooth part (Fig. 4.7-4(b)) or use a fine mesh close to the corners or edges (Fig. 4.7-4(c)).

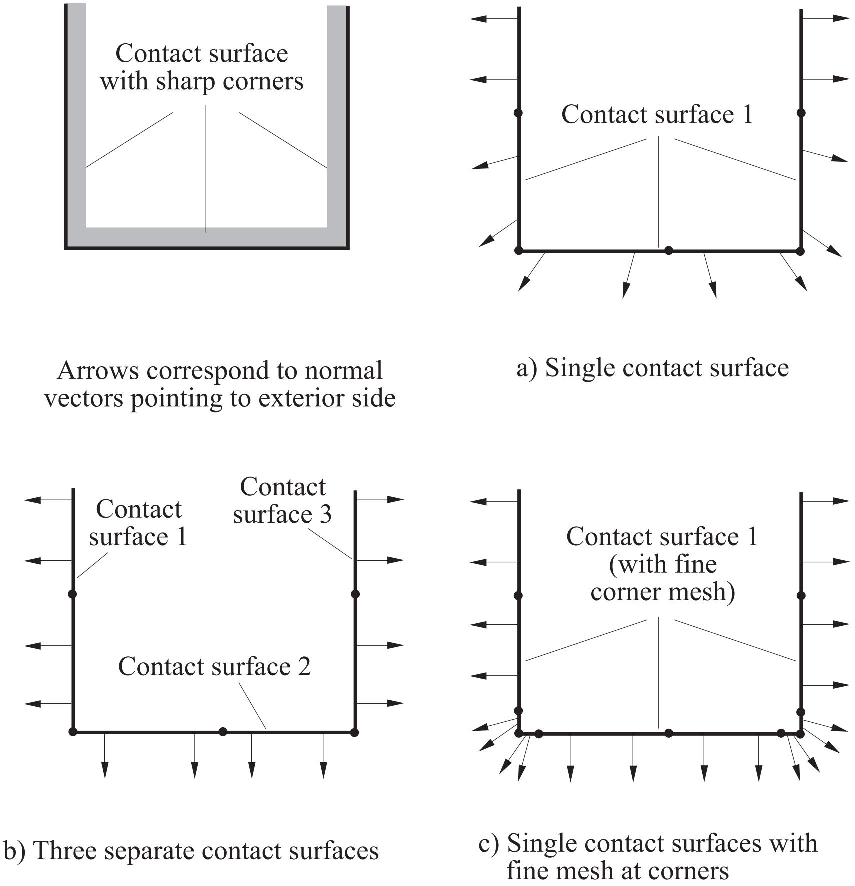

Contact surface with sharp corners
Contact surface 1
Arrows correspond to normal vectors pointing to exterior side
a) Single contact surface
Contact surface 1
Contact surface 3
Contact surface 2
Contact surface 1 (with fine corner mesh)
b) Three separate contact surfaces
c) Single contact surfaces with fine mesh at corners

Fig. 4.7-4: Defining contact surfaces (with continuous normal vectors) in the presence of corners

4.7.4 Modeling hints specific to Solution 701

- The penalty algorithm is preferred when both surfaces are rigid or have many fixed or prescribed nodes.

- Large oscillations in the contact forces may occur when using the penalty method even though the model is stable. These can be reduced by adding a penalty damping term and/or reducing the penalty stiffness.

- When using the penalty contact algorithm it is important to check that the contact stiffnesses are properly selected. Unduly small penalty stiffnesses will lead to excessive penetrations, while unduly large penalty stiffnesses will lead to excessive oscillations or unstable time integration.

- Large mismatches between the masses of contacting surfaces should be avoided when using the kinematic constraint method. This mismatch is common when contact involves a rigid surface with a small mass and an applied force, as shown in Fig. 4.7-5. The best solution in such cases is to minimize the mismatch by increasing the mass of the rigid surface.

  The inaccuracy in this case results from the way the contact is enforced. The kinematic constraint method first predicts displacements without contact then applies a contact correction. The contact conditions are satisfied more accurately when the penetrations in the predicted configuration are small which is usually the case due to the small time step size of explicit analysis. However, some cases such as that mentioned above lead to large projected penetrations which results in incorrect contact conditions and tensile contact forces.

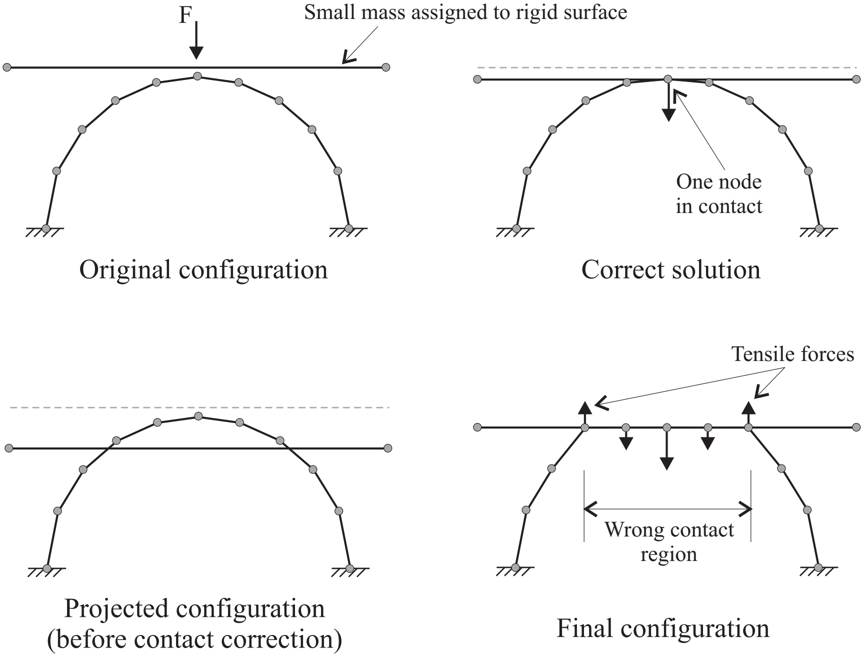


Fig. 4.7-5: Performance of kinematic contact algorithm when contact surfaces have a large mass mismatch

- Large mismatches between the masses of contacting surfaces can also lead to problems when using the penalty method. In this case, the normal penalty stiffness required to avoid instability (without reducing the time step) can be unduly small leading to excessive penetrations. The best solution in such cases is to minimize the mismatch by increasing the mass of the rigid surface, or increase the penalty stiffness by setting it manually or by reducing the time step.

4.7.5 Convergence considerations (Solution 601 only)

- When Solution 601 fails to converge during the incremental analysis, the intermediate printout given by Solution 601 in the output listing can provide some useful information (see Fig. 4.7-6).

```
         OUT-OF-        NORM OF                                            ...
         BALANCE     OUT-OF-BALANCE        NORM OF INCREMENTAL             ...
         ENERGY      FORCE    MOMENT      DISP.      ROTN.     CFORCE      ...
                  NODE-DOF  NODE-DOF   NODE-DOF   NODE-DOF    CFNORM       ...
                 MAX VALUE MAX VALUE  MAX VALUE  MAX VALUE
                   box b     box c      box d      box e      box f


ITE=  0  1.14E+00  1.41E+02  9.99E-17  5.35E-02  5.12E-02  1.27E-15 ...
                       36-Z      35-X      31-Z      31-X  0.00E+00 ...
                -1.00E+02  4.71E-17 -5.68E-03 -3.30E-02

ITE=  1 -1.29E-03  2.56E+01  1.92E-04  1.56E-02  2.45E-01  2.65E+03 ...
                      121-Z      31-X      64-Z      34-X  1.27E-15 ...
                -9.85E+00 -1.06E-04  5.07E-03  1.26E-01

ITE=  2  3.32E-04  2.51E+01  1.88E-04  1.80E-02  1.77E-01  1.95E+03 ...
                      117-Z      31-X      64-Z      32-X  5.08E+01 ...
                -9.66E+00 -1.04E-04  4.97E-03 -9.02E-02

ITE=  3  7.69E-02  4.46E+02  8.18E-04  1.04E-03  1.17E-02  1.95E+03 ...
                       64-Z      34-X     120-Z      33-X  2.00E+03 ...
                 3.21E+02  5.15E-04 -1.33E-04 -7.92E-03
```

```
... CONVERGENCE RATIOS  CONVERGENCE RATIOS   OUT-OF-BALANCE LOAD
... FOR OUT-OF-BALANCE    FOR INCREMENTAL    VECTOR CALCULATION
...    ENERGY      FORCE    DISP.     CFORCE      BETA      RATIO
...                MOMENT   ROTN.                (ITERNS)

      COMPARE WITH        COMPARE WITH
     ETOL       RTOL      DTOL       RCTOL
    1.00E-03  1.00E-02 (NOT USED) 5.00E-02

...    1.00E+00  1.41E+01  0.00E+00  1.27E-05
...              9.99E-17  0.00E+00


...   -9.69E-03  2.56E+00  0.00E+00  2.65E+05  1.00E+00 -5.54E-02
...              1.92E-04  0.00E+00               ( 1)


...    2.49E-03  2.51E+00  0.00E+00  3.85E+01  1.92E-02  5.08E-03
...              1.88E-04  0.00E+00               ( 9)


...    5.76E-01  4.46E+01  0.00E+00  9.77E-01  1.00E+00  3.94E+03
...              8.18E-04  0.00E+00               ( 2)
```

Fig. 4.7-6: Solution 601 output listing of convergence criteria during equilibrium iterations

- Three non-contact related norms are given: first, the energy convergence criterion, the displacement and rotation convergence criterion (boxes d and e), and the force and moment convergence criterion (boxes b and c). Each box has 3 lines of output with the top one giving the norm of the quantity, the second one giving the node number corresponding to the maximum value, and the third line giving the maximum value. See Chapter 6 for definitions and more details on these norms.

- Box f of Fig. 4.7-6 shows the contact related norms. Parameter CFORCE indicates the norm of the change in the contact forces (between two iterations), and parameter CFNORM gives the norm of the contact force vector.

- The following additional convergence criterion is used when contact is present:

$$\frac{\text{CFORCE}}{\max(\text{CFNORM},\text{RCONSM})} \le \text{RCTOL}$$

where RCTOL is the contact force convergence tolerance and RCONSM is the reference contact force to prevent possible division by zero in the contact convergence criterion above. RCONSM and RCTOL are set in the NXSTRAT entry.

- When the maximum number of iterations is reached without convergence, and all norms are decreasing, the maximum number of iterations should be increased.

- When the norms are rapidly changing before convergence fails, it is commonly caused by applying the load too quickly or using a large time step.

- When CFNORM is stable but CFORCE changes rapidly during equilibrium iterations, the contact can be oscillating between 2 or more close solutions. In this case, try to change the time stepping, or turn on the suppression of contact oscillations feature. When CFNORM varies rapidly, usually the other three norms also vary.

- The additional output from NXSTRAT DIAGSOL=2 can also assist in determining the reasons for convergence difficulties.

4.7.6 Handling improperly supported bodies

Many static problems depend on contact to provide the boundary conditions necessary for a stable problem (one in which there are no rigid body modes). Some examples are shown in Fig. 4.7-7.

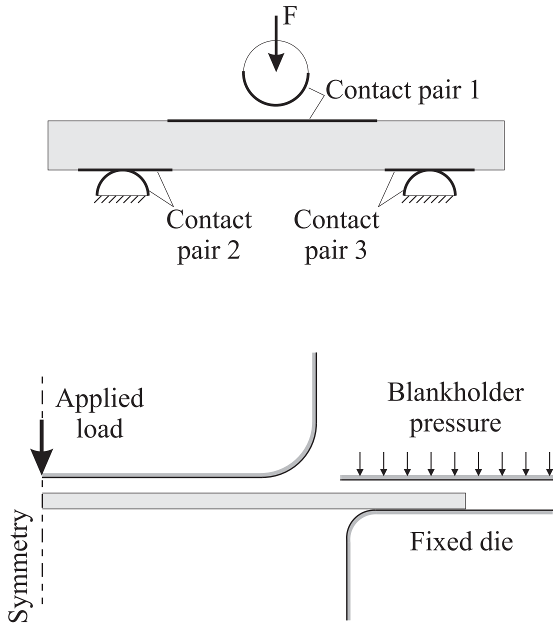


Fig. 4.7-7: Examples of improperly supported bodies

In such cases, the stiffness matrix is singular if the contact constraints are inactive. Even if the constraints are active the stiffness matrix is still not positive definite. The problem is more serious if natural boundary conditions are applied (forces, moments, or pressure). Weak springs can be added by the user to make the model stable. However, the selection of appropriate locations and stiffnesses for such springs may not be feasible.

Some models may be better suited for a dynamic analysis or the low-speed dynamics feature. However, in many cases, this too is not a feasible option. Therefore, several other modeling techniques are available in Solution 601 to handle such problems. These are stiffness stabilization, contact damping and limiting incremental displacements. These techniques can be used separately or combined in the same model.

Stiffness stabilization (see Section 10.6 for details).
This feature provides a stabilizing effect by scaling all diagonal stiffness terms without affecting the right-hand-side load vector. The outcome of each iteration will be affected, but the final

converged solution will not be (within the bounds of the convergence tolerances). Since the stabilization constant in non-dimensional, it should always be a small number. Typical values are between 10^{-12} and 10^{-9}.

Contact damping (see Section 4.6.5 for details).
Contact damping adds grounded viscous dampers to all contactor nodes. Setting the damping to be only at the initial time step is sufficient for some problems such as those in Fig. 4.7-7. When the first time step converges contact must be established and damping will have been removed. This way, the converged solution will be free of any contact damping. Other problems however, require the damping to be constantly present. In this case, the program outputs the damping forces at every time step. These forces should be compared with the reactions in order to ensure that damping is not excessive.

The damping constants have units of force per unit velocity. Hence, their proper value is problem dependent. If initial contact damping is used to stabilize a problem involving two contact bodies at least one of which is unsupported, and with a gap between them, then a good estimate of the damping constants C_N and C_T is one in which the gap is nearly closed in the first iteration. Starting with the dynamic equations of motion (see Equation 6.3-1) and canceling out the inertial term (static analysis), and the stiffness term (since one or both bodies are initially unsupported), we obtain

$$\mathbf{C}\ \dot{\mathbf{U}} = \mathbf{R}$$

where **C** is the total damping matrix, which in this case is diagonal, and $\mathbf{R}$ is the applied load vector. We can assume the normal and tangential damping constants to be equal, the total damping contribution to be the damping constant times the number of contact nodes on the unsupported contact surface N (the top circular body in the example in Fig. 4.7-7), and the velocity to be approximately equal to the minimum initial gap between the two bodies, g, divided by the time step size Δt. This leads to the following value of the damping constants

$$C_N = C_T = \frac{R\,\Delta t}{N g}$$

where R is the sum of the applied loads at the first time step.

Note that this is only an estimate, but is frequently an acceptable one.

Limiting maximum incremental displacement (see Section 6.2.1 for details).

Limiting the maximum incremental displacement per iteration is useful when a load is applied to a body that is not initially in contact. The model at that stage is unstable and even when stiffness stabilization or viscous damping is used, the initial displacement can be excessive leading the program away from the converged solution, and thus making the return to the proper solution difficult. Setting the limiting displacement to about the element length size in this case would scale down the potentially huge displacement in the first iteration so that the results remain close to the converged solution.

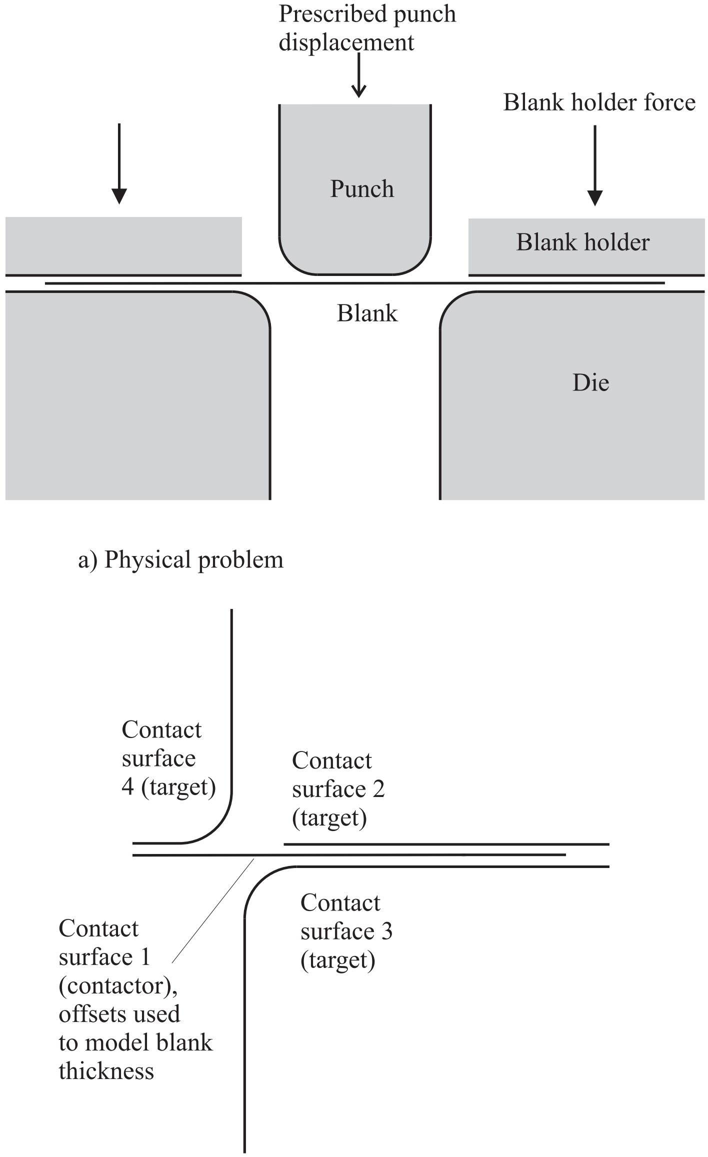
Note that this feature does not stabilize the stiffness matrix, so in many cases it may be necessary to use it together with stiffness stabilization or viscous contact damping or both.

4.8 Rigid target contact algorithm

4.8.1 Introduction

- The rigid target contact algorithm is intended for use in applications in which the target surfaces are considered to be rigid. It is only available for 3-D contact.

Fig. 4.8-1 shows a typical application in metal forming.


Prescribed punch displacement
Blank holder force
Punch
Blank holder
Blank
Die
Contact surface 4 (target)
Contact surface 2 (target)
Contact surface 3 (target)
Contact surface 1 (contactor), offsets used to model blank thickness

a) Physical problem

b) Modeling with contact surfaces

Fig. 4.8-1: Sample metal forming analysis using the rigid-target contact algorithm

- A target surface can either be stationary, can translate as a rigid body or can rotate as a rigid body.

- Contact can be frictionless or can include Coulomb friction.

- The rigid target contact algorithm is completely revised in Advanced Nonlinear Solution of NX 5. However the rigid target contact algorithm in Advanced Nonlinear Solution of NX 4 is retained in Advanced Nonlinear Solution for backwards compatibility. The revised rigid target contact algorithm of NX 5 is the default.

  Throughout this section, the rigid target contact algorithm in Advanced Nonlinear Solution of NX 4 is referred to as the “NX4” rigid target contact algorithm. This section does not describe the NX4 rigid target contact algorithm; for information on the NX4 rigid target contact algorithm, see the NX Nastran 4 Advanced Nonlinear Theory and Modeling Guide.

  Models that were set up using the NX4 rigid target contact algorithm may need to be revised when using the current rigid target contact algorithm. We suggest that new models not be set up using the NX4 rigid target contact algorithm.

- It is also possible to solve many problems involving rigid targets using the constraint function and segment contact algorithms described earlier in this chapter. However, the rigid target contact algorithm described here is frequently more effective, because the rigid target contact algorithm uses the assumption of rigid targets to simplify the contact searching.

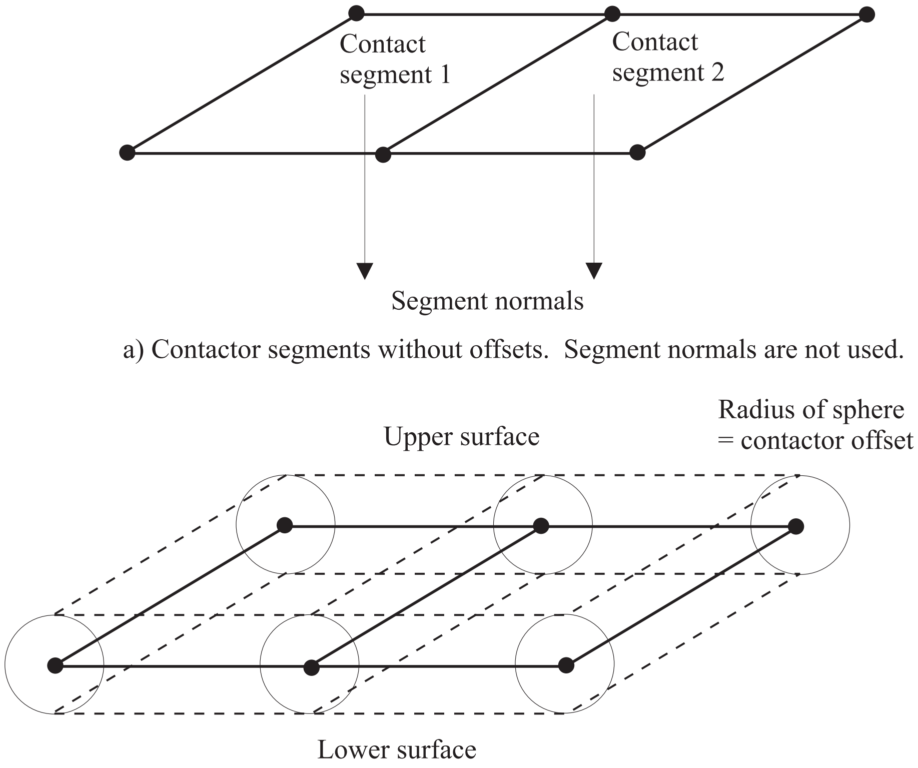
4.8.2 Basic concepts

4.8.2.1 Contactor surfaces

Similar to the other contact algorithms in Advanced Nonlinear Solution, the contact surfaces are organized into contact sets. Each contact surface consists of 3- or 4-node contact segments. A contact pair consists of a contactor surface and a target surface. In the rigid target algorithm, it is allowed for a contactor surface to be in contact with more than one target surface simultaneously.

Contactor surface: The contactor surface definition includes the possibility of offsets.

When there are no offsets specified, the contactor surface is described entirely by the contactor nodes. (Fig. 4.8-2(a)).

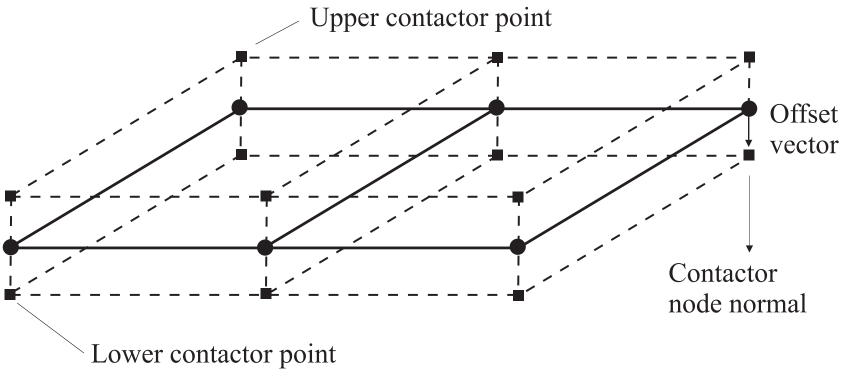


a) Contactor segments without offsets. Segment normals are not used.

b) Contactor segments with spherical offsets. Segment normals are not used.

Fig. 4.8-2: Contactor segments

When there are offsets specified, the offsets can either be described using spheres centered around the contactor nodes (Fig. 4.8-2(b)), or using the contactor normals (Fig. 4.8-2(c)). In either case, the offset magnitude is either constant or taken from the current thickness of attached shell elements, as described below.



Contactor node normal = average of all segment normals

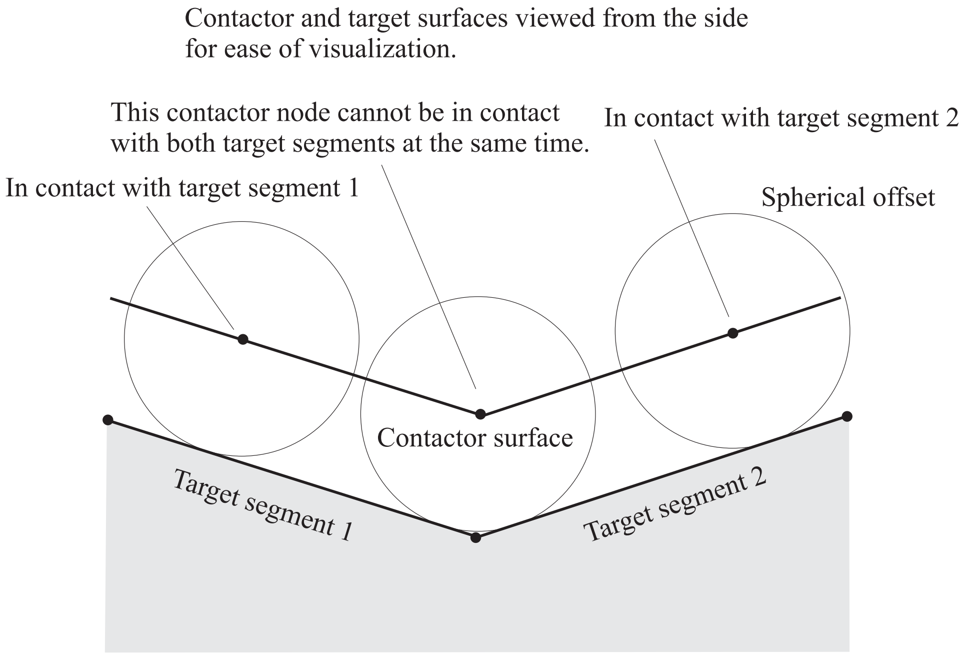
Offset vector = contactor node normal × offset magnitude

c) Contactor segments with offsets, normals used to describe offsets

Fig. 4.8-2: Contactor segments (continued)

When the offsets are described using the contactor normals, offset vectors are constructed using the averaged contactor normals and the offset magnitude. The upper and lower contactor points are constructed from the contactor nodes and the offset vectors.

When the target surface is concave, it is possible for the contact situation to be similar to the one shown in Fig. 4.8-3. In this case, when the offsets are described using spheres, the center contactor node cannot be in contact with both target segments at once, hence the center contactor node will oscillate between them. The center contactor node cannot be in contact with target edge 1 since edge 1 is farther away than either of the target segments. Equilibrium iterations in static and implicit dynamic analysis will not converge, because of the oscillation. However contact is correctly modeled when the offsets are described using normals, because the center contactor node can be in contact with target edge 1.


Contactor and target surfaces viewed from the side for ease of visualization.
This contactor node cannot be in contact with both target segments at the same time.
In contact with target segment 2
In contact with target segment 1
Spherical offset
Contactor surface
Target segment 1
Target segment 2

a) Spherical offsets, contact incorrectly modeled

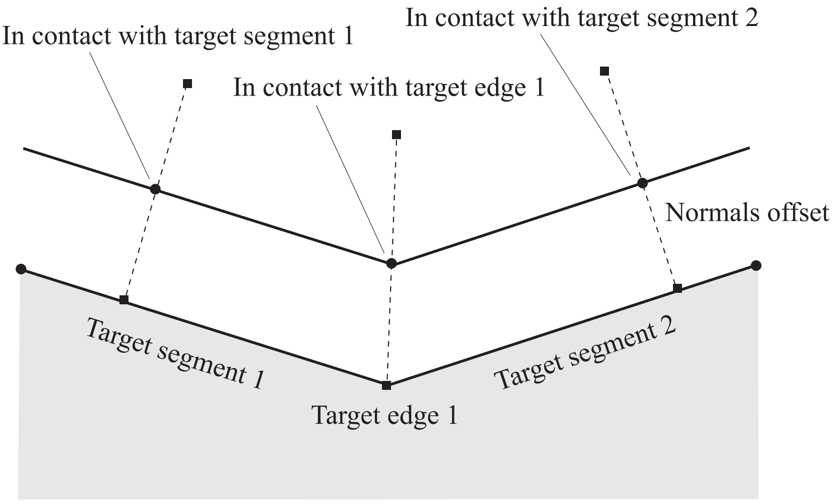
Fig. 4.8-3: Concave target surface, contactor surface with offsets

4.8.2.2 Target surfaces

Each target surface is either stationary, or can rigidly move (translate, rotate or a combination of translations and rotations).

4.8.2.3 Determination of contact between contactor and target

No contactor offsets: It is allowed for a contactor node to be in contact with a target segment, target edge or target node. The program searches for the target segment, edge or node for which the absolute value of the distance *d* between the contactor node and the target segment, edge or node is minimized, where the distance is measured in the direction opposite to the target normal (Fig. 4.8-4). A positive distance corresponds to a geometric gap; a negative distance corresponds to geometric overlap.


In contact with target segment 1
In contact with target segment 2
In contact with target edge 1
Normals offset
Target segment 1
Target segment 2
Target edge 1

b) Normals offsets, contact correctly modeled

Fig. 4.8-3: (continued)

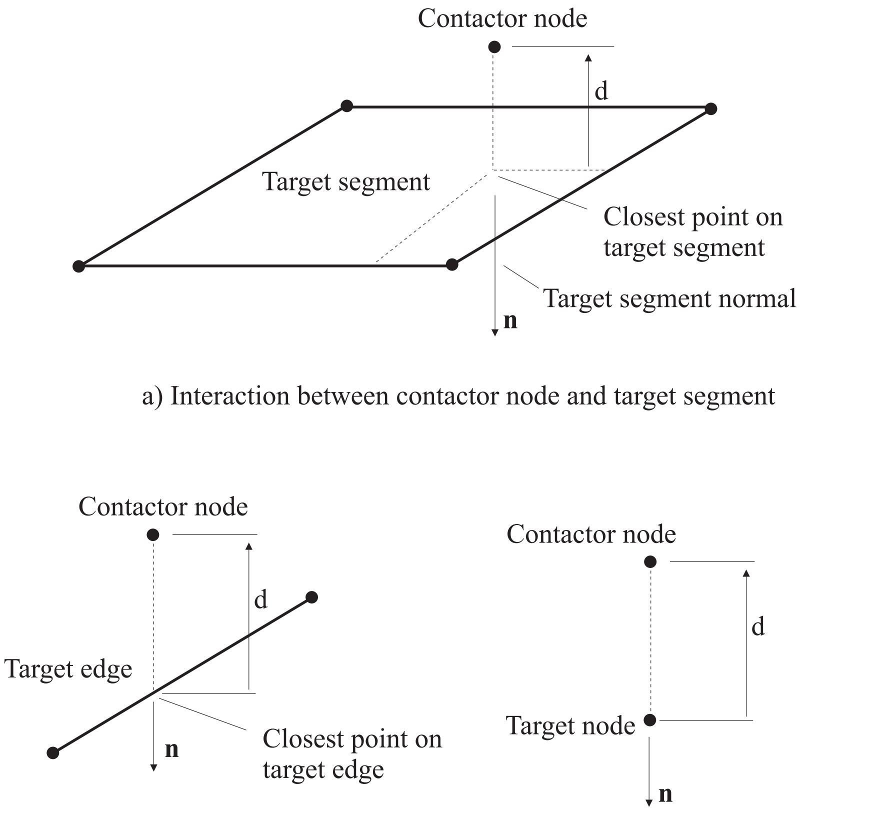
Notice that, for interaction between a contactor node and target edge, or between a contactor node and target node, the normal direction is taken from the line segment connecting the contactor node and the target, as shown in Fig. 4.8-4.

Fig 4.8-5 shows two target segments with a common target edge. The shaded volumes indicate which of the target entities any contactor node is closest to. Notice that the shaded volume in which the contactor node is closest to the target edge depends upon the angles between the target segments attached to the edge.

Once the target segment, edge or node is determined, then the contact gap is computed using

$$g = d - \text{GAPBIAS}$$

where GAPBIAS can be chosen to, for example, not model contact even if there is geometric overlap. (The default for GAPBIAS is 0.)



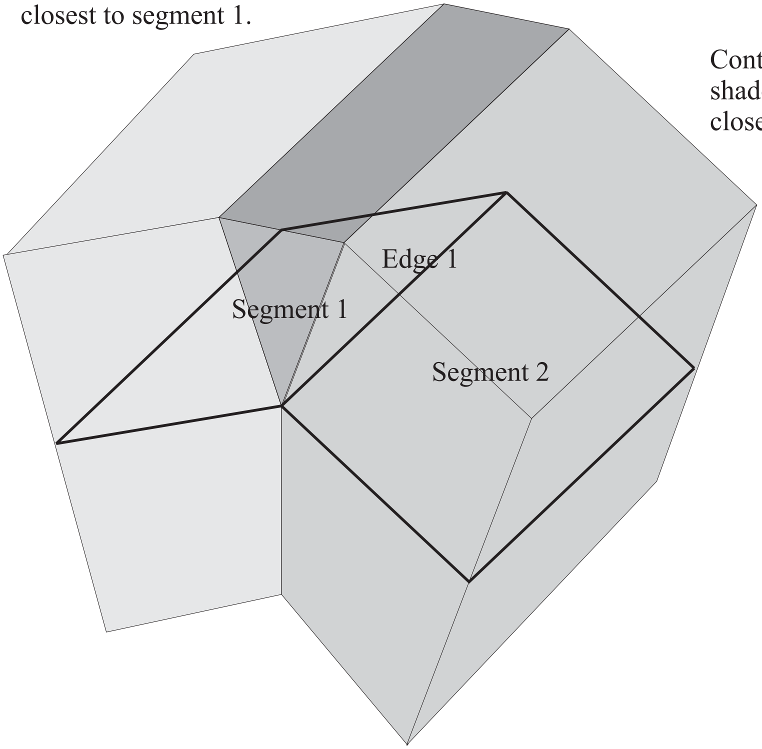
a) Interaction between contactor node and target segment

b) Interaction between contactor node and target edge

c) Interaction between contactor node and target node

Fig. 4.8-4: Interaction between contactor node and target surface

If the corresponding gap is negative, and less than DEPTH, then contact occurs, in other words $-\text{DEPTH} \le g \le 0$ is the contact condition. DEPTH can be chosen to limit the depth of the target surface, exactly as in the other contact algorithms.



Contactor node in this shaded volume is closest to segment 1.

Contactor node in this shaded volume is closest to edge 1.

Contactor node in this shaded volume is closest to segment 2.

Fig. 4.8-5: Interaction of contactor node with target segments and edges

Contactor offsets described using spheres: In this case, the distance *d* is determined exactly as if there are no offsets. Then the contact gap is computed using

$$g = d - \mathrm{OFFSET} - \mathrm{GAPBIAS}$$

where OFFSET is the offset magnitude. The process is illustrated for interaction between a contactor node and target segment, assuming that GAPBIAS = 0 (Fig. 4.8-6). The same idea is used for interaction between a contactor node and target edge or node.

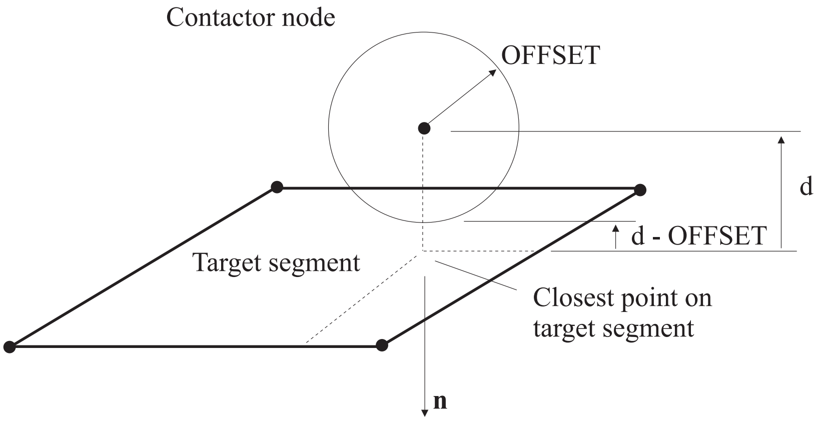


Fig. 4.8-6: Interaction between contactor node and target segment, spherical offsets

Contactor offsets described using normals: In this case, contact is detected using the upper and lower contactor points instead of the contactor nodes.

Oscillation checking: The search for the nearest target segment, edge or node is performed every equilibrium iteration in Solution 601. During the equilibrium iterations, it is possible for the contactor node to move in such a way as to be alternately in contact with two neighboring segments. This is especially true if the target surface is concave. When the contactor node oscillates between two neighboring segments, the solution cannot converge unless oscillation checking is turned on. When oscillation checking is turned on, then, when oscillation is detected between two neighboring segments, the contactor node is placed into contact with the shared target edge. In many cases, this procedure allows the iterations to converge, if in fact the contactor node “should” have been in contact with the shared target edge.

Oscillation checking only forces the contact between the contactor node and shared target edge for the current equilibrium iteration. For the successive equilibrium iterations, the contactor node is always in contact with the nearest target segment, edge or node. So oscillation checking cannot force contact to the “wrong” target segment, edge or node in a converged solution.

Contact normal force: The normal force corresponding to contact is computed as $F_n = -k_n g$ where the normal force acts in the direction opposite to the target normal direction (Fig. 4.8-7). k_n is the contact normal stiffness, entered as a parameter (see Section 4.8.3 for hints about choosing k_n). k_n can be considered to be a penalty parameter.

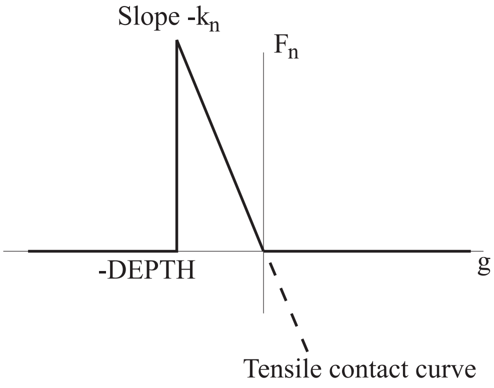


Fig. 4.8-7: Normal contact force vs. gap

Tensile contact: During equilibrium iterations in Solution 601, a node can temporarily be in “tensile contact”. The basic ideas for tensile contact are illustrated in Figs. 4.8-8 to 4.8-10.

Fig. 4.8-8 shows the iteration history when tensile contact is not used. For iteration ite-2, the contactor node and target segment overlap. Hence contact is assumed between the contactor node and target segment. For iteration ite-1, because of the relative motion of the contactor node and target segment, the contactor node and target segment do not overlap. For this iteration, no contact is assumed between the contactor node and target segment. For iteration ite, there is a large overlap because the contactor spring unloads, since there are no forces acting on the contactor spring, and the target does not provide any stiffness to the contactor node. This large overlap causes large contact forces, which can cause trouble in convergence in the successive iterations.

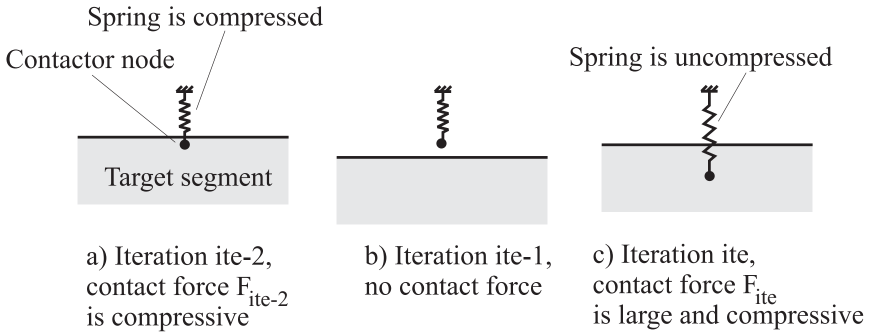


Fig. 4.8-8: Iterations when tensile contact is not used

Fig. 4.8-9 shows the iteration history when tensile contact is used. Now, in iteration ite-1, tensile contact is assumed between the contactor node and target segment. In tensile contact; the target surface still provides stiffness to the contactor node. Hence the overlap in iteration ite is small.

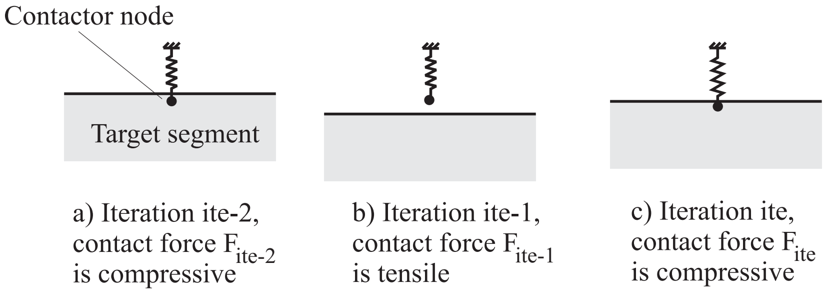
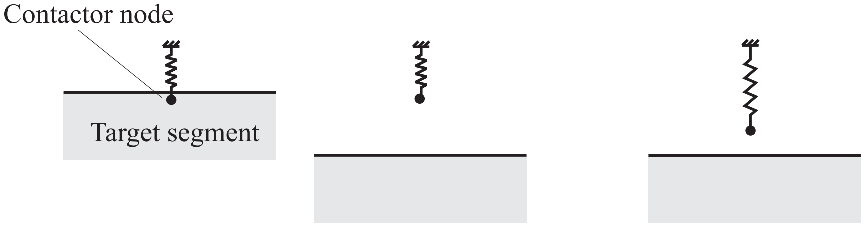


Fig. 4.8-9: Iterations when tensile contact is used

Fig.4.8-10 shows the iteration history when tensile contact is used, and the gap is large. In iteration ite-1, tensile contact is assumed, and in iteration ite, no contact is assumed.

It is seen that tensile contact speeds up the convergence when the converged solution is in contact, and slows down the convergence when the converged solution is not in contact.



a) Iteration ite-2, contact force $F_{\text{ite-2}}$ is compressive

b) Iteration ite-1, contact force $F_{\text{ite-1}}$ is tensile

c) Iteration ite, no contact force

Fig. 4.8-10: Iterations when tensile contact is used, converged solution not in contact

It is not permitted for a solution in which tensile contact is present to converge, unless the tensile forces are all less than the value of a user-input parameter (see Section 4.8.3). Hence the tensile contact feature does not affect the converged solution.

Tensile contact is always used in Solution 601.

4.8.2.4 Frictional contact

The friction force is calculated using the relative sliding velocity between the target and contactor. The relative sliding velocity $\dot{\mathbf{u}}_f$ is calculated from the velocities of the contactor and target using

$$\dot{\mathbf{u}}_f = \left(\dot{\mathbf{u}}_c - \dot{\mathbf{u}}_t\right) - \left(\left(\dot{\mathbf{u}}_c - \dot{\mathbf{u}}_t\right)\cdot\mathbf{n}\right)\mathbf{n}$$

where $\dot{\mathbf{u}}_c$ is the velocity of the contactor node, $\dot{\mathbf{u}}_t$ is the velocity of the target and $\mathbf{n}$ is the target normal.

In static analysis, the contactor and target velocities are calculated using the nodal incremental displacements divided by the time step. In dynamic analysis, the contactor and target velocities are taken from the nodal velocities.

The friction force magnitude is computed using

$$\begin{aligned} \left|F_f\right| &= \frac{\left\|\dot{\mathbf{u}}_f\right\|}{\dot{u}_{f\min}}\mu F_n, \left\|\dot{\mathbf{u}}_f\right\| < \dot{u}_{f\min} \\ &= \mu F, \left\|\dot{\mathbf{u}}_f\right\| \geq \dot{u}_{f\min} \end{aligned}$$

where F_n is the normal contact force and μ is the Coulomb friction constant (Fig. 4.8-11). $\dot{u}_{f\min}$ is the minimum sliding velocity, entered as a parameter (see Section 4.8.3 for hints about choosing $\dot{u}_{f\min}$). The direction of the friction force is always opposite to $\dot{\mathbf{u}}_f$.

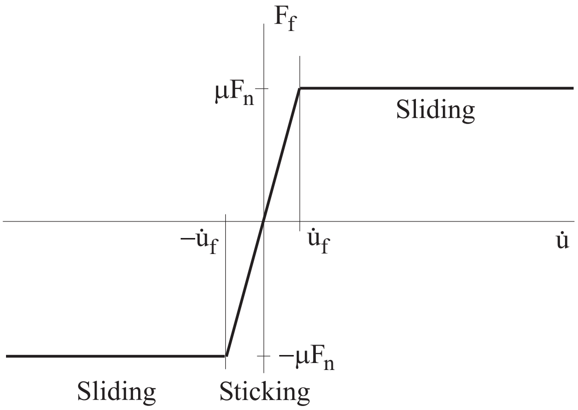


Fig. 4.8-11: Friction force vs. velocity

When $\left\|\dot{\mathbf{u}}_f\right\| < \dot{u}_{f\min}$, the friction is sticking, otherwise the friction is sliding.

Oscillation checking with friction: During equilibrium iterations in Solution 601, it is possible for the contactor node to undergo "sliding reversals". Namely, the contactor node slides in one direction for an equilibrium iteration, then reverses sliding direction for the next equilibrium iteration. When sliding reversals occur, the solution cannot converge unless oscillation checking is turned on. When oscillation checking is turned on, then, when sliding reversals are detected, the contactor node is placed into sticking contact, even if the sticking force is larger than the sliding force, and convergence is prevented for the current equilibrium iteration.

Oscillation checking only forces sticking friction for the current equilibrium iteration. For successive equilibrium iterations, the frictional state (sliding or sticking) is determined as usual from the sliding and sticking forces. So oscillation checking cannot converge to a solution in which the frictional state is wrong.

4.8.3 Modeling considerations

Selection of rigid target contact: For Solution 601, set TYPE=2 on the BCTPARA card. For Solution 701, set XTYPE=3 on the BCTPARA card.

Algorithm used: The current rigid target contact algorithm is selected by default. To select the NX4 rigid target contact algorithm, set RTALG=1 on the NXSTRAT card.

Modeling of target surfaces: If the target surface translates or rotates, all of the nodes on the target surface must be connected to a “master node”, either using constraint equations or using rigid links. For example, in Fig. 4.8-12, all of the nodes on the lower target surface are connected to a master node using rigid links.

It is not allowed for the nodes on a target surface to have independent degrees of freedom. All degrees of freedom for the nodes on a target surface must be fixed or constrained.

Modeling of contactor surfaces: The amount and description of offset is determined by the BCTPARA parameters OFFTYPE, OFFSET and OFFDET. If OFFTYPE=0 (the default), there is no offset. If OFFTYPE=1, a constant offset of value OFFSET is used. If OFFTYPE=2, an offset equal to half of the current shell element thickness is used.

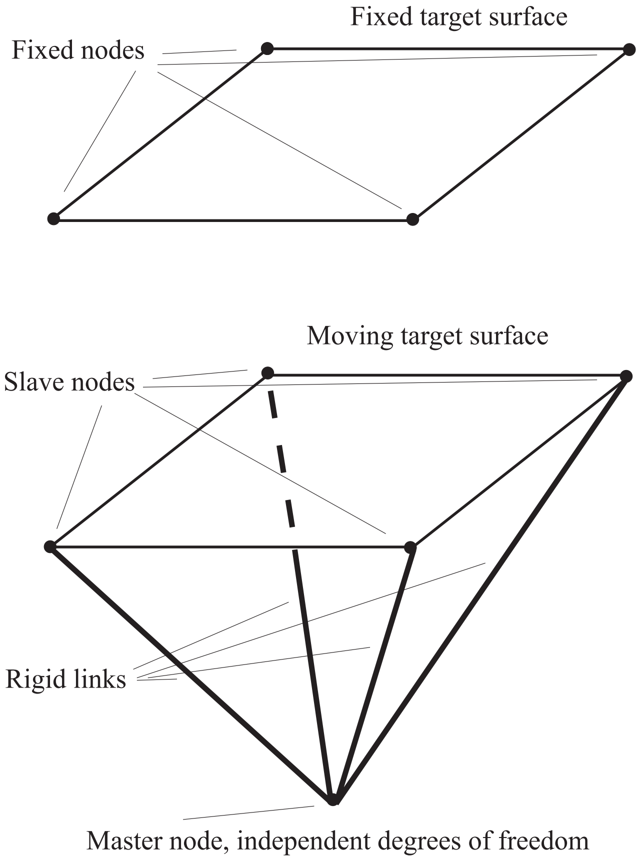


Fig. 4.8-12: Modeling of fixed and moving target surfaces

When there is an offset, then BCTPARA parameter OFFDET determines the description of the offset. If OFFDET=0, then Advanced Nonlinear Solution determines the offset description (either spheres or normals, see Section 4.8.2.1). The criterion used by Advanced Nonlinear Solution is that an offset description of spheres is used for each target surface that is convex or flat, and an offset description of normals is used for each target surface that is concave. If OFFDET=1, then the offset description is spheres, and if OFFDET=2, then the offset description is normals.

When normals are used for the offset description, small steps should be used in Solution 601. This is because the offset vectors are assumed to remain constant during the equilibrium iterations. In particular, at convergence, the offset vectors corresponding to the previous converged solution are used.

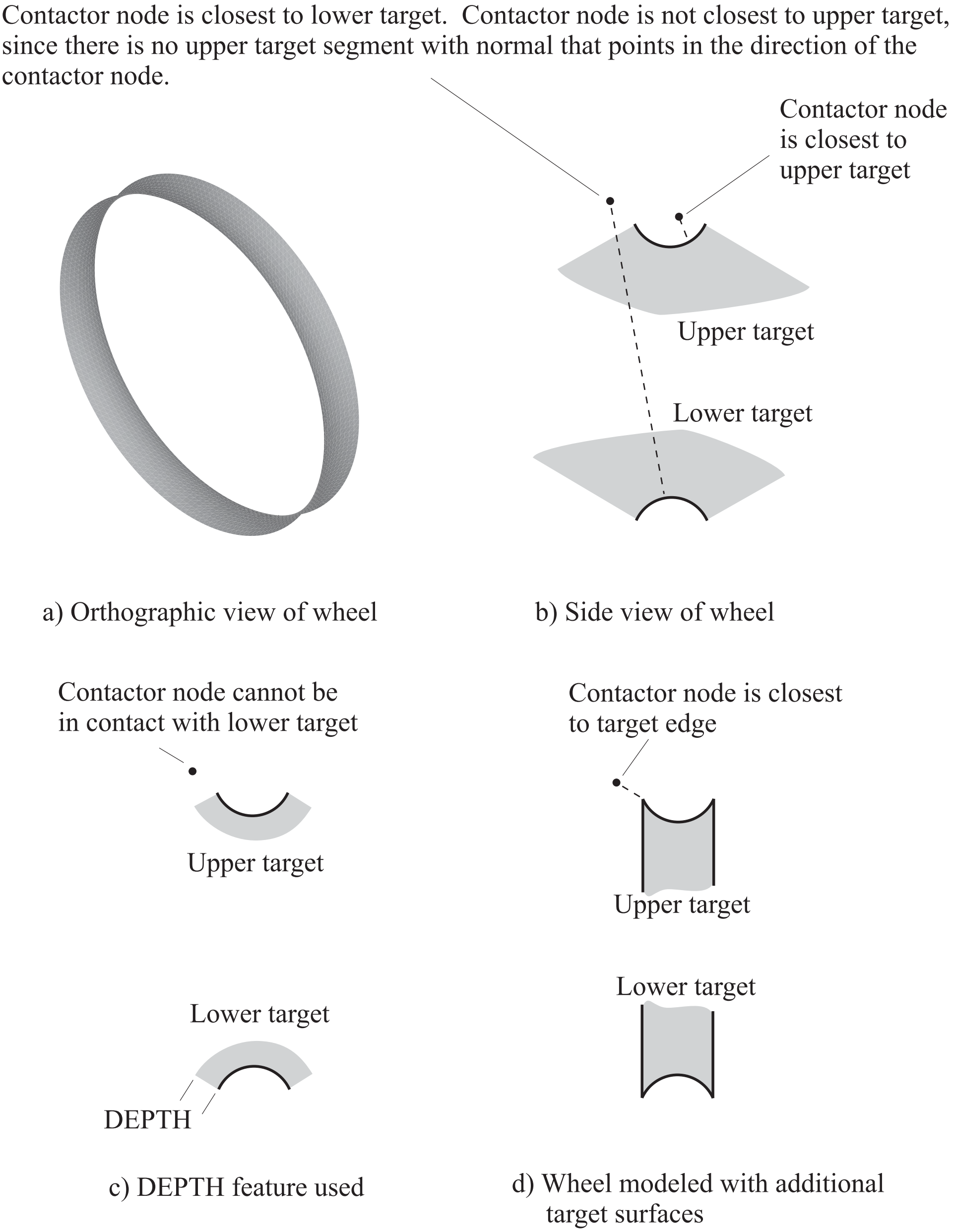
Determination of contact, modeling issues: Contact is affected by variables GAPBIAS and DEPTH, as described in Section 4.8.2.3. GAPBIAS is set using BCTPARA parameter GAPBIAS (default =0) and DEPTH is set using BCTPARA parameter PDEPTH (default=0).

It is possible for the closest target segment, edge or node to not be the expected one. An example is shown in Fig. 4.8-13. In this example, the rim of the wheel is modeled with target segments. Because the distance between a contactor node and a target segment is measured in the direction of the target segment normal, a contactor node interacts with the lower target surface, and the contact algorithm detects a large overlap between this contactor node and the lower target surface.

Another example is shown in Fig. 4.8-14. In Fig. 4.8-14(a), there is a gap between the contactor node and the closest target segment, as expected. In Fig. 4.8-14(b), the punch has moved upward relative to the contactor node. Now there is a large overlap between the contactor node and the closest target segment. This segment is the only segment with a normal that points in the direction of the contactor node.

In both Fig. 4.8-13 and Fig. 4.8-14, the large overlap is unintended. In Solution 601, the equilibrium iterations would most likely not converge. In Solution 701, the large forces between contactor and target would overdistort the elements attached to the contactor node.

One way to avoid the large overlaps is to use the DEPTH feature so that contact is not detected between the contactor node and the incorrect target segment. Another way to avoid this issue is to create additional target segments as shown. Then the contactor node is closest to one of the additional target segments.


Contactor node is closest to lower target. Contactor node is not closest to upper target, since there is no upper target segment with normal that points in the direction of the contactor node.
Contactor node is closest to upper target
Upper target
Lower target
Contactor node cannot be in contact with lower target
Upper target
Lower target
DEPTH
Contactor node is closest to target edge
Upper target
Lower target

a) Orthographic view of wheel

b) Side view of wheel

c) DEPTH feature used

d) Wheel modeled with additional target surfaces

Fig. 4.8-13: Modeling of a wheel

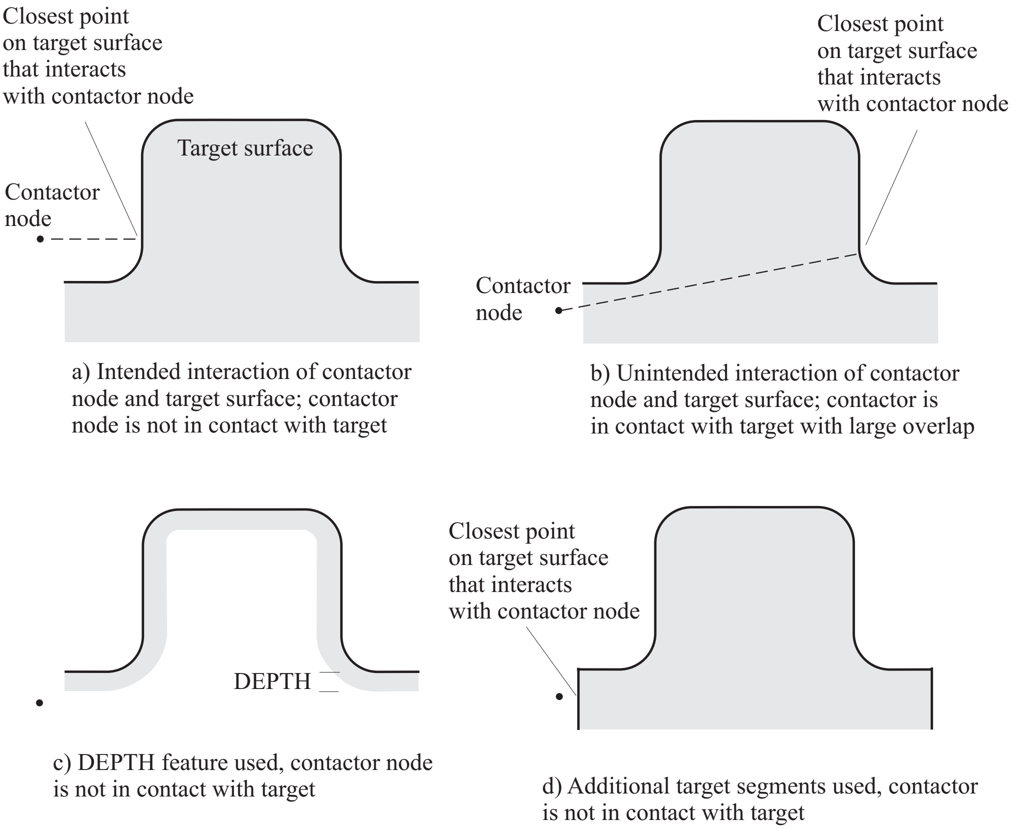

Closest point
on target surface
that interacts
with contactor node
Target surface
Contactor
node
a) Intended interaction of contactor node and target surface; contactor node is not in contact with target
Closest point
on target surface
that interacts
with contactor node
Contactor
node
b) Unintended interaction of contactor node and target surface; contactor is in contact with target with large overlap
DEPTH
c) DEPTH feature used, contactor node is not in contact with target
Closest point
on target surface
that interacts
with contactor node
d) Additional target segments used, contactor is not in contact with target

Fig. 4.8-14: Modeling of a punch

Choice of k_n: k_n is set using BCTPARA parameter NCMOD. The default value of the normal contact stiffness k_n is 1E11. However, k_n can be chosen for optimal convergence. Note that increasing k_n causes the maximum overlap between the contactor and the target to become smaller. Also, increasing k_n can lead to convergence difficulties.

We recommend that the smallest value of k_n be used such that the maximum overlap is still acceptably small. For example, if the target surface is curved, there will be a geometric error associated with using a coarse contractor surface (Fig. 4.8-15). There is no advantage if the maximum overlap is less than the geometric error. So, if the mesh is coarse, a large maximum overlap can be used.

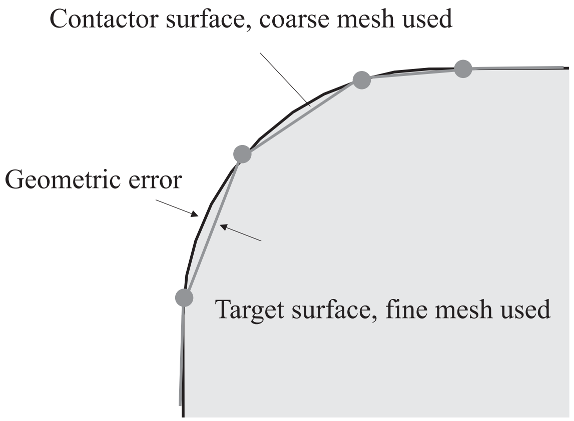
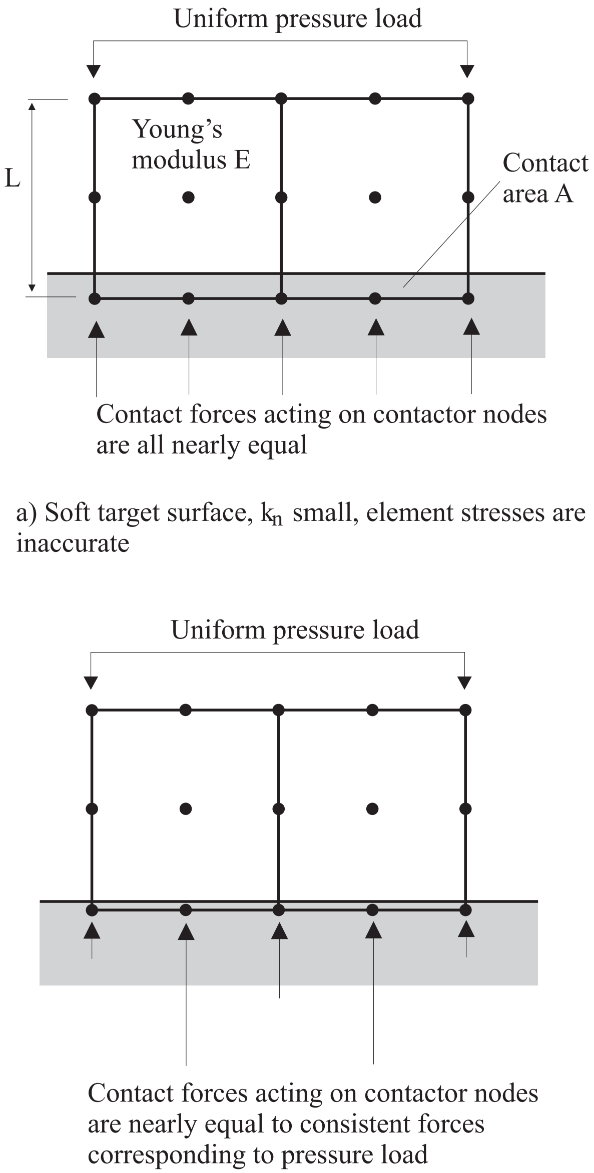

Contactor surface, coarse mesh used
Geometric error
Target surface, fine mesh used

Fig. 4.8-15: Modeling of a curved target surface

Another consideration for the choice of k_n is the following. Because the rigid target algorithm is node-based, and because the contact stiffness is the same for each node in contact, the stresses computed in higher-order elements on the contactor surface will be inaccurate, if k_n is too small. For example, in a problem involving pressing an element onto a contact surface, k_n should be greater than $100\dfrac{EA}{nL}$ where E is the Young's modulus, A is the contact area, L is the element thickness (in the contact direction) and n is the number of nodes on the contact area. The basic concept is illustrated in Fig. 4.8-16.

This issue also arises when lower-order elements are used, but when lower-order elements are used, the variation in the consistent nodal point forces is much less, so k_n can be smaller for the same accuracy in the element stresses.


Uniform pressure load
Young’s
modulus E
L
Contact
area A
Contact forces acting on contactor nodes
are all nearly equal
Uniform pressure load
Contact forces acting on contactor nodes
are nearly equal to consistent forces
corresponding to pressure load

a) Soft target surface, k_n small, element stresses are inaccurate

b) Hard target surface, k_n large, element stresses are accurate

Fig. 4.8-16: Higher-order elements and rigid-target contact

Time step for Solution 701: For Solution 701, the time step should be smaller than

$$\Delta t = \sqrt{2}\sqrt{\frac{m}{k_n}}$$

This formula is derived from the following considerations. Consider a single contactor node with mass m and no additional stiffness or damping, with a velocity normal to the target. If this node just touches the target at time $t - \Delta t$, and penetrates the target at time t, the node should remain in contact at time $t + \Delta t$. The choice of Δt in the above equation satisfies this condition.

Clearly, decreasing k_n will increase the time step Δt.

A node that is out of contact at time $t - \Delta t$, in contact at time t and out of contact at time $t + \Delta t$ is said to have had a contact reversal (Fig. 4.8-17).

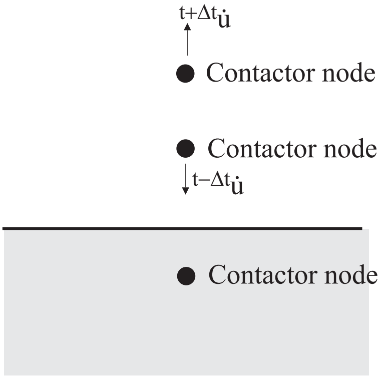


Fig. 4.8-17: Contact reversal due to too large time step in Solution 701

Time step selection in frictional contact: The time step size will affect the frictional velocities and hence the results. This is because, in static analysis, the nodal velocities used in the friction calculations are calculated as the incremental displacements divided by the time step.

In those parts of the analysis in which friction is important, a "realistic" time step should be used.

In those steps of the analysis in which friction is not important, a large time step can be used, which causes the velocities to be small. For example, in metal forming analysis, a large time step size can be used when establishing the blank holder force, and during springback calculations.

Choice of $\dot{u}_{f\,\min}$ *for frictional contact:* $\dot{u}_{f\,\min}$ is set using BCTPARA parameter SLIDVEL. The default value of the minimum sliding velocity $\dot{u}_{f\,\min}$ is 1E-10. However $\dot{u}_{f\,\min}$ can be chosen for optimal convergence. Decreasing $\dot{u}_{f\,\min}$ can lead to convergence difficulties.

We recommend that $\dot{u}_{f\,\min}$ be chosen from experimental data, or that the largest acceptable value of $\dot{u}_{f\,\min}$ be used.

Time step for Solution 701 for frictional contact: For Solution 701, the time step should be smaller than

$$\Delta t = \frac{2m\dot{u}_{f\,\min}}{\mu \left\| F_n \right\|}$$

to prevent reverse sliding. This formula is derived from the finite difference equation corresponding to explicit time integration, when applied to a single contactor node with mass m and no additional stiffness or damping, sticking to the target, but with a nonzero sticking velocity. If Δt is larger than the value in the above equation, the velocity will increase, and eventually the node will slide. The sliding will then tend to "reverse", that is, for a given time step, the sliding direction will be opposite to the sliding direction in the previous step (Fig. 4.8-18).

Note that when the time step is greater than $\Delta t = \dfrac{2m\dot{u}_{f\,\min}}{\mu \left\| F_n \right\|}$, the solution is still stable.

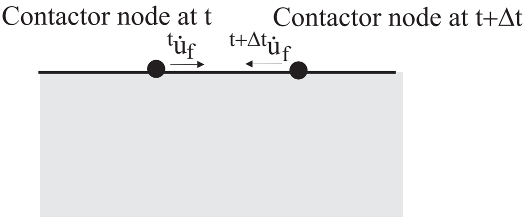


Fig. 4.8-18: Reverse sliding due to too large time step in Solution 701

Automatic time step selection in Solution 701: When using the automatic time step selection options in Solution 701, the time step returned from the rigid target contact algorithm is

$$\Delta t = \min_i \sqrt{\frac{m_i}{k_n}}$$

where the minimum is taken over all contactor nodes. Notice that friction is not considered in the automatic time step selection; this is because the model remains stable even if the time step is larger than the friction time step discussed above.

Birth/death: Rigid target contact surfaces can have a birth and death time, similar to other contact surfaces. The birth and death times are set by BCTPARA parameters TBIRTH and TDEATH (default =0.0, corresponding to no birth and death).

Other user-input parameters:

BCTPARA TFORCE

The maximum tensile force for a node in tensile contact for which convergence is allowed (default value 0.001). All nodes in tensile contact must have a tensile force less than this value for the solution to converge. Tensile contact is not used in Solution 701.

BCTPARA OCHECK

If OCHECK = 0, then oscillation checking (described in Section 4.8.2.3) is turned off. If OCHECK = ITE>0, then oscillation checking is activated after equilibrium iteration ITE. The default is 5. Oscillation checking is not used in Solution 701.

4.8.4 Rigid target contact reports for Solution 601

The following messages are output at the end of each converged solution.

Maximum overlap at convergence:

Meaning: self-explanatory

Recommend: If the maximum overlap is too large, increase k_n; if the maximum overlap is too small, decrease k_n.

Maximum tensile contact gap during iterations for nodes in contact at convergence:

Meaning: A node that is in contact at the start of the time step may temporarily move out of contact during the iterations, then go back into contact before convergence. This report item reports the maximum contact gap of all such nodes. When the tensile contact gap is large, then convergence may be difficult.

Recommend: Either reduce the time step or decrease k_n to reduce the tensile contact gap.

Maximum friction velocity at convergence:

Meaning: For nodes in frictional contact, this is the maximum friction velocity of a node (either sticking or sliding).

Recommend: If the maximum velocity is less than $\dot{u}_{f\,\min}$, and the corresponding node should be sliding, decrease $\dot{u}_{f\,\min}$.

Number of nodes in contact, number of nodes in sticking contact, number of nodes in sliding contact:

Meaning: Self-explanatory. Each node is counted once for each target surface that the node is in contact with. So a node that is in contact with two target surfaces simultaneously is counted twice.

Change of contact status during iterations:

Meaning: The number of nodes that switch contact status (not in contact to in contact, or vice versa), is reported. If there are many nodes that switch contact status, this may cause convergence difficulties.

Recommend: Either reduce the time step or decrease k_n.

In contact at convergence, in tensile contact during iterations.

Meaning: The number of nodes which were in tensile contact during the iterations (meaning that the nodes were almost out of contact) and in contact in the converged solution. When there are many such nodes then convergence may be difficult.

Recommend: Either reduce the time step or decrease k_n to reduce the likelihood that nodes go into tensile contact.

Change in frictional contact status during iterations:

Meaning: The number of nodes that change frictional contact status (from sticking to sliding or vice versa) is reported.

Recommend: If there are many nodes that switch frictional contact status, reduce the time step or increase $\dot{u}_{f\min}$.

The following messages are output at the end of each solution that did not converge.

Maximum change of contact force at end of iterations:

Meaning: The contactor node for which the contact force had the largest change is output.

Recommend: Examine the model near that contactor node for hints about why the solution did not converge.

Change of contact status at end of iterations:

Meaning: The number of nodes that are changing contact status at the end of the iterations.

Recommend: Reduce the time step or decrease k_n.

Sliding reversal at end of iterations:

Meaning: The number of nodes that are undergoing sliding reversals at the end of the iterations.
Recommend: Reduce the time step or increase $\dot{u}_{f\,\mathrm{min}}$

Change of target entity at end of iterations.

Meaning: The number of nodes that are oscillating between different target entities at the end of the iterations.

Recommend: If oscillation checking is not turned on, turn it on. Otherwise refine the target surfaces, or reduce the time step.

4.8.5 Rigid target contact report for Solution 701

The following items are output for each time step in which results are printed or saved:

Number of nodes in contact, number of nodes in sticking contact, number of nodes in sliding contact:

Meaning: See the corresponding message in Section 4.8.4.

Maximum overlap since solution start; maximum overlap since last report:

Meaning: See the corresponding message in Section 4.8.4.

Recommend: See the corresponding recommendations in Section 4.8.4.

Maximum friction velocity since solution start, maximum friction velocity since last report:

Meaning: See the corresponding message in Section 4.8.4.

Recommend: See the corresponding recommendations in Section 4.8.4.

Contact reversals since solution start, since last report:

Meaning: This is a count of the total number of contact reversals. Also the number of contact reversals for the node with the most contact reversals is given, along with the mass of the node.

Recommend: To reduce the number of contact reversals, either reduce the time step or decrease k_n .

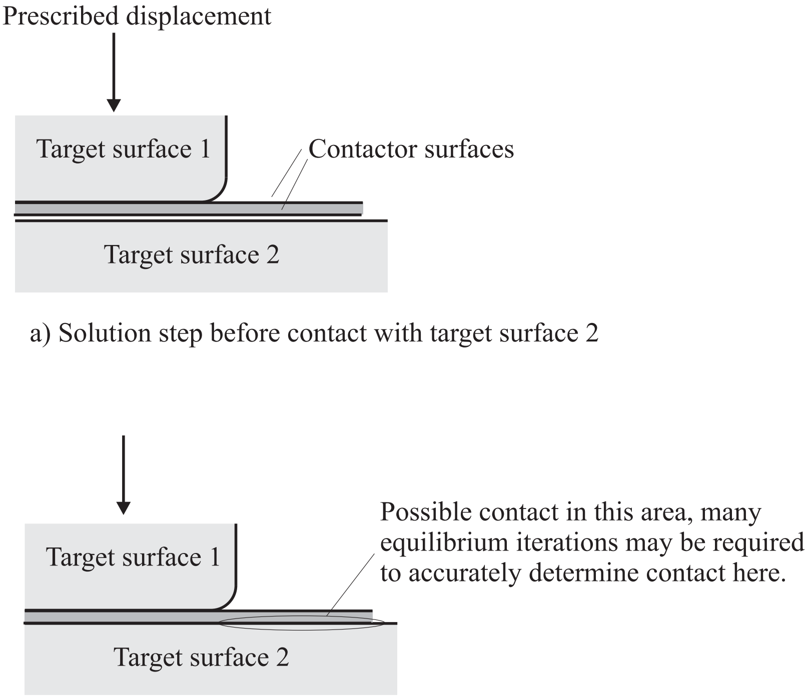
Sliding reversals since solution start, since last report:

Meaning: This is a count of the total number of sliding reversals. Also the number of sliding reversals for the node with the most sliding reversals is given, along with the mass of the node.

Recommend: To reduce the number of sliding reversals, either reduce the time step or increase $\dot{u}_{f\,\min}$.

4.8.6 Modeling hints and recommendations

- For a time step in which contact is established over a large area, many equilibrium iterations may be required in Solution 601. This is because the solution cannot converge until the nodes in and out of contact are determined, and it may take many equilibrium iterations to determine which nodes are in and out of contact. An example is shown in Fig. 4.8-19. The ATS cutback method will not be effective for this time step. Rather, the maximum number of iterations should be set very large, so that the program can find the converged solution.


Prescribed displacement
Target surface 1
Contactor surfaces
Target surface 2
Target surface 1
Possible contact in this area, many equilibrium iterations may be required to accurately determine contact here.
Target surface 2

a) Solution step before contact with target surface 2

b) Solution step after contact with target surface 2

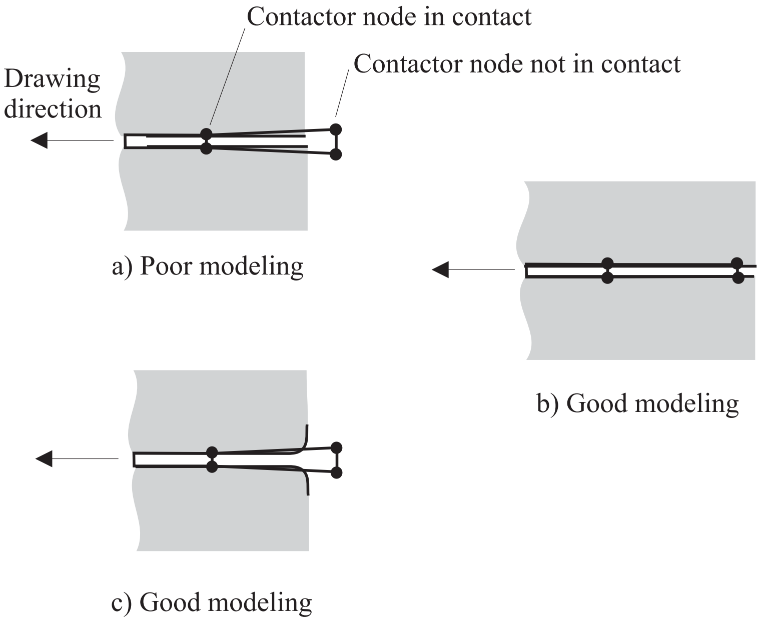
Fig. 4.8-19: Establishment of contact over a large area during a solution step

- When forming a part that is relatively thin, setting the PLASALG flag of the NXSTRAT card to 2 can allow the use of larger time steps.

- The contact search algorithm may take a relatively long time for the first iteration of the first time step. Similarly, the contact search algorithm may take a relatively long time for the first iteration of any time step in which a contact set is born.

- As the contactor surface is refined, keeping k_n constant, the overlap and contact force will decrease at each contactor node. Hence k_n may need to be adjusted as the mesh is refined. In general, as the mesh is refined, k_n can be decreased in order to keep the overlap reasonable.

- Convergence in Solution 601 may become difficult when contactor nodes that were not in contact with the target suddenly interact with the target. An example is illustrated in Fig. 4.8-20. Eventually the contactor nodes on the right will come into contact with the target, and convergence may be difficult. Alternate ways to model this situation are shown in Fig. 4.8-20.



a) Poor modeling

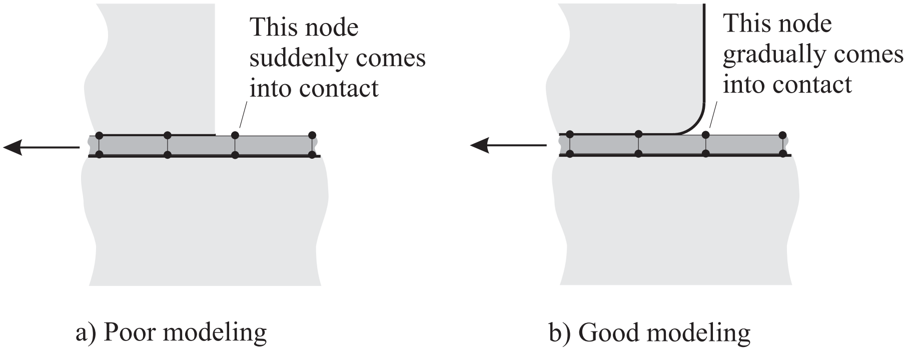
b) Good modeling

c) Good modeling

Fig. 4.8-20: Contactor nodes suddenly coming into contact

Another example is shown in Fig. 4.8-21. In Fig. 4.8-21(a), the top target surface is flat, and the indicated node suddenly comes into contact with the top target surface. Convergence is very difficult, because a very small change in the position of the indicated node can cause the contact status of that node (and hence the contact force) to change abruptly. In Fig. 4.8-21(b), the top target surface has a round corner, and the indicated node gradually comes into contact with the top target surface. Convergence is easier, because a very small change of the position of the indicated node results in only a very small change in the contact force.

In metal forming analysis, this situation in frequently encountered in the modeling of the blank holder. The modeling is easiest if the blank holder is modeled as a flat target surface. But convergence is easier if round corners are added to the blank holder wherever nodes on the blank are anticipated to contact the blank holder during drawing.



a) Poor modeling

b) Good modeling

Fig. 4.8-21: Additional example of nodes suddenly coming into contact

5. Loads, boundary conditions and constraint equations

5.1 Introduction

The objective of this chapter is to present the various options available in Advanced Nonlinear Solution for the description of applied loads, boundary conditions, and constraint equations.

- Table 5-1 lists the Case Control commands used for loading, boundary conditions, and initial conditions.

- Note that the selected DLOAD set can be used for any time varying loads in both static and dynamic analysis. Similarly, the selected LOAD set can be used for defining constant loads in both static and dynamic analyses.

- Table 5-2 lists the load, boundary condition and initial condition Bulk Data entries supported in Advanced Nonlinear Solution.

- Table 5-3 lists the Bulk Data entries used for combining applied loads and/or enforced displacements in Advanced Nonlinear Solution.

Table 5-1: Case Control commands in Advanced Nonlinear Solution

Case Control Command	Comments
DLOAD	Select load set (time varying)
LOAD	Select load set (non-time varying)
SPC	Select single-point constraint set (including enforced displacement)
MPC	Select multipoint constraint set
IC	Select initial conditions set (displacements and velocities)
TEMPERATURE	Select initial and applied thermal load sets
BOLTLD	Select bolt preload set
DTEMP	Select time-dependent temperature set

Table 5-2: Bulk data entries for defining loads, boundary conditions and constraints

Bulk Data Entry	**Comments**
FORCE, FORCE1, FORCE2	Concentrated force on nodes
MOMENT, MOMENT1, MOMENT2	Concentrated moment on nodes
SPC^{1,2,4}, SPC1^{4}, SPCADD^{4}	Fixed or enforced degrees of freedom on nodes
SPCD^{4}	Enforced displacement on nodes
PLOAD	Uniform pressure on shell element or 3-D solid face
PLOAD1	Distributed load on beam element Concentrated force on beam nodes
PLOAD2	Uniform pressure on shell element
PLOAD4	Pressure or distributed load on shell or 3-D solid face
PLOADE1	Varying pressure on plane stress or plane strain 2-D solid element
PLOADX1	Varying pressure on axisymmetric 2-D solid element
TEMP	Applied temperature on nodes
TEMPD	Applied default temperature
DTEMP	Time-dependent temperature loading
GRAV	Mass proportional inertial load
RFORCE	Centrifugal load
RFORCE1	Centrifugal load with more than one load applied to the model
MPC, MPCADD	Define multipoint constraints
TIC^{3}	Initial displacement and velocity on nodes
BOLTFOR	Preload force on bolts

Table 5-2: (continued)

Bulk Data Entry	Comments
TEMPBC	Applied temperature on nodes in heat transfer analysis
QHBDY, QBDY1	Uniform heat flux on boundary element
QBDY2	Varying heat flux on boundary element
QVOL	Uniform volumetric heat addition
CONV	Free convection on boundary element
RADBC	Space radiation on boundary element

Notes:

1. SPC can also enforce displacement.
2. If enforced displacements are always 0.0 they become a boundary condition.
3. Initial conditions are discussed in Section 10.1.
4. Can also be used to fix or enforce temperature in a heat transfer analysis.

- The LOAD entry is used for combining loads that are constant throughout the analysis while DLOAD is used for combining time-varying loads. The DLOAD entry references a load defined through a TLOAD1 entry. The TLOAD1 entry references the type of load (applied load or enforced displacement), as well as the table entry (TABLED1 or TABLED2) defining the time variation of the load.

- Both LOAD and DLOAD can be used in static and dynamic analyses in Advanced Nonlinear Solution.

Table 5-3: Bulk data entries for applying loads and enforced displacements

Bulk Data Entry	Comments
LOAD	Defines a linear combination of constant loads
DLOAD	Defines a linear combination of time varying loads (by combining different TLOAD1 entries)
TLOAD1	Defines time varying loads and enforced motion
TABLED1, TABLED2	Defines the time functions used by the loads

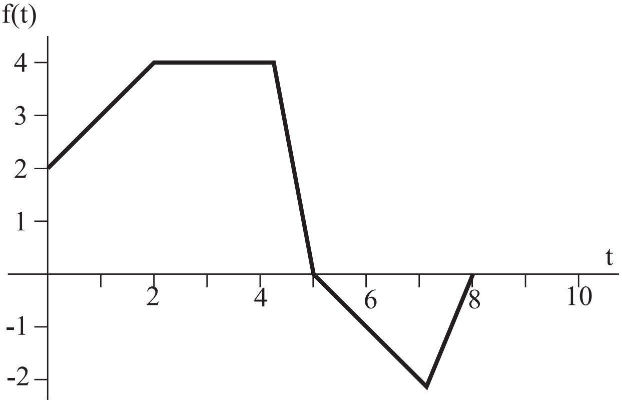
- A time function is defined as a series of points $\left(t, f_i(t)\right)$ in which *t* is time and $f_i(t)$ is the value of time function *i* at that time. Between two successive times, the program uses linear interpolation to determine the value of the time function.

- Advanced Nonlinear Solution does not support subcases. If subcases are only used to change the applied load in a static analysis, then they can be equivalently defined in Advanced Nonlinear Solution as time-varying loads in a single case.

- A typical time-varying load such as the enforced displacement shown in Fig 5.1-1 (on the y direction of node 100) will be applied as follows:

```
DLOAD, 1, 1.0, 10.0, 5
TLOAD1, 5, 3,, DISP, 7
SPCD, 3, 100, 2, 1.0
TABLED2, 7, 0.0,
,0.0, 2.0, 2.0, 4.0, 4.0, 4.0, 5.0, 0.0,
,7.0, -2.0, 8.0, 0.0, ENDT
TSTEP, 1, 8, 1.0, 4
```

Time functions:



Resulting load values for $^tR = 10f(t)$:

Time	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0
^tR	20	30	40	40	40	0	-10	-20	0

Fig. 5.1-1: Typical time-varying load

Note that the TSTEP entry is used for both linear and nonlinear analyses. In this case, 8 steps of size 1.0 are selected with output every 4 steps.

Note that in Solution 701 with automatic time step selection, the above input will not result in 8 steps. Instead, the critical time step for the model will be used and output of results will be done as soon as the solution time exceeds 4.0 and 8.0. See Section 7.1 for details.

- The LOAD case control command can point to a LOAD entry or to individual loads, and similarly the DLOAD case control command can point to a DLOAD entry or directly to a TLOAD1 entry. The initial and applied temperature load sets must be selected by the TEMPERATURE case control command if needed. The active initial conditions must be selected by the IC case control command.

- Boundary conditions can be grouped into two classes: essential and natural boundary conditions (see ref. KJB, Section 3.3.2).

Essential boundary conditions can be enforced displacements or rotations. Natural boundary conditions include all applied forces and moments.

- Displacement boundary conditions include fixed nodal degrees of freedom, enforced displacements and constraint equations.

- Force and moment boundary conditions include numerous types of applied loading available in Advanced Nonlinear Solution.

- All displacement and force boundary conditions are referred to the displacement coordinate system at the node at which they act.

- The externally applied load vector used in the governing equilibrium equations is established using contributions from the various applied loads.

  For concentrated loads, the contributions of these nodal loads are directly assembled into the externally applied load vector.

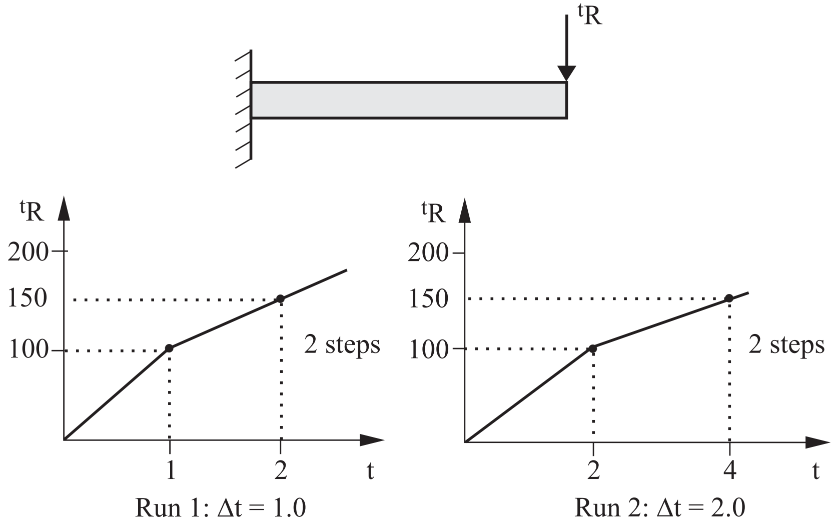
  For pressure loading, distributed loading, centrifugal loading and mass proportional loading, Advanced Nonlinear Solution first calculates the corresponding consistent nodal load vectors (consistent in the sense that the principle of virtual work is used) and then assembles these load vectors into the externally applied load vector. The evaluations of the consistent nodal load vectors for the various types of loading are described in the following sections.

- Temperatures in Advanced Nonlinear Solution are used in conjunction with material models which include temperature effects.

- The definition of the time variation of the externally applied load vector for the various time steps in the solution period depends on whether automatic step incrementation is used or not.

Time variation of externally applied loads when automatic step incrementation is not used: Each applied load or enforced displacement is associated with a time function which defines the time variation of the load throughout the solution period.

  ▸ In a static analysis in which time-dependent effects (such as creep or friction) are not included in the material models, time is a "dummy" variable which is used, via the associated time function of each applied load, to define the load intensity at a step. Thus, the time step increment directly establishes the load increments. So, in the example shown in Fig. 5.1-2, the same solution is obtained regardless of the size of the time step increment.



Run 1: $\Delta t = 1.0$

Run 2: $\Delta t = 2.0$

Note: identical results are obtained in Run 1 and Run 2
for a linear static analysis.

Fig. 5.1-2: Example of time varying loads

▸ In a dynamic analysis or if time-dependent effects are included in the material models in a static analysis, time is used in a similar way to define the load intensity of an applied load at a step. However, in these cases, time is a "real" variable because the time step increment is employed in the actual integration of the equations of motion in a dynamic analysis, and in the integration of the element stresses in a creep analysis. Hence, in these cases the choice of the time step increment is no longer arbitrary.

Time variation of externally applied loads when automatic step incrementation is used: Two options are available:

- Using the automatic-time-stepping (ATS) procedure (see Section 6.2.4), the loads are defined for all times Δ*t*, 2Δ*t*,... as for no automatic step incrementation. In addition, when the algorithm subdivides a time (load) step, the load vector is established by linear interpolation of the load vectors at times *t* and *t* + Δ*t*.
- Using the load-displacement-control (LDC) procedure (see Section 6.2.6), the applied loads are not associated with any time function and the time variation of the loads cannot be specified by the user. The contributions from all the loads are assembled into a constant load vector denoted as the reference load vector. During the response calculation, this reference load vector is scaled proportionally using a load multiplier (in general different from one step to the next) automatically computed by the program.

- The activation of the various applied loads can be delayed using the X1 field in the TABLED2 entry. The arrival/delay time option does not apply, however, to centrifugal and mass-proportional loading, see Section 5.4.

The specification of a nonzero arrival time corresponds to a shifting of the associated time function forward in time. If the time function is used by a force boundary condition, this corresponds to using a time function multiplier of zero for all times *t* smaller than the arrival time; see illustration given in Fig. 5.1-3. However, if the time function corresponds to a enforced displacement/rotation the associated degrees of freedom are assumed to be free prior to the arrival time (not having a zero prescribed value).

The arrival time feature is used only if NXSTRAT TFSHIFT=1 (the default). If NXSTRAT TFSHIFT=0, any arrival time data specified by TABLED2 X1 is ignored.

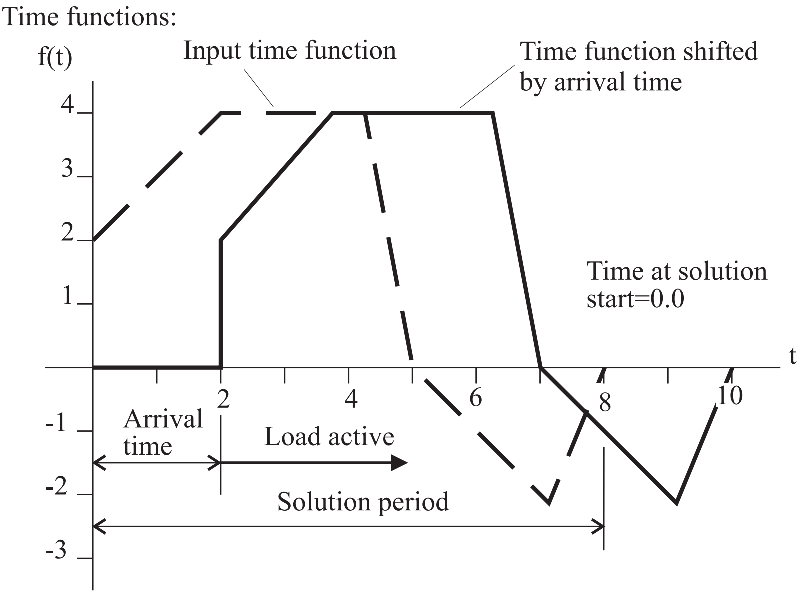


Fig. 5.1-3: Example of the use of the arrival time option

5.2 Concentrated loads

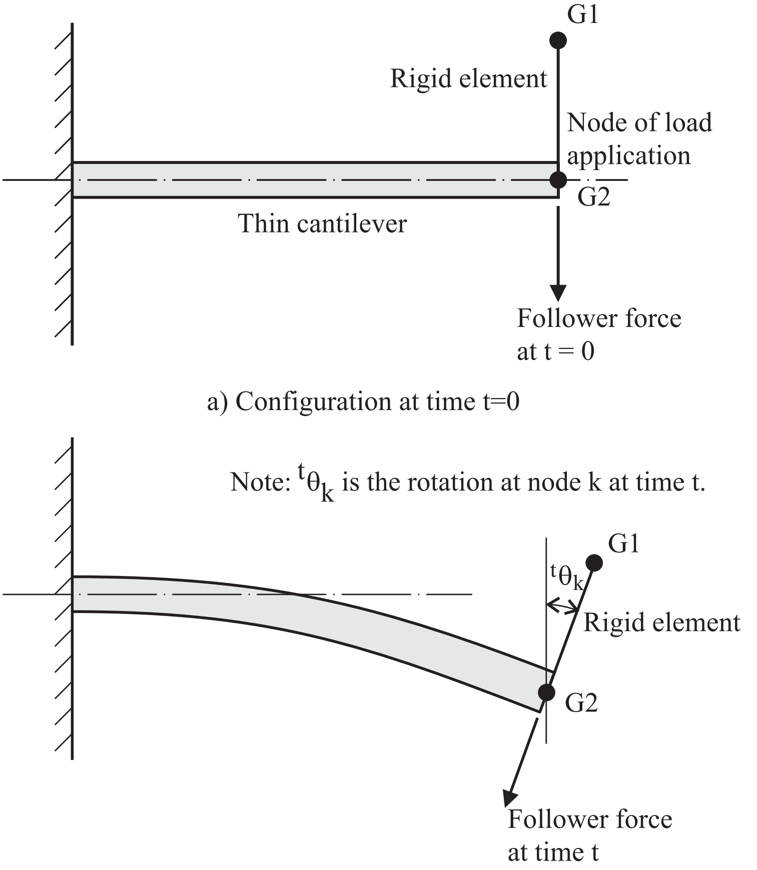
- Concentrated loads are nodal point forces applied at the specified nodes using the FORCE, FORCE1, or FORCE2 entries. Concentrated moments are also applied to specific nodes using the MOMENT, MOMENT1, or MOMENT2 entries. Concentrated forces on beam nodes can also be applied using the PLOAD1 entry.

- The direction in which a concentrated load acts depends on the displacement coordinate system assigned to the node.

- Note that concentrated moments applied to shell nodes convert the shell nodes automatically to 6 degree of freedom nodes. This is done since the local $\mathbf{V}_1$ and $\mathbf{V}_2$ directions at shell nodes are unknown to the user, and hence cannot be used in defining moments.

- When the FORCE1 or MOMENT1 entries are used in a large displacement analysis, they can be follower loads, meaning that the direction of the applied force or moment can be updated during the simulation based on the current coordinates of the G1 and G2 nodes. This however is only possible if either G1 or G2 is set to be the node of load application.

- The direction of a follower load can be controlled using RBAR or RBE2 rigid elements (see Section 2.7). An example is given in Fig. 5.2-1.



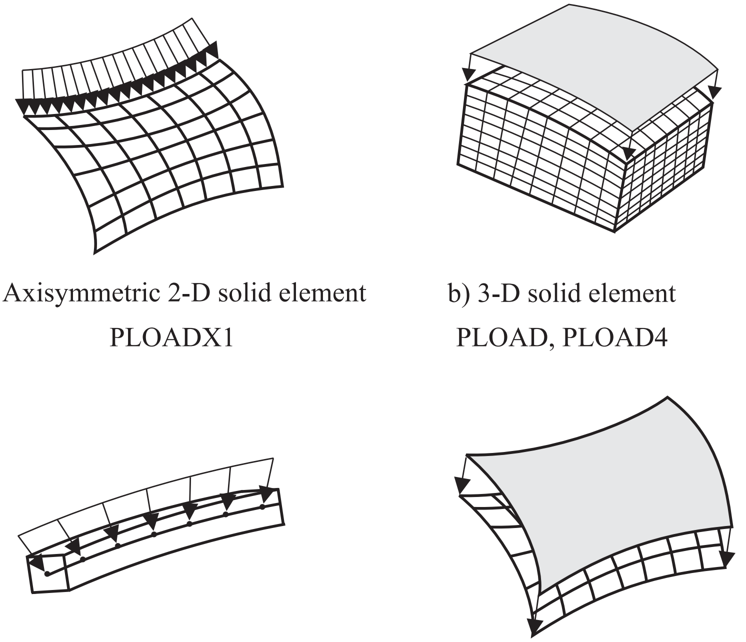
a) Configuration at time t=0

b) Configuration at time t

Fig. 5.2-1: Example of the use of a rigid element to establish the follower load direction

5.3 Pressure and distributed loading

- Some examples of pressure and distributed loading are shown in Fig. 5.3-1.



a) Axisymmetric 2-D solid element
PLOADX1

b) 3-D solid element
PLOAD, PLOAD4

c) Beam element
PLOAD1

d) Shell element
PLOAD, PLOAD2, PLOAD4

Fig. 5.3-1: Examples of distributed and pressure loading

- Distributed loads can be applied to beam elements using the PLOAD1 entry. This entry can also be used to apply concentrated forces on beam nodes.

- Pressure loads can be applied to shell elements using the PLOAD or PLOAD4 entries, and to shell 3-node and 4-node elements only (CTRIA3 and CQUAD4) using the PLOAD2 entry.

- Pressure loads can be applied to axisymmetric 2-D solid elements using the PLOADX1 entry. Pressure loads are input as force per unit area.

- Pressure loads can be applied to plane stress or plane strain 2-D solid elements using the PLOADE1 entry. Pressure loads are input as force per unit area.

  Deformation-dependent pressure loads that act onto the edges of plane stress elements do not take into account the change in element thickness due to in-plane deformations. For example, a plane stress element that undergoes uniaxial tension due to a deformation-dependent pressure load has internal stresses larger than the pressure load by the ratio (original thickness)/(current thickness).

- Pressure loads can be applied to the faces of 3-D elements (HEXA, CPENTA, CTETRA, CPYRAM) using the PLOAD or PLOAD4 entries.

- When applied through the PLOAD4 entry, the pressure can be normal to the face of the element, or along a specified direction.

- For each pressure/distributed load surface specified, a consistent nodal load vector is calculated to represent the pressure/distributed loading.

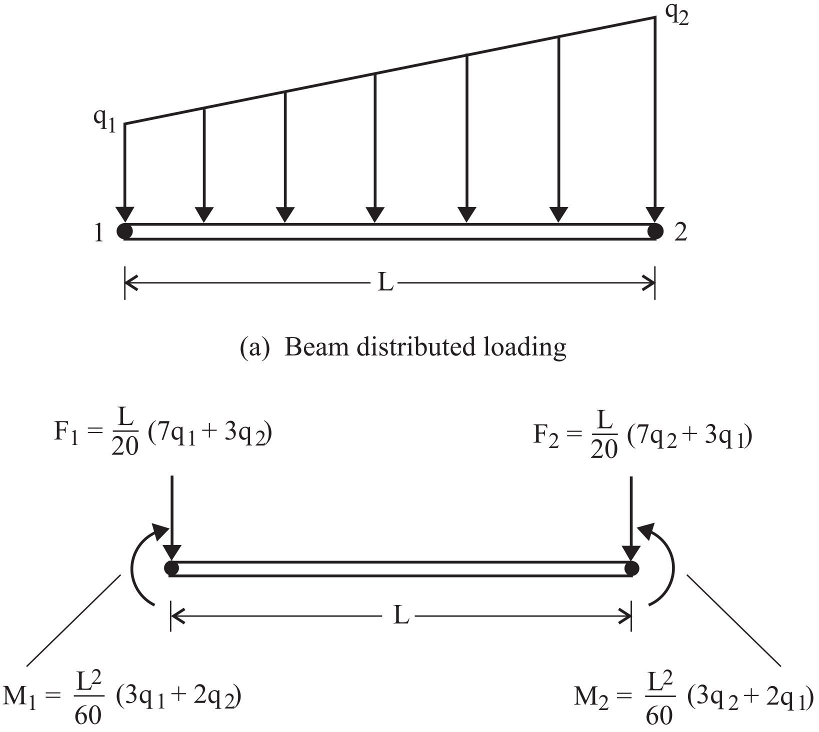
- In a large displacement analysis, the pressure/distributed loading can be specified as deformation dependent for all element types via the LOADOPT parameter in the NXSTRAT entry. In this case, the calculations of the consistent load vectors are based on the latest geometry and configuration of the loading surface.

- In Solution 601, deformation dependent loading should only be used in a large displacement analysis. Equilibrium iterations (see Chapter 6) should in general be performed if deformation dependent loading is present.

- The loading direction for distributed loads can be along the basic coordinate system or the element coordinate system. Loads along element coordinate systems can be deformation dependent in large displacement analysis.

- For pressure loading on 2-D and 3-D solid elements, the consistent load vector consists of nodal forces acting on the translational degrees of freedom only. The calculation of this load vector is given in ref. KJB, Section 4.2.1. The same effect occurs in shells since the nodal translations and rotations are interpolated independently.

- The distributed loading on a beam element results in equivalent concentrated forces and moments acting at the beam nodes as shown in Fig. 5.3-2. The calculation of these consistent forces and moments also follows the equations in ref. KJB Section 4.2.1.

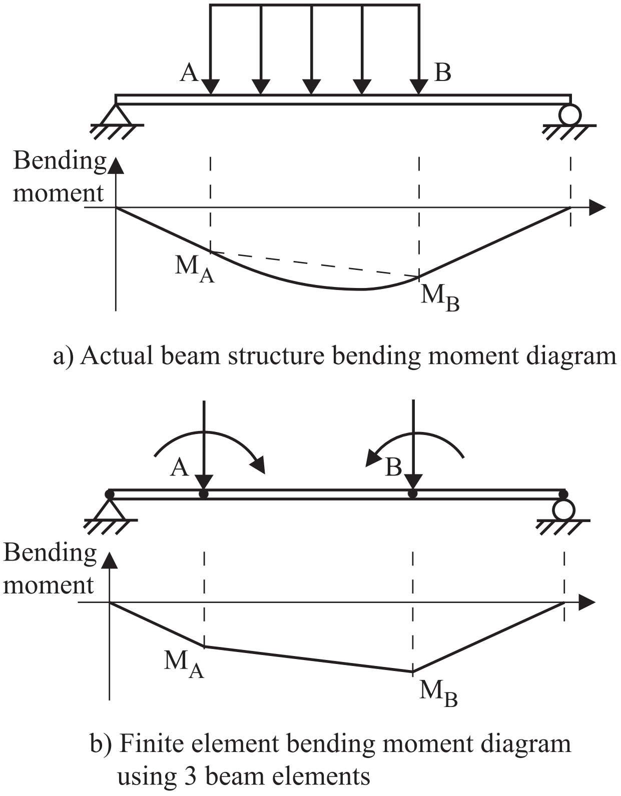

q_2
q_1
1
2
L
$F_1 = \frac{L}{20}(7q_1 + 3q_2)$
$F_2 = \frac{L}{20}(7q_2 + 3q_1)$
L
$M_1 = \frac{L^2}{60}(3q_1 + 2q_2)$
$M_2 = \frac{L^2}{60}(3q_2 + 2q_1)$

(a) Beam distributed loading

(b) Fixed-end forces/moments representation

Fig. 5.3-2: Representation of beam distributed loading

- Displacements and stresses in the model are calculated by representing the actual distributed loading using the consistent load vector defined above. Hence, the calculated solution corresponds only to these equivalent concentrated nodal forces and moments, and may not correspond entirely to beam theory results taking account of the distributed loading more accurately, see Fig. 5.3-3. In order to capture the applied moment more accurately, more beam elements are required between sections A and B.



a) Actual beam structure bending moment diagram

b) Finite element bending moment diagram using 3 beam elements

Fig. 5.3-3: Beam element bending moments when subjected to distributed loading

5.4 Inertia loads ─ centrifugal and mass proportional loading

- Centrifugal and mass proportional loading can be used to model the effect of body forces which arise from accelerations to which the structure is subjected.

- Centrifugal loading is generated using the RFORCE and RFORCE1 entries, and mass proportional loading is generated using the GRAV entry.

- The mass matrix used in the calculation of centrifugal and mass proportional loading can be lumped or consistent depending on the mass setting for the whole model. Note that the computational effort involved in the evaluation of a lumped mass matrix is, in general, much less than the effort for a consistent mass matrix.

- Centrifugal and mass proportional loading can both be present in a static or dynamic analysis. In a dynamic analysis, the type of mass matrix employed in the load calculation and in the dynamic response calculation are the same.

- When more than one centrifugal load is applied to part of the model, each load acts independently. This means that certain inertia forces due to the coupling of the velocities (e.g. gyroscopic forces) are not accounted for in the analysis. In certain problems, the coupling terms are significant. Therefore, in general, it is recommended to only apply one centrifugal load to any part of the model.

- The angular velocities in centrifugal loading are not coupled with the nodal velocities in the dynamic response. This means that certain inertia forces due to the coupling of the velocities (e.g. gyroscopic forces) are not accounted for in dynamic analysis. It is recommended to only use centrifugal loading in static analysis.

- When elements die (due to rupture), their contribution to the load vector is removed. Hence, the consistent load vector consists (at all times) only of the contributions from the elements currently alive.

- Centrifugal and mass proportional loading cannot be applied with a delay/arrival time. The time function has to be shifted manually to create this effect.

- Centrifugal or mass proportional forces at fixed nodes are taken into account in the calculation of reaction forces.

- Centrifugal l loading is not supported for potential-based fluid elements.

- Centrifugal loads are only approximately computed for the large displacement beam element. This is because the centrifugal loads are calculated assuming that the beam is straight. The error in the solution is reduced as the beam element mesh is refined.

Centrifugal loading

- The consistent load vector for centrifugal loading is computed as follows (see Fig. 5.4-1):

$$ {}^{t}\mathbf{R} = -\int \rho\Big[\ \underbrace{\left({}^{t}\boldsymbol{\alpha}\times{}^{t}\mathbf{r}\right)}_{\text{angular acceleration}} + \underbrace{{}^{t}\boldsymbol{\omega}\times\left({}^{t}\boldsymbol{\omega}\times{}^{t}\mathbf{r}\right)}_{\text{centripetal acceleration}}\ \Big]\, dV \qquad (5.4\text{-}1)$$

where ${}^{t}\boldsymbol{\alpha} = \left(2\pi\,\mathrm{RACC}\, f(t)\right)\mathbf{R}$ is the angular acceleration vector, ${}^{t}\boldsymbol{\omega} = \left(2\pi\,\mathrm{A}\sqrt{f(t)}\right)\mathbf{R}$ is the angular velocity vector, ${}^{t}\mathbf{r}$ is the radial vector from the axis of rotation to the node, A is the scale factor of the angular velocity (in revolutions/unit time), RACC is the scale factor of the angular acceleration (in revolutions / unit time squared), ρ is the density, *f*(*t*) is the time function, and **R** (no left superscript) is the rotation vector.

Note that the centrifugal force is directly proportional to the time function *f*(*t*).

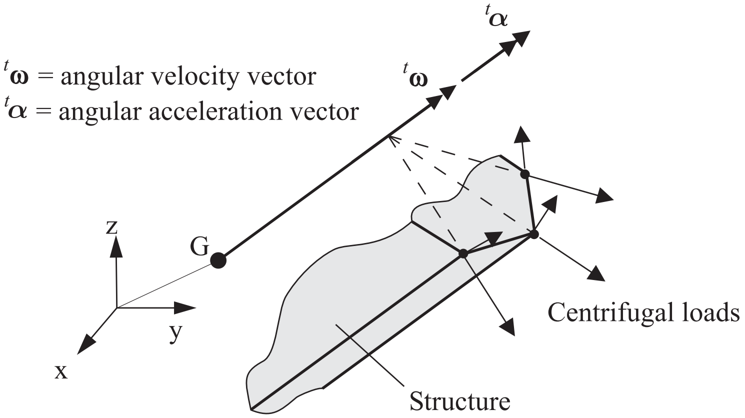

${}^{t}\boldsymbol{\omega}$ = angular velocity vector
${}^{t}\alpha$ = angular acceleration vector
${}^{t}\alpha$
${}^{t}\boldsymbol{\omega}$
z
G
y
x
Centrifugal loads
Structure

Fig. 5.4-1: Convention used for centrifugal loading

Note that the consistent load vector consists of nodal point forces applied to all the nodes of the finite element model. But loads applied to fixed degrees of freedom do not have any effects in the solution.

- When centrifugal loading is used in nonlinear analysis, additional nonlinear terms are added to the stiffness matrix and the load vector. These additional nonlinear contributions are described in the following. Let the equilibrium of the finite element system be calculated at time $t + \Delta t$, iteration i. Then

$$ {}^{t+\Delta t}\mathbf{M}\ {}^{t+\Delta t}\ddot{\mathbf{U}}^{(i)} + {}^{t+\Delta t}\mathbf{K}^{(i-1)}\,\Delta\mathbf{U}^{(i)} = {}^{t+\Delta t}\mathbf{R}^{(i)} - {}^{t+\Delta t}\mathbf{F}^{(i-1)} \qquad (5.4\text{-}2)$$

where **M** = mass matrix, **K** = stiffness matrix = $\left(\mathbf{K}_L + \mathbf{K}_{NL_0}\right)$, $\ddot{\mathbf{U}}$ = acceleration vector, $\Delta\mathbf{U}$ = incremental displacement vector, **R** = external load vector, **F** = nodal point forces corresponding to stresses, (i) = iteration i, and $t + \Delta t$ = time $t + \Delta t$.

The complete expression of the load contribution due to the angular velocity part of the centrifugal loading, including all nonlinear effects, is

$$ {}^{t+\Delta t}\mathbf{R}^{(i)} = -\int \rho\left(\boldsymbol{\omega}\times\left(\boldsymbol{\omega}\times\left({}^{0}\mathbf{r} + {}^{t+\Delta t}\mathbf{U}^{(i-1)} + \Delta\mathbf{U}^{(i)}\right)\right)\right) dV \qquad (5.4\text{-}3)$$

where ${}^{0}\mathbf{r}$ = initial radial vector from the axis of rotation to the node and $\boldsymbol{\omega}$ = angular velocity vector. From the expression (5.4-3), it can be seen that a deformation-dependent load is present, given by

$$ {}^{t+\Delta t}\mathbf{R}^{(i)}_{NL} = -\int \rho\left(\boldsymbol{\omega}\times\left(\boldsymbol{\omega}\times {}^{t+\Delta t}\mathbf{U}^{(i-1)}\right)\right) dV $$

and that an additional nonlinear contribution $\mathbf{K}_{NL_1}$ to the stiffness matrix is present, given by

$$ {}^{t+\Delta t}\mathbf{K}^{(i-1)}_{NL_1}\Delta\mathbf{U}^{(i)} = \int \rho\left(\boldsymbol{\omega}\times\left(\boldsymbol{\omega}\times\Delta\mathbf{U}^{(i)}\right)\right) dV $$

Both of these effects are included in the program.

The complete expression of the angular acceleration load contribution, including all nonlinear effects, is

$$^{t+\Delta t}\mathbf{R}^{(i)} = -\int \rho\left(\boldsymbol{\alpha}\times\left({}^{0}\mathbf{r} + {}^{t+\Delta t}\mathbf{U}^{(i-1)}\right)\right)\, dV$$

In this case, a deformation-dependent load is present, given by

$$^{t+\Delta t}\mathbf{R}_{NL}^{(i)} = -\int \rho\left(\boldsymbol{\alpha}\times {}^{t+\Delta t}\mathbf{U}^{(i-1)}\right)\, dV$$

However, Solution 601 does not include the nonlinear stiffness matrix contribution from the angular acceleration load. The stiffness matrix associated with this load is skew-symmetric due to the single cross-product term.

- Nonlinear centrifugal loading can be used in static analysis and implicit dynamic analysis. Nonlinear centrifugal loading cannot be used in explicit dynamics (Solution 701).

- The correction to the stiffness matrix and the correction to the loading are made when deformation dependent loading is requested (LOADOPT parameter in the NXSTRAT entry).

- More than one centrifugal load can be specified using the RFORCE1 entry.

- In the RFORCE1 entry, the GROUPID field can be used to choose the elements onto which the centrifugal loads will be applied. The GROUPID field selects a group of elements defined using the GROUP bulk entry.

Mass-proportional loading

- The consistent load vector for mass proportional loading in direction *i* is computed using the mass matrix of the entire finite element system and the specified accelerations (only in the translational degrees of freedom), as follows:

$$^{t}\mathbf{R}_i = {}^{t}\mathbf{M}\,\mathbf{d}_i\,{}^{t}a_i \tag{5.4-4}$$

where $\mathbf{d}_i$ is a direction vector with "1" in the portions of the translational degrees of freedom acting into the direction i and "0" in the other portions, and a_i is the acceleration magnitude in the direction i.

In the calculation, the mass coupling term between active and deleted degrees of freedom is included. This mass coupling term can be clearly seen in the discussion of ground motion loads later in this section.

- Mass proportional loading is commonly used to model gravity loading. For gravity loading, ${}^t a_i$ is the acceleration vector due to gravity. For example, for the z coordinate in the vertical direction (increasing z corresponds to movement away from the ground), enter ${}^t a_z = -g$, where g is the (positive) acceleration due to gravity.

- Mass-proportional loads are frequently used to model ground motions. The basis for using mass-proportional loads in modeling ground motions is given briefly now. The equations of motion for linear dynamics, not including damping but including ground motions, are

$$\begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{12}^T & \mathbf{M}_{22} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{x}}_1 \\ \ddot{\mathbf{x}}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{12}^T & \mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix} \tag{5.4-5}$$

where $\mathbf{x}_1$ is the vector of nodal point displacements for nodes not attached to the ground and $\mathbf{x}_2$ is the vector of nodal point displacements for nodes attached to the ground. $\mathbf{R}_1$, $\mathbf{R}_2$ are externally applied forces (for example, concentrated forces).

When the ground motions are the same at all nodes attached to the ground, $\mathbf{x}_1 = \mathbf{u}_1 + \mathbf{d}_{1i} x_{gi}$, $\ddot{\mathbf{x}}_1 = \ddot{\mathbf{u}}_1 + \mathbf{d}_{1i} \ddot{x}_{gi}$, $\mathbf{x}_2 = \mathbf{u}_2 + \mathbf{d}_{2i} x_{gi}$, $\ddot{\mathbf{x}}_2 = \ddot{\mathbf{u}}_2 + \mathbf{d}_{2i} \ddot{x}_{gi}$, where $\mathbf{u}_1$ is the vector of nodal point displacements relative to the ground for nodes not attached to the ground and $\mathbf{u}_2$ is the vector of nodal point displacements relative to the ground for nodes attached to the ground. Clearly, $\mathbf{u}_2 = \mathbf{0}$. Also $\mathbf{d}_{1i}$ is the direction vector for the nodes not attached to the ground and $\mathbf{d}_{2i}$ is the direction vector for the nodes attached to the ground, with "1" in the portions of the translational degrees of freedom acting into the direction i and "0" in the other portions.

The matrix equation of motion becomes

$$\begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{12}^T & \mathbf{M}_{22} \end{bmatrix} \left(\begin{bmatrix} \ddot{\mathbf{u}}_1 \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{d}_{1i}\ddot{x}_{gi} \\ \mathbf{d}_{2i}\ddot{x}_{gi} \end{bmatrix} \right) + \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{12}^T & \mathbf{K}_{22} \end{bmatrix} \left(\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{d}_{1i}x_{gi} \\ \mathbf{d}_{2i}x_{gi} \end{bmatrix} \right) = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix} \tag{5.4-6}$$

Now $\begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{12}^T & \mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{d}_{1i} \\ \mathbf{d}_{2i} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}$ since the vector $\mathbf{d}_i = \begin{bmatrix} \mathbf{d}_{1i} \\ \mathbf{d}_{2i} \end{bmatrix}$ corresponds to a rigid body motion. The matrix equation of motion becomes

$$\begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{12}^T & \mathbf{M}_{22} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}}_1 \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{12}^T & \mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix} - \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{12}^T & \mathbf{M}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{d}_{1i}\ddot{x}_{gi} \\ \mathbf{d}_{2i}\ddot{x}_{gi} \end{bmatrix} \tag{5.4-7}$$

and therefore the system of equations solved is

$$\mathbf{M}_{11}\ddot{\mathbf{u}}_1 + \mathbf{K}_{11}\mathbf{u}_1 = \mathbf{R}_1 - \mathbf{M}_{11}\mathbf{d}_{1i}\ddot{x}_{gi} - \mathbf{M}_{12}\mathbf{d}_{2i}\ddot{x}_{gi} \tag{5.4-8}$$

The mass coupling term between active and deleted degrees of freedom ($\mathbf{M}_{12}\mathbf{d}_{2i}\ddot{x}_{gi}$) is included.

It is seen that the ground acceleration can be applied to the model as a mass-proportional load, provided that the resulting nodal point motions are interpreted as motions relative to the ground.

Please note:

- To enter a positive ground acceleration $\ddot{x}_{gi}$, specify a negative mass-proportional load a_i.
- All single-point fixities are relative to the ground. In other words, fixing a node attaches it to the ground.
- All enforced displacements are relative to the ground.
- All single DOF scalar elements are attached to the ground.

Damping can be used. However, scalar dampers and single DOF damping scalar elements are attached to the ground. Mass-proportional Rayleigh damping acts relative to the ground motion.

Although we have illustrated the procedure only for linear dynamics, the procedure is also valid in nonlinear dynamics.

5.5 Enforced motion

- Enforced displacements at specified degrees of freedom can be applied in Advanced Nonlinear Solution using the single point constraint entries (SPC, SPC1, or SPCADD) or the enforced motion (SPCD) entry. The applied displacement can be constant or described by a time function.

- Enforced velocities and accelerations are not supported.

- Nodal point translations and rotations can be enforced. The degree of freedom is in the direction of the displacement coordinate system assigned to the node.

- A nodal point can be "fixed" by prescribing a zero displacement component for all degrees of freedom at this node. This is, however, different from imposing a permanent single-point constraint on the GRID entry because the enforced degrees of freedom are retained in the system matrices (i.e., equation numbers are assigned) whereas the degrees of freedom at which permanent GRID constraints are imposed are deleted from the system matrices.

- Note that enforced displacements are not recommended on contactor surfaces (see Chapter 4).

- *Arrival time option:* Delay or arrival times can be used for enforced displacements. In this case, the displacements are free before the arrival time. Once the arrival time is reached, the displacements are set to their enforced values. However, the enforced value can be interpreted as an absolute or total displacement or as a relative displacement based on the configuration at the arrival time.

- *Relative enforced displacement option:* Normally the specified enforced displacement is directly applied to the model, so that the enforced displacement at the node is equal to the specified enforced displacement. However, there is a "relative enforced displacement option" that works as follows. When the relative enforced displacement option is active, then the enforced displacement at the node is calculated using the specified enforced displacement plus a preexisting displacement. The preexisting displacement is the displacement at the arrival time (if the arrival time is nonzero), or at the beginning of the current analysis (which might be a restart analysis).

The time function for the specified enforced displacement can be constant or time-dependent.

As a special case, the specified enforced displacement can be zero. Then the enforced displacement is equal to the preexisting displacement. This option is useful for "freezing" the motion of a node, so that before the arrival time, the node has no enforced displacement, and after the arrival time, the node has a enforced displacement equal to the displacement at the arrival time.

The relative enforced displacement option is controlled by the DISPOPT flag in the NXSTRAT entry. DISPOPT=0 selects absolute enforced displacements (the default) and DISPOPT=1 selects relative enforced displacements.

- Enforced rotations can be applied to the rotational degrees of freedom of large displacement formulation structural elements (such as Hermitian beam elements), and also to the master rotational degrees of freedom of large displacement rigid links. The formula used is now discussed.

In large displacement analysis, the increment in enforced rotation is computed as $\Delta\boldsymbol{\theta} = {}^{t+\Delta t}\boldsymbol{\theta} - {}^{t}\boldsymbol{\theta}$, where ${}^{t}\boldsymbol{\theta}$, ${}^{t+\Delta t}\boldsymbol{\theta}$ are the enforced rotations at times t, $t+\Delta t$. The components of the increment in enforced rotation are $\Delta\boldsymbol{\theta} = (\Delta\theta_x, \Delta\theta_y, \Delta\theta_z)$.

Consider an incremental rotation with components $\Delta\boldsymbol{\theta} = (\Delta\theta_x, \Delta\theta_y, \Delta\theta_z)$ applied to the master node of a large displacement rigid link with end coordinates ${}^{t}\mathbf{x}_m$, ${}^{t}\mathbf{x}_s$. ("m" = master, "s" = slave). The vector connecting these coordinates is ${}^{t}\mathbf{x}_{sm} = {}^{t}\mathbf{x}_s - {}^{t}\mathbf{x}_m$. When this incremental rotation is applied to the master node of the rigid link, the vector ${}^{t}\mathbf{x}_{sm}$ changes to

$$ {}^{t+\Delta t}\mathbf{x}_{sm} = \mathbf{Q}(\Delta\boldsymbol{\theta})\,{}^{t}\mathbf{x}_{sm}\text{, where } \mathbf{Q}(\Delta\boldsymbol{\theta}) = \mathbf{I} + \frac{\sin\gamma}{\gamma}\mathbf{S} + \frac{1}{2}\left(\frac{\sin\frac{\gamma}{2}}{\frac{\gamma}{2}}\right)\mathbf{S}^2 $$

and $\mathbf{S} = \begin{bmatrix} 0 & -\Delta\theta_z & \Delta\theta_y \\ \Delta\theta_z & 0 & -\Delta\theta_x \\ -\Delta\theta_y & \Delta\theta_x & 0 \end{bmatrix}$, $\gamma = \sqrt{\Delta\theta_x^2 + \Delta\theta_y^2 + \Delta\theta_z^2}$. The above formula can be found in ref KJB, p 580, exercise 6.56.

This formula gives the exact answer for the case when the incremental rotations are finite. Physically, the incremental rotations are interpreted as a vector, with the direction of the vector interpreted as the axis of rotation, and the magnitude of the vector interpreted as the amount of rotation about that axis.

The same formula is applied to the end-node triad of a beam element local node, and to the director vector of a shell element local node.

5.6 Applied temperatures

- In Advanced Nonlinear Solution, temperature can be prescribed in any structural analysis. In addition, in heat transfer and TMC analyses, temperature can be prescribed at certain parts of the model, and the program will solve for the complete temperature field.

- Temperature can be applied directly to a node using the TEMP entry or to the whole model using the TEMPD entry. Direct nodal values applied with TEMP override the default TEMPD value. This is applicable to structural analysis, and to initial conditions for a heat transfer analysis.

- The TEMPERATURE case control command selects the initial or reference temperature field.

- In heat transfer analysis, temperature boundary conditions are applied using the TEMPBC entry or the SPC entry.

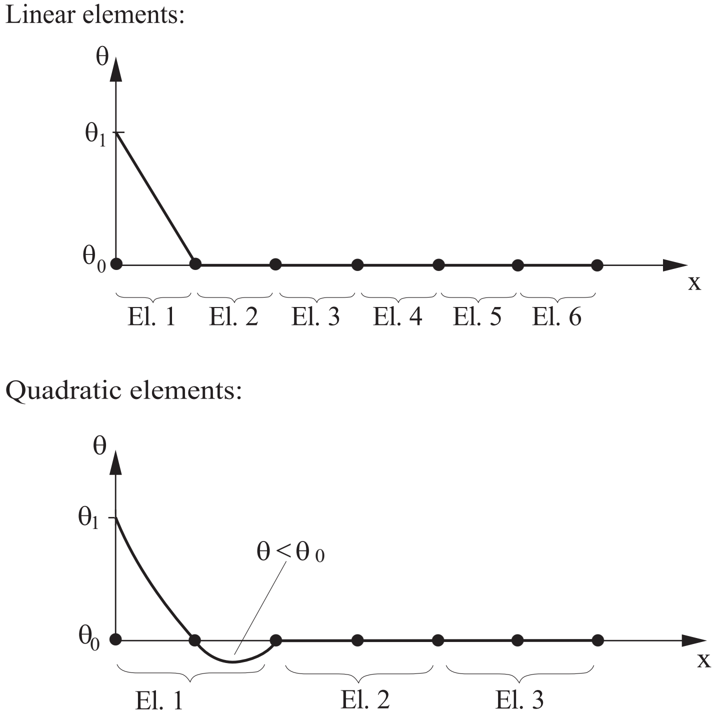
- A time-dependent temperature load can be applied by using the DTEMP bulk entry and referencing it in a DTEMP case control. In previous versions, a time-dependent temperature load can be defined with the TLOAD1 entry, and referencing the TLOAD1 in the DLOAD case control. This is still supported for backwards compatibility but it is recommended to use DTEMP instead. Note that DTEMP cannot be used with TEMP(LOAD) or DLOAD for temperature loading.

- It should also be noted that when using higher-order elements, the temperatures can be significantly different within the element than at the nodal points. For example, the temperature can be negative at points within an element, although the nodal point temperatures are all non-negative. This is illustrated in Fig. 5.6-1 (see element 1). This observation can be important when performing an analysis with temperature-dependent material properties.

- Applied temperature gradients cannot be specified on shell element nodes.

- A node cannot be specified more than once in a temperature set.

- Temperature loads cannot be specified with both TEMP(LOAD) and DLOAD case control commands. Exception: if TEMPOPT=1 in the NXSTRAT entry, then both TEMP(LOAD) and DLOAD are allowed. This option is provided for backwards compatibility with previous versions of NXN. However, it is recommended to specify only TEMP(LOAD) for a constant temperature load or DTEMP for a time-dependent temperature load in a model.



Linear elements:

Quadratic elements:

Fig. 5.6-1: Interpolation of temperature boundary conditions

5.7 Bolt preload

- Advanced Nonlinear Solution supports the preloading of bolt elements. The bolt preloads or forces are applied during extra solution steps performed at the very beginning of the analysis prior to the rest of the step-by-step solution.

- The bolt preloads are applied via the BOLTFOR entry which should be used together with a BOLTLD case control command.

- Bolt preload (and bolts in general) are only available in Solution 601.

- The BOLTSTP feature of the NXSTRAT entry allows the bolt preload to be applied over BOLTSTP solution steps (default is

BOLTSTP=1.

- See Section 10.7 for more details on the bolt feature.

5.8 Constraint equations

- Advanced Nonlinear Solution supports single-point and multipoint constraints. The single-point constraints are defined using the SPC, SPC1, SPCADD entries. Permanent single-point constraints can also be defined using the GRID entry. This case, however, totally removes the associated degree of freedom from the solution.

- Multipoint constraints are defined directly using the MPC and MPCADD entries. They can also be defined through R-type elements (see Section 2.7).

- The following relationship holds for multipoint constraints:

$$\sum_j R_j u_j = 0$$

- Constraints can be enforced in two ways. The first approach (called regular constraints approach) is for the first degree of freedom in each constraint (u_1) to be a dependent degree of freedom. The second approach (called general constraints approach) is to add a Lagrange multiplier to enforce each constraint, and hence keep all constraint degrees of freedom independent.

  Parameter GENMPC in the NXSTRAT entry sets the approach used for all constraint equations. GENMPC=0 selects regular constrants (the default) and GENMPC=1 selects general constraints.

- Note that in the regular constraints approach, each constraint reduces the number of independent equations by one, while in the general constraints approach, each constraint adds one extra degree of freedom (the Lagrange multiplier). Hence, the regular constraints approach should be used whenever possible. In some cases, however, one cannot easily express a constraint in a way such that dependent degrees of freedom are not constrained to other dependent degrees of freedom.

- As a simple example, the constraints given by the equations below cannot be expressed as regular constraints:

$$u_z^2 - u_z^1 - 3\theta_x^1 = 0 \qquad \leftarrow \text{valid regular constraint}$$

$$\underset{\substack{\uparrow \\ \text{This DOF is already dependent}}}{}\; u_z^3 - \underset{\substack{\uparrow}}{u_z^2} - u_z^1 + 4\theta_x^2 = 0 \qquad \leftarrow \text{invalid regular constraint}$$

in which the subscript gives the direction and the superscript gives the node number. These constraints can be expressed as general constraints.

However, the second constraint can be written as a regular constraint by manually applying the first constraint to obtain

$$u_z^3 - 2u_z^1 - 3\theta_x^1 + 4\theta_x^2 = 0$$

- Mesh glueing (Section 5.9) internally creates general constraint equations that are enforced using Lagrange multipliers. All independent degrees of freedom associated with the glued mesh remain independent.

- Regular constraints can be applied both in Solution 601 (in static and implicit dynamic analysis) and in Solution 701.

- General constraints cannot be used in Solution 701.

- Multipoint constraints are only approximately satisfied in an explicit analysis (Solution 701), since imposing the constraint exactly requires a non-diagonal mass matrix.

- For R-type elements in large deformation the multipoint constraint can have variable coefficients that are updated based on the deformation of the structure.

- Note that enforced displacements detailed in Section 5.5 are internally enforced using single-point constraints.

- For an R-type element to produce multipoint constraints with changing coefficients that capture large deformations, the constraints must be between only 2 nodes. In addition, one of the nodes should possess all the independent degrees of freedom and the other node should only possess dependent degrees of freedom. These large displacement multipoint constraints are internally called rigid links.

- The following comments apply to regular constraints of in which the dependent degree of freedom is constrained to more than one independent degree of freedom.

    - If a consistent mass matrix is used and concentrated masses are applied to constrained nodes, then the off-diagonal terms are included in the global mass matrix.
    - Similarly, if concentrated dampers are applied to constrained nodes, then the off-diagonal terms are included in the global damping matrix.

5.9 Mesh glueing

- The mesh glueing feature is used to attach two surfaces (or lines in 2-D) together. These two surfaces usually involve different finite element meshes (see Fig. 5.9-1). The glueing procedure results in a smooth transition of displacements and tractions between the glued surfaces. Mesh glueing sets are defined in the BGSET entry, and the glued surfaces are defined via the BEDGE, BSURFS or BCPROPS entries.

- This feature is useful for several applications:

    - When a fine mesh is desired in a certain region and coarser meshes are desired in other regions.
    - When different regions are meshed independently with unstructured free meshes.
    - When different regions are meshed with different element types (such as a tetrahedral mesh attached to a brick mesh).

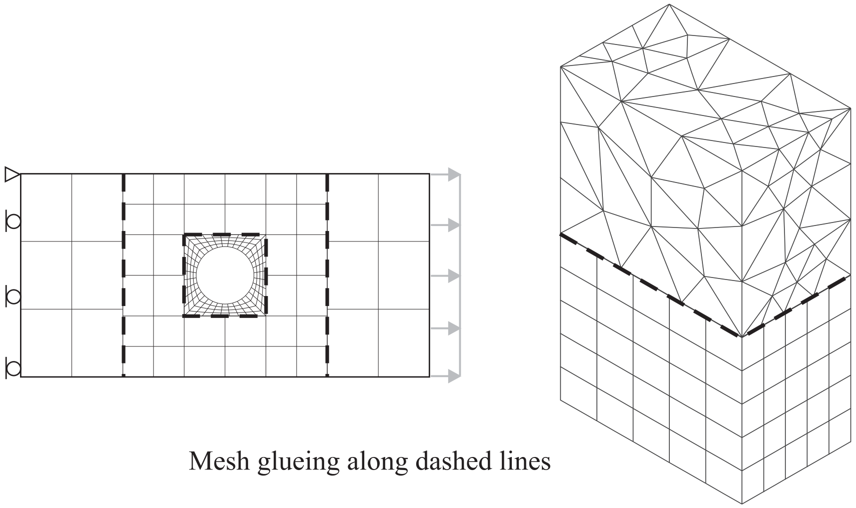


Fig. 5.9-1: Examples requiring mesh glueing

- The proper glueing constraint between the two surfaces can be expressed as

$$\int_{\Gamma} \lambda \cdot (u^1 - u^2) d\Gamma = 0 \qquad (5.9\text{-}1)$$

where u^1 is the displacement of the first glued surface, u^2 is the displacement of the second surface and λ is the Lagrange multiplier field imposing the constraint.

One of the glued surfaces is designated as the master and the other as the slave. The Lagrange multiplier field involves nodal degrees of freedom at the nodes of the slave surface, and the integration is also performed over the slave surface. Hence Eq. (5.9-1) becomes

$$\int_{\Gamma^S} \lambda^S \cdot (u^M - u^S) d\Gamma^S = 0 \qquad (5.9\text{-}2)$$

The accurate integration of Eq. (5.9-2) is not trivial since the displacements u^M and u^S are generally interpolated over different domains. This integration is automatically performed by Advanced Nonlinear Solution.

- Mesh glueing is not available in Solution 701.

- Glueing is superior to tied contact and should be used in its place whenever applicable.

- Only 2-D solid elements can be used in the gluing of 2-D lines. The glued element side can have linear or quadratic displacement interpolation. Rod and beam elements are not supported.

- Only 3-D solid elements can be used in the gluing of 3-D surfaces. The glued element faces can be triangles or quads, and they can have linear or quadratic displacement interpolation. Shell elements are not supported.

- Nodes on glued surfaces (both master or slave) cannot have dependent translation degrees of freedom. Therefore, they cannot be slaves in multi-point constraints involving translations.

- If one glue surface is smaller than the other, the smaller surface should preferably be the slave. However, the glueing will also work if the smaller surface is the master. The two glued surfaces can also be partially overlapping. These cases are shown in Fig. 5.9-2.

- The two glued surfaces should ideally be smooth surfaces (no sharp corners). If corners exist it is better to create multiple glued meshes, as shown in Fig. 5.9-3.

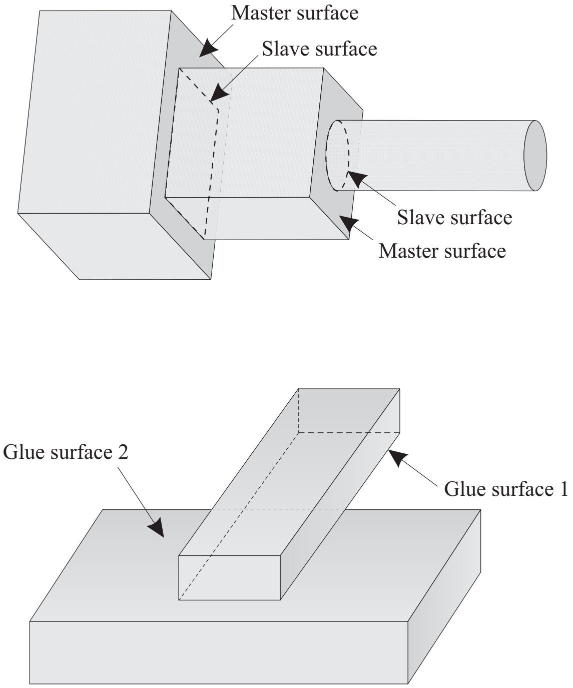


Fig. 5.9-2: Examples of master and slave glue surfaces

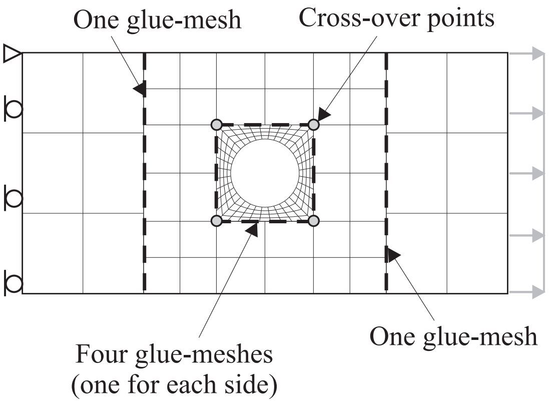


Fig. 5.9-3: Glueing non-smooth surfaces

- The Lagrange Multiplier field is modified at nodes where multiple glue meshes intersect. These nodes are called cross-over points.

- If the two surfaces have different mesh densities, either one can be used as slave. However, it is recommended that the finer mesh surface be the slave surface. Using the finer meshed surface as a slave will produce more equations, since the Lagrange multipliers degrees of freedom are on the nodes of the slave surface.

- The master glue surface can be enlarged beyond its geometric bounds, so that the slave points that project slightly outside the master can still be considered glued. This is done via the EXTi parameter in the BGSET card.

- If the independent parts of the model have rigid body motions (not considering the effect of glueing), the equation solver might stop due to zero pivots. In this case, stiffness stabilization can be employed (see Section 10.6).

- The mesh glue feature admits rigid body translations. Therefore, when the model is rigidly translated, no spurious stresses and strains are generated, and force equilibrium is satisfied between the glued surfaces.

- The mesh glue feature only admits rigid body rotations when the two glued surfaces exactly coincide, that is, when the two glued surfaces are flat. Curved surfaces do not exactly coincide due to the discretization error, as illustrated in Fig. 5.9-4.

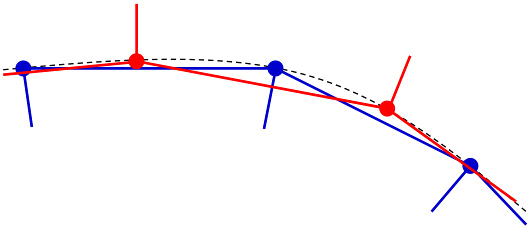
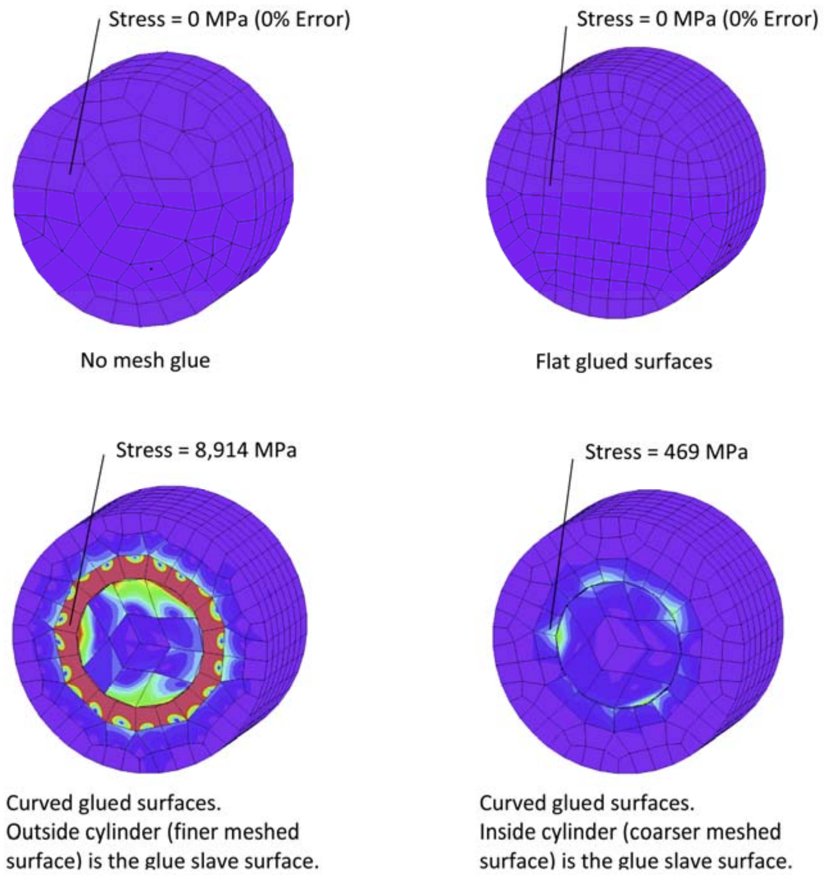


Fig. 5.9-4: Glued surfaces do not exactly coincide when the surfaces are curved.

As a result, if the glued surfaces are curved, moment equilibrium is not satisfied between the glued surfaces, spurious stresses and strains are generated when the model rotates, and rotational motions are resisted, just as if the glued surfaces are attached to a grounded rotational spring. The stiffer the glued surfaces, the stiffer the rotational spring. This effect is less severe when the coarser meshed surface is the slave surface of the glue mesh pair because in this case the surfaces are not as “tightly” glued (fewer glueing constraints are created), and hence the distortional motion imposed by the mesh glue is less severe.

Fig. 5.9-5 shows the stress results for a large displacement static analysis where a steel cylinder of radius 1mm is rotated by 10 degrees by prescribing the rotation on the outside surface of the cylinder. Other than the prescribed rotation, the cylinder is completely free. We observe that the exact solution is obtained when there is no mesh glue and when the glued surfaces are flat, but spurious stresses are generated when the glued surfaces are curved, especially when the finer meshed surface is the slave surface in the mesh glue pair.



No mesh glue

Flat glued surfaces

Curved glued surfaces.
Outside cylinder (finer meshed
surface) is the glue slave surface.

Curved glued surfaces.
Inside cylinder (coarser meshed
surface) is the glue slave surface.

Fig. 5.9-5: Stress results when cylinder is rotated 10 degrees.

5.10 Convection boundary condition

- Convection boundary conditions take the following form

$$q^S = h\left(\theta_e - \theta^S\right)$$

where h is the convection coefficient, θ_e is the external ambient temperature, and θ^S is the unknown body surface temperature.

- Convection boundary conditions are applied using surface elements generated using the CHBDYE or CHBDYG entities which point to a CONV entry. This entry provides some of the necessary inputs and in turn points to a PCONV entry providing more input. The heat convection coefficient *h* is provided in the material definition entries.

- The following types of convection boundary conditions are available:

  Line convection boundary conditions, used in conjunction with 2-D planar elements and 2-D axisymmetric elements.

  Surface convection boundary conditions, used in conjunction with 3-D solid or shell elements.

- The convection coefficient *h* can be either temperature-dependent (through a MATT4 or MATT5 entry), or time-dependent. This is achieved via the Control Node setting in the CONV entry. It cannot, however, be both temperature and time dependent.

- The ambient temperature θ_e is obtained from node using parameter TA1 in the CONV entry. The temperature at node TA1 must be prescribed, and can be time-varying.

- The heat flux, q^S, is converted to nodal heat fluxes by consistent integration over the convection boundary. See ref. KJB, Section 7.2.3 for details. In this integration, the temperatures are interpolated from their nodal values, and if the heat transfer coefficient, *h*, is temperature dependent, it is calculated for each integration point based on its interpolated temperatures.

5.11 Radiation boundary condition

- Radiation boundary conditions take the following form

$$q^S = \sigma f e\left(\theta_r^4 - \left(\theta^S\right)^4\right)$$

where σ is the Stefan-Boltzmann constant, *f* is a view factor or shape factor, *e* is the material emissivity, θ_r is the temperature of the radiative source (or sink) and θ^S is the unknown body surface temperature. Both temperatures are in the absolute scale. Note that in the above equation the absorptivity is assumed to be equal to the emissivity.

- Radiation boundary conditions are applied using surface elements generated using the CHBYDE or CHBYDG entries which point to a RADBC entry. This entry provides some of the necessary inputs and in turn points to a RADM or RADMT entry providing the rest of the inputs.

- The Stefan-Boltzmann constant (σ) and the absolute temperature offset are set in PARAM entries (SIGMA and TABS parameters). Note that although σ is a constant it must be input in the proper units.

- The following types of radiation boundary conditions are available:

  Line radiation boundary conditions, used in conjunction with 2-D planar elements and 2-D axisymmetric elements.

  Surface radiation boundary conditions, used in conjunction with 3-D solid or shell elements.

- The emissivity coefficient *e* can also be temperature-dependent by using the RADMT entry.

- The radiative source/sink temperature θ_r is specified in the NODAMB parameter in the RADBC entry. The temperature at this node NODAMB must be prescribed, and can be time-varying.

- The view or shape factor *f* is input via the FAMB parameter in the RADBC entry.

- Default values of some radiation settings are defined using the BDYOR entry.

- The heat flux, q^S, is converted to nodal heat fluxes by consistent integration over the radiation boundary. See ref. KJB, Section 7.2.3 for details. In this integration, the temperatures are interpolated from their nodal values, and if the emissivity, *e*, is temperature dependent, it is calculated for each integration point based on its interpolated temperatures.

5.12 Boundary heat flux load

- Applied boundary heat flux is specified by equation (8.1-3):

ref. KJB Section 7.2.1

$$k_n \left.\frac{\partial \theta}{\partial n}\right|_{S_2} = q^S$$

where q^S is the surface heat flux input to the body across some part S_2 of the body surface, k_n is the body thermal conductivity in direction *n*, the outward normal to the surface, and θ is the temperature.

- Boundary heat flux loads are applied either directly to the nodes defining a face of an element using the QHBDY entry, or by pointing to existing surface elements (CHBDY$_{\mathrm{i}}$ type) using the QBDY1 or QBDY2 entries.

- The heat flux, q^S, is converted to nodal heat fluxes by consistent integration over the boundary. See ref. KJB, Section 7.2.3 for details.

- Note that any boundary of the domain which does not have either the heat flux or temperature specified will be assumed by virtue of the formulation to have

$$q^S = 0$$

i.e., this part of the boundary is insulated, allowing no heat transfer across it.

5.13 Internal heat generation

ref. KJB Section 7.2.3

- This form of thermal loading results from the generation of heat within the domain, which is introduced into the governing equation system by the term q^B of equation (8.1-1).

- Internal heat generation is applied via the QVOL entry which provides a load multiplier to the heat generation parameter HGEN set in the MAT4 or MATT4 entries.

- A negative heat generation term q^B indicates a loss of heat within the body.

- The heat generation term can be temperature-dependent by making the HGEN parameter temperature dependent using the MATT4 entry.

- The heat flux generated per unit volume, q^B, is converted to nodal heat fluxes by consistent integration over the element volume. See ref. KJB, Section 7.2.3 for details.

6. Static and implicit dynamic analysis

This chapter presents the formulations and algorithms used to solve static and dynamic problems using Solution 601. This includes convergence checking and the available solvers. Most flags or constants that need to be input in this chapter are in the NXSTRAT bulk data entry.

Information about the progress of the solution is always output to the .f06 file. A shorter summarized output is provided in the .log file.

6.1 Linear static analysis

ref. KJB Sections 8.2.1, 8.2.2 and 8.2.3

- In linear analysis using Solution 601, the finite element system equilibrium equations to be solved are

$$\mathbf{KU} = \mathbf{R}$$

- A direct sparse solver, iterative multigrid solver or 3D iterative solver can be used to solve this system of equations, see Section 6.5.

- The equation solvers assume that the system stiffness matrix is symmetric.

- The equation solvers assume that the system stiffness matrix is positive definite. This requirement can be summarized as follows: The Rayleigh quotient

$$\rho(\boldsymbol{\phi}) = \frac{\boldsymbol{\phi}^T \mathbf{K}\boldsymbol{\phi}}{\boldsymbol{\phi}^T \boldsymbol{\phi}}$$

must be greater than zero for any displacement vector $\boldsymbol{\phi}$. Since $\rho(\boldsymbol{\phi})$ is equal to twice the strain energy stored in the system (for $\boldsymbol{\phi}^T \boldsymbol{\phi} = 1$), this is equivalent to the requirement that the strain energy stored in the finite element system when subjected to any displacement vector $\boldsymbol{\phi}$ must be greater than zero.

- Hence, the finite element system must be properly supported, so that the system cannot undergo any rigid body displacements or rotations.

- It also follows that no part of the total finite element system must represent a mechanism, see ref. KJB, Fig. 8.7, p. 704, for such a case.

- Nodal point degrees of freedom for which there is no stiffness must be restrained. A degree of freedom does not carry any stiffness if all of the elements connected to the nodal point do not carry stiffness into that degree of freedom. In this case the degree of freedom must be restrained using boundary conditions.

  Note that nodal degrees of freedom which are not connected to any elements and are not used as dependent nodes in constraint equations are automatically deleted by the program.

- The elements joining into a nodal point and contributing to the stiffness of the nodal point degrees of freedom must all be defined properly and in a physically correct manner. For example, if there are program input errors that yield a zero or negative Young's modulus or an incorrect nodal point numbering for an element, the stiffness at a system degree of freedom may be zero or negative.

- More details on the solvers available in Solution 601 are provided in Section 6.5.

6.2 Nonlinear static analysis

ref. KJB Section 8.4

- In nonlinear static analysis the equilibrium equations to be solved are:

$$^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F} = \mathbf{0}$$

where $^{t+\Delta t}\mathbf{R}$ is the vector of externally applied nodal loads at time (load) step $t+\Delta t$, and $^{t+\Delta t}\mathbf{F}$ is the force vector equivalent (in the virtual work sense) to the element stresses at time $t+\Delta t$.

- The nonlinearity may come from the material properties, the kinematic assumptions, deformation dependent loading, the presence of contact, or the use of special features such as the element birth/death feature.

- The solution to the static equilibrium equations can be obtained in Solution 601 using

  - Full Newton iterations, with or without line searches
  - Automatic step incrementation (automatic-time-stepping and load-displacement-control methods)

These methods are described in detail in the following sections and also in Sections 6.1 and 8.4 of ref. KJB.

- The same equation solvers are used for both linear and nonlinear analysis. However, the automatic time stepping algorithms do not require the stiffness matrix to be positive definite, thus allowing for the solution of bifurcation problems.

- A special case is reached when the element birth or death option is used for some elements. In this case the equation solver allows that during the solution a degree of freedom may not carry any stiffness at a particular time, because the adjoining element(s) may not have been born yet (or may have died already). Note that the program does not check if any load is applied in that degree of freedom, and that the appropriate time functions should be used to generate/delete that load according to the element birth or death.

- The stiffness stabilization feature can be used to treat some nonlinear static problems involving a non-positive definite stiffness matrix. See Section 10.6 for details.

6.2.1 Solution of incremental nonlinear static equations

Full Newton iterations: In the full Newton iteration method, the following algorithms are employed:

- Without line search

$$ {}^{t+\Delta t}\mathbf{K}^{(i-1)}\,\Delta\mathbf{U}^{(i)} = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(i-1)} \qquad (6.2\text{-}1a)$$

$$ {}^{t+\Delta t}\mathbf{U}^{(i)} = {}^{t+\Delta t}\mathbf{U}^{(i-1)} + \Delta\mathbf{U}^{(i)} \qquad (6.2\text{-}1b)$$

- With line search

$$ {}^{t+\Delta t}\mathbf{K}^{(i-1)}\,\Delta\mathbf{U}^{(i)} = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(i-1)} \qquad (6.2\text{-}1c)$$

$$ {}^{t+\Delta t}\mathbf{U}^{(i)} = {}^{t+\Delta t}\mathbf{U}^{(i-1)} + \beta^{(i)}\Delta\mathbf{U}^{(i)} \qquad (6.2\text{-}1d)$$

where ${}^{t+\Delta t}\mathbf{K}^{(i-1)}$ is the tangent stiffness matrix based on the solution calculated at the end of iteration (i - 1) at time t+Δt, ${}^{t+\Delta t}\mathbf{R}$ is the externally applied load vector at time $t + \Delta t$; ${}^{t+\Delta t}\mathbf{F}^{(i-1)}$ is the consistent nodal force vector corresponding to the element stresses due to the displacement vector ${}^{t+\Delta t}\mathbf{U}^{(i-1)}$; $\Delta\mathbf{U}^{(i)}$ is the incremental displacement vector in iteration (i) and $\beta^{(i)}$ is an acceleration factor obtained from line search. Note that, since the full Newton iteration method is employed, a new stiffness matrix is always formed at the beginning of each new load step and in each iteration.

- An upper bound for the incremental displacements in Δ**U** can be set by the user (via the MAXDISP parameter in the NXSTRAT entry). If the largest increment displacement component exceeds the limiting value, the whole incremental displacement vector is scaled down to satisfy the upper bound.

This feature is useful for problems where one or more iterations can produce unrealistically large incremental displacements. This may happen, for example, if a load is applied to contacting bodies before contact is properly established, or in the first unloading steps after a material has undergone plastic deformation.

The default (MAXDISP=0.0) is

- Dynamic analysis or analysis without contact: no limit on incremental displacements
- Static analysis with contact: maximum incremental displacement is 1% of the maximum model dimension.

6.2.2 Line search

The line search feature is activated by setting LSEARCH=1 in NXSTRAT. In this case, the incremental displacements obtained from the solver are modified as follows

$$^{t+\Delta t}\mathbf{U}^{(i)} = {}^{t+\Delta t}\mathbf{U}^{(i-1)} + \beta^{(i)}\Delta\mathbf{U}^{(i)}$$

where β is a scaling factor obtained from a line search in the direction $\Delta\mathbf{U}^{(i)}$ in order to reduce out-of-balance residuals, according to the following criterion

$$\frac{\Delta\mathbf{U}^{(i)^T}\left[{}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(i)}\right]}{\Delta\mathbf{U}^{(i)^T}\left[{}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(i-1)}\right]} \le \text{STOL} \qquad (6.2\text{-}2)$$

where STOL is a user-input line search convergence tolerance (in NXSTRAT), and $^{t+\Delta t}\mathbf{F}^{(i)}$ is calculated using the total displacement vector $^{t+\Delta t}\mathbf{U}^{(i)}$.

The magnitude of β is also governed by the following bounds

$$LSLOWER \le \beta \le LSUPPER \qquad (6.2\text{-}3)$$

where LSUPPER and LSLOWER are user-input parameters in NXSTRAT.

The incremental displacements are not modified (i.e., $\beta = 1$) if no suitable line search parameter satisfying Equations (6.2-2) and (6.2-3) is found within a reasonable number of line search iterations, or if the unbalanced energy falls below a certain user-specified energy threshold ENLSTH.

- Line search is off by default. It is useful for problems involving plasticity, as well as large displacement problems involving beams and shells. It is also helpful in many contact problems. In the case of contact problems, it is sometimes better to set LSUPPER to 1.0 so that the line search only scales down displacements.

- The effect of line search is more prominent when the current displacements are far from the converged solution. This usually happens in the first few iterations of a time step, or when a major change occurs in the model, due for example, to contact initiation/separation, or the onset of plasticity.

- Note that line search increases the computational time for each iteration. Most of the extra time goes towards the evaluation of $^{t+\Delta t}\mathbf{F}^{(i)}$ in Equation (6.2-2). However, for the types of problems mentioned above the reduction in the number of iterations and the ability to use bigger time steps leads to an overall reduction in solution time.

6.2.3 Low speed dynamics feature

- A low speed dynamics option is available for static analysis (can only be used with ATS or TLA/TLA-S features). It is activated with ATSLOWS in the NXSTRAT entry. Low speed dynamics is a special technique developed to overcome convergence difficulties in collapse, post-collapse and certain contact problems.

  In essence, this feature includes dynamics effects in an otherwise static problem. Solution 601 solves

$$\delta \mathbf{M} \; {}^{t+\Delta t}\ddot{\mathbf{U}}^{(i)} + \mathbf{C} \; {}^{t+\Delta t}\dot{\mathbf{U}}^{(i)} + {}^{t+\Delta t}\mathbf{K}^{(i-1)} \, \Delta\mathbf{U}^{(i)} = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(i-1)} \quad (6.2.4)$$

where **M** is the mass matrix and δ is a mass scaling factor that can vary from 0.0 to 1.0 (default 1.0), to partially account for the dynamic inertia effect . The **C** matrix is evaluated using

$$\mathbf{C} = \beta \mathbf{K}$$

where β is a user-specified parameter (default 10^{-4}), and $\mathbf{K}$ is the (initial) total stiffness matrix (corresponding to zero initial displacements). For more details on this dynamics equation refer to Section 6.4. The δ and β parameters are set via the ATSMASS and ATSDAMP parameters in the NXSTRAT entry.

- When low speed dynamics is used with ATS, the time step size will influence the results. It is recommended that the time step size be at least $10^5\beta$. Or, it is recommended that the loads be held constant for a period of time of at least 10β so that the dynamic effects die out.

- Notice that when $\mathbf{C} = \beta \mathbf{K}$, rigid body motions are not damped out. In order to achieve damping of rigid-body motions, $\mathbf{C} = \alpha \mathbf{M} + \beta \mathbf{K}$ must be used.

- Note that mass effects may not be needed in the low speed dynamic analysis. In this case, set the δ parameter in Eq. (6.2-4) to zero, or, alternatively, set the material densities to zero. That way, only structural damping effects will be present in the otherwise static analysis.

- The program outputs the damping and inertia force norms at every time step. The damping and inertia forces should be compared with the external forces in order to insure that they are not excessive.

6.2.4 Automatic-Time-Stepping (ATS) method

- The automatic-time-stepping (ATS) method controls the time step size in order to obtain a converged solution. It is activated with AUTO=1 in the NXSTRAT entry (which is the default). If there is no convergence with the user-specified time step, the program automatically subdivides the time step until it reaches convergence. In some cases, the time step size may be increased to accelerate the solution.

- Parameter ATSDFAC in the NXSTRAT entry sets the division factor that Solution 601 uses to subdivide the time step when there is no convergence. Successive subdivisions can be performed, if necessary, provided that the time step size is not smaller than a minimum value. This minimum value is set as the original step size divided by a scaling factor provided by the user (ATSSUBD in NXSTRAT).

- Note that the loads at any intermediate time instant created by the ATS method are recalculated based on the current value of the time functions.

- The solution output is only furnished at the user-specified times, except when the solution is abandoned due to too many time step subdivisions without convergence. In this case, the solution output is also given for the last converged time instant.

- There are three options for controlling the time step size once convergence is reached after ATS subdivisions. Any of these options can be selected, or Solution 601 can make the selection automatically (ATSNEXT in NXSTRAT).

1. Use the time increment that gave convergence
In this case, the program continues to use the reduced (subdivided) time step size that gave convergence for the number of consecutive substeps specified by the NXSTRAT ATSNSUB parameter if ATSNSUB is greater than zero (default, ATSNSUB=0). Once this number of consecutive substeps is reached, or once the end of the user-specified time step is reached, the program might increase the time step size based on the iteration history, but the program ensures that the analysis always proceeds through all user-specified solution times. This option is the default in analyses without contact.

2. Return to original time step size
In this case, the program continues the analysis using the original user-provided time increment. This option is the default when contact is present.

3. Proceed through user-defined time points
In this case, the program sets the time step size such that the final time is that initially provided by the user. Hence, the analysis always proceeds through all user-specified time steps.

The program automatically switches to option 3 if, when option 2 is specified, one of the following features is used: Bathe method of time integration or TMC coupling.

- The ATS method can also increase the time step beyond the user-specified value if the iteration history shows that such an increase is useful. This is only possible in a static analysis without low-speed dynamics. The increase in time step cannot be larger than a user-specified factor (via ATSMXDT in NXSTRAT). Due to this increase, the analysis may be completed in fewer time steps than requested. This time step increase is only possible when the ATS subdivision is set to “Return to original time step size”.

- Following is an example to illustrate the basic options of ATS subdivisions. Assume we are in load step 15 of a particular problem with initial time t = 15.0 and a time step of 1.0. The solution does not converge and the time step is set to 0.5 (assuming a time step division factor of 2.0). If that too does not converge, the time step is set to 0.25. If that converges, the results are not yet saved. Another sub-step is performed for load step 15. The size of this step depends on which of the three options above is selected:

- In option 1, the next sub-step will use a time step size of 0.25. Two other sub-steps will be performed within load step 15 both of size 0.25 (assuming they all converge).
- In option 2, the next sub-step will use a time step size of 1.0. If this converges load step 16 starts with t = 16.25.
- In option 3, the sub-step will use a time step size of 0.75. If this converges load step 16 starts with t = 16.0.

- Note that while options 1 and 2 may result in outputted solution times that are different from those initially specified by the user, there are certain time values that are not skipped. These are the time values at the end of “time step blocks”. In this case, the time step size is reduced such that the solution time at the end of the block is satisfied. The program determines these time step blocks based on the time step pattern input by the user. The final solution time is always assumed to be an end of a time step block.

- Option 2 is useful for contact because of the highly nonlinear response (sudden changes in stiffness) that occurs when a node comes into contact, or is released from contact, or even moves from one contact segment to another. A small step size may sometimes be needed only in the vicinity of this contact incident. Once contact is established/released the problem is “less nonlinear” and the original time step size can be used.

- Whenever the ATS method is used, the program always checks if a time function point could be skipped for a time step used, and makes an appropriate adjustment to the magnitude of the time step if necessary. Consider the time function shown in Fig. 6.2-1.

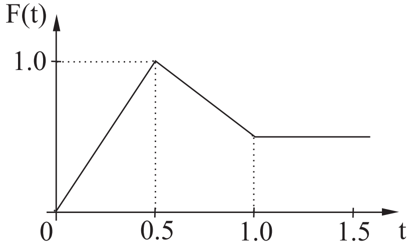


Fig. 6.2-1: Time function illustrating ATS program checking

With a time step equal to 0.2, assuming that convergence is obtained, the program will provide solutions for times t = 0.2 and t = 0.4. If the time step of 0.2 is used again, then the maximum time function value at time t = 0.5 will be missed. Therefore, the program uses two substeps of value 0.1 to get to time t = 0.6.

6.2.5 Total Load Application (TLA) method and Stabilized TLA (TLA-S) method

- The Total Load Application method is useful for nonlinear static analysis problems where all applied loads do not require the user to explicitly specify the time step sequence. It is activated with AUTO=3 in the NXSTRAT entry. In this case, the user applies the full load value and Solution 601 automatically applies the load through a ramp time function, and increases or reduces the time step size depending on how well the solution converges. This method cannot be used in dynamic analysis.

The TLA method automatically activates the following features that are suitable for this type of uniform loading static problems:

- The first time step has a size of 1/50 of the total time. May be modified by TLANSTP in NXSTRAT.
- Maximum number of equilibrium iterations is 30. May be modified by TLAMXIT in NXSTRAT.
- Line search is used.
- Limiting incremental displacements per iteration is set to 5% of the largest model dimension. May be modified by TLAMXDF in NXSTRAT.
- The maximum number of time step subdivisions is set to 64.

- The time step cannot be increased more than 16 times its initial size.

- The Stabilized TLA method (TLA-S) is identical to the regular TLA method with the addition of various stabilizing features to create a more stable system and aid convergence. The TLA-S method is activated with AUTO=4 in the NXSTRAT entry. The sources of stabilization are low speed dynamics which adds inertia and stiffness proportional damping (see Section 6.2.3), contact damping (see Section 4.6.5), and stiffness stabilization (see Section 10.6). The amount of stiffness stabilization, low speed dynamics inertia and damping, and contact damping can be adjusted by the parameters TLASTBF, TLALSMF, TLAMSDF and TLACTDF in the NXSTRAT entry. Indicators are provided in the output file after each converged solution to show if the forces due to the various stabilization effects are excessive.

The following solution indicators are printed:

- external forces

$$I_{EF} = \mathbf{R} \bullet \Delta\mathbf{U}$$

where $\mathbf{R}$ is an external load vector, and $\Delta\mathbf{U}$ is a displacement vector in a given step.

- damping forces

$$I_{DF} = \mathbf{R}_D \bullet \Delta\mathbf{U}$$

where $\mathbf{R}_D$ is a damping force vector.

- bolt forces

$$I_{BOLT} = \mathbf{R}_{BOLT} \bullet \Delta\mathbf{D}_{BOLT}$$

where $\mathbf{R}_{BOLT}$ are bolt forces and $\Delta\mathbf{D}_{BOLT}$ are bolt shortenings.

- inertia forces

$$I_I = \mathbf{R}_I \bullet \Delta\mathbf{U}$$

where $\mathbf{R}_I$ are inertia forces.

- contact damping forces

$$I_{CD} = \mathbf{R}_{CD} \bullet \Delta\mathbf{U}$$

where $\mathbf{R}_{CD}$ are contact damping forces.

- stiffness stabilization

$$I_{SST} = \left(\tilde{\mathbf{K}} \bullet \Delta\mathbf{U}\right) \bullet \Delta\mathbf{U}$$

where $\tilde{\mathbf{K}}$ is the extracted stabilization part from the stiffness matrix.

- drilling forces (applicable to shell elements only)

$$I_{DR} = \mathbf{R}_{DR} \bullet \Delta\mathbf{U}$$

where $\mathbf{R}_{DR}$ are drilling forces.

Note that during bolt iterations, all indicators are referenced against the bolt force indicator. For step-by-step solutions, external force indicators are used for references.

- The TLA-S method can serve several purposes. If the stabilization indicators are all within reasonable bounds, typically less than 1% of the external force indicator, then the TLA-S solution may be reasonably accurate. However, even when the indicators are large, the TLA-S method provides a useful approximate solution that can frequently be used to detect modeling errors such as incorrect contact definition, applied load, or boundary conditions.

- The following features cannot be used with the TLA and TLA-S methods:

  - All materials with creep effects
  - All materials with viscosity effects
  - The temperature-dependent multilinear plastic material
  - Rigid target contact

6.2.6 Load-Displacement-Control (LDC) method

- The load-displacement-control (LDC) method (arc length method) can be used to solve for the nonlinear equilibrium path of a model until its collapse. If desired, the post-collapse response of the model can also be calculated. The main feature of the method is that the level of the externally applied loads is adjusted automatically by the program.

  The LDC method can only be used in nonlinear static analysis in which there are no temperature or creep effects. The LDC method can be used in contact problems.

- The formulations used in the LDC method used in Solution 601 are described in ref. KJB Section 8.4.3 and the following reference:

  ref. Bathe, K.J. and Dvorkin, E.N., "On the Automatic Solution of Nonlinear Finite Element Equations," *J. Computers and Structures*, Vol. 17, No. 5-6, pp. 871-879, 1983.

- The LDC method is activated via the AUTO flag in the NXSTRAT entry. An enforced displacement on a user-specified degree of freedom is used to evaluate the initial load vector, and analysis continues until a specified displacement is reached at a certain node, or a critical point on the equilibrium path is reached. Variants of the LDC method are commonly referred to as arc-length methods.

- The equations employed in the equilibrium iterations are

$$
\begin{aligned}
{}^{t+\Delta t}\mathbf{K}^{(i-1)}\,\Delta\mathbf{U}^{(i)} &= \left({}^{t+\Delta t}\lambda^{(i-1)} + \Delta\lambda^{(i)}\right)\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(i-1)} \\
{}^{t+\Delta t}\mathbf{U}^{(i)} &= {}^{t+\Delta t}\mathbf{U}^{(i-1)} + \Delta\mathbf{U}^{(i)} \qquad (6.2\text{-}5) \\
f\left(\Delta\lambda^{(i)}, \Delta\mathbf{U}^{(i)}\right) &= 0
\end{aligned}
$$

where

$^{t+\Delta t}\mathbf{K}^{(i-1)}$ = tangent stiffness matrix at time $t+\Delta t$, end of iteration (i-1)

$\mathbf{R}$ = constant reference load vector

$^{t+\Delta t}\lambda^{(i-1)}$ = load scaling factor (used on **R**) at the end of iteration (i-1) at time $t+\Delta t$

$\Delta\lambda^{(i)}$ = increment in the load scaling factor in iteration (i)

The quantities $^{t+\Delta t}\mathbf{F}^{(i-1)}$ and $\Delta\mathbf{U}^{(i)}$ are as defined for Eq. (6.2-1).

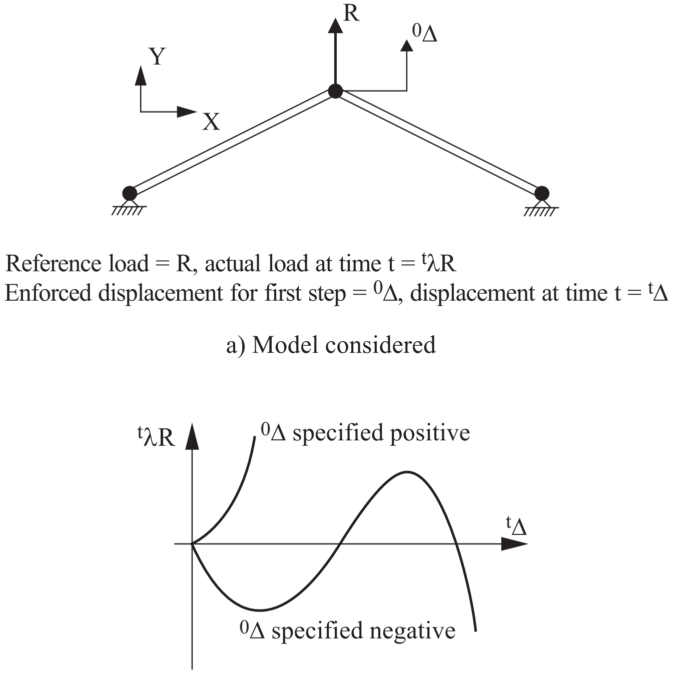
Note that in Eq. (6.2-5), the equation $f = 0$ is used to constrain the length of the load step. The constant spherical arc length constraint method is usually used and the constant increment of external work method is used if the arc length method has difficulties to converge.

The reference load vector **R** is evaluated from all the mechanical loads (except for the enforced displacement loads).

- To start the LDC method, the load multiplier for the first step $^{\Delta t}\lambda$ (used to obtain the corresponding load vector $^{\Delta t}\lambda$ **R**) is calculated using a user-specified enforced displacement (LDCDISP) acting on a given degree of freedom (LDCDOF) on a specific node (LDCGRID). All parameters are in the NXSTRAT entry. The direction of the displacement is given by its sign.

The degree of freedom is relative to the displacement coordinate system specified by LDCGRID.

As shown in Fig. 6.2-2, the input for the initial enforced displacement (in particular whether it is positive or negative) is critical in establishing successive equilibrium positions using the LDC method.


R
${}^0\Delta$
Y
X
Reference load = R, actual load at time t = ${}^t\lambda R$
Enforced displacement for first step = ${}^0\Delta$, displacement at time t = ${}^t\Delta$
${}^t\lambda R$
${}^0\Delta$ specified positive
${}^t\Delta$
${}^0\Delta$ specified negative

a) Model considered

b) Equilibrium paths

Fig. 6.2-2: Example of the dependence of solution path on the displacement enforced in the first step for the LDC method

As an example, two entirely different solution paths will be obtained for the same model shown in Fig. 6.2-2 if initial displacements of different signs are enforced for the first solution step.

- After the first step, the program automatically traces the nonlinear response by scaling the external load vector **R** proportionally, subject to the constraint of Eq. (6.2-5), so that at any discrete time *t* in iteration (*i*), the external load vector is ${}^t\lambda^{(i)}\mathbf{R}$.

The scaling of the reference load vector using ${}^t\lambda$ is analogous to the scaling of the applied loads **R** using a user predefined time function when the LDC method is not used (see Chapter 5). In the case of the LDC method, the scaling function is determined internally by the program instead of being user-specified.

- The converged displacement must satisfy the following relation:

$$\left\|\mathbf{U}\right\|_2 \le 100\alpha \left\| {}^{\Delta t}\mathbf{U}\right\|_2$$

where $\mathbf{U} = {}^{t+\Delta t}\mathbf{U} - {}^{t}\mathbf{U}$ is the incremental displacement vector for the current step, α is a displacement convergence input factor (LDCIMAX parameter in NXSTRAT), and ${}^{\Delta t}\mathbf{U}$ is the displacement vector obtained in the first step. If the above inequality is not satisfied, an internal restart of the iteration for the current step is performed by the program.

- The LDC solution terminates normally when any one of the following conditions is satisfied:

  ▸ The maximum specified displacement is reached (LDCDMAX in NXSTRAT).
  ▸ A critical point on the equilibrium path has been passed. If LDCCONT=1 in NXSTRAT is specified, this condition is skipped.
  ▸ The number of converged solution steps is reached.
  ▸ The maximum number of subdivisions (LDCSUBD in NXSTRAT) has been attempted using different strategies but each time the solution has failed to converge within the number of allowed iterations.

6.2.7 Convergence criteria for equilibrium iterations

- The following convergence criteria can be specified in Solution 601 (via CONVCRI in NXSTRAT):

  ▸ energy only,
  ▸ energy and force/moment,
  ▸ energy and translation/rotation,
  ▸ force/moment only, and
  ▸ translation/rotation only.

- If contact is defined in an analysis, the contact force convergence criterion is always used in addition to the above criteria (see Chapter 4).

- The values of all convergence norms, whether used or not, are provided in the .f06 file. For more details on the .f06 output format see Section 6.2.9.

LDC method not used

- If the LDC method is not used, the convergence in equilibrium iterations is reached when the following inequalities are satisfied:

Energy convergence criterion
For all degrees of freedom:

$$\frac{\Delta\mathbf{U}^{(i)^T}\left[{}^{t+\Delta t}\mathbf{R}-{}^{t+\Delta t}\mathbf{F}^{(i-1)}\right]}{\Delta\mathbf{U}^{(1)^T}\left[{}^{t+\Delta t}\mathbf{R}-{}^{t}\mathbf{F}\right]}\le \text{ETOL} \qquad (6.2\text{-}6)$$

where ETOL is a user-specified energy convergence tolerance.

Force and moment convergence criteria
For the translational degrees of freedom:

$$\frac{\left\|{}^{t+\Delta t}\mathbf{R}-{}^{t+\Delta t}\mathbf{F}^{(i-1)}\right\|_2}{\text{RNORM}}\le \text{RTOL}$$

For the rotational degrees of freedom:

$$\frac{\left\|{}^{t+\Delta t}\mathbf{R}-{}^{t+\Delta t}\mathbf{F}^{(i-1)}\right\|_2}{\text{RMNORM}}\le \text{RTOL}$$

where RTOL is a user-specified force convergence tolerance, RNORM and RMNORM are user-specified reference force and moment norms. If left undefined the program automatically determines RNORM and RMNORM during execution.

Translation/rotation convergence criteria
For the translational degrees of freedom:

$$\frac{\left\|\Delta\mathbf{U}^{(i)}\right\|_2}{\text{DNORM}} \le \text{DTOL}$$

For the rotational degrees of freedom:

$$\frac{\left\|\Delta\mathbf{U}^{(i)}\right\|_2}{\text{DMNORM}} \le \text{DTOL}$$

where DTOL is a user-specified force convergence tolerance, DNORM and DMNORM are user-specified reference displacement and rotation norms. If left undefined the program automatically determines DNORM and DMNORM during execution.

Note that in each of these convergence criteria the residual norm is measured against a user-specified maximum residual value; for example, the force criterion could be interpreted as

(norm of out-of-balance loads) ≤ RTOL × RNORM

where RTOL × RNORM is equal to the user-specified maximum allowed out-of-balance load.

Note also that these convergence criteria are used in each subdivision of time or load step when the ATS method of automatic step incrementation is used.

- If contact is present in the analysis the following additional criterion is always used in measuring convergence

$$\frac{\max\left(\left\|\mathbf{R}_c^{(i-1)} - \mathbf{R}_c^{(i-2)}\right\|_2, \left\|\boldsymbol{\lambda}^{(i)} - \boldsymbol{\lambda}^{(i-1)}\right\|_2\right)}{\max\left(\left\|\mathbf{R}_c^{(i-2)}\right\|_2, \text{RCONSM}\right)} \le \text{RCTOL}$$

where $\mathbf{R}_c^{(i-1)}$ is the contact force vector at the end of iteration

$(i-1)$, $\boldsymbol{\lambda}^{(i)}$ is the Lagrange multiplier vector at the end of iteration (i), RCONSM is a reference contact force level used to prevent possible division by zero and RCTOL is a user-specified contact force convergence tolerance.

- By default, the program uses the force vector at iteration 0 of each time step to calculate the values of RNORM and RMNORM. Similarly the displacement vector at iteration 0 is used to calculate default values of DNORM and DMNORM. You can also estimate RNORM, DNORM, etc. based on the size of applied forces and enforced displacements, and define RNORM, DNORM, etc. on the NXSTRAT entry.

Non-convergence: Convergence might not occur when the maximum number of iterations is reached or when the solution is diverging. The maximum number of iterations is set by MAXITE in the NXSTRAT entry.

- If the specified convergence criteria are not satisfied within the allowed number of iterations, but the solution is not diverging, the following can be attempted:

  - Check the model according to the guidelines in Section 6.2.8.
  - Use a smaller time step.
  - Increase the number of allowable iterations.
  - Change the ATS parameters.
  - Change convergence tolerances. In most cases, looser tolerances help. However, in some problems, tighter tolerances help by not allowing approximate solutions that could potentially prevent convergence in future time steps.
  - Change line search. Some problems, such as those involving plasticity, perform better with line search.
  - Change contact settings. The optimal contact settings and features depend on the model. See Chapter 4 for more details.

- Divergence of solution terminates the iteration process before the maximum number of iterations is reached. It is sometimes detected when the energy convergence ratio in Eq. (6.2-6) becomes unacceptably large, or when the excessive displacements lead to distorted elements and negative Jacobians. In this case, the following can be attempted:

- Check the model according to the guidelines in Section 6.2.8.
- Use a smaller time step.
- Make sure there are sufficient constraints to remove rigid body modes from all components in the model. Presence of rigid body modes usually results in a large ratio of maximum to minimum pivot during factorization (with sparse solver).

LDC method used

- Convergence in the equilibrium iterations is reached when the following inequalities are satisfied:

Energy convergence criterion: For all degrees of freedom

$$\frac{\Delta \mathbf{U}^{(i)^T}\left[{}^{t+\Delta t}\lambda^{(i-1)}\mathbf{R}-{}^{t+\Delta t}\mathbf{F}^{(i-1)}\right]}{\Delta \mathbf{U}^{(1)^T}\left[\Delta\lambda^{(1)}\mathbf{R}\right]}\leq \text{ETOL}$$

where ETOL is a user-specified energy convergence tolerance.

Force and moment convergence criteria: For the translational degrees of freedom

$$\frac{\left\|{}^{t+\Delta t}\lambda^{(i-1)}\mathbf{R}-{}^{t+\Delta t}\mathbf{F}^{(i-1)}\right\|_2}{\text{RNORM}}\leq \text{RTOL}$$

For the rotational degrees of freedom

$$\frac{\left\|{}^{t+\Delta t}\lambda^{(i-1)}\mathbf{R}-{}^{t+\Delta t}\mathbf{F}^{(i-1)}\right\|_2}{\text{RMNORM}}\leq \text{RTOL}$$

where RTOL is a user-specified force convergence tolerance, RNORM and RMNORM are user-specified reference force and moment norms. If left undefined the program automatically determines RNORM and RMNORM during execution.

The translation/rotation convergence criteria, and the contact convergence criterion, are the same as when the LDC method is not used, see above.

Non-convergence: If convergence has not been reached from an established equilibrium configuration after the maximum restart attempts, the program saves the required restart information and program execution is terminated.

The solution can be continued by performing a restart run. Note that in this case the LDC method must be used in the restart run. A different value for the initial displacement can be enforced at a different nodal point in the first step of the restart run. The enforced initial displacement then corresponds to a displacement increment from the last converged equilibrium position, that is, at the time of solution start for the restart analysis.

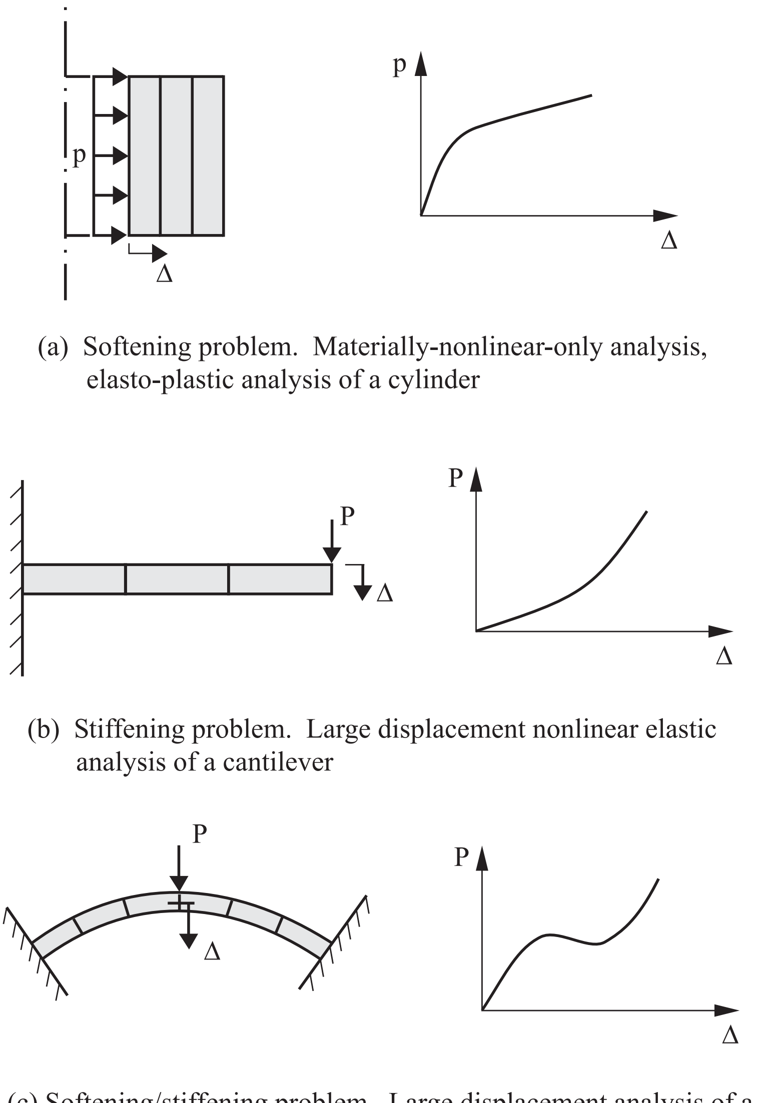
6.2.8 Selection of incremental solution method

- This section gives recommendations on which incremental solution method to use for a given analysis.

- Every nonlinear analysis should be preceded by a linear analysis, if only to check that the model has been set up correctly. The linear analysis results will highlight many important factors such as the proper application of boundary conditions, deletion of all degrees of freedom without stiffness, the quality of the finite element mesh, etc.

- If the use of a sufficient number of load steps and equilibrium iterations with tight convergence tolerances at each load step is considered to yield an accurate solution of the model, then the basic aim is to obtain a response prediction close to this accurate one at as small a solution cost as possible.

- It is helpful to know if the model softens or hardens under increasing load. Structures can soften due to the spread of plasticity, and they can soften or harden due to geometric nonlinear effects. Contact usually leads to hardening. Fig. 6.2-3 shows some examples.



(a) Softening problem. Materially-nonlinear-only analysis, elasto-plastic analysis of a cylinder

(b) Stiffening problem. Large displacement nonlinear elastic analysis of a cantilever

(c) Softening/stiffening problem. Large displacement analysis of a thin arch

Fig. 6.2-3: Different types of nonlinear analyses

- Displacement controlled loading generally converges faster than force controlled loading. For example, in the model shown in Fig. 6.2-3(b), applying an increasing tip displacement to the beam will converge faster than an applied load P, and both will follow the same load displacement curve. For the model in Fig. 6.2-3(c) force control would fail past the local maximum on the load-displacement curve. Displacement controlled loading (apply an increasing Δ) would work in this case. Note that this case is also suitable for the LDC method.

- When the ATS method is used, together with a reasonable time step size, the ATS method will result in almost the same "iteration path" as when not using the method. Namely, no step subdividing will be performed if convergence is always directly reached at the user-specified load levels.

  Hence, in general, it is most convenient to use a reasonable number of load steps together with the ATS method.

- If the problem involves localized buckling, or sudden changes due to contact, or other discontinuities, consider using the low speed dynamic feature. Make sure that the selected structural damping is not excessive.

- The LDC method is useful if collapse of the structure occurs during the (static) solution. However, the use of this algorithm can be very costly. The LDC method is recommended if the user doesn’t want to specify the load increments for the solution period and computational costs are not of primary concern. The LDC method allows also the calculation of the post-collapse response. Note, however, that the displacement solution at a specific load level cannot be obtained using the LDC method because the load increments are calculated by the program.

- Note that usually it is quite adequate to employ the energy convergence tolerance only. The need to use one of the other convergence criteria arises when the energy convergence is not tight (small) enough. In addition, there exist special loading conditions under which the denominator of the inequality (6.2-6) in Section 6.2.7 becomes small and hence the inequality is difficult to satisfy.

6.2.9 Example

This section presents a worked example that illustrates the nonlinear iteration and convergence concepts previously discussed. Fig. 6.2-4 shows the iteration history for a load step. The standard Newton method with line searches is used with the energy and force convergence criterion (ETOL=1×10^{-6}, RTOL=0.01, RNORM=10.0, RMNORM =10.0). For contact RCTOL=1×10^{-3} and the reference contact force RCONSM=0.01. For line search STOL=0.01.

Row ITE=0: This row shows the result of the initial iteration called iteration 0. For this iteration, the program performs the following steps:

Compute ${}^{t+\Delta t}\mathbf{U}^{(0)} = {}^{t}\mathbf{U}$.

Compute ${}^{t+\Delta t}\mathbf{F}^{(0)}$ and ${}^{t+\Delta t}\mathbf{K}^{(0)}$ using ${}^{t+\Delta t}\mathbf{U}^{(0)}$.

Compute the out-of-balance force vector ${}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(0)}$. Only considering translational degrees of freedom, the norm of the out-of-balance force vector is $\left\| {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(0)} \right\|_2 = 2.32\times10^{6}$ and the largest magnitude in the out-of-balance force vector is -2.32×10^{6} at the Z translational degree of freedom of node 740. Only considering rotational degrees of freedom, the norm of the out-of-balance force vector is $\left\| {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(0)} \right\|_2 = 3.27\times10^{-3}$ and the largest magnitude in the out-of-balance force vector is -6.22×10^{-4} at the Y rotational degree of freedom of node 319.

```
 STEP NUMBER =         4     ( TIME STEP = 0.20000E-03   SOLUTION TIME = 0.80000E-03 )
DIAG ELEMENT (WITH MAX ABS VALUE) OF THE FACTORIZED MATRIX =    0.25547E+10     NODE =       740     DOF = Z-TRANSLATION
DIAG ELEMENT (WITH MIN ABS VALUE) OF THE FACTORIZED MATRIX =    0.58413E+02     NODE =       161     DOF = X-ROTATION


METHOD  STEP-NUMBER  SUBINCREMENT    TIME STEP    SOLUTION TIME   INITIAL ENERGY
*ATS*          4          1       0.200000E-03     0.800000E-03     0.223798E+04


I N T E R M E D I A T E   P R I N T O U T   D U R I N G   E Q U I L I B R I U M   I T E R A T I O N S


         OUT-OF-         NORM OF                                       CONVERGENCE RATIOS  CONVERGENCE RATIOS   OUT-OF-BALANCE LOAD
         BALANCE     OUT-OF-BALANCE        NORM OF INCREMENTAL         FOR OUT-OF-BALANCE    FOR INCREMENTAL    VECTOR CALCULATION
         ENERGY      FORCE  MOMENT       DISP.      ROTN.    CFORCE      ENERGY     FORCE    DISP.     CFORCE      BETA      RATIO
                  NODE-DOF   NODE-DOF   NODE-DOF   NODE-DOF   CFNORM                MOMENT   ROTN.               (ITERNS)
                 MAX VALUE  MAX VALUE  MAX VALUE  MAX VALUE
                                                                            COMPARE WITH        COMPARE WITH
                                                                           ETOL       RTOL     DTOL      RCTOL
                                                                          1.00E-06  1.00E-02 (NOT USED) 1.00E-03

ITE=  0  2.24E+03  2.32E+06   3.27E-03   4.79E-03   7.45E-01  4.74E+03    1.00E+00  2.32E+05  6.92E-02  1.15E+00
                      740-Z      319-Y      159-Z      239-Y  4.11E+03              3.27E-04  8.63E-01
                  -2.32E+06  -6.22E-04  -1.14E-03   1.64E-01

ITE=  1  3.95E+00  1.96E+04   9.93E+00   6.16E-04   2.03E-01  4.74E+03    1.77E-03  1.96E+03  8.90E-03  9.01E-01  9.99E-01 -5.95E-08
                      239-X       39-Y      239-X      279-Y  5.26E+03              9.93E-01  2.35E-01               ( 3)
                   4.14E+03  -2.37E+00   1.52E-04   6.71E-02

ITE=  2  2.67E-01  5.89E+03   4.45E+00   1.29E-04   5.98E-02  4.54E+02    1.19E-04  5.89E+02  1.86E-03  9.06E-02  4.84E-01  2.85E-06
                      740-Z      199-Y       42-X      279-Y  5.01E+03              4.45E-01  6.92E-02               ( 5)
                  -1.35E+03  -1.16E+00  -1.67E-05  -1.95E-02

ITE=  3  4.28E-03  1.00E+03   5.20E-01   2.45E-05   7.30E-03  4.46E+02    1.91E-06  1.00E+02  3.53E-04  9.67E-02  9.81E-01 -3.45E-05
                       39-Z      479-Y       39-X       79-Y  4.61E+03              5.20E-02  8.45E-03               ( 3)
                  -2.80E+02   1.72E-01   3.53E-06  -1.86E-03

ITE=  4  2.48E-06  1.77E+01   1.35E-02   3.47E-07   1.71E-04  3.12E+01    1.11E-09  1.77E+00  5.01E-06  6.80E-03  1.00E+00  8.78E-03
                       39-Z      239-Y      159-Z      119-Y  4.58E+03              1.35E-03  1.98E-04               ( 1)
                  -5.76E+00  -5.12E-03  -1.16E-07  -5.42E-05

ITE=  5  8.61E-10  1.51E-01   1.53E-04   1.14E-08   6.76E-06  1.99E-01    3.85E-13  1.51E-02  1.65E-07  4.35E-05  1.00E+00  1.54E-03
                       79-Z      119-Y      159-Z      119-Y  4.58E+03              1.53E-05  7.83E-06               ( 1)
                   5.41E-02   7.55E-05   3.59E-09   2.76E-06

ITE=  6  1.87E-12  5.22E-03   9.02E-06   4.84E-10   2.62E-07  2.64E-02    8.35E-16  5.22E-04  6.99E-09  5.76E-06  1.00E+00 -2.39E-02
                      119-Z      119-Y      159-Z      119-Y  4.58E+03              9.02E-07  3.03E-07               ( 1)
                   2.05E-03  -4.21E-06  -1.80E-10  -1.30E-07
```

Fig. 6.2-4: Example of iteration history printout

```
I T E R A T I O N   T I M E   L O G

SOLUTION TIME (SECONDS)   . . . . . . . . . . . . . . . . . . . . =       5.13

PERCENT OF TIME SPENT IN LINE SEARCHING . . . . . . . . . . . . . . . =  54.00
PERCENT OF TIME SPENT IN LOAD VECTOR/STIFFNESS MATRIX CALCULATION . =  39.57
PERCENT OF TIME SPENT IN SOLUTION OF EQUATIONS  . . . . . . . . . . =   6.43


    6 EQUILIBRIUM ITERATIONS PERFORMED IN THIS TIME STEP TO REESTABLISH EQUILIBRIUM
 STIFFNESS REFORMED FOR EVERY ITERATION OF THIS STEP
 NUMBER OF SUBINCREMENTS IN THIS TIME STEP =    1
```

Fig. 6.2-4: (continued)

Compute $\mathbf{\Delta U}^{(0)}$ using ${}^{t+\Delta t}\mathbf{K}^{(0)}\mathbf{\Delta U}^{(0)} = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(0)}$. Only considering translational degrees of freedom, the norm of the incremental displacement vector is $\left\|\mathbf{\Delta U}^{(0)}\right\|_2 = 4.79\times10^{-3}$ and the largest magnitude in the incremental displacement vector is -1.14×10^{-3} at the Z displacement of node 159. Only considering rotational degrees of freedom, the norm of the incremental displacement vector is $\left\|\mathbf{\Delta U}^{(0)}\right\|_2 = 7.45\times10^{-1}$ and the largest magnitude in the incremental displacement vector is 1.64×10^{-1} at the Y rotation of node 239.

Compute the “out-of-balance energy”
$\mathbf{\Delta U}^{(0)^T}\left({}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(0)}\right) = 2.24\times10^{3}$.

Compute the norm of the change in contact forces CFORCE=4.74×10^{3}, and the norm of the contact forces CFNORM=4.11×10^{3}.

Compute the energy convergence criterion

$$\frac{\mathbf{\Delta U}^{(0)^T}\left[{}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(0)}\right]}{\mathbf{\Delta U}^{(0)^T}\left[{}^{t+\Delta t}\mathbf{R} - {}^{t}\mathbf{F}\right]} = 1.00$$

Compute the force and moment convergence criteria

$$\frac{\left\| {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(0)} \right\|_2}{\text{RNORM}} = 2.32\times 10^{5} \text{ and}$$

$$\frac{\left\| {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(0)} \right\|_2}{\text{RMNORM}} = 3.27\times 10^{-4}.$$

Compute the contact convergence criterion

$$\frac{\text{CFORCE}}{\max(\text{CFNORM,RCONSM})} = 1.15\times 10^{0}$$

The energy convergence criterion is greater than ETOL, the force convergence criterion is greater than RTOL and the contact convergence criterion is greater than RCTOL. Therefore, convergence is not satisfied.

Note that the displacement and rotation norms are also substituted into the displacement convergence criterion, which results in

$$\frac{\left\| \Delta\mathbf{U}^{(0)} \right\|_2}{\text{DNORM}} = 6.92\times 10^{-2} \text{ and}$$

$$\frac{\left\| \Delta\mathbf{U}^{(0)} \right\|_2}{\text{DMNORM}} = 8.63\times 10^{-1}.$$

Since DNORM and DMNORM are not provided by the user, they are automatically estimated by the program. The above displacement convergence values however are not used in determining convergence.

Row ITE=1: This row shows the results of the first equilibrium iteration. In this iteration, the program performs the following steps:

Compute ${}^{t+\Delta t}\mathbf{U}^{(1)} = {}^{t+\Delta t}\mathbf{U}^{(0)} + \Delta\mathbf{U}^{(0)}, {}^{t+\Delta t}\mathbf{F}^{(1)}$ and the line search ratio $\dfrac{\Delta\mathbf{U}^{(0)^T}\left[{}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(1)} \right]}{\Delta\mathbf{U}^{(0)^T}\left[{}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(0)} \right]}$. This ratio turns out to be greater than STOL=0.01, so line search is performed for 3 steps and ends up with a line search energy ratio of 5.95×10^{-8} which is less than STOL, corresponding to $\beta^{(1)} = 0.999$.

Compute ${}^{t+\Delta t}\mathbf{U}^{(1)} = {}^{t+\Delta t}\mathbf{U}^{(0)} + \beta^{(1)}\Delta\mathbf{U}^{(0)}$.

Compute ${}^{t+\Delta t}\mathbf{F}^{(1)}$ using ${}^{t+\Delta t}\mathbf{U}^{(1)}$, and compute ${}^{t+\Delta t}\mathbf{K}^{(1)}$.

Note that the stiffness matrix is updated since the standard Newton method is used.

Compute the out-of-balance force vector ${}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(1)}$. Only considering translational degrees of freedom, the norm of the out-of-balance force vector is $\left\| {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(1)} \right\|_2 = 1.96\times10^4$ and the largest magnitude in the out-of-balance force vector is 4.14×10^3 at node 239 (X translation). Only considering rotational degrees of freedom, the norm of the out of balance force vector is $\left\| {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(1)} \right\|_2 = 9.93\times10^0$ and the largest magnitude in the out-of-balance force vector is –2.37×10^0 at node 39 (Y rotation).

Compute $\mathbf{\Delta U}^{(1)}$ using ${}^{t+\Delta t}\mathbf{K}^{(1)}\mathbf{\Delta U}^{(1)} = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(1)}$. Only considering translational degrees of freedom, the norm of the incremental displacement vector is $\left\| \mathbf{\Delta U}^{(1)} \right\|_2 = 6.16\times10^{-4}$ and the largest magnitude in the incremental displacement vector is 1.52×10^{-4} at node 239 (X translation). Only considering rotational degrees of freedom, the norm of the incremental displacement vector is $\left\| \mathbf{\Delta U}^{(1)} \right\|_2 = 2.03\times10^{-1}$ and the largest magnitude in the incremental displacement vector is 6.71×10^{-2} at node 279 (Y rotation).

Compute CFORCE=4.74×10^3 and CFNORM=5.26×10^3.

Compute the "out-of-balance energy" $\mathbf{\Delta U}^{(1)^T}\left({}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(1)} \right) = 3.95\times10^0$.

Compute the energy convergence criterion

$$\frac{\Delta\mathbf{U}^{(1)^T}\left[{}^{t+\Delta t}\mathbf{R}-{}^{t+\Delta t}\mathbf{F}^{(1)}\right]}{\Delta\mathbf{U}^{(1)^T}\left[{}^{t+\Delta t}\mathbf{R}-{}^{t}\mathbf{F}\right]}=1.77\times10^{-3}.$$

Compute the force and moment convergence criteria

$$\frac{\left\|{}^{t+\Delta t}\mathbf{R}-{}^{t+\Delta t}\mathbf{F}^{(1)}\right\|_2}{\text{RNORM}}=\ 1.96\times10^{3}\ \text{and}$$

$$\frac{\left\|{}^{t+\Delta t}\mathbf{R}-{}^{t+\Delta t}\mathbf{F}^{(1)}\right\|_2}{\text{RMNORM}}=9.93\times10^{-1}.$$

Compute the contact convergence criterion

$$\frac{\text{CFORCE}}{\max(\text{CFNORM,RCONSM})}=9.01\times10^{-1}$$

The energy convergence criterion is greater than ETOL, the force convergence criterion is greater than RTOL, the moment convergence criterion is greater than RTOL, and the contact convergence criterion is greater than RCTOL. Therefore, convergence is not satisfied.

The displacement convergence criterion is also evaluated for informational purposes.

Row ITE=2: This row shows the results from the second equilibrium iteration. This row is interpreted exactly as is row ITE=1. The line search factor in this case is 4.84×10^{-1} obtained in 5 line search iterations.

Again, none of the convergence criteria are satisfied. However, they are all getting smaller.

Row ITE=3: This row shows the results from the third equilibrium iteration. This row is interpreted exactly as is row ITE=2.

Again, none of the convergence criteria are satisfied.

Row ITE=4: This row shows the results from the fourth equilibrium iteration. In this case, the previous increment of displacement from the solver $\Delta\mathbf{U}^{(3)}$ satisfies the line search energy tolerance STOL so no line search is performed.

$$^{t+\Delta t}\Delta\mathbf{U}^{(4)} = {}^{t+\Delta t}\Delta\mathbf{U}^{(3)} + \Delta\mathbf{U}^{(3)}$$

At the end of the that equilibrium iteration, the energy convergence criterion is

$$\frac{\Delta\mathbf{U}^{(4)^T}\left[{}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(4)}\right]}{\Delta\mathbf{U}^{(4)^T}\left[{}^{t+\Delta t}\mathbf{R} - {}^{t}\mathbf{F}\right]} = 1.11\times 10^{-9} \text{ which is less than ETOL.}$$

The force convergence value is 1.77×10^{0} and the moment convergence value is 1.35×10^{-3}. The contact convergence value is 6.80×10^{-3}.

Two of these four criteria (force and contact) are not satisfied, so convergence is not satisfied.

Row ITE=5: The row shows the results for the fifth equilibrium iteration. Contact convergence criterion is now also satisfied leaving only force convergence unsatisfied. The solution continues.

Row ITE=6: The row shows the results from the sixth equilibrium iteration. In this case, all convergence criteria are satisfied and convergence is reached.

6.3 Linear dynamic analysis

- Linear dynamic analysis in Solution 601 is performed by implicit integration using the Newmark method or the Bathe method.

- The notation given below is used in the following sections in the descriptions of the equilibrium equations:

$\mathbf{M}$	=	constant mass matrix
$\mathbf{C}$	=	constant damping matrix
$\mathbf{K}$	=	constant stiffness matrix
${}^{t}\mathbf{R}, {}^{t+\Delta t}\mathbf{R}$	=	external load vector applied at time *t, t*+Δ*t*
${}^{t}\mathbf{F}$	=	nodal point force vector equivalent to the element stresses at time *t*
${}^{t}\ddot{\mathbf{U}}, {}^{t+\Delta t}\ddot{\mathbf{U}}$	=	vectors of nodal point accelerations at time *t, t*+Δ*t*
${}^{t}\mathbf{U}, {}^{t+\Delta t}\mathbf{U}$	=	vectors of nodal point displacements at time *t, t*+Δ*t*

${}^{t}\dot{\mathbf{U}}, {}^{t+\Delta t}\dot{\mathbf{U}}$ = vectors of nodal point velocities at time t, $t+\Delta t$

$\mathbf{U}$ = vector of nodal point displacement increments from time t to time $t+\Delta t$, i.e., $\mathbf{U} = {}^{t+\Delta t}\mathbf{U} - {}^{t}\mathbf{U}$.

- The governing equilibrium equations at time $t+\Delta t$ are given by

$$\mathbf{M}\,{}^{t+\Delta t}\ddot{\mathbf{U}} + \mathbf{C}\,{}^{t+\Delta t}\dot{\mathbf{U}} + \mathbf{K}\,{}^{t+\Delta t}\mathbf{U} = {}^{t+\Delta t}\mathbf{R} \qquad (6.3\text{-}1)$$

The procedures used in the time integration of the governing equations for dynamic analysis are summarized in ref. KJB, Chapter 9.

- The time integration of the governing equations can use the Newmark method or the Bathe time integration method. The method can be selected using TINTEG in the NXSTRAT entry. The Newmark method is explained in ref. KJB, Section 9.2.4, and the Bathe method is explained in the following paper.

ref. K.J. Bathe, “Conserving Energy and Momentum in Nonlinear Dynamics: A Simple Implicit Time Integration Scheme” *J. Computers and Structures*, Vol. 85, Issue 7-8, pp. 437-445. (2007)

- In the Bathe method, the time increment Δt is divided into two, the displacements, velocities, and accelerations are solved for at a time $t + \gamma\Delta t$, where $\gamma \in (0,1)$ using the standard Newmark method. The γ parameter is always set to $2 - \sqrt{2} \approx 0.5858$ to keep the same effective stiffness matrix in the two substeps and to avoid recalculating that matrix and refactorization. In the second substep a 3-point Euler backward method is used to solve for the displacements, velocities and accelerations at time $t + \Delta t$ using the results at time t and $t + \gamma\Delta t$.

Note that, for a given step size, the Bathe scheme is about twice as expensive as the Newmark method due to the extra solution step at time $t + \gamma\Delta t$.

- The following assumptions are used in the Newmark method:

$${}^{t+\Delta t}\dot{\mathbf{U}} = {}^{t}\dot{\mathbf{U}} + \left[\left(1-\delta\right){}^{t}\ddot{\mathbf{U}} + \delta\,{}^{t+\Delta t}\ddot{\mathbf{U}}\right]\Delta t \qquad (6.3\text{-}2)$$

$$ {}^{t+\Delta t}\mathbf{U} = {}^{t}\mathbf{U} + {}^{t}\dot{\mathbf{U}}\,\Delta t + \left[\left(\frac{1}{2} - \alpha\right){}^{t}\ddot{\mathbf{U}} + \alpha\,{}^{t+\Delta t}\ddot{\mathbf{U}}\right]\Delta t^2 \qquad (6.3\text{-}3)$$

where α and δ are the Newmark time integration parameters.

This transforms Eq. (6.3-1) to

$$\hat{\mathbf{K}}\,{}^{t+\Delta t}\mathbf{U} = {}^{t+\Delta t}\hat{\mathbf{R}} \qquad (6.3\text{-}4)$$

where

$$\hat{\mathbf{K}} = \mathbf{K} + a_0\mathbf{M} + a_1\mathbf{C} \qquad (6.3\text{-}5)$$

$$\hat{\mathbf{R}} = {}^{t+\Delta t}\,\mathbf{R} + \mathbf{M}\left(a_0\,{}^{t}\mathbf{U} + a_2\,{}^{t}\dot{\mathbf{U}} + a_3\,{}^{t}\ddot{\mathbf{U}}\right) + \mathbf{C}\left(a_1\,{}^{t}\mathbf{U} + a_4\,{}^{t}\dot{\mathbf{U}} + a_5\,{}^{t}\ddot{\mathbf{U}}\right) \qquad (6.3\text{-}6)$$

and where a_0, a_1, ... , a_5 are integration constants for the Newmark method (see Ref. KJB, Section 9.2.4).

A similar procedure can be followed for the Bathe time integration scheme.

ref. KJB Sections 9.2.4 and 9.4.4

- The trapezoidal rule (also called the constant-average-acceleration method of Newmark) obtained by using $\delta = 0.5$, $\alpha = 0.25$ is recommended for linear dynamic analysis (when the Newmark method is used).

- The trapezoidal rule has the following characteristics:

  ▸ It is an implicit integration method, meaning that equilibrium of the system is considered at time $t+\Delta t$ to obtain the solution at time $t+\Delta t$.

  ▸ It is unconditionally stable. Hence, the time step size Δt is selected based on accuracy considerations only, see ref. KJB, Section 9.4.4.

- The Newmark method is in general stable when the following constraints are satisfied: $\delta \geq 0.5$, $\alpha \geq 0.25(\delta + 0.5)^2$.

- Newmark parameters different from the trapezoidal rule can be specified using ALPHA and DELTA in the NXSTRAT entry. Other Newmark values can add some numerical damping for high frequencies which is useful for some models.

- The Newmark method can be effective in wave propagation problems, but only if the finite element system has a narrow bandwidth. In this case the use of higher-order elements with consistent mass idealization can be a good choice.

- The Newmark method is usually more effective for structural vibration problems. In these analyses, the use of higher-order elements, just as in static analysis, and the use of a consistent mass discretization can be effective.

- The time step increment (Δt) recommended for dynamic analysis with the Newmark method is given by $\omega_{co}\Delta t \le 0.20$ where ω_{co} is the highest frequency of interest in the dynamic response.

- Whether the mass and damping matrices are diagonal or banded (lumped or consistent discretization), the solution always requires that a coefficient matrix be assembled and factorized.

6.3.1 Mass matrix

- The mass matrix of the structure may be based on a lumped or consistent mass calculation. The type of mass matrix to use is selected with MASSTYP in the NXSTRAT entry.

- The consistent mass matrix for each element $\mathbf{M}^{(i)}$ is calculated using

$$\mathbf{M}^{(i)} = \int \rho^{(i)} \mathbf{H}^{(i)T} \mathbf{H}^{(i)} dV$$

where $\rho^{(i)}$ is the density, and $\mathbf{H}^{(i)}$ is the displacement interpolation matrix specific to the element type.

- The construction of the lumped mass matrix depends on the type of element used. Each of the elements in Chapter 2 detail how its lumped mass matrix is calculated. For elements with translational degrees of freedom only, the total mass of the element is divided equally among its nodes.

6.3.2 Damping

ref. KJB Section 9.3.3

- Damping can be added directly to the model through Rayleigh damping. Additional indirect damping results from the selected time integration parameters, plasticity and friction.

- If Rayleigh damping is specified, the contributions of the following matrix $\left(\mathbf{C}_{Rayleigh}\right)$ are added to the total system damping matrix **C** described in Section 6.3:

$$\mathbf{C}_{Rayleigh} = \alpha\mathbf{M} + \beta\mathbf{K}$$

where **M** is the total system mass matrix which can be lumped or consistent, and **K** is the total system stiffness matrix. α and β are specified through the entry PARAM, ALPHA1, ALPHA2.

- Note that $\mathbf{C}_{Rayleigh}$ is in general a consistent damping matrix ($\mathbf{C}_{Rayleigh}$ is diagonal if β is zero and a lumped mass matrix is used).

- $\mathbf{C}_{Rayleigh}$ is constant throughout the solution and it is formed only once – before the step-by-step solution of the equilibrium equations.

- See Ref. KJB, Section 9.3.3, for information about selecting the Rayleigh damping constants *α, β*. In the modal basis, the damping ratio can be written as

$$\xi_i = \frac{\alpha}{2\omega_i} + \frac{\beta\omega_i}{2}$$

where ξ_i is the damping ratio for mode ω_i. It is clear that α tends to damp lower modes and β tends to damp higher modes.

If α is not used, then a value of $\beta = \dfrac{T_p}{\pi}$ will overdamp all motions with periods smaller than T_p. Hence motions with periods smaller than T_p can be suppressed by choosing $\beta = \dfrac{T_p}{\pi}$. This may be of interest when using damping to suppress numerical oscillations.

The above comments apply only when the stiffness matrix does not change significantly during the analysis, however.

6.4 Nonlinear dynamic analysis

6.4.1 Step-by-step implicit time integration

ref. KJB Section 9.5

- Nonlinear dynamic analysis in Solution 601 is performed by direct implicit integration using the Newmark method or the Bathe time integration method, similar to linear dynamic analysis.

- The use of Rayleigh damping $\left(\mathbf{C}_{Rayleigh}\right)$ is the same as described in Section 6.3.2. In this case, the total mass matrix and the initial total stiffness matrix are used to evaluate the Rayleigh damping matrix.

- The governing equations at time $t+\Delta t$ are

$$\mathbf{M}\,{}^{t+\Delta t}\ddot{\mathbf{U}}^{(i)} + \mathbf{C}\,{}^{t+\Delta t}\dot{\mathbf{U}}^{(i)} + {}^{t+\Delta t}\mathbf{K}\,\Delta\mathbf{U}^{(i)} = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(i-1)}$$

where ${}^{t+\Delta t}\ddot{\mathbf{U}}^{(i)}$, ${}^{t+\Delta t}\dot{\mathbf{U}}^{(i)}$, ${}^{t+\Delta t}\mathbf{U}^{(i-1)} + \Delta\mathbf{U}^{(i)}$ are the approximations to the accelerations, velocities, and displacements obtained in iteration (i) respectively.

The vector of nodal point forces ${}^{t+\Delta t}\mathbf{F}^{(i\text{-}1)}$ is equivalent to the element stresses in the configuration corresponding to the displacements ${}^{t+\Delta t}\mathbf{U}^{(i\text{-}1)}$.

- The trapezoidal rule obtained by using $\delta = 0.5$ and $\alpha = 0.25$ is recommended if the Newmark method is used.

- In the Bathe method the time increment Δt is divided into two substeps. In the first substep, the displacements, velocities, accelerations are solved for at a time $t + \gamma\Delta t$, where $\gamma = 0.5$, using the standard Newmark method. In the second substep, a 3-point Euler backward method is used to solve for the displacements, velocities, accelerations at time $t + \Delta t$, using the results at both time t and $t + \gamma\Delta t$.

For large deformation problems, the Newmark method can become unstable, while the Bathe scheme remains stable. While for a given time step size, the Bathe method uses two substeps, the extra solution time in nonlinear analysis can be much less than twice the time used with the Newmark method because of better convergence in the Newton-Raphson iterations.

Some properties of the Bathe method are discussed in

ref. K. J. Bathe and G. Noh, “Insight into an Implicit Time Integration Scheme for Structural Dynamics”, *Computers & Structures*, Vol. 98-99, pp. 1-6, 2012.

ref. G. Noh, S. Ham and K.J. Bathe, “Performance of an Implicit Time Integration Scheme in the Analysis of Wave Propagations”, *Computers & Structures*, Vol. 123, pp. 93-105, 2013.

- The dynamic equilibrium equations are solved using the same iterative procedures used in static analysis, including the ATS method and line search, see Sections 6.2.1 and 6.2.2 for more details. However, the LDC method cannot be used in dynamics.

- The energy and force/moment convergence criteria used in nonlinear dynamic analysis are:

Energy convergence criterion
For all degrees of freedom:

$$\frac{\Delta\mathbf{U}^{(i)^T}\left[\,{}^{t+\Delta t}\mathbf{R}-\mathbf{M}\,{}^{t+\Delta t}\ddot{\mathbf{U}}^{(i-1)}-\mathbf{C}\,{}^{t+\Delta t}\dot{\mathbf{U}}^{(i-1)}-{}^{t+\Delta t}\mathbf{F}^{(i-1)}\right]}{\Delta\mathbf{U}^{(1)^T}\left[\,{}^{t+\Delta t}\mathbf{R}-\mathbf{M}\,{}^{t+\Delta t}\ddot{\mathbf{U}}^{(0)}-\mathbf{C}\,{}^{t+\Delta t}\dot{\mathbf{U}}^{(0)}-{}^{t}\mathbf{F}\right]}\le \text{ETOL}$$

Force and moment convergence criteria
For the translational degrees of freedom:

$$\frac{\left\| {}^{t+\Delta t}\mathbf{R} - \mathbf{M}\,{}^{t+\Delta t}\ddot{\mathbf{U}}^{(i-1)} - \mathbf{C}\,{}^{t+\Delta t}\dot{\mathbf{U}}^{(i-1)} - {}^{t+\Delta t}\mathbf{F}^{(i-1)} \right\|_2}{\text{RNORM}} \le \text{RTOL}$$

For the rotational degrees of freedom:

$$\frac{\left\| {}^{t+\Delta t}\mathbf{R} - \mathbf{M}\,{}^{t+\Delta t}\ddot{\mathbf{U}}^{(i-1)} - \mathbf{C}\,{}^{t+\Delta t}\dot{\mathbf{U}}^{(i-1)} - {}^{t+\Delta t}\mathbf{F}^{(i-1)} \right\|_2}{\text{RMNORM}} \le \text{RTOL}$$

The other convergence criteria and the notation and considerations for the use of the convergence criteria are the same as in nonlinear static analysis; see Sections 6.2.7 and 6.2.8.

- In dynamic analysis the solution is sensitive to the time step size. Using a large step leads to inaccurate time integration regardless of the tightness of the convergence tolerances.

6.4.2 Global mass matrix

The global mass matrix used in nonlinear dynamic analysis is the same as the global mass matrix used in linear dynamic analysis.

Unlike the global stiffness matrix, which is typically recalculated during each equilibrium iteration, the global mass matrix is recalculated only for the following case:

Element birth-death: only the elements that are currently alive contribute to the global mass matrix. (This includes the case of elements that have ruptured; ruptured elements also do not contribute to the global mass matrix.)

In the following discussion, we consider an analysis without element birth-death. In this case, the global mass matrix is calculated only once, at the beginning of the analysis.

For continuum elements, in which the unknowns are the displacement degrees of freedom, the mass matrix does not change due to large deformations (ref KJB, p 542). Hence, in this case, it is theoretically justified to not recalculate the global mass matrix.

However, it should be recognized that there are some situations in which, theoretically, the global mass matrix changes during the analysis. These situations are all associated with rotational degrees of freedom.

Structural elements with rotational degrees of freedom, consistent mass matrix. The effect is most pronounced when the element cross-section or thickness is large, and is also most pronounced for the Hermitian beam element.

Rigid elements and multi-point constraints with rotational degrees of freedom

Typically the effect of neglecting the time-varying parts of the global mass matrix is small. However, for certain analyses in which there are large rotations, the effect can be significant.

6.5 Solvers

Three solvers are available in Solution 601. These are the direct sparse solver (default), the iterative multigrid solver and the 3D-iterative solver. The SOLVER parameter in the NXSTRAT entry is used to select which solver to use. Details on parallel processing can be found in Section 10.9.

The spare solver is the only choice for heat transfer analysis.

6.5.1 Direct sparse solver

- The direct solution method in Solution 601 is a sparse matrix solver. A hybrid ordering scheme of the nested dissection and the minimum degree algorithms is used to greatly reduce the amount of storage required and the total number of operations performed in the solution of the equations.

- The sparse matrix solver is invoked using SOLVER=0 in the NXSTRAT entry.

- The sparse matrix solver is very reliable and robust and should generally be used for most problems in Solution 601. It is the default solver.

- The sparse solver memory is separate from that memory allocated by the rest of the program. It is also dynamically allocated by the solver as needed. The total memory allocated by the Nastran program for Solution 601 covers both the program's memory and the solver's memory.

- The sparse solver can be used both in-core and out-of-core. It is more efficient to run an out-of-core sparse solver using real (physical) memory than it is to run an in-core sparse solver using virtual memory. Therefore, for large problems, we recommend increasing the memory size (via the Nastran command) until it fits the problem in-core, or it reaches approximately 85% of the real memory.
- When a non-positive definite stiffness matrix (i.e. one with a zero or negative diagonal element) is encountered during solution, the program may stop or continue, according to the following rules:

  ▸ If a diagonal element is exactly equal to 0.0, Solution 601 stops unless

    - The equation number corresponding to the zero diagonal element is only attached to inactive elements (elements that are dead due to rupture or the element birth/death feature).
    - The user has requested that Solution 601 continue execution using the NPOSIT flag in the NXSTRAT entry.

  ▸ If the value of a diagonal element is smaller than 10^{-12} but not equal to zero, or the value of a diagonal element is negative, Solution 601 stops unless one of the following options is used:

    - Automatic load-displacement (LDC)
    - Automatic time-stepping (ATS)
    - Potential-based fluid elements
    - Contact analysis
    - The user has requested that Solution 601 continue execution using the NPOSIT flag in the NXSTRAT entry.

- When Solution 601 stops, it prints informational messages for the zero or negative diagonal elements.

- When Solution 601 continues execution and the diagonal element is smaller than 10^{-12}, Solution 601 assigns a very large number to the diagonal element, effectively attaching a very stiff spring to that degree of freedom.

- Note that the stiffness matrix can be non-positive definite due to a modeling error, for example if the model is not sufficiently restrained in static analysis. In this case the results obtained can be misleading.

6.5.2 Iterative multigrid solver

- In the analysis of very large problems, the amount of storage required by a direct solution solver may be too large for the available computer resources. For such problems, the use of the iterative method of solution is necessary.

- The multigrid solver available in Solution 601 is an algebraic solver, and can be used with all solution options of Solution 601.

- The multigrid solver is invoked using SOLVER=1 in the NXSTRAT entry.

- The multigrid solver is sensitive to the conditioning of the coefficient matrix. It generally performs better (requires fewer solver iterations) for well-conditioned problems. Ill-conditioned problems may require a large number of iterations or may not converge at all. The maximum number of iterations is set by ITEMAX in the NXSTRAT entry.

- The conditioning sensitivity of the multigrid solver makes it more suited for bulky 3-D solid models compared to thin structural models where the membrane stiffness is much higher than the bending stiffness. It also makes it more efficient in dynamic analysis (compared to static), because of the stabilizing effect of the mass matrix (inertia effect) on the coefficient matrix.

- Note that the multigrid solver cannot recognize that the stiffness matrix is singular. For such problems, the solver will iterate without converging.

- The multigrid solver is sometimes also less efficient for problems with

  - Displacement coordinate systems that vary significantly along the model
  - A large number of rigid elements or constraint equations
  - A large number of rod or beam elements
  - Some contact problems.

In such cases, the 3D-iterative solver might be used, see Section 6.5.3.

- The main practical differences between the use of the direct solver and the multigrid solver are as follows:

  - The direct solver executes a predetermined number of operations after which the solution is obtained. It is less sensitive to the conditioning of the coefficient matrix.
  - The multigrid solver performs a predetermined number of operations per iteration, but the number of iterations is not known beforehand. The number of iterations depends on the condition number of the coefficient matrix. The higher the condition number, the more iterations are needed. The number of iterations required varies from a few hundred to a few thousand.

- Regarding the convergence of the multigrid method, assume that the system of equations to be solved is $\mathbf{Ax} = \mathbf{b}$, $\mathbf{D}$ is the diagonal vector of $\mathbf{A}$, *N* is the dimension of **x**, $\mathbf{x}^{(k)}$ is the approximate solution at solver iteration *k,* and the residual vector is $\mathbf{r}^{(k)} = \mathbf{b} - \mathbf{Ax}^{(k)}$. We can define:

$$RDA^{(k)} = \left\| r^{(k)} / D \right\|_2 \Big/ \sqrt{N} ,$$

$$RDB^{(k)} = \left\| x^{(k)} - x^{(k-1)} \right\|_2 ,$$

$$RDC^{(k)} = \left\| x^{(k)} \right\|_2 ,$$

$$RDR = \min\left(RDA^{(k)} / RDA^{(1)}, RDB^{(k)} / RDB^{(1)} \right), \text{ and}$$

$$RDX = RDB^{(k)} / RDC^{(k)}.$$

The multigrid method converges when one of the following criteria is reached:

$$RDA^{(k)} \leq \text{EPSII, and } RDX \leq \text{EPSB,}$$

$$RDA^{(k)} \leq \text{EPSA, } RDR \leq \text{EPSB, and } RDX \leq \text{EPSB,}$$

$$RDR \leq \text{EPSA, and } RDX \leq \text{EPSB,}$$

$$\max\left|x^{(k)} - x^{(k-1)}\right| \leq \text{EPSII} * 0.1$$

where EPSIA, EPSIB, and EPSII are convergence tolerances set via the NXSTRAT entry. The defaults are $\text{EPSIA} = 10^{-6}$, $\text{EPSIB} = 10^{-4}$, $\text{EPSII} = 10^{-8}$. However, for nonlinear analysis with equilibrium iterations, looser tolerances can be used.

6.5.3 3D-iterative solver

- The 3D-iterative solver has been developed to efficiently solve large models (i.e. models with more than 500,000 equations) containing mainly higher order 3-D solid elements (e.g., 10-node CTETRA, 20-node CHEXA, etc.).

- The 3D-iterative solver is invoked using SOLVER=2 in the NXSTRAT entry.

- In addition to the higher order 3-D solid elements, the models can contain other elements available in the program (e.g., shells, rods, beams, rebars, etc.), including contact conditions.

- The 3D-iterative solver is effective in linear or nonlinear static analysis and in nonlinear dynamic analysis. For linear dynamic analysis, the sparse solver is usually more effective.

- The 3D-iterative solver, like all iterative solvers, performs a number of iterations until convergence is reached. The maximum number of iterations is set by ITEMAX in the NXSTRAT entry.

- In linear analysis, the 3D-iterative solver convergence tolerances affect the accuracy of the solution. If the convergence tolerances are too loose, inaccurate results are obtained, and if the tolerances are too tight, much computational effort is spent to obtain needless accuracy.

- In nonlinear analysis, the 3D-iterative solver is use to solve the linearized equations in each Newton-Raphson equilibrium iteration. Hence, in this case, the 3D-iterative solver convergence tolerances only affect the convergence rate, not the accuracy of the solution. The accuracy of the solution is specified by the equilibrium iteration tolerances (see Section 6.2).

- The benefits of the 3D-iterative solver are:

  - ▸ When the 3D-iterative solver is used, the solution time and memory requirements scale approximately linearly with the number of equations. When the sparse solver is used, the solution time and memory requirements scale approximately quadratically with the number of equations. Therefore the 3D-iterative solver allows the solution of very large problems.
  - ▸ The 3D-iterative solver is significantly more stable than the iterative multigrid solver; hence the 3D-iterative solver should always be used.

- The limitations of the 3D-iterative solver are:

  - ▸ The 3D-iterative solver convergence tolerances must be appropriately set (see above).
  - ▸ The 3D-iterative solver does not scale well for shared memory parallel processing. However the sparse solver scales well to about 8 cores.
  - ▸ Nearly incompressible hyperelastic materials may slow down the convergence of the 3D-iterative solver. For these material models, the bulk modulus κ should be restricted to a value corresponding to $\nu = 0.49$, instead of the default 0.499 (see Eq. 3.7-13).

- Convergence control in the 3D-iterative solver is as follows:

Considering the linearized equation $Ax = b$, let x^i be its approximate solution at inner-iteration i and $r^i \left(= b - Ax^i\right)$ be its corresponding residual. The convergence in the iterative solver is said obtained if any one of the following criteria is satisfied:

$$\left\|b\right\| \le \varepsilon \quad or \quad \left\|r^i\right\| \le \varepsilon \quad or \quad \left\|x^i - x^{i-1}\right\| \le \varepsilon$$

or equation residual: $\left\|r^i\right\| \big/ r_{scale} \le \begin{cases} 0.001\varepsilon_b & i < 3 \\ \varepsilon_b & i \ge 3 \end{cases}$

or solution residual: $\left\|x^i - x^{i-1}\right\| \big/ x_{scale} \le \begin{cases} 0.001\sigma_v & i < 3 \\ \sigma_v & i \ge 3 \end{cases}$

The solution norm is defined as $\left\|x\right\| = \frac{1}{n}\sum_1^n \left|x_i\right|$. In the above, $\varepsilon \equiv 10^{-16}$, the equation scale $r_{scale} = \left\|b\right\|$, and the variable scale

$$x_{scale} = \begin{cases} \max\left\{\varepsilon, \frac{1}{3}\left(\left\|x^1\right\| + \left\|x^2\right\| + \left\|x^3\right\|\right)\right\} & for\ linear\ problems \\ \max\left\{\varepsilon, \frac{1}{3}\left(\left\|x^1\right\| + \left\|x^2\right\| + \left\|x^3\right\|\right), \left\|\Delta x\right\|\right\} & for\ nonlinear\ problems \end{cases}$$

where $\Delta x = {}^{t+\Delta t}U - {}^{t}U,\ x^i = \Delta U^{(i)}$ is the current solution increment in the Newton-Raphson iteration *i*, $\varepsilon_b = \text{EPSIB}$ of the NXSTRAT entry, and $\sigma_v = \min(10^{-6}, \varepsilon_b \cdot 10^{-3})$.

The equation residual (EQ) and the solution residual (VAR) are written to the .f06 file.

6.6 Tracking solution progress

- Important model parameters such as the memory used by the model, memory used by the solver, number of degrees of freedom, solution times, warning messages, and error messages are all provided in the .f06 file.
- Detailed iteration by iteration convergence information is also written to the .f06 file as illustrated in the example of Section 6.2.9.

- Additional iteration by iteration convergence information can be requested using NXSTRAT DIAGSOL=1 or NXSTRAT

DIAGSOL=2. DIAGSOL=2 is especially useful in contact analysis. This additional information is also written to the .f06 file.

- The program outputs a more summarized time step information to the .log file. This outputs focuses on the time steps and the ATS history.

- The program terminates when the final solution time is reached or when it cannot reach a converged solution. The user can also terminate the program during execution. This can be done gracefully by creating a runtime option file “tmpadvnlin.rto” with a line “STOP=1”. This forces the program to stop after cleaning up all temporary and results files. This method is more useful than “killing” the solution process if the results at the previously converged times steps are needed.

- Several NXSTRAT solution parameters can be modified during execution via the runtime option file “tmpadvnlin.rto”. The NXSTRAT parameters that can be modified are MAXITE, DTOL, ETOL, RCTOL, RTOL, STOL, RCONSM, RNORM, RMNORM, DNORM, and DMNORM. Only one parameter can be specified in each line of the .rto file.

7. Explicit dynamic analysis

This chapter presents the formulations and algorithms used to solve explicit dynamic problems using Solution 701 including time step calculation. Most flags or constants that need to be input in this chapter are in the NXSTRAT bulk data entry. The elements and material properties available for explicit analysis with Solution 701 are listed in Table 2-3.

Information about the progress of the solution is always output to the .f06 file. A shorter summarized output is provided in the .log file.

Tables 7.1 and 7.2 lists element types and options not available in Solution 701.

Table 7.1: Element types not available in Solution 701

Potential-based fluid elements
Multilayered shell elements

Table 7.2: Options not available in Solution 701

Consistent mass matrix
Consistent Rayleigh damping
General constraints
Mesh glueing

7.1 Formulation

The central difference method (CDM) is used for time integration in explicit analysis (see ref. KJB, Section 9.2.1). In this case, it is assumed that

$$ {}^{t}\ddot{\mathbf{U}} = \frac{1}{\Delta t^{2}}\left({}^{t-\Delta t}\mathbf{U} - 2\,{}^{t}\mathbf{U} + {}^{t+\Delta t}\mathbf{U} \right) \tag{7.1-1} $$

and the velocity is calculated using

$$ {}^{t}\dot{\mathbf{U}} = \frac{1}{2\Delta t}\left(-{}^{t-\Delta t}\mathbf{U} + {}^{t+\Delta t}\mathbf{U}\right) \qquad (7.1\text{-}2)$$

The governing equilibrium equation at time *t* is given by

$$\mathbf{M}\,{}^{t}\ddot{\mathbf{U}} + \mathbf{C}\,{}^{t}\dot{\mathbf{U}} = {}^{t}\mathbf{R} - {}^{t}\mathbf{F} \qquad (7.1\text{-}3)$$

Substituting the relations for ${}^{t}\ddot{\mathbf{U}}$ and ${}^{t}\dot{\mathbf{U}}$ in Eq. (7.1-1) and (7.1-2), respectively, into Eq. (7.1-3), we obtain

$$\left(\frac{1}{\Delta t^2}\mathbf{M} + \frac{1}{2\Delta t}\mathbf{C}\right){}^{t+\Delta t}\mathbf{U} = {}^{t}\mathbf{R} - {}^{t}\mathbf{F} + \frac{2}{\Delta t^2}\mathbf{M}\,{}^{t}\mathbf{U} - \left(\frac{1}{\Delta t^2}\mathbf{M} + \frac{1}{2\Delta t}\mathbf{C}\right){}^{t-\Delta t}\mathbf{U} \qquad (7.1\text{-}4)$$

from which we can solve for ${}^{t+\Delta t}\mathbf{U}$.

- The central difference method has the following characteristics:

ref. KJB Sections 9.2.1, 9.4 and 9.5.1

- ▸ It is an explicit integration method, meaning that equilibrium of the finite element system is considered at time *t* to obtain the solution at time *t*+Δ*t*.
- ▸ When the mass and damping matrices are diagonal, no coefficient matrix needs to be factorized, see ref. KJB, p. 772. The use of the central difference method is only effective when this condition is satisfied. Therefore, only lumped mass can be used in Solution 701. Also damping can only be mass-proportional.
- ▸ No degree of freedom should have zero mass. This will lead to a singularity in the calculation of displacements according to Eq. 7.1-4, and will also result in a zero stable time step.
- ▸ The central difference method is conditionally stable. The time step size Δ*t* is governed by the following criterion

$$\Delta t \le \Delta t_{CR} = \frac{T_{Nmin}}{\pi}$$

where Δt_{CR} is the critical time step size, and T_{Nmin} is the smallest period in the finite element mesh.

- The central difference method is most effective when low-order elements are employed.

- The time step in Solution 701 can be specified by the user, or calculated automatically (via the XSTEP parameter in NXSTRAT). When the user specifies the time, Solution 701 does not perform any stability checking. It is the user's responsibility, in this case, to ensure that an appropriate stable time step is used.

- When automatic time step calculation is selected, the TSTEP entry is only used to determine the number of nominal time steps and the frequency of output of results. The stable time step is used instead of the value in TSTEP (unless the value in TSTEP is smaller).

  For example, if the following TSTEP entry is used

```
TSTEP, 1, 12, 1.0, 4
```

there will be 12 nominal steps each of size 1.0. If the stable time step is smaller than 1.0 it will be used instead and results will be saved as soon as the solution time exceeds 4.0, 8.0 and exactly at 12.0 since it is the last step of the analysis.

7.1.1 Mass matrix

- The construction of the lumped mass matrix depends on the type of element used. Details are provided in the appropriate section in Chapter 2.

  For elements with translational degrees of freedom only, the total mass of the element is divided equally among its nodes. For elements with rotational masses (beam and shell elements), the lumping procedure is element dependent.

  Note that the lumping of rotational degrees of freedom is slightly different in implicit and explicit analysis. The rotational masses in explicit analysis are sometimes scaled up so that they do not affect the element's critical time step.

7.1.2 Damping

- Damping can be added directly to the model through Rayleigh damping. Additional indirect damping results from plasticity, friction and rate dependent penalty contact.

- Only mass-proportional Rayleigh damping is available in explicit analysis. Hence, the damping matrix **C** in Eq. 6.3-1 is set to:

$$\mathbf{C}_{Rayleigh} = \alpha \mathbf{M}$$

where **M** is the total lumped mass matrix.

See Ref. KJB, Section 9.3.3, for information about selecting the Rayleigh damping constant *α*.

7.2 Stability

ref. KJB Section 9.4.2

- The stable time step for a single degree of freedom with central difference time integration is

$$\Delta t_{CR} = \frac{T_N}{\pi} = \frac{2}{\omega_N}$$

The stable time step for a finite element assembly is

$$\Delta t \le \Delta t_{CR} = \frac{T_{N\min}}{\pi} = \frac{2}{\omega_{N\max}} \le \frac{2}{\omega_{E\max}}$$

where $\omega_{N\max}$ is the highest natural frequency of the system, which is bound by the highest natural frequency of all individual elements in a model $\omega_{E\max}$ (see Ref. KJB, Example 9.13, p. 815).

- When automatic time step is selected, the time step size is determined according to the following relationship

$$\Delta t = K \times \Delta t_{E\min} = K \times \frac{2}{\omega_{E\max}} \qquad (7.2\text{-}1)$$

where *K* is a factor (set via the XDTFAC parameter in NXSTRAT) that scales the time step.

- For most element types the critical time step can be expressed in terms of a characteristic length and a material wave speed

$$\Delta t_E = \frac{L}{c} \tag{7.2-2}$$

where the definition of the length *L* and the wave speed *c* depend on the element and material type. For all elastic-plastic materials the elastic wave is used. This condition is used in Solution 701 instead of actually evaluating the natural frequency in Eq. (7.2-1).

- Note that the critical time step calculated for all elements is only an estimate . For some elements and material combinations it is exact, and for others it is slightly conservative. However, it may not be small enough for excessively distorted elements (3-D solid and shells), and it will therefore need scaling using the *K* factor in Eq (7.2-1).

- The time step also changes with deformation, due to the change in the geometry of the elements and the change in the wave speed through the element (resulting from a change in the material properties).

Rod elements
The critical time step for a 2-node rod element is

$$\Delta t_E = \frac{L}{c}$$

where L is the length of the element, and c is the wave speed through the element

$$c = \sqrt{\frac{E}{\rho}}$$

Beam elements
The critical time step for the (Hermitian) beam element is

$$\Delta t_E = \frac{L}{c} / \sqrt{1 + \frac{12I}{AL^2}}$$

where L is the length of the element, A is the element area, I is the largest moment of inertia, and c is the wave speed through the element

$$c = \sqrt{\frac{E}{\rho}}$$

Shell elements

The critical time step for shell elements is

$$\Delta t_E = \frac{L}{c}$$

where L is a characteristic length of the element based on its area and the length of its sides, and c is the planar wave speed through the shell, which for linear isotropic elastic materials is

$$c = \sqrt{\frac{E}{\rho(1-\nu^2)}}$$

The critical time step estimated here is only approximate, and may be too large for excessively distorted shell elements.

3-D solid elements

The critical time step for the 3-D solid elements is

$$\Delta t_E = \frac{L}{c}$$

where L is a characteristic length of the element, based on its volume and the area of its sides, and c is the wave speed through the element. For linear isotropic elastic materials c is given as

$$c = \sqrt{\frac{E(1-\nu)}{\rho(1+\nu)(1-2\nu)}}$$

The critical time step estimated here is only approximate, and may be too large for excessively distorted 3-D solid elements.

Spring elements
The critical time step for a spring element is

$$\Delta t_E = \frac{2}{\omega_N} = 2\sqrt{\frac{M_1 M_2}{K(M_1 + M_2)}}$$

where M_1 and M_2 are the masses of the two spring nodes and K is its stiffness. Massless springs are not taken into account in the calculation of the stable time step.

R-type elements
These elements are perfectly rigid and therefore do not affect the stability of explicit analysis.

Gap and bushing elements
These elements use the same criterion as the spring element.

7.3 Time step management

- The stable time step size has a major influence on the total simulation time. Since this time step is determined based on the highest eigenvalue of the smallest element, a single small or excessively distorted element could considerably increase the solution time, even if this element is not relevant to the full model.

  Note that the element having the smallest critical time step size is always provided in the output file.

- Ideally, all elements should have similar critical time steps. If the material properties are uniform throughout the model this means that elements should approximately have the same lengths (see Eq. 7.2-2).

- The evaluation of the critical time step for each element takes some computational time. Therefore, it does not need to be performed at every time step. The parameter XDTCAL in NXSTRAT determines how frequently the critical time step is reevaluated.

- The time step size for explicit analysis can be unduly small for a realistic solution time. Three features are provided to deal with this problem.

- A global mass scaling variable can be applied to all elements in the model (the XMSCALE parameter in NXSTRAT). This scale factor is applied to the densities of all elements, except scalar elements where it is applied directly to their mass.

- Mass scaling can also be applied to elements whose automatically calculated initial time step is below a certain value (XDTMIN1 parameter in NXSTRAT). A mass scale factor is then applied to these elements to make their time steps reach XDTMIN1. The mass scaling ratio is then held constant for the duration of the analysis. This option is not used when the time step size is user-specified.

- Elements with automatically calculated time step smaller than a specified value (XDTMIN2 parameter in NXSTRAT) can be completely removed from the model. This parameter is useful for extremely small or distorted elements that do not affect the rest of the model. This option is not used when the time step size is user-specified.

- The three parameters explained above (XMSCALE, XDTMIN1 and XDTMIN2) should all be used with great care to ensure that the accuracy of the analysis is not significantly compromised.

7.4 Tracking solution progress

- Important model parameters such as the memory used by the model, number of degrees of freedom, solution times, minimum stable time step, warning and error messages are all provided in the .f06 file.

- The program outputs a more summarized time step information to the .log file.

- The program terminates when the final solution time is reached or when it cannot reach a converged solution. The user can also terminate the program during execution. This can be done gracefully by creating a runtime option file “tmpadvnlin.rto” with a line “STOP=1”. This forces the program to stop after cleaning up all temporary and results files. This method is more useful than "killing" the solution process if the results at the previously converged times steps are needed.

8. Heat transfer analysis (Solution 601 only)

8.1 Formulation

ref. KJB Section 7.2.1

- For heat transfer in a body, we assume that the material of the body obeys Fourier's law of heat conduction, i.e.,

$$q = -k\frac{\partial \theta}{\partial x}$$

where

q = heat flux (heat flow conducted per unit area)
θ = temperature
k = thermal conductivity (material property)

- The law states that the heat flux is proportional to the temperature gradient, the constant of proportionality being the thermal conductivity, *k*, of the material. The minus sign indicates the physical fact that a positive heat flux along direction >*x*= is given by a drop in temperature θ in that direction $\partial\theta/\partial x < 0$.

Consider a three-dimensional solid body as shown in Fig. 8.1-1. In the principal axis directions *x*, *y*, and *z* we have

$$q_x = -k_x\frac{\partial \theta}{\partial x};\quad q_y = -k_y\frac{\partial \theta}{\partial y};\quad q_z = -k_z\frac{\partial \theta}{\partial z}$$

where q_x, q_y, q_z and k_x, k_y, k_z are the heat fluxes and conductivities in the principal axis directions. Equilibrium of heat flow in the interior of the body thus gives

$$\frac{\partial}{\partial x}\left(k_x\frac{\partial \theta}{\partial x}\right)+\frac{\partial}{\partial y}\left(k_y\frac{\partial \theta}{\partial y}\right)+\frac{\partial}{\partial z}\left(k_z\frac{\partial \theta}{\partial z}\right) = -q^B \qquad (8.1\text{-}1)$$

where q^B is the rate of heat generated per unit volume.

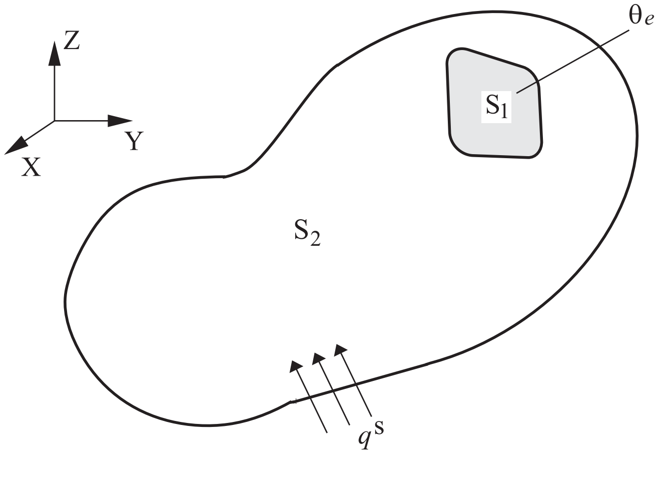


Fig. 8.1-1: Body subjected to heat transfer

- At the boundaries of the body one of the following conditions must be satisfied:

$$\theta\big|_{S_1} = \theta_e \tag{8.1-2}$$

$$k_n \left.\frac{\partial \theta}{\partial n}\right|_{S_2} = q^S \tag{8.1-3}$$

where θ_e is the external surface temperature (on surface S_1), k_n is the body thermal conductivity in the direction n of the outward normal to the surface, and q^S is the heat flow input to the body across surface S_2. This quantity may be constant or a function of temperature as in the case of convection and radiation boundary conditions.

- The governing principle of virtual temperatures corresponding to the above equation can be found in Section 7.2.1 of ref. KJB. The incremental form of the equations is provided in Section 7.2.2, and the discretized finite element equations are provided in Section 7.2.3.

- Note that any region of the boundary where no boundary conditions or loads are explicitly applied is assumed, by virtue of the formulation, to have

$$q^S = 0$$

This implies that the boundary is insulated, allowing no heat transfer across it.

- Note that a time-dependent temperature distribution has not been considered in the above equations – i.e., steady-state conditions have been assumed. For transient problems the heat stored within the material is given by

$$q^C = c\rho\dot{\theta} \tag{8.1-4}$$

where c is the material specific heat capacity and ρ is the density. q^C can be interpreted as forming part of the heat generation term q^B, i.e.,

$$q^B = \tilde{q}^B - c\rho\dot{\theta} \tag{8.1-5}$$

where $\tilde{q}^B$ does not include any heat capacity effect.

- Note that all terms involving stored heat always involve the product $c\rho$. Hence, it is an acceptable modeling technique to set ρ to 1.0 and c to the heat capacity per unit volume (instead of the specific heat).

8.2 Loads, boundary conditions, and initial conditions

ref. KJB Section 7.2.1

- In heat transfer analysis loads and boundary conditions can be specified. More details on these loads and boundary conditions are provided in Chapter 5.

- In all cases, the heat flux or heat generated is converted to nodal heat fluxes by consistent integration of the finite element load vector over the domain of the load application. See ref. KJB Section 7.2.3 for details.

Temperature conditions: The temperature is prescribed on the boundary denoted by S_1 in equation (8.1-2).

Heat flow conditions: The heat flow input is prescribed on the boundary denoted by S_2 in equation (8.1-3).

Convection boundary conditions: The heat flow input is specified on the boundary denoted by S_2 in (8.1-3) according to the following convection condition

$$q^S = h\left(\theta_e - \theta^S\right) \tag{8.2-1}$$

with h being the convection coefficient (possibly temperature dependent), θ_e the ambient (external) temperature, and θ^S the body surface temperature.

Radiation boundary conditions: The heat flow input is specified on the boundary denoted by S_2 in (8.1-3) according to the following radiation condition

$$q^S = \sigma f e\left(\theta_r^4 - \left(\theta^S\right)^4\right) \tag{8.2-2}$$

where σ is the Stefan-Boltzmann constant, f is a view factor or shape factor, e is the material emissivity, θ_r is the temperature of the radiative source (or sink) and θ^S is the unknown body surface temperature. Both temperatures are in the absolute scale. Note that in the above equation the absorptivity is assumed to be equal to the emissivity.

Internal heat generation: Internal heat is generated inside the body. This is introduced as the q^B term in equation (8.1-1).

Initial conditions: For a transient analysis the temperature distribution at the start of the analysis must be specified.

8.3 Steady state analysis

- For a steady-state problem there is no heat capacity effect, i.e., the time derivative term $\dot{\theta}$ does not appear in the governing equation system. See Section 7.2 of ref. KJB for more details.

- Time becomes a dummy variable which is used to indicate different load levels in an incremental load analysis (just as in static structural analysis).

- In linear thermal analysis, the finite element system of equations to be solved is

$$\hat{\mathbf{K}}\boldsymbol{\theta} = \mathbf{Q} \tag{8.3-1}$$

where $\hat{\mathbf{K}}$ is the effective conductance matrix and $\mathbf{Q}$ is the nodal heat flow vector from all thermal load sources.

- In nonlinear thermal analysis, the finite element system of equations to be solved at iteration i of time step $t + \Delta t$ is

$$^{t+\Delta t}\hat{\mathbf{K}}^{(i-1)}\Delta\boldsymbol{\theta}^{(i)} = {}^{t+\Delta t}\mathbf{Q} - {}^{t+\Delta t}\mathbf{Q}_I^{(i-1)} \tag{8.3-2}$$

where $^{t+\Delta t}\hat{\mathbf{K}}^{(i-1)}$ is the effective conductance matrix with contributions from thermal conduction, boundary convection and radiation, $^{t+\Delta t}\mathbf{Q}$ is the nodal heat flow vector with contributions from all thermal load sources such as convection, radiation, boundary heat flux and internal heat generation and $^{t+\Delta t}\mathbf{Q}_I^{(i-1)}$ is the internal heat flow vector corresponding to the element temperatures.

The temperatures are then updated as

$$^{t+\Delta t}\boldsymbol{\theta}^{(i)} = {}^{t+\Delta t}\boldsymbol{\theta}^{(i-1)} + \Delta\boldsymbol{\theta}^{(i)} \tag{8.3-3}$$

These two equations correspond to the full Newton method without line search.

- In the full Newton method, the effective conductance matrix is updated every iteration, and in the modified Newton method, the conductance matrix is only updated every time step. The selection of the full or modified Newton method is controlled by the ITSCHEM parameter in the TMCPARA entry, with full Newton as the default.

- If line search is used, Equation (8.3-3) is replaced by

$$ {}^{t+\Delta t}\boldsymbol{\theta}^{(i)} = {}^{t+\Delta t}\boldsymbol{\theta}^{(i-1)} + \beta^{(i)}\Delta\boldsymbol{\theta}^{(i)} \qquad (8.3\text{-}4)$$

where a line search scaling factor is obtained from a line search in the direction of $\Delta\boldsymbol{\theta}^{(i)}$ in order to reduce out-of-balance residuals according to the following criterion

$$\frac{\Delta\boldsymbol{\theta}^{(i)^T}\left[{}^{t+\Delta t}\mathbf{Q} - {}^{t+\Delta t}\mathbf{Q}_I^{(i)}\right]}{\Delta\boldsymbol{\theta}^{(i)^T}\left[{}^{t+\Delta t}\mathbf{Q} - {}^{t+\Delta t}\mathbf{Q}_I^{(i-1)}\right]} \le \text{TOL}$$

where TOL is a hard-coded tolerance equal to 5 x 10^{-3}, and the magnitude of β is bounded as follows

$$0.001 < \beta < 8.0$$

Line search is off by default, and it is activated via the LSEARCH parameter in the TMCPARA entry.

- The size of the time step increment should be carefully selected in nonlinear heat transfer analysis. If a time step is too large the equilibrium iterations may not converge; on the other hand, too small a time step may result in many more increments being required to reach the desired load level than are necessary.

8.4 Transient analysis

- For a transient analysis, the effect of heat capacity is included in the governing equation system; thus the time derivative, $\dot{\theta}$, term appears in the equations.

- In linear transient thermal analysis, the finite element system of equations to be solved is

$$\mathbf{C}\dot{\boldsymbol{\theta}} + \hat{\mathbf{K}}\boldsymbol{\theta} = \mathbf{Q}$$

where **C** is the heat capacity matrix.

- The heat matrix can be calculated as lumped or consistent (set via the HEATCAP flag in the TMCPARA entry).

- In nonlinear transient thermal analysis, the finite element system of equations to be solved is

$${}^{t+\Delta t}\mathbf{C}^{(i-1)}\dot{\boldsymbol{\theta}}^{(i)} + {}^{t+\Delta t}\hat{\mathbf{K}}^{(i-1)}\Delta\boldsymbol{\theta}^{(i)} = {}^{t+\Delta t}\mathbf{Q} - {}^{t+\Delta t}\mathbf{Q}_I^{(i-1)}$$

- Both full or modified Newton methods can be used, and line search can also be used, as explained in the previous section.

- The time integration of the governing equations can be performed using one of three available time integration schemes: the Euler backward method, the trapezoidal rule, or the Bathe time integration method. The time integration scheme is controlled by the TINTEG parameter of the TMCPARA entry. All three methods are implicit. Explicit analysis is not supported for heat transfer problems.

8.5 Choice of time step and mesh size

ref. KJB Section 9.6.1

- The choice of time step size Δt is important; if Δt is too large then the equilibrium iteration process may not converge for nonlinear problems. For transient problems, the accuracy will also be sacrificed with an excessively large time step. On the other hand, too small a time step may result in extra effort unnecessarily being made to reach a given accuracy.

Therefore it is useful to provide some guidelines as to the choice of time step size Δt. We would like to use as large a time step as the accuracy/stability/convergence conditions allow. Thus the guidelines are phrased as upper limits on the time step size Δt, i.e.

$$\Delta t \le \Delta t_{\max}$$

- Consider the governing differential equation for constant thermal conductivity and heat capacity in one dimension (extrapolation to higher dimension is possible)

$$c\rho \frac{\partial \theta}{\partial t} = k \frac{\partial^2 \theta}{\partial x^2}$$

Non-dimensionalizing this equation, we use

$$\hat{\theta} = \frac{\theta - \theta_0}{q_w\, L/k} \quad ; \quad \hat{t} = \frac{t}{\tau} \quad ; \quad \hat{x} = \frac{x}{L}$$

where θ_0 is the initial temperature, τ a characteristic time, L a characteristic length, and q_w a characteristic heat flux input. This yields the equation

$$\frac{\partial \hat{\theta}}{\partial \hat{t}} = \frac{a}{L^2} \frac{\partial^2 \hat{\theta}}{\partial \hat{x}^2}$$

where $a = \dfrac{k}{c\rho}$ is the thermal diffusivity. We take the characteristic time to be

$$\tau = \frac{L^2}{a}$$

giving the dimensionless time $\hat{t}$ and the dimensionless Fourier number F_0

$$F_0 = \frac{at}{L^2}$$

This number gives the ratio of the rate of heat transferred by conduction to the rate of heat stored in the medium.

To obtain a time step value, a related parameter is introduced

$$F_{0_\Delta} = \frac{a(\Delta t)}{(\Delta x)^2}$$

where Δx is a measure of the element size. Thus, given an element size Δx and a value of F_0, a time step size can be determined. The recommended value of F_{0_Δ} given below comes from stability and accuracy considerations. However, since all available time integration schemes are implicit, accuracy becomes the primary consideration.

- Setting

$$F_{0_\Delta} \leq 1$$

or equivalently

$$\Delta t \leq \frac{(\Delta x)^2}{a}$$

gives reasonably accurate solutions (again, overall solution accuracy depends on the "mesh size" Δx). The minimum value of $\frac{(\Delta x)^2}{a}$ over all the elements of the mesh should be employed. The "element size" Δx is taken, for low or high-order elements, as the minimum distance between any two adjacent corner nodes of the element.

- To provide guidelines for the choice of element size Δx, we consider the case of a semi-infinite solid initially at a uniform temperature, whose surface is subjected to heating (or cooling) by applying a constant temperature or constant heat-flux boundary condition.

We define a "penetration depth", γ, which represents the distance into the solid at which 99.9% of the temperature change has occurred at a time t. For the above posed problem, which has an analytical solution, we have

$$\gamma = 4\sqrt{at}$$

where a is the thermal diffusivity. Thus the penetration zone of the domain must have a sufficient number of elements to model the spatial temperature variation, but beyond that zone larger elements can be used without loss of accuracy.

Since the penetration zone increases with time, we define a time t_{min} which is the minimum >time of interest= of the problem. t_{min} may be the first time at which the temperature distribution over the domain is required, or the minimum time at which discrete temperature measurements are required.

Given this time t_{min} we divide the penetration zone into a number of elements, e.g., for a one-dimensional model, such that

$$\Delta x \le \frac{4}{N}\sqrt{at_{\min}}$$

Usually $N = 10$ gives an effective resolution of the penetration zone for a variety of boundary conditions and time integration schemes i.e.,

$$\Delta x \le \frac{2}{5}\sqrt{at_{\min}}$$

- Note that for a given (large) t_{min}, the element size upper bound may be greater than the physical dimensions of the problem. In this case it is obvious that the element size must be significantly reduced.

- Although consideration was given to one-dimensional problems only, the generalization of Δx to two- and three-dimensional problems has been shown to be valid. Hence the above element size can also be used for two- and three-dimensional problems.

- In coupled TMC analysis the element size will frequently be governed by the structural model. The same will frequently also apply to the time step size (for iterative TMC coupling).

8.6 Automatic time stepping method

- The heat transfer automatic-time-stepping (ATS) method can be used to vary the time step size in order to obtain a converged solution. It is set via the AUTO parameter in the TMCPARA entry. If there in no convergence with the user-specified time step, the program automatically subdivides the time step.

- Further subdivision can be done until convergence is reached or the time step size becomes smaller than a minimum value. This minimum value is set as the original time step size divided by a scaling factor provided by the user (ATSSUBD in TMCPARA).

- This automatic time stepping procedure is used in the solution of heat transfer analyses and one-way coupled TMC (thermo-mechanically coupled) analyses. For iteratively coupled TMC analyses the structural ATS procedure of Section 6.2.4 is used instead. Note that the structural ATS procedure has many more features, and is better suited for nonlinear problems involving contact, geometric and material structural nonlinearities.

9. Coupled thermo-mechanical analysis (Solution 601 only)

ref. KJB Section 7.3

- Advanced Nonlinear Solution can handle two forms of coupling between thermal (heat transfer) and structural analyses (COUP parameter in TMCPARA entry).

- The first is one-way coupling, where the thermal solution affects the structural solution, but the structural solution does not affect the thermal solution.

- The second is iterative coupling which is a two-way coupling where both the thermal and structural solutions are interdependent.

- TMC coupling can involve any combination of static or implicit dynamic structural analysis, and steady state or transient heat transfer analysis. This feature is useful due to the potential for different physical time scales between the structural and heat transfer models.

  The settings needed for each combination are listed below (TRANOPT parameter is in TMCPARA entry).

Table 9.1: Settings for structural and heat transfer combinations

Structural	**Heat transfer**	**Settings**	
		SOL	**TRANOPT**
Static	Steady	153	-
Static	Transient	159	1
Dynamic	Steady	159	2
Dynamic	Transient	159	0 (default)

- Note that since the temperatures are interpolated in the same manner as the displacements, but the mechanical strains are obtained by differentiation of the displacements, it follows that the thermal strains (which are proportional to the temperatures) are in effect interpolated to a higher order than the mechanical strains. The consequence is that for coarse finite element idealizations, the stress predictions show undesirable errors (e.g., nonzero stresses, when the stresses should be zero). These errors vanish as finer finite element idealizations are employed.

Fig. 9.1-1 summarizes the results of a simple analysis that illustrates these concepts.

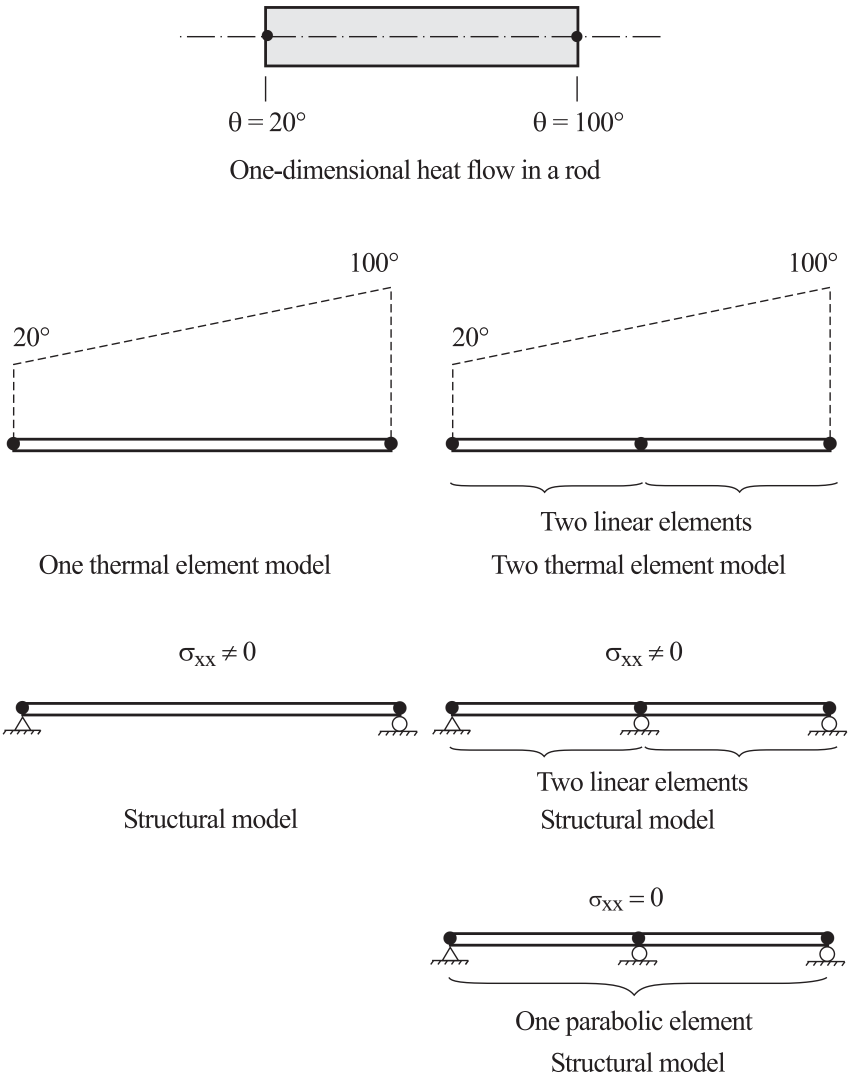


Fig. 9.1-1: Simple problem to schematically demonstrate solution inaccuracies that can arise due to discretizations used in heat flow and stress analyses

9.1 One-way coupling

- In this case, the heat transfer and structural equations are solved separately, with the temperatures from the heat transfer passed back to the structural problem for calculation of thermal expansion and temperature-dependent material properties.

- Currently, the time steps used in the structural and heat transfer equations are assumed to be the same, as set via the TSTEP entry. However, if ATS is present the time steps may differ during the solution. In this case, the heat transfer solution is always ahead of the structural one, and the structural solution uses temperature interpolated from the two closest heat transfer solutions.

9.2 Iterative coupling

- In iterative coupling, the thermal solution can affect the structural solution and the structural solution can affect the thermal solution.

- The coupling from structural to thermal models includes the following effects:

  - Internal heat generation due to plastic deformations of the material
  - Heat transfer between contacting bodies
  - Surface heat generation due to friction on the contact surfaces.

- At the beginning of each time step, the structural model is solved for the displacements using the current temperatures. Then the heat transfer model is solved for the temperatures using the current displacements. This cycle constitutes one TMC equilibrium iteration. TMC convergence is then assessed, and if it is not reached, then the structural and heat transfer models are solved again using the new current displacements and new current temperatures. This process is repeated until TMC convergence is reached. Note that within each TMC equilibrium iteration, the heat transfer and structural models each have their own internal iteration procedure and convergence criteria.

- The same TMC convergence parameter is used in the displacement and temperature convergence checks.

- The temperature convergence is checked as follows:

$$\frac{\left\| {}^{t+\Delta t}\boldsymbol{\theta}^{(i)} - {}^{t+\Delta t}\boldsymbol{\theta}^{(i-1)} \right\|_2}{\left\| {}^{t+\Delta t}\boldsymbol{\theta}^{(i)} \right\|_2} < \text{TOLL}$$

- The displacement convergence is checked as follows:

$$\frac{\left\| {}^{t+\Delta t}\mathbf{U}^{(i)} - {}^{t+\Delta t}\mathbf{U}^{(i-1)} \right\|_2}{\left\| {}^{t+\Delta t}\mathbf{U}^{(i)} \right\|_2} < \text{TOLL}$$

where *i* denotes the TMC iteration. TOLL is set using the TMCTOL parameter in the TMCPARA entry.

- In strongly coupled problems, a temperature relaxation factor can be used to help reach convergence. This is set via the TRELAX parameter in the TMCPARA entry and defaults to 1.0, which corresponds to no relaxation. The temperatures used in the structural analysis in the case of temperature relaxation at a TMC iteration *k* are based on the temperatures in the last heat transfer TMC iteration *k*-1 as well as the prior heat transfer TMC iteration *k*-2.

$$\theta_{structure}^{(k)} = \left(1-\lambda\right)\theta_{heat}^{(k-2)} + \lambda\,\theta_{heat}^{(k-1)}$$

where λ is the temperature relaxation factor.

- Note that decreasing the relaxation factor usually reduces the chances of an oscillating solution, but if decreased too much will also slow down convergence.

Internal heat generation rate due to plastic deformations of the material: The internal heat generation rate per unit volume due to plastic deformations q_M is computed as

$$q_M = \omega\,\overline{\boldsymbol{\tau}} : \overline{\mathbf{D}}^p \qquad (9.1)$$

where $\bar{\boldsymbol{\tau}}$ is the Cauchy stress tensor and $\bar{\mathbf{D}}^p$ is the plastic velocity strain tensor. The overbar denotes Acorresponding to the intermediate configuration@. ω is a parameter, $0 \le \omega \le 1$, to account for the fraction of plastic work that gets converted to internal heat. It is set via the HGENPL parameter in the TMCPARA entry.

This feature is only available for 2-D solid, 3-D solid and shell elements.

Internal heat generation rate due to inelastic deformations of rubber-like materials: When there are viscoelastic or Mullins effects included in rubber-like materials, these effects can cause heat generation, see Sections 3.7.7 and 3.7.8.

Heat transfer between contacting bodies: Contact heat transfer is governed by an equation similar to that used for convection boundary conditions: the heat flux entering contacting body *I* is

$$q_c^{IJ} = \hat{h}\left(\theta^J - \theta^I\right) \tag{9.2}$$

where $\hat{h}$ is the contact heat transfer coefficient (set via the TMCHHAT parameter in the BCTPARA entry) and θ^I and θ^J are the surface temperatures of the contacting bodies.

In the limit as $\hat{h}$ approaches infinity, the temperatures of the contacting bodies become equal to each other. With $\hat{h}$ large, equation (9.2) can be considered a penalty method approximation to the equation $\theta^I = \theta^J$.

Surface heat generation rate due to friction: The frictional contact heat generation rate at a contactor node *G* is computed as

$$q_G^{IJ} = \boldsymbol{\tau} \cdot \dot{\mathbf{U}} \tag{9.3}$$

where $\boldsymbol{\tau}$ is the frictional contact force and $\dot{\mathbf{U}}$ is the relative velocity between the contacting bodies at the point of contact.

The heat rate going to the contactor body is $f_c q_G^{IJ}$ and the heat rate going to the target body is $f_t q_G^{IJ}$, where f_c and f_t are the fractions of generated heat reaching the contactor and target surfaces, respectively. These user input parameters are set via the TMCFC and TMCFT parameters in the BCTPARA entry. The following relations must hold:

$$0 \le f_c \le 1, \qquad 0 \le f_t \le 1, \qquad 0 \le f_c + f_t \le 1$$

The contactor heat rate is applied to the contactor node. The target heat rate is distributed among the target segment nodes.

10. Additional capabilities

10.1 Initial conditions

10.1.1 Initial displacements and velocities

- Initial displacements and velocities at nodes can be specified using the TIC entry together with the IC case control command.

- Any initial displacements or velocities specified in a restart run are ignored, except when restarting from a static to a dynamic analysis; in this case, initial velocities are taken into account.

- Initial rotations should only be applied in small displacement analysis.

- The initial rotations at a node are interpreted differently depending upon the elements attached to the node:

  - Element without rotational dofs: rotations are ignored.
  - Linear or MNO element with rotational dofs: initial rotations are used in the calculation of the initial force vector (but not used thereafter).
  - Large displacement element with rotational dofs: initial rotations are ignored.
  - Small displacement R-type element that is internally represented as a constraint equation or rigid link: the initial position of the slave node is determined by the initial rotation of the master node.
  - Large displacement R-type element that is internally represented as a rigid link: the initial position of the slave node is determined by the initial rotation of the master node, as if the rigid link is a small displacement rigid link. This will cause the length of the rigid link to be incorrect if the initial rotation is large.

10.1.2 Initial temperatures

- Initial temperatures, for both structural and heat transfer analyses, are specified via the TEMPERATURE (INITIAL) case control command. The actual temperature values are specified via the TEMPD and TEMP entries.

- For transient heat transfer analysis (SOL 601,159) the initial temperatures can also be specified using the IC case control command. In this case, it takes precedence over the TEMPERATURE (INITIAL) command.

- The thermal strains are always assumed to be zero initially, see Section 3.1.6.

10.2 Restart

- Restart is a useful feature in Advanced Nonlinear Solution. It can be used when the user wishes to continue an analysis beyond its previous end point, or change the analysis type, loads or boundary conditions or tolerances. A restart analysis is selected by setting MODEX = 1 in the NXSTRAT entry. Recovering results from a restart file without continuing the analysis can also be done setting MODEX = 2.

- All relevant solution data needed for a restart run are saved in a file (with extension .res) in case they are needed in a restart analysis.

- Note that multiple restart data can be appended to the restart file. This enables the restart analysis to be based on a solution step different from the last converged solution. Saving multiple time step solutions to a restart file can be expensive, however, as it leads to a large restart file size. The frequency of data writing to a restart file is set via the IRINT flag in the NXSTRAT entry.

- The restart time is set via the TSTART parameter in the NXSTRAT entry. If no restart time is provided in the restart run (achieved by setting the restart time to 0.0), the program uses the data for the latest restart time on the .res file.

- Note that once the second analysis starts, it will overwrite the .res file with new data. Therefore, if the user wishes to redo the second run, then the .res file must be copied again from the first model.

- The geometry, and most element data, cannot be changed in a restart analysis. However, the following changes are allowed:

  - Type of analysis can change. Static to dynamic and dynamic to static restarts are allowed.
  - Solution type can be changed. Solution 601 (static or dynamic) to Solution 701 restarts are allowed and vice-versa. In this case, features not available in either solution type cannot be used.
  - Solution control variables can change. The flags, constants and tolerances for the iteration method, convergence, time integrations, automatic time stepping and load-displacement-control can be changed.
  - Externally applied loads and enforced displacements can be changed.
  - The material constants can be changed. However, note that in a restart run the same material model (with the same number of stress-strain points and the same number of temperature points, if applicable) must be used for each element as in the preceding run.
  - Boundary conditions can be changed.
  - Constraint equations and rigid elements can be changed.
  - Contact settings can be changed. This includes most contact set, contact pair and contact surface parameters. See section 4.6.4 for restrictions.
  - Rayleigh damping coefficients can be changed.
  - Time increment and number of solution steps can be modified.
  - Time functions describing the load variations can be changed.
  - It is not allowed to have no birth-death in the first run, then birth-death in the restart run. If there is no birth-death in the first run, give a large value for TDEATH for the element groups that have birth-death in the restart run.

- Note that some default settings are different between Solution 601 and Solution 701. Some of these have to be manually set by the user to enable restarts. The most common such settings are:

  - Incompatible modes default on in Solution 601 and off in Solution 701

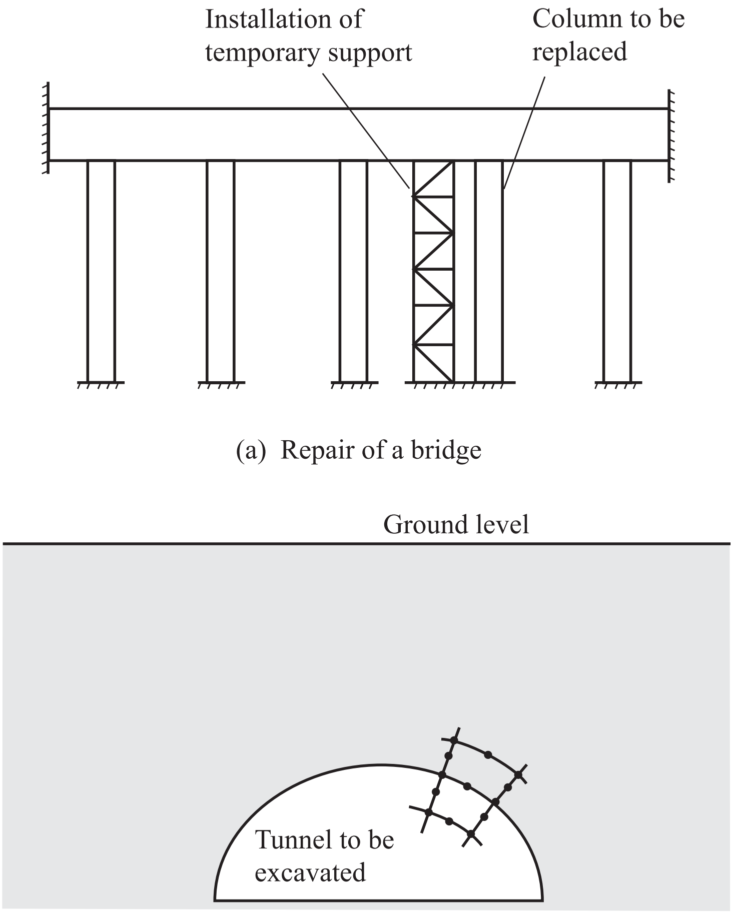
- ▸ Default large strain formulation (ULH in Solution 601 and ULJ in Solution 701)

- When restarting from static to dynamic analysis (both implicit and explicit dynamics), the initial velocities and accelerations are assumed to be zero. However, if an initial velocity is prescribed in the restart run, it will be used instead. When restarting from one dynamic analysis to another, initial velocities and accelerations are transferred from the first to the second run.

- A results recovery mode is available by setting MODEX =2 in the NXSTRAT entry. In this case, the program reads the restart file and recovers the results at the final restart time available in the restart file. Results at a specific time can also be recovered by setting the TSTART parameter in NXSTRAT to the desired time.

10.3 Element birth and death feature

- The element birth and death option is available for modeling processes during which material is added to and/or removed from the structure (set via the EBDSET case control and bulk data commands). Such processes, for example, are encountered in the construction of a structure (structural members are added in succession), the repair of a structure (structural components are removed and new ones are added) or during the excavation of geological materials (a tunnel is excavated). If the element birth and death option is used, the corresponding elements become automatically nonlinear. Fig. 10.3-1 illustrates two analyses that require the element birth and death options.



(a) Repair of a bridge

(b) Excavation of a tunnel

Fig. 10.3-1: Analyses that require the element birth and death options

- The main features of element birth and death are as follows:
    - If the element birth option is used, the element is added to the total system of finite elements at the time of birth and all times thereafter.
    - If the element death option is used, the element is taken out of the total system of finite elements at times equal to and

larger than the time of death.

- If both element birth and death options are used, the element is added to the total system of finite elements at the time of birth and remains active until the time of death. The time of death must be greater than the time of birth. The element is taken out of the total system of finite elements at all times equal to and larger than the time of death.

- Once an element is born, the element mass matrix, stiffness matrix and force vector are added to the mass matrix, stiffness matrix and force vector of the total element assemblage (until the death time, if any). Similarly, once an element dies, the element mass matrix, stiffness matrix and force vector are removed from the total assembled mass matrix, stiffness matrix and force vectors for all solution times equal to or larger than the time of death of the element.

- The element birth/death option applies to any mass effect, i.e., gravity loading, centrifugal loading and inertia forces. The mass matrix, therefore, does not remain constant throughout the solution.

- Note also that the damping matrix is not modified when elements die or when they are born. Therefore, Rayleigh damping should not be used. For example, if a pipe break is simulated by setting the death time for certain elements in an implicit dynamic analysis, then, if Rayleigh damping is used, the pipe might not separate.

- When the element birth/death option is used, the tangent stiffness matrix may at some solution times contain zero rows and corresponding columns. The equation solver disregards any zero diagonal element in the tangent stiffness matrix if no elements are attached to the associated degrees of freedom.

- Advanced Nonlinear Solution enables the user to set an element death decay time parameter (DTDELAY in NXSTRAT) which causes the gradual reduction of the element stiffness matrix to zero over a finite time rather than instantly. The reduction starts at the death time and progresses linearly with time until the decay time has passed. The element therefore totally vanishes at a time equal to the sum of the death time and the death decay time. This option is useful for mitigating the discontinuity that the structure may experience due to the death of some of its elements.

- The birth/death feature is available for contact sets.

- The birth/death feature is not available for mesh glued surfaces.

- The birth/death feature is not available for the potential-based fluid elements.

- To provide the appropriate concentrated and element loading that takes into account the element birth/death option, time functions on the loading need to be used that correspond to the element birth and death times.

- The time at which an element becomes active or inactive is specified by the parameters TBIRTH and TDEATH respectively (in the EBDSET entry).

Birth option active

In the discussion of element birth, it is useful to refer to the “preborn” time. The program determines the preborn time to be the solution time that just precedes the user-input time of element birth. We will discuss the selection of the preborn time in more detail below.

Now we describe how the element displacements at the preborn time are used in the element strain calculations. Recall that the current element coordinates, original element coordinates and displacements are related by

$$^{t+\Delta t}\mathbf{x} = {}^{0}\mathbf{x} + {}^{t+\Delta t}\mathbf{u}$$

and that the element strains are calculated using the displacements and original coordinates, using, for example in geometrically linear analysis,

$$^{t+\Delta t}\mathbf{e} = \frac{\partial\, ^{t+\Delta t}\mathbf{u}}{\partial\, ^{0}\mathbf{x}}$$

with similar calculations in geometrically nonlinear analysis.

For an element that is born, these relationships are modified as follows:

$$^{t+\Delta t}\mathbf{x} = \left({}^{0}\mathbf{x} + {}^{pb}\mathbf{u} \right) + \left({}^{t+\Delta t}\mathbf{u} - {}^{pb}\mathbf{u} \right)$$

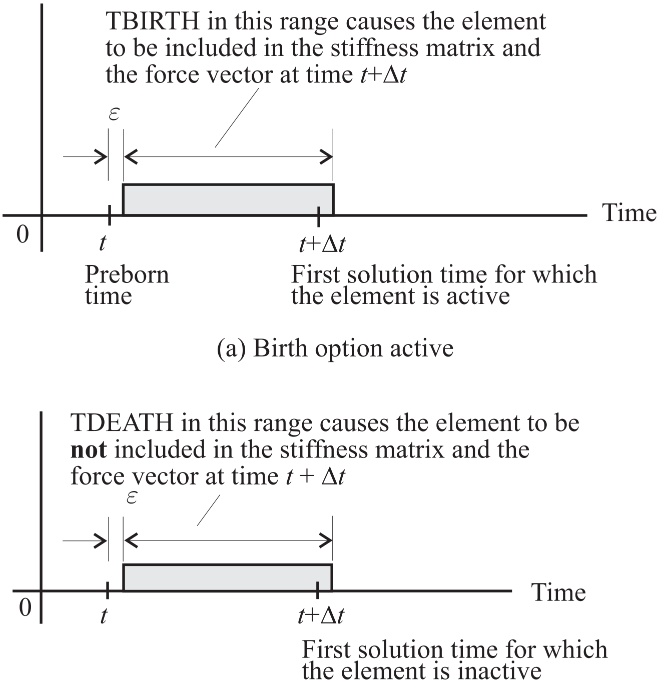
$$^{t+\Delta t}\mathbf{e} = \frac{\partial \left({}^{t+\Delta t}\mathbf{u} - {}^{pb}\mathbf{u} \right)}{\partial \left({}^{0}\mathbf{x} + {}^{pb}\mathbf{u} \right)}$$

in which ${}^{pb}\mathbf{u}$ are the displacements at the preborn time. The quantity ${}^{0}\mathbf{x} + {}^{pb}\mathbf{u}$ are the coordinates at the preborn time and the quantity ${}^{t+\Delta t}\mathbf{u} - {}^{pb}\mathbf{u}$ are the displacements relative to the coordinates at the preborn time.

From the above, we observe that if the current displacements are the same as the preborn displacements, the element strains are zero. And, if there are no thermal or initial strains in the element, the element stresses are also zero.

Now we discuss in detail how the program determines the preborn time.

Fig. 10.3-2(a) shows the activity of an element for which the birth option is active. Note that if TBIRTH is input for the range shown (where $\mathrm{TBIRTH} > t + \varepsilon$ and $\mathrm{TBIRTH} < t + \Delta t + \varepsilon$, then the preborn time for the element is t and the element is first active at time $t + \Delta t$. Here ε is a program-calculated tolerance, typically $\Delta t / 1000$. The results obtained are independent of the exact position of TBIRTH within the range of solution times $t + \varepsilon$ to $t + \Delta t + \varepsilon$.


TBIRTH in this range causes the element to be included in the stiffness matrix and the force vector at time $t+\Delta t$
ε
0
Time
t
$t+\Delta t$
Preborn time
First solution time for which the element is active
TDEATH in this range causes the element to be **not** included in the stiffness matrix and the force vector at time $t + \Delta t$
ε
0
Time
t
$t+\Delta t$
First solution time for which the element is inactive

(a) Birth option active

(b) Death option active

Fig. 10.3-2: Use of element birth and death option

Two special cases are noteworthy:

- TBIRTH just slightly greater than $t+\varepsilon$, for example $t+\Delta t/100$. In this case, the preborn time is t, which is close to TBIRTH. Because TBIRTH is close to t, the preborn time is essentially selected by the choice of TBIRTH, and the element can be thought of as being born “strain-free”. The element is first active at time $t+\Delta t$. If the element displacements remain unchanged between times t and $t+\Delta t$, the element remains strain-free. See also the example given below.

- TBIRTH = $t+\Delta t$. In this case, the preborn time is t (and not $t+\Delta t$). The element is first active at time $t+\Delta t$, however the element need not be strain-free at time $t+\Delta t$.

Regarding the mass matrix, when the element is born, the mass matrix for the element is computed using the original coordinates of the element ${}^{0}\mathbf{x}$, and not the coordinates at the preborn time ${}^{0}\mathbf{x}+{}^{pb}\mathbf{u}$.

Death option active

Fig.10.3-2(b) shows the activity of an element for which the death option is active. Note that if TDEATH is input for the range shown (where $\text{TDEATH} > t+\varepsilon$ and $\text{TDEATH}<t+\Delta t+\varepsilon$, the element is first inactive at time $t+\Delta t$.

Birth then death option active

This is a direct combination of the birth and death options. Initially some elements are inactive. At a particular solution time determined by the time of birth TBIRTH, the elements become active and remain so until a subsequent solution time determined by the time of death TDEATH, where TDEATH > TBIRTH.

Example of the element birth option: Consider the materially linear rod element model shown in Fig. 10.3-3(a) in which the time of birth for element 2 is slightly larger than t, e.g. $\text{TBIRTH} = t+\Delta t/100$. Element 2 is unborn at time t, as shown in Fig. 10.3-3(b).

At the beginning of the solution for time $t+\Delta t$, the program determines that element 2 is active for time $t+\Delta t$, and that the preborn time t_{pb} is equal to the solution time t. The program stores ${}^{pb}u={}^{t}u$. In the subsequent equilibrium iterations, the element relative displacement is ${}^{t+\Delta t}u-{}^{pb}u={}^{t+\Delta t}u-{}^{t}u$, the element initial length is ${}^{0}L+{}^{pb}u={}^{0}L+{}^{t}u={}^{t}L$ and the element strain is ${}^{t+\Delta t}e=\dfrac{{}^{t+\Delta t}u-{}^{pb}u}{{}^{0}L+{}^{pb}u}$. This procedure is physically identical to adding an element of length ${}^{t}L$ to the assemblage at time $t+\Delta t$. Note that the stiffness of element 2 is based on the length ${}^{t}L$.

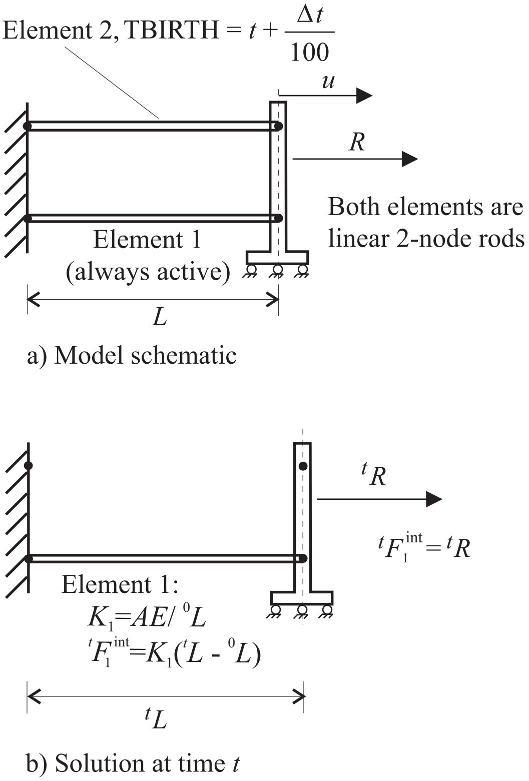
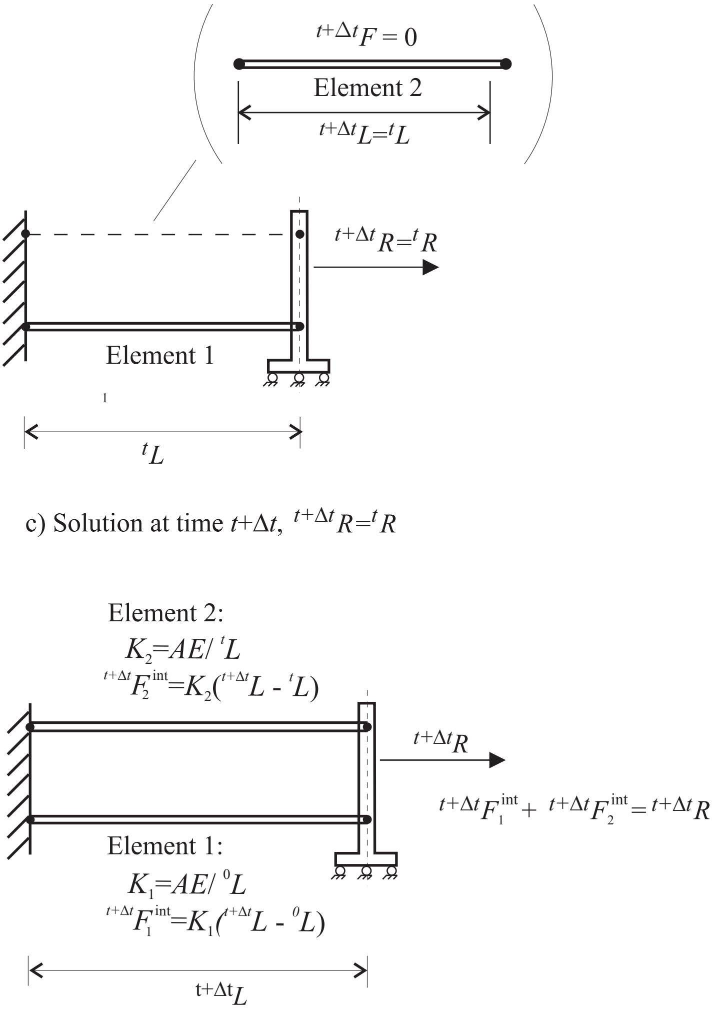

Element 2, TBIRTH = $t + \frac{\Delta t}{100}$
u
R
Both elements are linear 2-node rods
Element 1 (always active)
L
a) Model schematic
${}^{t}R$
${}^{t}F_1^{\text{int}} = {}^{t}R$
Element 1:
$K_1 = AE/\,{}^{0}L$
${}^{t}F_1^{\text{int}} = K_1({}^{t}L - {}^{0}L)$
${}^{t}L$
b) Solution at time t

Fig. 10.3-3: Example of the use of the element birth option

If the external force ${}^{t+\Delta t}R = {}^{t}R$, then element 2 is stress-free, as the system was in equilibrium at time t without any forces from element 2 (Fig 10.3-3(c)).

If the external force ${}^{t+\Delta t}R \neq {}^{t}R$, then element 2 is not stress-free, and the force in element 2 is determined based on its deformation with respect to its preborn state (the solution at time t), as shown above (Fig 13.3-3(d)).

Hence, the total increment in displacement from time t to time $t + \Delta t$ determines the force in element 2. Identically, the same solution would be obtained using any value for TBIRTH which satisfies the relation $\text{TBIRTH} > t + \varepsilon$ and $\text{TBIRTH} < t + \Delta t + \varepsilon$.



$^{t+\Delta t}F = 0$
Element 2
$^{t+\Delta t}L = {}^{t}L$
$^{t+\Delta t}R = {}^{t}R$
Element 1
1
^{t}L
Element 2:
$K_2 = AE/\,^{t}L$
$^{t+\Delta t}F_2^{\text{int}} = K_2(^{t+\Delta t}L - {}^{t}L)$
$^{t+\Delta t}R$
$^{t+\Delta t}F_1^{\text{int}} + {}^{t+\Delta t}F_2^{\text{int}} = {}^{t+\Delta t}R$
Element 1:
$K_1 = AE/\,^{0}L$
$^{t+\Delta t}F_1^{\text{int}} = K_1(^{t+\Delta t}L - {}^{0}L)$
$^{\mathrm{t}+\Delta \mathrm{t}}L$

c) Solution at time $t+\Delta t$, $^{t+\Delta t}R = {}^{t}R$

d) Solution at time $t+\Delta t$, $^{t+\Delta t}R \neq {}^{t}R$

Fig. 10.3-3 (continued)

10.4 Element death due to rupture

- For the materials and elements listed in Table 10.4-1, element death is automatically activated when rupture is detected at any integration point of the element. The element is then considered "dead" for the remainder of the analysis, and, in essence, removed from the model (mass and stiffness contributions).

- When elements die, contactor segments connected to these elements are also removed from the model.

- Dead elements may be gradually removed from the model in order to avoid sudden changes in stiffness and acceleration. This feature is activated by setting a non-zero DTDELAY time in the NXSTRAT entry.

Table 10.4-1: Elements and material models that include "death upon rupture"

	Rod[1]	2-D solid	3-D solid	Beam	Shell
Plastic-multilinear	✓	✓	✓	-	✓
Thermo-elastic-plastic, plastic-creep, thermal plastic-creep	✓	✓	✓	-	✓
Plastic-cyclic	✓	✓	✓	✓	✓

1) The rupture option is not applicable to a rod element with a gap.

10.5 Reactions calculation

- Output of the reaction forces and moments is governed by the SPCFORCES case control command.

- Note that loads applied to fixed degrees of freedom do not contribute to the displacement and stress solutions. However, these loads are accounted for in the reaction calculations.

- Reaction forces and moments at a node are computed using the consistent force vectors (calculated from the element internal stresses) of elements attached to the node. Hence, a check on the balance of the support reactions and the applied loads often provides a good measure on the accuracy of the solution (in terms of satisfying equilibrium in a nonlinear analysis).

- Reaction calculations in dynamic analysis with consistent mass matrix take into account the mass coupling to the deleted degrees of freedom. The reactions exactly equilibrate the applied forces in all cases.

- If Rayleigh damping is used, then the damping contribution to deleted degrees of freedom is taken into account for implicit time integration.

10.6 Stiffness stabilization (Solution 601 only)

- During the solution of equations in static analysis, zero pivots may arise, for example in the following cases:

Unsupported body: If the forces acting on the body are not in equilibrium, one or more rigid body motions of the body are activated and no solution can be expected. Even if the forces acting on the body are in equilibrium, so that no rigid body motion is in fact activated, zero pivots are present corresponding to the rigid body modes.

Contact analysis, in which one or more of the individual parts of the model (not considering contact) contain rigid body motions. When the parts are not in contact, then there is nothing to prevent the rigid body motions. (This includes tied contact.)

Mesh glueing, when one or more of the individual parts of the model (not considering glueing) contain rigid body motions.

General constraints, when one or more of the individual parts of the model (not considering the general constraints) contain rigid body motions.

These zero pivots will stop the solution, unless the zero pivots are prevented from occuring.

- Stiffness stabilization is used to prevent the equation solver from encountering zero pivots. Stiffness stabilization is available for static analysis, with or without low-speed dynamics. (In dynamic analysis, these zero pivots are not present due to the mass matrix, so stiffness stabilization is not available for dynamic analysis.)

- Parts of the model with rigid body motions can alternatively be treated by adding weak springs at various locations in the model. The advantages of using stiffness stabilization, instead of using weak springs, are:

  - Determining the number, location and stiffness of the springs requires a lot of user intervention.
  - There may be no suitable locations for the springs.
  - The stiffness of each spring has to be entered as an absolute value (with dimensions of force/length) while the stiffness stabilization factor is dimensionless (see below).
  - The springs generate internal forces which affect the final solution while stiffness stabilization does not affect the internal forces. It is sometimes hard to assess how much the springs affect the final solution.

- Stiffness stabilization modifies the diagonal stiffness terms (except for those belonging to contact equations) as follows:

$$\mathbf{K}_{ii} = (1+\varepsilon_{STAB})\ \mathbf{K}_{ii}$$

where ε_{STAB} is a dimensionless stabilization factor. The right-hand side load vector is not modified.

- There are three stiffness stabilization options available, which are selected using the MSTAB and MSFAC parameters in the NXSTRAT entry:

MSTAB=0 (no stabilization)
MSTAB=1 (stabilization, with ε_{STAB} = MSFAC)
MSTAB=2 (stabilization is activated if needed)

The defaults are MSTAB=0, MSFAC=1E-10.

- When MSTAB=2, the use of stabilization is determined based on the ratio of the factorized maximum and minimum diagonals of the stiffness matrix. This determination is made for every equilibrium iteration in nonlinear analysis. When stabilization is used, stabilization is applied to all degrees of freedom using the value ε_{STAB} (as if MSTAB=1).

- In linear analysis, stabilization should be used with caution, since the right-hand-side load vector is not modified. The solution can therefore be affected by stiffness stabilization.

It is recommended to try the analysis first without stabilization. If the equation solver encounters zero pivots, then try one of the following methods:

- Use stabilization with the smallest possible value of ε_{STAB} for which the equation solver gives a solution, or
- Use stabilization, and change the analysis to a nonlinear analysis, for example specify element birth-death in one of the elements; or specify a nonlinear material, with material constants chosen so that the material response is linear (for example, an elastic-plastic material with a very high yield stress).

- In nonlinear analysis, since the right-hand-side load vector is not modified, the final converged solution is the same as without stabilization (assuming that the tolerances are tight enough). However, the rate of convergence can be worsened due to the stiffness stabilization, so that more equilibrium iterations are required.

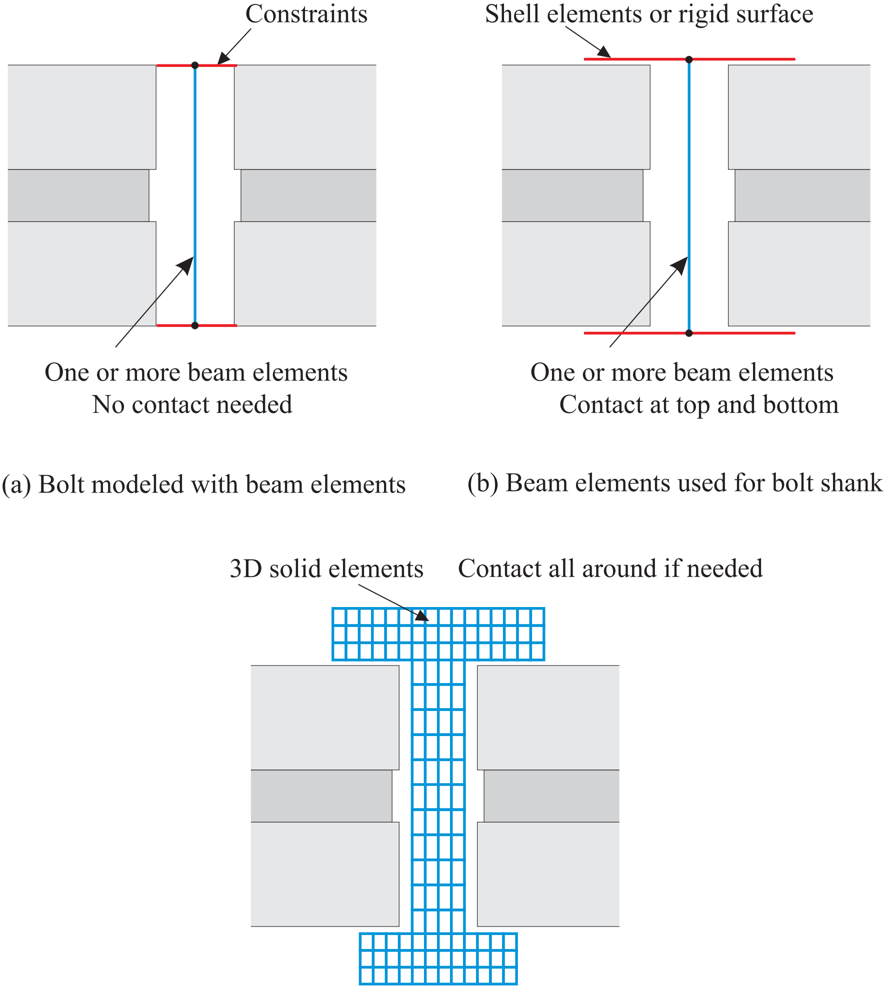
- Stiffness stabilization is only useful for the sparse and 3D-iterative solvers. The iterative multigrid solver does not fully factorize the stiffness matrix and hence cannot properly trigger the automatic stabilization.

10.7 Bolt feature (Solution 601 only)

Overview

- The bolt option is a modeling feature which uses elements with a specific bolt-type loading. The bolt-type loading is bolt-tensioning, i.e., the axial force in the bolt is specified. The deformations of the rest of the structure are considered. For example, if a single bolt is loaded with a user-specified bolt tension, then, after the bolt loading is applied, the axial force of the bolt has the given value, and the axial force acts upon the rest of the structure, causing structural deformations.

- Different modeling techniques with varying complexity can be used to model a bolt. Three such techniques are shown in Fig. 10.7-1, in increasing level of complexity.


Constraints
One or more beam elements
No contact needed
Shell elements or rigid surface
One or more beam elements
Contact at top and bottom
3D solid elements
Contact all around if needed

(a) Bolt modeled with beam elements

(b) Beam elements used for bolt shank

(c) Bolt modeled with 3-D solid elements

Fig 10.7-1: Different bolt modeling techniques

- Any number of bolts can be included in the model. The bolts are all loaded simultaneously.

- The bolt type loading is applied during special 'bolt loading" iterations. During the bolt loading iterations, time is frozen and the bolt parameters are automatically iteratively adjusted as described below. After the bolt loading iterations have converged, then the time steps proceed as usual.

  The bolt loading is performed at the beginning of the analysis, i.e. at time zero.

- The bolt feature can be used only in static and implicit dynamic analysis (Solution 601).

- A bolt iteration consists of the same process as is normally used for static analysis, namely the solution of the equilibrium equations as discussed in Section 6.2. In each iteration all elements, including bolt elements, contribute to the global force vector, and the increment in displacements, including the displacements of the bolt element nodes, is obtained.

  The remainder of the structure (that is all of the rest of the model, including bolts not being loaded) behaves exactly as usual. For example, plasticity can occur. Therefore, it is possible that the bolt iterations might not converge. In this case, the bolt loading can be divided into sub-steps, so that the change in bolt loads is smaller during each sub-step.

10.7.1 Beam-bolt element

- The beam-bolt element is implemented as an elastic Hermitian beam element with a modified axial force - length relationship. The formulation used for the beam-bolt elements can either be small displacements or large displacements. Any cross-section available for the beam element can be used. However, the material model must be linear elastic.

- The axial force - length relationship of the beam-bolt element is

$$F = k({}^{t}L - L_b) + F_b \qquad (10.7\text{-}1)$$

where F is the bolt axial force, k is the bolt axial stiffness

EA/L_b, ${}^tL$ is the bolt current length, L_b is the bolt adjusted length (defined below) and F_b is the bolt force corresponding to L_b. (Thermal effects are ignored in this discussion.) The bolt adjusted length is defined as $L_b = {}^0L - \Delta_b$, where ${}^0L$ is the bolt original length and Δ_b is the bolt shortening.

Using ${}^tL = {}^0L + {}^t\Delta$ where ${}^0L$ is the bolt initial length and ${}^t\Delta$ is the bolt axial displacement gives

$$F = k({}^t\Delta + \Delta_b) + F_b \qquad (10.7\text{-}2)$$

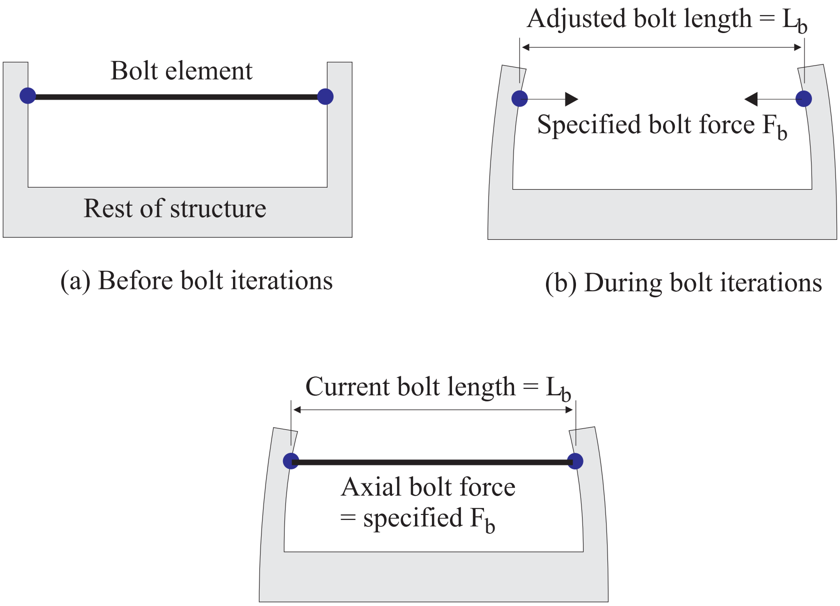
Initially (before the first bolt-loading is applied), F_b and Δ_b are zero, which means that $F = k\,{}^t\Delta$ and the element behaves exactly like a usual beam element.

- We now discuss the details of the bolt iterations for beam-bolt elements. For simplicity, in this discussion, we show only a single bolt element.

- The process of bolt tensioning is shown in Fig. 10.7-2. During the bolt iterations, the axial stiffness of the bolt element is set very small (to $f_1 k$, where $f_1 = 10^{-6}$), and the axial force in the bolt element is replaced by the user-specified axial force F_b. The bolt shortening Δ_b is determined from the condition $\Delta_b = -{}^t\Delta$ in which ${}^t\Delta$ is determined from the displacements of the bolt element nodes. The bolt iterations continue until equilibrium is satisfied and also the condition

$$\frac{\left|\Delta_b^{(ite)} - \Delta_b^{(ite-1)}\right|}{\max\left(\left|\Delta_b^{(ite-1)}\right|, \left|\Delta_b^{(ite)}\right|\right)} < 0.01$$

is satisfied. At this point, the bolt axial stiffness is restored to its normal value. Since ${}^t\Delta + \Delta_b = 0$, the bolt axial force F computed from (10.7-2) equals the user-specified axial force F_b.


Bolt element
Rest of structure
Adjusted bolt length = L_b
Specified bolt force F_b
Current bolt length = L_b
Axial bolt force
= specified F_b

(a) Before bolt iterations

(b) During bolt iterations

(c) After bolt iterations

Fig. 10.7-2: Bolt force loading

- For all other iterations, in which the bolt is not being adjusted, the bolt axial force - length relationship is given by (10.7-1), where F_b and Δ_b are in general non-zero and are held constant.

- In the beam-bolt element, no additional global equations are used to include the additional bolt variables Δ_b and F_b.

10.7.2 3D-bolt

- A set of 3-D solid elements can be combined into a bolt (called a 3D-bolt). The elements can use any formulation (small or large displacements, small or large strains), any material model and any number of nodes per element. The mixed u/p formulation or the incompatible modes formulation can also be used.

The geometry of the bolt is also arbitrary. The element sides and faces can be straight or curved.

- The basic ideas used in the 3D-bolt are shown in Fig 10.7-3. This figure shows a very simple bolt model of four elements, with the ends of the bolt fixed. For simplicity, it is assumed that the Poisson's ratio is zero, so that the cross-section of the bolt doesn't change during the deformations.

Fig. 10.7-3(a) shows the 3D-bolt along with its bolt plane. This bolt plane is an additional input for the 3D-bolt. As seen, the bolt plane determines the bolt direction and the bolt split faces. (For more information about the bolt plane, see Section 10.7.5.)

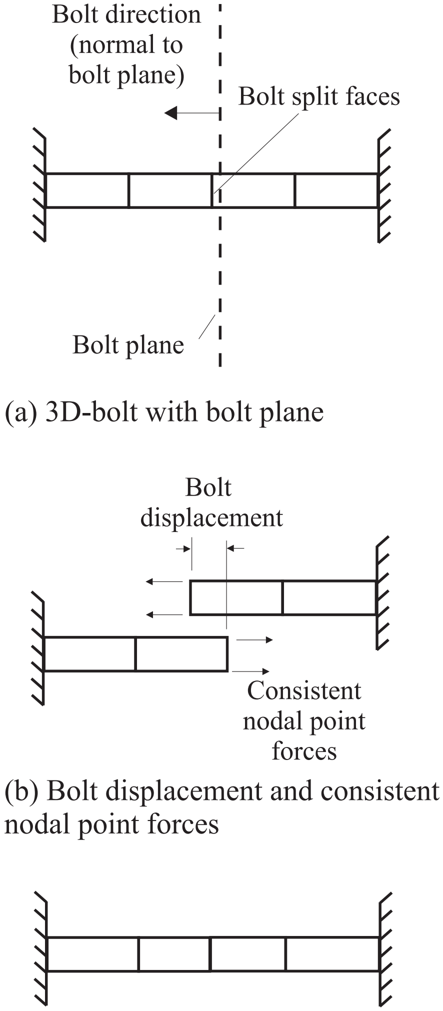
Fig 10.7-3(b) shows an incompatible displacement, called the bolt displacement, applied to the nodes on the bolt split faces. Because the bolt model is fixed at its ends, the elements must increase in length, and, to satisfy equilibrium, the elements must all be under uniform tension. The nodal point forces acting at the nodes on the bolt split faces are combined into a single bolt force.

Fig 10.7-3(c) shows the 3D-bolt after deformations, as visualized during postprocessing. The elements appear to be different lengths because the bolt displacement is not included in the plot. However, the strains are the same in all elements.

In this example, it is clear that the bolt force is determined by the bolt displacement, and that the bolt force increases as the bolt displacement increases.

- Fig 10.7-4 shows the same problem, but with one end of the bolt free. Now when the bolt displacement is applied, the elements on the free end move stress-free, and the resulting plot is shown in Fig 10.7-4(c). This example justifies the use of the term "bolt displacement".

When the bolt end is free, it is clear that the bolt force is zero for any value of the bolt displacement.

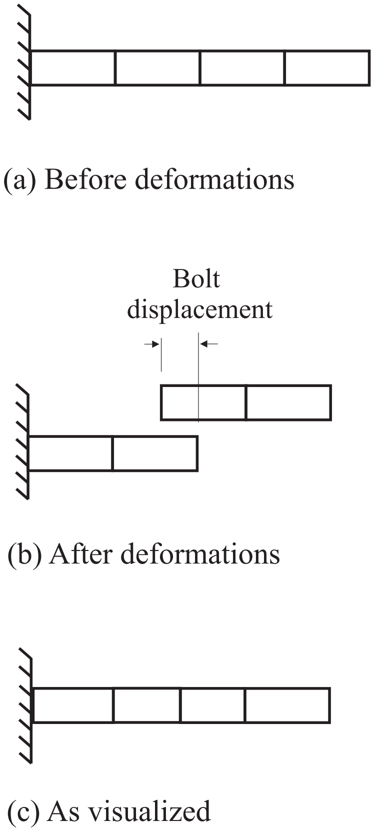


(a) 3D-bolt with bolt plane

(b) Bolt displacement and consistent nodal point forces

(c) 3D-bolt, as visualized

Fig. 10.7-3: 3D-bolt, fixed ends



(a) Before deformations

(b) After deformations

(c) As visualized

Fig. 10.7-4: 3D-bolt, free end

- In the general case, a 3D-bolt has two additional quantities: the bolt displacement (denoted v_b) and the bolt force (denoted F_b). These quantities are used in the global solution of equations as follows:

1) Bolt force specified:

$$\begin{bmatrix} \mathbf{K} & \mathbf{KUB} \\ \mathbf{KBU} & \mathbf{KBB} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{u} \\ \Delta v_b \end{bmatrix} = \begin{bmatrix} \mathbf{R}-\mathbf{F} \\ R_b - F_b \end{bmatrix} \tag{10.7-3}$$

where $\mathbf{K}$ is the usual global stiffness matrix, $\mathbf{R}$ is the usual external loads vector, $\mathbf{F}$ is the usual internal loads vector, $\Delta\mathbf{u}$ is the usual increment in displacements, $\mathbf{KBU, KUB, KBB}$ are coupling matrices, and R_b is the specified bolt force. During the equilibrium iterations, both $\Delta\mathbf{u}$ and Δv_b are obtained, and at equilibrium, $F_b = R_b$. Thus the value of v_b that results in the specified R_b is obtained by this procedure.

In addition to the usual convergence criteria, the convergence criterion

$$\frac{\left|R_b - F_b\right|}{\max\left(\left|R_b\right|,\left|F_b\right|\right)}<0.01$$

is used.

2) Usual equilibrium iterations in which the bolt is not being adjusted:

$$\begin{bmatrix} \mathbf{K} & \mathbf{KUB} \\ \mathbf{KBU} & \alpha\mathbf{KBB} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{u} \\ \Delta v_b \end{bmatrix} = \begin{bmatrix} \mathbf{R}-\mathbf{F} \\ 0 \end{bmatrix} \qquad (10.7\text{-}4)$$

where α is very large. The last row of this system of equation results in $\Delta v_b = 0$, so the global equilibrium equation reduces to the usual case.

- When there are several 3D-bolts, then each bolt contributes an additional equation to the global system of equations. There is no limit to the number of 3D-bolts that can be present in the model.

10.7.3 Usage of bolt loadings

- The bolt force is defined using the BOLTFOR entry. The bolt preload set must be selected via the BOLTLD case control command.

- Bolt force iterations can be performed in one step (default) or in a number of “bolt steps” (set via the BOLTSTP parameter in

NXSTRAT). This feature should be used if the bolt conditions are too severe to converge in one step.

- During bolt loading in implicit dynamic analysis, the program temporarily switches to static analysis (unless the low-speed dynamics option is employed). Bolt iterations that are performed before the first time step are always static or low-speed dynamic, never truly dynamic, so as to not introduce any dynamic effects during bolt tightening such as ringing. Note that in practice bolt tightening is typically a quasi-static process.

- If the ATS method is used, it is also applied to the bolt loading procedure.

- Bolt loading can be used along with both one-way or fully coupled TMC solutions.

- Damping can be applied to the model during the bolt iterations before the first time step using the NXSTRAT BOLTDAMP entry. This can be useful to stabilize any rigid-body motions that may be present before contact is established. It is possible to apply damping only during the bolt iterations, not to the time stepping, such that the solution at the end of the time step is free of any damping.

  The default is BOLTDAMP=0 (no bolt damping).

- In static analysis, a static analysis is performed during the bolt iterations if BOLTDAMP = 0. If BOLTDAMP > 0, a low-speed dynamic analysis is performed during the bolt iterations before the first time step with the specified bolt damping factor and an inertia factor = 1, and with the Newmark time integration method.

- In low-speed dynamic analysis, a low-speed dynamic analysis is performed during the bolt iterations with the same settings as during the time steps if BOLTDAMP = 0. If BOLTDAMP > 0, the bolt damping factor overrides the low-speed dynamics damping factor during the bolt iterations.

- In dynamic analysis, a static analysis is performed during the bolt iterations before the first time step if BOLTDAMP = 0. If BOLTDAMP > 0, a low-speed dynamic analysis is performed during the bolt iterations before the first time step with the specified bolt damping factor and an inertia factor = 1, and with the same time integration method as for the dynamic analysis during the time steps. Hence, in dynamic analysis, bolt damping is often required to remove rigid body modes during the bolt iterations before the first time step.

- Table 10.7-1 shows the behavior of bolt damping.

Table 10.7-1: Behavior of bolt damping

	BOLTDAMP = 0	BOLTDAMP > 0
True dynamics without ATS: Solution 601,129 NXSTRAT AUTO=0	Bolt iterations use static analysis without low-speed dynamics and without ATS.	Bolt iterations use low-speed dynamics with the user-specified time integration method and the specified bolt damping factor, with ATS forced on.
True dynamics with ATS: Solution 601,129 NXSTRAT AUTO=1	Bolt iterations use static analysis without low-speed dynamics, but with ATS.	Bolt iterations use low-speed dynamics with the user-specified time integration method and the specified bolt damping factor, with ATS.
Statics without ATS and without low-speed dynamics: Solution 601,106 NXSTRAT AUTO=0, ATSLOWS=0	Bolt iterations use static analysis without low-speed dynamics and without ATS.	Bolt iterations use low-speed dynamics with the Newmark method and the specified bolt damping factor, with ATS forced on.
Statics with ATS and without low-speed dynamics: Solution 601,106 NXSTRAT AUTO=1, ATSLOWS=0	Bolt iterations use static analysis without low-speed dynamics and with ATS.	Bolt iterations use low-speed dynamics with the Newmark method and the specified bolt damping factor, with ATS.
Statics with ATS and low-speed dynamics (ATS is always on for low-speed dynamics). Solution 601,106 NXSTRAT ATSLOWS=1	Bolt iterations use low-speed dynamics with the same settings as during the time steps, and with ATS.	Bolt iterations use low-speed dynamics with the same time integeration method as during the time steps, but with the specified bolt damping factor, with ATS.

10.7.4 Usage of beam-bolts

- The beam elements that make up the bolt are selected using the BOLT entry with ETYPE=1. All of the elements must have the same PID.

- If more than one element has the same PID, it is assumed that the elements are all connected sequentially and all have the same length, as shown in Fig. 10.7-5.

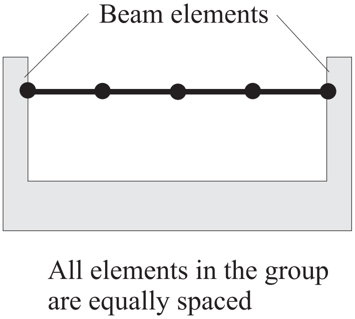


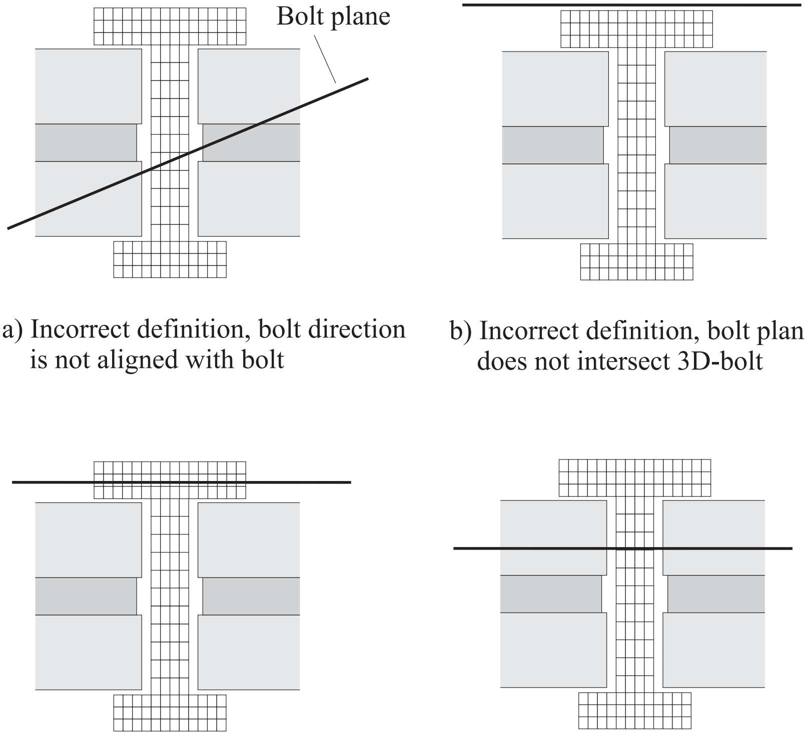
Fig. 10.7-5: Modeling of a bolt using several beam elements.

10.7.5 Usage of 3D-bolts

- The 3-D solid elements that make up the bolt are selected using the BOLT entry with ETYPE=3. All of the elements must have the same PID.

- The 3D-bolt also uses a bolt plane as part of its definition. The bolt plane gives the bolt direction and the approximate location of the bolt split faces. The bolt direction is used to determine the direction of the bolt displacement and the direction of the bolt force. The bolt plane is used to determine the bolt split faces. Note that the bolt split faces do not necessarily coincide with the bolt plane. Rather, the program determines the bolt split faces to lie "near" the bolt plane.

- It is necessary for the bolt plane to intersect the elements, in such a way that the bolt direction is aligned with the bolt, and the entire bolt force is transmitted across the bolt plane. Fig 10.7-6 shows some examples of incorrect and correct definitions.



a) Incorrect definition, bolt direction is not aligned with bolt

b) Incorrect definition, bolt plane does not intersect 3D-bolt

c) Incorrect definition, entire bolt force is not transmitted across bolt plane

d) One possible correct definition

Fig. 10.7-6: Examples of bolt plane definition

- The bolt-plane is defined using the IDIR parameter of the BOLT entry:

  - IDIR=1,2,3: the bolt plane normal is aligned with the x, y or z directions of the coordinate system referenced by CSID.
  - IDIR=blank (default): the program automatically determines the bolt plane as follows (Fig 10.7-7). First the centroid and moments of inertia of the 3D-bolt are obtained (assuming unit density). Then the bolt direction is obtained as the direction of the minimum principal moment of inertia, and the bolt plane passes through the centroid.

The inertial properties are determined using numerical integration within the finite element mesh. Because the mesh itself is used, the inertial properties might not exactly match the properties corresponding to the underlying geometry. For this reason, if the direction of minimum principal moment of inertia is within 1 degree of a global coordinate direction, the global coordinate direction is used instead.

This algorithm gives a reasonable choice for the bolt plane in many cases.

- The program outputs the bolt direction and bolt cross-sectional area for each bolt during data file generation. This information can be used to confirm that the bolt plane is defined correctly.

- When the bolt force is specified for a system in which one end of the bolt is free (for example, the system shown in Fig 10.7-4), the global system of equations (10.7-3) is singular. This is physically correct as there is no non-zero bolt force that can satisfy equilibrium.

  In this case, stiffness stabilization can be used to prevent numerical difficulties.

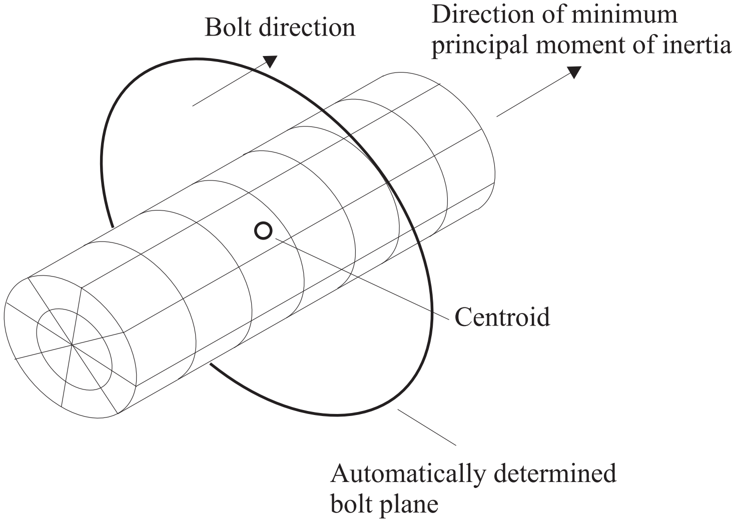


Fig. 10.7-7: Automatic determination of bolt-plane

- During large displacement analysis, the bolt direction can change during the solution process. The program computes the updated bolt direction using three nodes on the bolt split faces; as these three nodes move, the bolt direction is correspondingly updated.

- It is recommended that the meshing of the 3D-bolts be compatible, in the sense that all of the adjacent internal element faces must match.

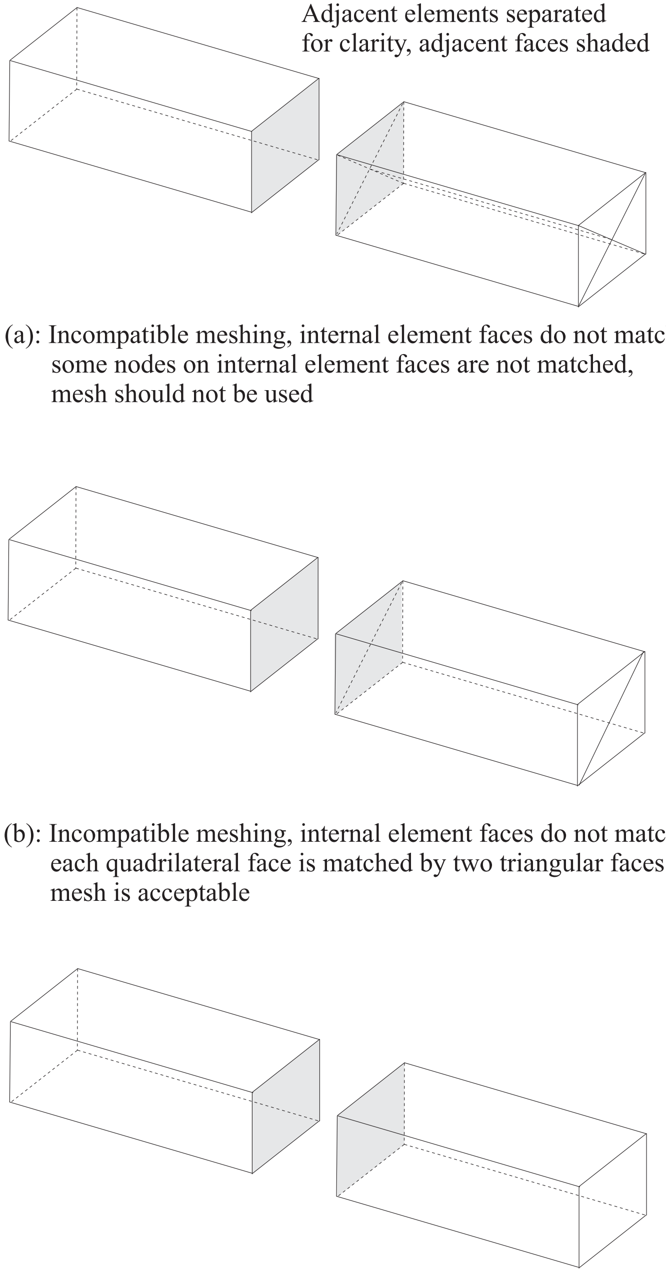
  The reason for this recommendation is that the compatibility of the mesh is assumed when determining the bolt split faces. If the mesh is incompatible, the bolt split faces might not be determined correctly.

  Three different cases are shown in Fig 10.7-8. In the first case (Fig 10.7-8a), the mesh is incompatible, and there is an unmatched node on one of the adjacent element faces. If this mesh was used in a 3D-bolt group, Solution 601 would not be able to determine the bolt split faces.

  In the second case (Fig. 10.7-8b), the mesh is, strictly speaking, incompatible; however Solution 601 recognizes that two triangular faces can match an adjacent quadrilateral face, so Solution 601 considers the mesh to be compatible. The second case occurs often in practice.

  In the third case (Fig 10.7-8c), the mesh is totally compatible.

- It is possible that meshes in which some faces are compatible and other faces are incompatible might cause Solution 601 to determine the bolt split faces incorrectly. However Solution 601 checks the bolt split faces for correctness: any node on a bolt split face must be attached to elements on both sides of the bolt plane, and any node attached to elements on both sides of the bolt plane must be on a bolt split face. Solution 601 gives an error message if these checks are not passed.



Adjacent elements separated for clarity, adjacent faces shaded

(a): Incompatible meshing, internal element faces do not match,
some nodes on internal element faces are not matched,
mesh should not be used

(b): Incompatible meshing, internal element faces do not match, but
each quadrilateral face is matched by two triangular faces,
mesh is acceptable

(c): Compatible meshing, internal element faces match, mesh is acceptable

Fig. 10.7-8: Incompatible and compatible meshing for 3D-bolt

10.7.6 Modeling issues

- One modeling pitfall is illustrated in Fig. 10.7-9. The figure shows a bolt with contact conditions. It is intended that, after the bolt force is specified, the bolt be in contact with the rigid target. However, this model will not work as intended, for the following reason. During the bolt force iterations, the bolt provides no stiffness between the top and bottom of the bolt (points A and B in the figure). And since the bolt is not initially in contact with the target, the contact also does not provide stiffness. Therefore, in the first equilibrium iteration, point A moves downwards as a rigid body under the prescribed bolt force, and this motion is very large since there is no stiffness. The remaining equilibrium iterations will probably not converge.

To prevent this effect from occurring, make sure that the contact is established before beginning bolt force iterations. For example, set up the model so that there is a very small overlap between the contacting parts.

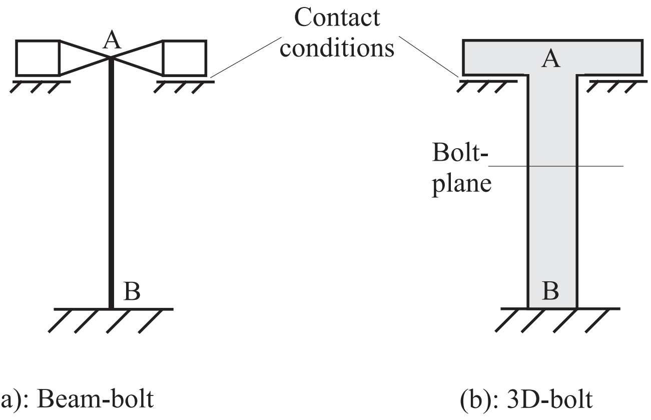


Fig. 10.7-9: Bolts and contact example

10.8 Direct matrix input (Solution 601 only)

- Advanced Nonlinear Solution supports direct matrix input using the K2GG, B2GG, M2GG case control commands and the DMIG bulk data entry.

10.9 Parallel processing

- Solution 601 supports parallel processing on all supported platforms, for the in-core and out-of-core sparse solvers.

- Solution 701 also supports parallel processing on all supported platforms.

- Parallelized assembly of the global system matrices is supported on all platforms.

- To benefit from parallel element assembly, groups of elements with the same property ids must be divided into subgroups. The number of subgroups should be equal to or greater than the number of processors (preferably a multiple of the number of processors). This is set via the NSUBGRP parameter in NXSTRAT.

10.10 Usage of memory and disk storage

Solution 601
Depending on the size of the problem and the memory allocated to Solution 601, it can perform the solution either in-core (entirely within real or virtual memory) or out-of-core (reading from and writing to disk files). Whenever possible the solution is performed in-core.

- The program memory usage is divided into two parts:

  ▸ memory usage not considering the equation solver
  ▸ memory usage of the equation solver

Each of these parts can be performed in-core or out-of-core, as described below.

- *Memory usage not considering the equation solver:* There are two options:

  ▸ The global system matrices and element information are all stored in-core (IOPTIM=3) .
  ▸ The global system matrices are stored in-core, and the element information is stored out-of-core (IOPTIM=2).

The program automatically chooses the appropriate option based on the size of the problem and the available memory. The chosen option is reported in the .f06 file as the value of IOPTIM.

- *Memory usage of the equation solver*

    - Sparse solver and 3D-iterative solver: Each of these solvers can either run in-core or out-of-core. The program chooses whether the solver runs in-core or out-of-core, based on the size of the problem and the available memory.
    - Iterative multigrid solver. The solver always runs in-core. The out-of-core solution procedure would take an unreasonably long time in most cases.

Solution 701

Solution 701 can only run in-core. Enough memory must be provided.

Additional reading

This section lists some references related to Solution 601 and 701.

Books
K.J. Bathe, *Finite Element Procedures, 2nd ed.*, Cambridge, MA, Klaus-Jürgen Bathe, 2014.

D. Chapelle and K.J. Bathe, *The Finite Element Analysis of Shells - Fundamentals*, Springer, 2nd ed, 2011.

M.L. Bucalem and K.J. Bathe, *The Mechanics of Solids and Structures - Hierarchical Modeling and the Finite Element Solution*, Springer, 2011.

Web
Additional references, including downloadable papers, can be found at the MIT web site of Prof. K. J. Bathe:

http://meche.mit.edu/people/faculty/kjb@mit.edu

Index