Academic Program

△ Altair | University



Introduction to Nonlinear Finite Element Analysis

using OptiStruct

Released: 2nd Edition 11/2018





Table of Contents

About This Book	7
1 Introduction to Nonlinearity	13
1.1 Sources of Nonlinearity	14
1.2 Features of Nonlinear Problems	15
1.3 When to Choose a Nonlinear Analysis?	17
1.4 Small vs Large Displacement Nonlinear Analysis	18
2 Solution Techniques	21
2.1 Incremental-Iterative Procedure	22
2.2 Automatic Increment Size Control	23
2.3 Convergence Criteria	24
2.4 Analysis Controls	24
2.5 Path Dependent Loading	31
2.6 Restarting a Nonlinear Analysis	33
2.7 Energy Output for Nonlinear Analysis	34
2.8 On the Fly Output of Displacement Results for Nonlinear Analysis	34
3 Material Nonlinearity	38
3.1 Elastic-Plastic Material	39
3.2 Hyperelastic Material	55
4 Geometric Nonlinearity	64
4.1 Large Deflection	64
4.2 Large Strain	66
4.3 Non-Uniqueness of the Solution: Snap-Through	68

🛆 Altair | University

	4.4 Follower Forces	69
5	5 Contact Nonlinearity	77
	5.1 Contact Discretization	77
	5.2 Contact Types & Properties	90
	5.3 Contact Utilities	99
	5.4 Contact Setup – General Steps	117
	5.5 Analysis Controls for Contacts	123
	5.6 Contact Output	124
	5.7 General Tips for Contact Setup	127
	5.8 Tutorial : Nonlinear Analysis of an Interference Fit	127
	5.9 Capstone Tutorial Part 1: Setting Up Frictional Contact in a Bolted Pipe	140
6	6 Bolt Pretension	145
	6.1 Pretensioning Process	145
	6.1 Pretensioning Process	145 146
	 6.1 Pretensioning Process 6.2 Bolt Pretension in OptiStruct 6.3 Capstone Tutorial Part 2: Creating Pretensioned 3D Bolts in the Bolted 	145 146 151
7	 6.1 Pretensioning Process 6.2 Bolt Pretension in OptiStruct 6.3 Capstone Tutorial Part 2: Creating Pretensioned 3D Bolts in the Bolted 7 Analyzing Gaskets 	145 146 151 156
7	 6.1 Pretensioning Process 6.2 Bolt Pretension in OptiStruct	145 146 151 156 156
7	 6.1 Pretensioning Process	145 146 151 156 156 158
7	 6.1 Pretensioning Process	145 146 151 156 158 160
7	 6.1 Pretensioning Process	145 146 151 156 158 160 161
7	 6.1 Pretensioning Process	145 146 151 156 158 160 161 167
7	 6.1 Pretensioning Process	145 146 151 156 156 160 161 167 168
7	 6.1 Pretensioning Process	145 146 151 156 156 158 160 161 167 168 169

🛆 Altair | University

9 Tips and Tricks	
9.1 Essential Steps To Start With Nonlinear FEA	
9.2 Debugging	179
9.3 Useful Tutorials and Industry Examples	
10 Appendix	



About This Book

This study guide aims to provide a fundamental to advanced approach into the exciting and challenging world of **NonLinear Analysis**. The focus will be on aspects of NonLinear Dynamic Analysis. As with our other eBooks we have deliberately kept the theoretical aspects as short as possible.

The tool of choice used in this book is **OptiStruct. Altair ® OptiStruct®** is an industry proven, modern structural analysis solver for linear and nonlinear structural problems under static and dynamic loadings. OptiStruct is used by thousands of companies worldwide to analyze and optimize structures for their strength, durability and NVH (noise, vibration and harshness) characteristics.

In this eBook, we will describe in some detail:

- Nonlinear Materials
- Nonlinear Geometry
- Contacts
- Bolt Pretension
- Gasket Analysis
- Nonlinear Direct Transient Analysis

Please note that a commercially released software is a living "thing" and so at every release (major or point release) new methods, new functions are added along with improvement to existing methods. This document is written using HyperWorks Solvers 2018.0. Any feedback helping to improve the quality of this book would be very much appreciated.

Thank you very much. Dr. Matthias Goelke On behalf of the **Altair University Team**



Model Files

The models (geometry) referenced in this book can be downloaded using the link provided in the exercises, respectively. These model files are based on HyperWorks OptiStruct 2017. You can download the HyperWorks 2017 model files from <u>here ...</u> These models are compatible with the latest version.

Software

Obviously, to practice you need to have access to HyperWorks. As a student, you are eligible to download and install the free Student Edition:



<u>https://altairuniversity.com/free-hyperworks-2017-student-edition</u> Note: From the different software packages listed in the download area, you just need to download and install HyperWorks.

Support

In case you encounter issues (during installation but also how to utilize Altair HyperWorks) post your question in the moderated Support Forum <u>https://forum.altairhyperworks.com</u> It's an active forum with several thousands of

posts - moderated by Altair experts!





Free eBooks

In case you are interested in more details about the "things" happening in the background we recommend our free eBooks

https://altairuniversity.com/free-ebooks-2





🛆 Altair | University

Learning & Certification Program

Many different eLearning courses are available for free in the

Learning & Certification Program.

For instance, you may find this eLearning course helpful:

Learn Mechanics of Solids Fundamentals



https://certification.altairuniversity.com/course/view.php?id=66

This course is to introduce the subject of mechanics of solids with a focus on theory. Of course, there is also an eLearning course about HyperWorks available

For OptiStruct Nonlinear Analysis, the prerequisite (or recommended) course is Structural Analysis:

Learn Structural Analysis with OptiStruct



https://certification.altairuniversity.com/course/view.php?id=71



Acknowledgement

"If everyone is moving forward together, then success takes care of itself" Henry Ford (1863 -1947)

A very special **Thank You** goes to all the many colleagues who contributed in different ways:

Gabriel Stankiewicz for writing the first draft of the book. Sanjay Nainani for the review of manuscript for the first edition. Rahul Rajan for editing, reviewing & updating the new features in the second edition of this book. Prakash Pagadala for helpful discussions and explanations.

Rahul Ponginan for overall review of the book. Premanand Suryavanshi, Priyanka Nagraj, Pranav Harikrishnan and Nimisha Srivastava for the valuable support. For sure, your feedback and suggestions had a significant impact on the "shape" and content of this book.

Sean Putman and Elizabeth White for all the support, especially with respect to the video recordings used in here.

Junji Saiki, Warren Dias, Ujwal Patnaik and Girish Mudigonda from OptiStruct team.

Mike Heskitt, Nelson Dias, Pavan Kumar, Rajneesh Shinde & Dev Anand for all the support.

The entire Altair HyperWorks Documentation Team for putting together 1000's of pages of documentation.

Lastly, the entire Altair OptiStruct team deserves huge credit for their passion & dedication! It is so exciting to see how OptiStruct has evolved throughout the last couple of years.

🛆 Altair | University

Disclaimer

Every effort has been made to keep the book free from technical as well as other mistakes. However, publishers and authors will not be responsible for loss, damage in any form and consequences arising directly or indirectly from the use of this book. © 2018 Altair Engineering, Inc. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, or translated to another language without the written permission of Altair Engineering, Inc. To obtain this permission, write to the attention Altair Engineering legal department at:

1820Big Beaver, Troy, Michigan, USA, or call +1-248-614-2400.

1 Introduction to Nonlinearity

In this book, we discuss the practical aspects of Nonlinear Finite Element Analysis. But how do we know that our problem is nonlinear? The best way is to look at the load-displacement response of one or more characteristic load introduction points. As discussed in one of our earlier books on Type of Analyses, when the structural response (deformation, stress and strain) is linearly proportional to the magnitude of the load (force, pressure, moment, torque, temperature etc.), then the analysis of such a structure is known as linear analysis. When the load to response relationship is not linearly proportional, then the analysis falls under nonlinear analysis (see figure below). For example, when a compact structure made of stiff metal is subjected to a load relatively lower in magnitude as compared to the strength of the material, the deformation in the structure will be linearly proportional to the load and the structure is known to have been subjected to linear static deformation. But most of the time either the material behavior is not linear in the operating conditions or the geometry of the structure itself keeps it from responding linearly. Due to the cost or weight advantage of nonmetals (polymers, woods, composites etc.) over metals, nonmetals are replacing metals for a variety of applications. These applications have nonlinear load to response characteristics, even under mild loading conditions. Also, the structures are optimized to make most of its strength, pushing the load level so close to the strength of the material, that it starts behaving nonlinearly. In order to accurately predict the strength of the structures in these circumstances, it is necessary to perform a nonlinear analysis.

As discussed in the earlier chapter, the stiffness matrix relating to the load and response is assumed to be constant for static analysis; however, all the real-world structures behave nonlinearly. The stiffness matrix consists of geometric parameters such as length, cross sectional area and moment of inertia, etc. and material properties such as elastic modulus, rigidity modulus etc. The static analysis assumes that these parameters do not change when the structure is loaded. On the other hand, nonlinear static analysis considers the changes in these parameters as the load is

13



applied to the structure. These changes are accommodated in the analysis by rebuilding the stiffness matrix with respect to the deformed structure (i.e. altered properties) after each incremental load application. Although, the world is nonlinear, it should be mentioned that in many cases the assumption of linearity is valid and in that way a linear analysis can be done instead. Also, from a computation point of view, it is a much less expensive approach.

1.1 Sources of Nonlinearity

Nonlinear analysis should be introduced when the linear solution is not valid anymore, this can be caused by any one of the three main sources of nonlinearity:

• Material nonlinearity

Material nonlinearity is caused by plasticity i.e. material stiffness changes as the strain increases.



Comparison of linear and nonlinear response

Material nonlinearity involves the nonlinear behavior of a material based on current deformation, deformation history, rate of deformation, temperature, pressure, and so on.

• Geometric nonlinearity

In analyses involving geometric nonlinearity, changes in geometry as the structure deforms are considered in formulating the constitutive and equilibrium equations. Many engineering applications require the use of large deformation analysis based on geometric nonlinearity. Applications such as metal forming, tire analysis, and medical



device analysis. Small deformation analysis based on geometric nonlinearity is required for some applications, like analysis involving cables, arches and shells. Such applications involve small deformation, except finite displacement or rotation.

• Boundary and Contact Nonlinearity

Presence of contact definition between two bodies imposes an additional condition that they do not penetrate each other. During relative displacement, the bodies may come in contact which in fact happens to be an additional displacement constraint for them, this causes a nonlinearity called boundary nonlinearity. During large displacements, we may also require that pressure is normal to the moving surface, so that it moves with respect to the global coordinate system. This is also classified as boundary nonlinearity.



Images after K.J. Bathe, Finite Elemente Methoden

1.2 Features of Nonlinear Problems

Nonlinear problems need consideration of several features that are especially characteristic:

• Non-uniqueness: For an applied load P there may be no solution, one solution, or many displacement solutions **u**

A great example is snap through behavior of a shallow curved plate. Notice that for certain **P** values, there can even be 3 different solutions of displacement u, such an example presents a **history-dependent** behavior.

△ Altair | University



- Non-scalability: If an applied load P causes displacement u, then an applied load x times P may not cause displacement x time u,
- No superposition: If an applied load P causes displacement u and load F causes displacement d, then P+F may not cause displacement u+d

Overview





1.3 When to Choose a Nonlinear Analysis?

Choice of a proper analysis type is sometimes not done correctly due to insufficient knowledge about the problem being examined. Often a simpler solver solution is chosen and complexity of the situation is neglected, such as choosing linear analysis when the problem needs plasticity considerations. Below a chart is presented that shows how one should investigate the type of analysis that is applicable for a structural problem. Five types of analyses are presented that in fact are somehow similar - it is sometimes parameters, like velocity, strains and forces that are critical to a proper choice of solution from the below presented analyses.



Difference between Nonlinear Quasi-Static Analysis and Nonlinear Transient Analysis:

The Nonlinear Quasi-Static Analysis differs from a transient analysis due to its assumption that time and hence inertial forces and momentums are negligible. One may even wonder why we compare nonlinear analysis to time-dependent solutions. As explained before, the solution for nonlinear analysis also considers **history**-



dependent behavior. Therefore, both solutions consider an incremental approach, but nonlinear quasi-static analysis increments are loading-based instead of time-based. This topic will be discussed further in the next chapter.

1.4 Small vs Large Displacement Nonlinear

Analysis

OptiStruct can perform two types of Quasi-Static analyses, depending on the types of nonlinearities available in the model. Below is a table that describes most relevant differences for small and large displacement analyses.

Туре	Small Displacement Nonlinear	Large Displacement Nonlinear		
	Analysis	Analysis		
Overview	Finds application mainly when	Accurately solves problems		
	the nonlinearity is limited to	containing large deformations,		
	contact and small strains. This	rotations, displacements.		
	type will not accurately solve	Applicable to all types of		
	for larger displacements,	nonlinearity.		
	significant geometry change			
	and when following forces are			
	required.			
Remarks	Active by default (LGDISP,0)	Must be activated by changing		
		the LGDISP control card to 1 .		
		Needs to be used together with		
		hash assembly option		
		(PARAM,HASHASSM,YES).		
Strain range	0 < ε < 5% (recommended)	ε > 5% (recommended)		
Material	Supported only for MATS1	Supported for both MATS1		
nonlinearity	elastic-plastic material.	elastic-plastic and MATHE		
		hyperelastic material of		
		polynomial form.		



Туре	Small Displacement Nonlinear	Large Displacement Nonlinear
	Analysis	Analysis
Contact	Supported	Supported
nonlinearity	(However, no update of	
	gap/contact element locations	
	or orientation due to geometry	
	change)	
Follower	Not supported	Supported
loads		
Supported	All generally used element	Solid elements, Shell (first order
elements	types are supported	only) elements, CBUSH, RROD,
		CELASi, RBAR, RBE2, and RBE3.
		Composites using PCOMP,
		PCOMPG, and PCOMPP entries
		are supported

🛆 Altair | University

2 Solution Techniques

Generally, there are two main techniques to solve nonlinear problems:

- **Explicit**, which is used to solve highly nonlinear dynamic problems implemented in RADIOSS,
- Implicit, which assumes that the simulated process runs infinitely slowly (Nonlinear Quasi-Static Analysis) or in a longer period of time (Nonlinear Direct Transient Analysis).

Method	Implicit	Explicit
Solver	OptiStruct, RADIOSS	RADIOSS
Equilibrium	Static equilibrium (Quasi-Static)	Dynamic equilibrium
equations	Dynamic equilibrium (Transient)	
Stability	Unconditionally stable (stability	Conditionally stable (stability
	of the system does not depend	of the system depends on an
	on increment size)	increment time step, which in
		turn depends on smallest
		element size and material
		properties)
Procedure	Incremental and iterative (each	Incremental
	load increment consists of	
	several iterations until	
	convergence is reached)	
Time step	Larger time steps are used, since	Much smaller time steps -
size	after each increment Newton-	small enough to assume the
	Raphson iterations are	solution is accurate, no
	performed until the solution is	iterations are performed.
	convergent enough.	

Key differences between these two methods are presented in the table:



Application	Static nonlinear events	Highly dynamic events,
	Relatively slow speed events	occurring in relatively short
	(Transient) or allowing	period of time
	assumption of fully static (Quasi-	
	Static) features.	

To solve Nonlinear Quasi-Static problems, OptiStruct applies implicit method.

2.1 Incremental-Iterative Procedure

Although in Nonlinear Quasi-Static Analysis time is not considered and we omit any dynamic effects, the solution is always **history-dependent** (non-uniqueness of the solution), hence, we need to apply loading in small increments.

Since the increments are much larger than in the explicit method accuracy is not achieved and therefore several iterations need to be performed to achieve certain, user-defined convergence criteria. Such iterations may sometimes be time consuming, but on the other hand the increment size can be even a few orders bigger than for explicit. In the figure below, an increment calculation procedure is shown.



Consider a nonlinear problem:

$$L(u) = f$$



Where, u is the displacement vector, f is the global load vector, and L(u) is the nonlinear response of the system (nodal reactions). Note that for a linear problem, L(u) would simply be Ku (as described in Linear Static Analysis). Application of Newton's method to this equation leads to an iterative solution procedure:

$$\mathbf{K}_{n} \Delta \mathbf{u}_{n} = \mathbf{R}_{n}$$
$$\mathbf{u}_{n+1} = \mathbf{u}_{n} + \Delta \mathbf{u}_{n}$$

Where,

$$\mathbf{K}_{n} = \left(\frac{\partial \mathbf{L}(\mathbf{u})}{\partial \mathbf{u}}\right)_{\mathbf{u}_{n}}$$
$$\mathbf{R}_{n} = \mathbf{f} - \mathbf{L}(\mathbf{u}_{n})$$

In the above formulas, Kn represents a "slope" matrix, defined as a tangent to the L(u) curve at a point un, and Rn is the nonlinear residual. Repeating this procedure iteratively, under certain convergence conditions, leads to systematic reduction of residual Rn and hence, convergence.

Note: Specifically, only for small displacement nonlinear analysis, that the above scheme is somewhat modified to an equivalent format wherein, instead of calculating Δ un, the new solution un+1 is directly obtained:

$$\mathbf{K}_{n}\mathbf{u}_{n+1} = \mathbf{R}_{n} + \mathbf{K}_{n}\mathbf{u}_{n}$$

This form is readily produced by adding Kn un to both sides of Newton's equation, and has certain advantages in practical implementations.

2.2 Automatic Increment Size Control

Choosing a suitable time increment is very important. In OptiStruct, an automatic time increment control is available. It should be suitable for a wide range of nonlinear problems and, in general, is a very reliable approach. In future OptiStruct versions, additional user control options to set up automatic time increments may be provided.



The automatic time increment control functionality measures the difficulty of convergence at the current increment. If the calculated number of iterations is equal to optimal number of iterations for convergence, OptiStruct will proceed with the same increment size. If a lesser number of iterations is required to achieve convergence, the increment size will be increased for next increment. Similarly, if it is determined that too many iterations are required, the current increment will be attempted again with a smaller increment size.

2.3 Convergence Criteria

As mentioned before, the iteration process for a single increment is repeated until certain convergence criteria are met, by default there are three convergence criteria activated, so that means all of them must be met simultaneously. They are: Relative error in displacement EU (printed as EUI in OUT file):

$$E_U = \frac{q}{1-q} \frac{\|A \cdot \Delta u\|}{\|A \cdot u\|} \text{ where } \|A \cdot u\| = \sum_i |A_i u_i| \text{ and } q = \frac{\|\Delta u_n\|}{\|\Delta u_{n-1}\|}$$

Relative error in terms of load EP (printed as EPI in OUT file):

$$\mathbf{E}_{\mathbf{P}} = \frac{\|\mathbf{R} \cdot \mathbf{u}\|}{\|\mathbf{P} \cdot \mathbf{u}\|}$$

Relative error in work EW (printed as EWI in OUT file):

$$\mathbf{E}_{\mathbf{W}} = \frac{\|\mathbf{R} \cdot \Delta \mathbf{u}\|}{\|\mathbf{P} \cdot \mathbf{u}\|}$$

For more information regarding convergence criteria, please refer to Help documentation.

2.4 Analysis Controls

The incremental-iterative procedure due to its complexity needs to be controlled by several parameters. Hence, user can specify them through load collectors with proper card images, which are later referenced by NLQS load step. The basic card image is NLPARM, which controls the main parameters. Besides that, user can additionally



specify NLADAPT to define more detailed time step and convergence controls. NLOUT allows one to request for additional results to an h3d file.

2.4.1 Main Parameters - NLPARM

Main parameters that control the Nonlinear Quasi-Static Analysis load step are defined within NLPARM card, which is activated as a Load Collector. Some of the most useful parameters, referenced in this card, are shown below (parameters which are important to understand during the first steps of NL analysis are presented in bold letters):

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
NLPARM	ID	NINC	DT		KSTEP	MAXITER	CONV		
	EPSU	EPSP	EPSW			MAXLS		LSTOL	
		TTERM							

Where:

ID	Identification number (must be unique)
NINC	Number of implicit load sub-increments.
DT	Initial load increment
MAXITER	Limit on number of implicit iterations for each load increment.
CONV	Flags to select implicit convergence criteria.
EPSU, EPSP,	Error tolerance for displacement (U), load (P), and work (W)
EPSW	criteria.
MAXLS	Maximum number of line searches allowed for each iteration.
LSTOL	Line search tolerance.
TTERM	Termination time.



• Number of load increments (NINC or alternatively DT + TTERM)

When **NINC** is not specified, there will be only one increment (full loading at once) applied during simulation, this causes high risk of divergence and error. With significantly nonlinear problems, it is recommended to use even **NINC = 1000** (0.001 load increments), mildly nonlinear problems can be solved with **NINC = 10**.

NINC number defines **only initial increment size**. Since automatic control is used for increment size – it may change over the analysis progress.

It is recommended that the analysis usually should start with smaller increments than the targeted one.

Alternatively, analysis increment size can be defined by use of **DT** (initial increment size) and **TTERM** (termination time) entries. These options are used for user-defined load curve, i.e. instead of defining only the end value of the load, one can also specify the load path **TLOAD1** bulk data entry via **DLOAD** in the subcase information entry. When both **DT** and **NINC** are specified, **DT** will be used.

2.4.2 Additional Time Step and Convergence Controls – NLADAPT

Sometimes additional parameters may help achieve convergence for e.g. terminate diverging simulation earlier, saving time spent waiting until the simulation ends. Through this card user can also specify whether increment size should be adaptive (automatic control, as described before) or fixed (constant increment size).

ID	Unique identification number
NCUTS	Number of cutbacks allowed to reduce the time increment.
DTMAX	Maximum time increment allowed.
DTMIN	Minimum time increment allowed.
NOPCL	Number of grids allowed to have open-close contact status change.
NSTSL	Number of grids allowed to have stick-slip contact status change when the current time step converged.
EXTRA	Determines whether to use linear extrapolation for load increments.

DIRECT Sets adapti

Sets adaptive or fixed time increment scheme.

- NCUTS available only for LGDISP. This parameter controls how many times solver can reduce the load increment (within one increment) after divergence occurs. By default, it is 5, which is reasonable for most of the cases.
- DTMAX and DTMIN available only for LGDISP. There are cases, when we will want to specify the maximum time increment allowed, for instance, to obtain accurate information about intermediate results, view smoother animation during postprocessing. DTMIN can also be specified to omit large analysis time for instance.
- DIRECT as mentioned earlier, user can specify a fixed increment (DIRECT,YES) for the whole analysis (NLPARM,NINC + NLADAPT,DIRECT,YES), when certain intermediate results are required. However, this often leads to divergence, as a larger nonlinearity is present. In that case, either NINC should be lowered or DIRECT set to default NO.

2.4.3 Request for Intermediate Results with NLOUT

ID	Unique identification number
NINT	Number of intervals specified to output intermediate results.
SVNONCNV	Flag to output the non-convergent solution, if nonlinear iterations does not converge.

- **NINT** user can request for results after each increment, so that the loading process can be displayed later in HyperView.
- SVNONCNV also, a nonconvergent result from the last increment can be requested, it sometimes may help debugging by viewing what happened in the



model during last converged increments. Activate **SVNONCNV,YES** to write h3d file containing non-convergent solution

2.4.4 Helpful Analysis Control Tool – EXPERTNL

Nonlinear expert system allows users flexibility in solving nonlinear solutions

- PARAM, EXPERTNL, AUTO (default)/YES/CNTSTB/NO
- The expert system monitors the convergence of nonlinear processes and tries to improve the convergence for poorly converging cases
- "YES" activates an 'expert system' that aids in the convergence. Possible actions like performing additional iterations, under-relaxation, automatic adjustment of the load increment, backing off to the last converged solution and retrying. May lead to a large number of nonlinear iterations.
- "AUTO" activates a 'light' version of the expert system which facilitates converging nonlinear process in reasonably close to minimum number of iterations.
- "CNTSTB" introduces temporary stabilization on contact interfaces to improve nonlinear convergence (discussed further in contact topic).

Adaptive load increment technique with the expert system (PARAM, EXPERTNL) is currently not supported with large displacement nonlinear analysis. However, the stabilization with PARAM, EXPERTNL, CNTSTB is supported for large displacement analysis.

2.4.5 Loadsteps Sequence

It may so happen that a simulation run must involve a sequence of different loading cases (loadcase) one after another. That means, a result from some specific loadcase needs to be taken as the initial model condition for the next loadcase. Such a configuration is used when a loading-unloading cycle is evaluated for e.g. some metal forming processes consist of a few steps in a sequence. Below is an example of a loading-unloading cycle of a chair subjected to plastic deformation:



Here one thing should be mentioned additionally. Despite being unloaded in the final state, the chair remains largely deformed. It is actually another very good example that proves the history-dependent character of NL analysis. Initial and final states are not subjected to any load, but due to load path that was applied to the model in the meantime, chair model ended up with pure plastic deformation. Loaded state deformation is a sum of elastic and plastic strains, here one can see that elastic strain was just a minor part (small difference between loaded and unloaded state).

Continuation of multiple subcases is activated through CNTNLSUB card, this can be done in two ways:

- a. **CNTNLSUB** activation in **SUBCASE OPTIONS**, where following options are available
 - CNTNLSUB = YES:

This nonlinear subcase continues the solution from the nonlinear subcase preceding. "Preceding" refers to the sequence in the input deck and NOT the numbering of the subcases

• CNTNLSUB = NO:

This nonlinear subcase executes a new solution sequence starting from the initial, stress-free state of the model.

• CNTNLSUB = SID:



This nonlinear subcase continues the nonlinear solution from the subcase with the ID given through SID. The subcase must precede the current subcase in the deck and must be a nonlinear subcase of the same type.

b. **CNTNLSUB** defined as global control, through **GLOBAL CASE CONTROL** card, which states that all available loadsteps are involved in loading sequence.

2.4.5.1 Boundary Fixing for CNTNLSUB

During analysis with loadsteps sequence user might sometimes need to retain displacements from one load case to the next one. Imagine a situation, in which beam is initially subjected to compression, but in the next step we want to bend it, keeping the compressive deformation. It is then possible to define DOFs in the second subcase that will be applied to **already deformed** structure from the first case. It is done by activating option **fixed** during SPC creation for the second subcase.

Example: A bar is placed under an axial load followed by a bending load subcase. The end of the bar under axial loading is boundary fixed in DOF 1 to retain the compression length reduction.



🛆 Altair | University



2.5 Path Dependent Loading

Normally in Nonlinear Quasi-Static Analysis setup a load is defined by inputting the final value that needs to be reached within a number of increments. These loads and pressures are then organized into one load collector and referred in Nonlinear Quasi-Static loadstep as LOAD. If we want to simulate more complex loading conditions, for instance loading/unloading sequence, we may use the technique described above – CNTNLSUB. However, still our possibilities are limited to giving just the final values for each subcase.

OptiStruct provides a possibility to define 'time' dependent loading in Nonlinear Quasi-Static loadstep. You may wonder why 'time' dependent case should be solved within NL Quasi-Static analysis, since there is actually no time.

Yes, time is in fact neglected here, but there are actually cases, where such time dependency is relevant and the dynamic effects are so little that they can be neglected at all. A great example is snap-through behavior already described as geometric nonlinearity (look at the picture below), where for the same load we can obtain 3 different final states. Although in real world such loading can only be dependent on time, we can accurately model it as a completely static event and that is why it is called 'Quasi-Static' and not 'Static' analysis.



Of course, in case of Nonlinear Quasi-Static analysis a better name would be 'path' dependent loading, since OptiStruct will follow a certain path when adding increments.



2.5.1 Path Dependent Load Setup in OptiStruct

- Definition of one reference load in a separate load collector without card image, whose value will be scaled in tabular data. This can be a maximum force value where scaling will be within [0, 1] range or can equal just 1 (a unit value) – then as scales real force values will be used.
- Creation of load (scaling) path TABLED card image load collector with tabular data. X-data corresponds to 'time', by default 1 is a termination (final) time, therefore all load scales should be defined for time within [0, 1] range. Y-data are load scales.
- 3) Creation of TLOAD1/TLOAD2 load collector, which matches reference load (under EXCITEID) with TABLED load path (under TID). Then all scales defined in TABLED will be multiplied by reference load. TYPE allows one to define whether it is a force or enforced displacement, temperature, velocity or acceleration.
- When creating a Nonlinear Quasi-Static loadstep, DLOAD (not LOAD) section is where load path (TLOAD1/TLOAD2) should be referred.



2.6 Restarting a Nonlinear Analysis

A nonlinear analysis can be restarted or continued from the last point at which the previous analysis was interrupted. When running an analysis, you can write the model information and analysis state information which can be used for restart.

Restart a Nonlinear Analysis with RESTARTW and RESTARTR entries.

RESTARTR: The RESTARTR entry can be used in the I/O Options section to define the reading requests for nonlinear restart.

RESTARTW: The RESTARTW entry can be used in the I/O Options section to define the control parameters for nonlinear restart.

Scenarios

- Restart from interruption continue an interrupted nonlinear analysis due to a power outage and so on → v2017
- Restart for subcase appending append steps to the load history after a successful nonlinear analysis → v2017.1
- Restart for subcase truncation and appending continue a nonlinear analysis from an intermediate point and change the remaining load history → v2017.1
- Restart for reusing nonlinear solution in linear perturbation append linear buckling/preloading steps after a successful nonlinear analysis → v2017.1

Support adding bulk data (listed below) in restart run

- SPCADD, SPC, SPC1
- LOADADD, FORCE
- DLOAD, TLOAD1, TLOAD2, RLOAD1, RLOAD2, DAREA
- NLPARM, NLADAPT, NLMON, NLOUT, TSTEPNL, TSTEP, SOLVTYP
- CNTSTB, MODCHG
- EIGRL, EIGRA, EIGRC
- FREQ, FREQ2
- TABLED1, TABLED2, TABLED3, TABLED4



2.7 Energy Output for Nonlinear Analysis

Starting from V2017.2, Contact stabilization energy and internal energy were output for large displacement nonlinear analysis and from V2017.2.3, Energy output is now supported for small displacement nonlinear analysis.

The output is only available when NLOUT is defined. The energy curves give the summary information based on the total energy variables, and so it provides accuracy in an overall sense.

The results are output to an ASCII file entitled filename_e.nlm and an HyperView/HyperGraph session file entitled filename_nlm.mvw.



2.8 On the Fly Output of Displacement Results for

Nonlinear Analysis

For nonlinear static (both small and large displacement) and nonlinear transient analysis, the displacement results are written to an .h3d file at each iteration or increment. The name of the .h3d file is filename_nl.h3d.These results can be viewed in HyperView when the simulation is still in progress. This helps to monitor the solution process.

• To request this output from every SUBCASE at each load increment and at each iteration, use PARAM, NLMON, DISP

- To request this output from a specific SUBCASE at each load increment or at each iteration, use NLMON=SID in the SUBCASE and the bulk data NLMON data
- If both PARAM,NLMON and NLMON in a subcase are used, the PARAM,NLMON request only applies to SUBCASE that do not have NLMON data.

Monitoring and Debugging Results

You can choose from:

- Monitoring Results (INT field on NLMON bulk entry is set to INC)
- Debugging Results (INT field on NLMON bulk entry is set to ITER or PARAM, NLMON, DISP is used)

Monitoring Results in HyperView (INT field is set to INC)

If the INT field on NLMON bulk data entry is set to INC, you can monitor the progression of the model during the run by following the steps below:

- Open HyperView and load the <filename>_nl.h3d file result file.
- You will see the Displacement results in HyperView until the point at which the model has run.
- In this way, you can monitor the deformation of the model during runtime.

Debugging the Results in HyperView (INT field is set to ITER)

If the INT field of the NLMON Bulk Data Entry is set to ITER, you can generate results and diagnostic data for each iteration of the nonlinear run.

Diagnostic data for debugging purposes such as Max Residual Force for LGDISP analysis can be viewed with a HyperView script to parse the data to highlight the problematic nodes. The procedure is as follows:

- Open HyperView and load the <filename>_nl.h3d file result file. You will see the model results in HyperView at this point.
- Import the TCL script. The script is located in the installation at: <install_directory>/altair/hwsolvers/scripts/os_out_file_parser.tcl

- Click File > Run > Tcl/Tk Script and select the os_out_file_parser.tcl script and click Open.
- In the subsequent OptiStruct OUT File Parser window, select <filename>.out from your working directory for the OUT file: field.

Activate the checkbox of the result type whose peak value occurrence you wish to investigate and click on the corresponding Iteration in the Convergence Table. The corresponding grid point at which the result type attains its peak value is displayed in the Graphics browser of HyperView. Click Open and the monitored results are now loaded into the Window, as shown below:


🛆 Altair | University

3 Material Nonlinearity

All engineering materials are inherently nonlinear as it is not feasible to characterize a nonlinear material by a single constitutive law for the entire range of environmental conditions such as loading, temperature and rate of deformation. We can idealize or simplify the material behavior to account for only certain effects which are important for the analysis. The linear elastic (also called Hookean) material assumption is the simplest of all. The material is nonlinear elastic if the deformation is recoverable and plastic if it is irrecoverable. If the temperature effects on the material properties are important, then the coupling between the mechanical and thermal behavior should be properly taken into consideration through thermo-elasticity and thermo-plasticity. If the strain rate has significant effects on the material, we have to consider the theories of visco-elasticity and visco-plasticity. An example of material nonlinear behavior has been given in the above figures.

A brief classification can be given as follows:

- Nonlinear elastic
- Hyperelastic
- Elastic-perfectly plastic
- Elastic-time independent plastic
- Time dependent plastic (Creep)
- Strain rate dependent elasticity plasticity
- Temperature dependent elasticity and plasticity

Typically, in OptiStruct we divide these materials in two main groups:

• Elastic-plastic

This material is modelled as **MATS1** continuation to **MAT1** card and it contains different types of plasticity, e.g. perfectly plastic, strain dependent plasticity etc.

• Hyperelastic

Defined through **MATHE** card and used in conjunction with **MAT1** card (membrane properties). Typical application is rubbers used as gasket materials.

3.1 Elastic-Plastic Material

Elastic-plastic material definition informs us right from the start that it contains two main, but different behavior models. Elasticity stands for deformations that are reversible (usually constant stiffness, i.e. linear stress to strain response) and plasticity means that the change is irreversible (can be linear and nonlinear). Even though both functions describing elastic and plastic characteristics of a given material might be linear in some cases, the fact that the stress-strain slopes are different makes the material already nonlinear. The most basic stress-strain chart presenting elasticplastic material is given below.



- I. Elastic-plastic material provides linear, elastic behavior until elastic limit in point 1 is reached.
- II. Above point 1 material starts deforming plastically, the strain gained between point 1 and 2 is irreversible portion of deformation.
- III. At the point 2, material's strain is a sum elastic and plastic strain:

$$\varepsilon = \varepsilon_e + \varepsilon_p$$

- IV. While unloaded, only elastic portion of deformation is regained. The strain that remained is ϵ_{p} .
- V. Point 2 is now a new elastic limit, i.e. material needs this time to reach elastic strain and stress defined by point 2 (assumed the chart is moved now to the right by ε_p distance) to be subjected to further plastic deformation.



The phenomenon of increased elastic limit is known as hardening.

The equation regarding strain sum provided above in this chapter is only valid when the strains are of no significant amount, i.e. 5% to eventually 10% is accurate enough. For large strains, OptiStruct uses a different equation. For more information one can refer to large strain theory

3.1.1 Engineering vs True Stress and Strain

This is a major distinguishing factor between linear static, small displacement nonlinear and large displacement nonlinear FEA (LGDISP). In linear analysis and small displacement nonlinear we always use the engineering stress – strain definitions, whereas in large displacement nonlinear analysis typically the true stress and strain is used.

We will examine these various stress and strain definitions through a simple onedimensional example.



A one dimensional rod undergoing axial deformation

1) Engineering Strain and Engineering Stress

Engineering strain is a small strain measure, which is computed using the original geometry. The engineering strain measure is a linear measure since it depends on the original geometry, (i.e. length) which is known beforehand. It is limited to small rotations of the material because a moderate rigid body rotation will result in non-zero strains.

$$\varepsilon_{\rm x} = \frac{\Delta l}{l_0}$$

Engineering stress (σ), is the conjugate stress measure to engineering strain (ϵ). It uses the current force F and the original area A₀ in its computation.

△ Altair | University

$$\sigma = \frac{F}{A_o}$$

2) Logarithmic Strain and True Stress.

Logarithmic strain/natural strain/true strain is a large strain measure, which is computed as

$$\epsilon_l = \int_{l_o}^l \frac{dl}{l_o} = \left(\frac{l}{l_o}\right)$$

This measure is a nonlinear strain measure since it is a nonlinear function of the unknown final length. It is also referred to as the log strain (or true strain) and represents an additive strain measure as compared to linear strain. Let us consider a bar with say initial length of 1m undergoing deformation in 3 steps as follows:

Step 1 : deformation from 1 m to 1.2 m

Step 2 : deformation from 1.2 to 1.5 m

Step 3 : deformation from 1.5 to 2 m

In the following table we compare the engineering strains against true strains and one can clearly see that the additivity is retained in only true strain and hence this one should be used in nonlinear analysis.

Step	Engineering Strain		True Strain	
1		0,2	Ln(1,2/1) =	0,1823
2	0,3/1,2 =	0,25	Ln(1,5/1,2) =	0,2231
3	0,5/1,5 =	0,33	Ln(2/1,5) =	0,2876
Sum of the	0,2+0,25+0,33	0,78	0,1823+0,2231+0,2876 =	0,6931
strains				
Total: From 1	(2,0-1,0)/1,0 =	1,0	Ln(2/1) =	0,6931
to 3				
Conclusion	Eng. Strain is not additive		True Strain is additive and thus	
	and doesn't preserve the		preserves the history of	
	history of deformation		deformation correctly.	
	correctly.			



True stress (s) is the conjugate 1-D stress measure to the log strain (ϵ_l), which is computed by dividing the force F by the current (or deformed) area A. This measure is also commonly referred to as the Cauchy stress.

$$s = \frac{F}{A}$$

3.1.2 Hardening Models

Since OptiStruct allows users to define different hardening behaviors through material card (this is with respect to tension-compression cycles), a brief explanation of how these models are working should be presented.

• Isotropic Hardening

For best understanding of the hardening mechanism, one should investigate the stress-strain chart.

If the Specimen is loaded up to point A and the load is reversed, it will unload along the slope of elastic part.

If it is loaded again, no additional plastic deformation occurs until the stress reaches point A.



If the part is under compression, it compresses

elastically along the elastic modulus until it reaches B, and then yields.

The stress change from A to B is twice the stress at A.

• Kinematic Hardening

If the specimen is loaded up to point A and the load is released, it will unload along the slope of elastic part. If it is loaded again, no additional plastic deformation occurs until the stress reaches point A.

If the part is under compression, it compresses elastically along the elastic modulus until it reaches point B, and then yields.



The stress change from A to B is twice the

initial yield stress. Yielding in tension has lowered the compressive strength. Kinematic hardening is preferred for analyses which undergo cyclical loading.

Bauschinger Effect

The Bauschinger effect describes the decrease in compression yield stress due to hardening which occurs in tension.

The graph shown describes a material initially loaded in tension in the plastic range.The material is then unloaded and reversed through to compression.Because of the Bauschinger effect, the yield stress in compression is less than the yield stress in tension

If the initial load is compressive, the reverse may also happen: a decrease in tensile yield stress due to hardening in compression.



Mixed Hardening

Mixed Hardening describes a more general case which is applicable for most materials.Mixed hardening combines isotropic hardening (expansion of the yield





surface) with kinematic hardening (translation of the yield surface). The figure presenting real material behavior is as follows:

3.1.3 Plasticity Setup in OptiStruct – MATS1

MAT1 material card image initially allows a setup of linear elastic material model, to define plasticity, MATS1 extension should be checked. The MATS1 card specifies stress-dependent and temperature-dependent material properties for use in applications involving nonlinear materials. It enhances the linear elastic material MAT1 by plasticity effects, i.e. a MAT1 Material definition is needed. The MATS1 Card is then a kind of extension within the MAT1 definition.

What you need to know about MATS1 (with regard to OptiStruct 2017.1):

- MATS1 is not available for 1D
- MATS1 is supported for 3D elements and large displacements
- MATS1 is not supported for second order shell elements
- Kinematic hardening and Mixed hardening are supported only for solids

It contains following options:



MID	Identification number (must be a MAT1 entry ID)
TID	Identification number of a TABLES1 or TABLEST entry. If H is given, then this field must be blank.
ТҮРЕ	Type of material nonlinearity – PLASTIC or NLELAST
н	Work hardening slope (slope of stress versus plastic strain) in units of stress.
YF	Yield function criterion (by default, YF = 1 (VonMises))
HR	Hardening Rule selection: Isotropic (1), Kinematic (2), Mixed (3) or user- selectable mixed hardening ratio
LIMIT1	Initial yield point
TYPSTRN	Specifies the type of strain used on the x-axis of the table pointed to by TID: total or plastic

- **MID** since MATS1 is defined throught MAT1 card, the ID is automatically assigned from MAT1.
- TID in this field we can refer to TABLES1 tabular data for stress-strain curve. For TYPE,PLASTIC the curve may be defined in two ways:
 - For **TYPSTRN = 1** the point (0,0) in the curve refers to **Yield strength.** That means the x-axis contains only plastic strain values.
 - For TYPSTRN = 0 the point the first point (0,0) is at origin (no stress and strain) and the second one must be at Yield Strength point (x₂,y₂), so that these two points define elastic slope (must equal E defined in MAT1)

For more information refer to MATS1 Bulk Data Entry in Help documentation.

• **TYPE** – for **NLELAST** the TABLES1 data should concern nonlinear elastic material (not hyperelastic, for that MATHE card is used – described later in this chapter)

 H – if TABLES1 data is not provided, linear hardening slope might be defined. Notice that more complex plasticity behavior must be referenced through TABLES1 data, not H. (By default - H = 0.0, perfectly plastic material).

The plasticity modulus (H) is related to the tangential modulus (ET) by



A simply modeled plasticity region defined by E, and Y,.

ET is the slope of the uniaxial stress-strain curve in the plastic region as a continuation of E past the yield point. Using ET and E, it is possible to calculate H (i.e. plasticity hardening) by the equation:

$$H = \frac{E_T}{1 - \frac{E_T}{E}}$$

HR – Mixed hardening (**HR** = **3**) corresponds to 30% of kinematic hardening and 70% of isotropic hardening. In many cases it allows good assumption of true material behavior. User may also manually specify the percentage of kinematic hardening by typing a value between 0 and 1 (e.g. **HR** = **0.26** corresponds to 26% of kinematic hardening and 74% of isotropic hardening).

LIMIT1 – must be defined only with H. If TABLES1 is provided it can be left blank.

3.1.4 Plasticity Results

Element strain results are computed as follows:

- Total strain computed from displacement gradients at the integration points and averaged over the element volume.
- Total strain (small-strain plasticity) is the sum of elastic and plastic strain components.

$$\varepsilon_{ij} = \varepsilon_{ij}^{pl} + \varepsilon_{ij}^{el}$$

where: ε

 ε_{ij} is the total strain,

 ε_{ij}^{el} is the elastic strain, ε_{ij}^{pl} is the plastic strain, and 0 < i and j ≤ 3

Total strain for large-strain plasticity is calculated through the following:

 $\mathbf{E}_{log} \stackrel{\text{\tiny def}}{=} \ln \mathbf{U} = \sum_{i=1}^{3} \ln \lambda_i \mathbf{N}_i \otimes \mathbf{N}_i$

where: N_i are the eigenvectors of $C = F^T F$, and λ_i^2 , its eigenvectors $F := \delta x / \delta X$ – deformation gradient F = RU – polar decomposition of F

Equivalent plastic strain is the trajectory length in the plastic strain space (mod sqrt(2/3))

- Equivalent plastic strain is the scalar measure of the accumulated plastic strain. For loading with reversals, equivalent plastic strain will continue to increase if the plastic strain rate is non-zero.
- Calculated at the integration points first, and then averaged over the element volume.

🛆 Altair | University

$$\varepsilon_{ps} = \sqrt{\frac{2}{3}} \int_0^t \sqrt{\dot{e}_{ij}^{pl} \dot{e}_{ij}^{pl}} dt$$

where e_{ij}^{pl} is the plastic strain deviator and \dot{e}_{ij}^{pl} is its rate.

Equivalent plastic stress is the stress intensity according to the yield criterion

- OS currently supports von Mises yield criterion
- For von Mises, the equivalent plastic stress is the same as regular von Mises stress
- This value is used to compare against the initial yield stress to determine occurrence of plastic flow

Plastic strain options are available for viewing tensor information

- Component values/directions are available as an OptiStruct output
- Specified in the control cards through STRAIN (PLASTIC)=YES

3.1.5 Tutorial: Small-Strain and Large-Strain Plasticity in a Test

Coupon

The purpose of this exercise is to illustrate the setup of a small-strain plasticity enforced displacement loading on a test coupon bar. The model is a solid bar composed of first order hexagonal elements which will be fixed at one end and subjected to an enforced displacement in-plane.



🛆 Altair | University



Problem Setup

You should copy the file: tension_sm_str.fem

Step 1: Import the model: tension_sm_str.fem into HyperMesh Desktop



Step 2: Import the table with the plasticity curve and add a new MAT1 material with

plastic material behavior

- Enable the Utility tab by checking the menu drop-down View > Browsers > HyperMesh > Utility.
- 2. Click on the **Utility** tab in the Browser area.
- 3. Ensure that the **FEA** button option at the bottom of the Browser area is selected.
- In the Utility tab, select the TABLE Create utility. This opens the Create Table dialog box.
- 5. Select **Import Table** as the option and click on the table type TABLES1.
- 6. Click Next.
- Browse for and select the plasticity.csv file in the model directory. Name the new table MATS1TS1 and click **Apply** to import the table.
- 8. Close the **Import TABLES1** window.
- In the Model tab, click on the existing MAT1 material named MAT1 to load it into the Entity Editor.
- 10. Activate the **MATS1** option to put in plasticity parameters and set the card values as shown on the right:

∆ c	reate Table				23
Optio	ns:				
ſ	Import Table				
0	Create/Edit Tab	le			
Table	IS:				
0	TABDMP1	$^{\circ}$	TABLED3	С	TABLEM3
0	TABFAT	C	TABLED4	С	TABLEM4
0	TABLED1	$^{\circ}$	TABLEM1	(•	TABLES1
0	TABLED2	C	TABLEM2	С	TABRND1
			Next		Cancel

Name	Value
Solver Keyword	MAT1
Name	MAT1
ID	1
Color	
Include File	[Master Model]
Defined	
Card Image	MAT1
User Comments	Do Not Export
E	210000.0
G	
NU	0.33
RHO	7.85e-009
A	
TREF	
GE	
ST	
SC	
SS	
MATS1	
TID	MATS1TS1 (1)
TYPE	PLASTIC
н	
YF	
HR_REAL	
HR	
LIMIT1	350.0
TYPSTRN	1
MATT1	
MAT4	
MAT5	
MATFAT	
MATF1	
MATX	

Step 3: Create a solid property card and apply it to the solid elements

- Create a new **PSOLID** property named PSOLID that references the material card MAT1.
- 2. Assign this property to all of the elements in the TENSIONTEST component.



Step 4: Create the boundary conditions and loading conditions

- 1. Create a new load collector named SPCs.
- 2. In the **constraints** panel, select node 14598 **by id** and constrain DOFs 1-6.

This node is the independent node of the RBE2 and will constrain all nodes connected to that rigid element.

3. Using a window selection, select all of the nodes shown in the image below. Constrain them in DOFs 1-3.



- 4. Create a new load collector named LOAD_20.
- 5. In the **constraints** panel, set the **load type** to SPCD and create a constraint in DOF

1 on node 14598 of **magnitude** 20.0.

🤄 create 🔻	🔹 🔹 nodes	Η	🔽 dof1	=	20.000	create
C update			clof2	=	0.000	create/edit
	relative size =		10.000 🔽 label constraint 🗌 dof3	=	0.000	reject
ŧ	constant value		Tixed dof4	=	0.000	review
			dof5	-	0.000	
			dof6	-	0.000	
			load types -	5	SPCD	return

6. Create a new NLPARM load collector named NLPARM with **NINC** of 20.

Step 5: Create the output requests for the measurement set

- 1. In the **control cards** panel, enter the GLOBAL_OUTPUT_REQUEST section.
- 2. Ensure that the following outputs are requested:

Output Type	Format	Option
DISPLACEMENT	H3D	ALL
SPCF	H3D	ALL
ELFORCE	H3D	SID (2)



Step 6: Add an NLOUT load collector to enable 100 increments of intermediate step results and set the NLOUT option to the load collector in GLOBAL_CASE_CONTROL.

Step 7: Create a new Non-linear quasi-static analysis load step named Load_20. Set the SPC to SPCs, the LOAD to LOAD_20, and the NLPARM to NLPARM.

Step 8: Run the analysis with 4 CPUs

- In the OptiStruct panel under the Analysis page, ensure that the export options: is set to all, the run options: is set to analysis. Name the file tensile_sm_str.fem and enter the options –optskip –out –cpu 4.
- 2. Click **OptiStruct** to run the nonlinear analysis.

Step 9: Review the results in HyperView

- When the run has completed, from the **OptiStruct** panel, click on the **HyperView** button to launch the post-processing report tensile_sm_str.mvw which was automatically created from the analysis.
- 2. Set the animation type to **Transient** and click the **Play** button to view the displacement results. Overlay the contours of various results onto the load case animation to view how they change as the enforced displacement scales.





A contour of element stresses of the model at its last enforced displacement (20 units)

Step 10: Setup Large Displacement Analysis

- 1. In the **control cards** panel, enter the PARAM section.
- 2. Set **PARAM, HASHASSM** to YES and **PARAM, LGDISP** to 1.

Step 11: Run the analysis with 4 CPUs

- In the OptiStruct panel under the Analysis page, ensure that the export options: is set to all, the run options: is set to analysis. Name the file tensile_lg_str.fem and enter the options –optskip –out –cpu 4.
- 2. Click **OptiStruct** to run the nonlinear analysis.

Step 12: Review the results in HyperView

- When the run has completed, from the **OptiStruct** panel, click on the **HyperView** button to launch the post-processing report tensile_lg_str.mvw which was automatically created from the analysis.
- 2. Set the animation type to **Transient** and click the **Play** button to view the displacement results. Overlay the contours of various results onto the load case animation to view how they change as the enforced displacement scales.





Step 13: Use HyperGraph to compare the SPC force displacement from the model against test data.

1. Use the Window Layout tool on the top button bar to create a 2-window layout

in the results page.

2. Click in the second window to activate it and use the Client Selector on the

bottom button bar to launch the **HyperGraph 2D** client in that window.

In the Build Plots panel , select the data file test_results.csv. Select Column 1 as the X Type, Select Unknown as Y Type, Block 1 as Y Request, and Column 2 as Y Component. Click Apply to create the force-displacement curve from the test data.

Data file: 📑	🗃 Winlap367\weedti\14.0 Cours	e Materia/Opfistruct Nonlinear/Solved/3c,	_SMALL_STR	AIN Vest_results.csy		 Apply
Subclase:	÷ .	Y Type Filter:	Y Request:	Filter:	Y Component Filter	- E Preview
⊠ Тура:	🛨 Column 1 🛛 🔻	Uriknown	Block 1		Column 1	۱
XRequest	× ·	I			Column 2	Adv. Options
KComponen	×]				
Layout	Use current plot		AI	None Flip ····	Al None Flip	

4. Use the **Scales, Offsets, and Axis Assignments** panel by to select the curve in the **Curves** list and change the **X-Scale** to 2 and the **Y-Scale** to 1000.

Data file: 📑	🔾 \\triap367\\vcott\14.0 Cours	e Materia/Optistruct Nonlinear/Solved/3c_	SMALL_STRAIN \test_results.csy	-	Apply
Subclase:	×	Y Type: Filter:	Y Request: Filter:	Y Component: Filter:	E Preview
⊠ Тура:	🜩 Eolumn 1 🔷 🔻	Unknown	Block 1	Column 1	
XRequest	*			Column 2	Adv. Options
KComponent:	n 7				
Layout	Use current plot		Al None Flip …	Al None Flip …	

- Return to the Build Plots panel and select the data file tensile_lg_str.h3d. Select Element Forces (1D) as the X Type, E3460 as the X Request, and X(1D) as X Component. Select SPCF Forces (1D) as Y Type, N14598 as Y Request, and X as Y Component. Click Apply to create curve from the large displacement analysis run.
- Use the Scales, Offsets, and Axis Assignments panel to select the second curve in the Curves list and change the X-Scale to 2e6.





Tip: This scale factor converts the spring stiffness from the CELAS1 element in the model to act as a strain gauge within the model.

3.2 Hyperelastic Material

The second main group of materials that are widely supported in OptiStruct are Hyperelastic materials, which usually are used to model foams and rubbers. Hyperelastic materials are:

- Nonlinear stress-strain characteristics
- Fully **reversible deformation** (no plasticity), e.g. loading-unloading simulation will always lead back to initial state, no constant deformation. In reality, it is close to ideally elastic, that is why such assumption is allowed for FEM application.
- Fully or approximately **incompressible** (rubbers), which means that the volume of the material is (almost) changed, e.g. large compression in one direction causes proportionally large expansion in transverse directions; or **compressible** (foams)
- Rubbers are isotropic; foams can be anisotropic or isotropic.



3.2.1 Hyperelastic Material Models in OptiStruct

Following mathematical models, that describe hyperelastic materials, are available in OptiStruct:

- Arruda-Boyce (ABOYCE) Generally recommended for models undergoing large tensile strains (accurately describes large tensile strains above 200%); recommended when test data is limited, uniaxial test data can be sufficient.
- Generalized Mooney-Rivlin (MOONEY) Polynomial model, which constants are determined to fit the test data
- **Physical Mooney-Rivlin (MOOR)** Equivalent to generalized Mooney-Rivlin with first order deviatoric and volumetric strain energy polynomials
- Reduced Polynomial (RPOLY)
- Neo-Hookean (NEOH) Reduced polynomial model
- Yeoh (YEOH) Similar to Arruda Boyce model, reduced polynomial, can be used with limited test data
- Ogden (OGDEN) Ogden model is based on principle stretch ratios, should not use when test data is limited, e.g. when only uniaxial data is available. Ogden is accurate even for very large strains. Ogden captures stiffening upturn of stressstrain curve, not captured by N-H or M-R



Note that only Arruda-Boyce and Yeoh models are capable of modelling large tensile deformations.



3.2.2 Defining Hyperelastic Material Properties

In OptiStruct, hyperelastic material is defined through MATHE material card image. For a proper parameter set users need to input distortional deformation properties for incompressible materials. If volume change is considered, additionally volumetric deformation properties are defined.

- a. User needs to define the material **distortional** deformation properties either by:
 - Providing tabular data for:





These entries are provided by TABLES1, where x-column corresponds to stretch ratios and y-column contains engineering stress. **Or**:



- Providing material constants:
 - C1 locking stretch; λm required for defining β value (Arruda-Boyce)
 - NA order of distortional strain energy polynomial; C_{pq} material constants (Mooney-Rivlin, Physical Mooney-Rivlin, Reduced Polynomial, Neo-Hookean); for further information about NA and C_{pq} orders please refer to MATHE card information in Help Documentation
 - MUi, ALPHAi (Odgen)
- b. For definitions: user may need to input **volumetric** deformation properties either by:
 - Providing tabular data for:
 - **TABD** TABLES1 entry, x-column corresponds to pressure, y-column contains volumetric change values.

Or:

- Providing material constants:
 - ND order of volumetric strain energy polynomial; Dp material constants (Mooney-Rivlin, Physical Mooney-Rivlin, Reduced Polynomial, Neo-Hookean) for further information about ND and Dp orders please refer to MATHE card information in Help Documentation

3.2.3 General Remarks for Hyperelastic Models

- MATHE materials are referenced by PLSOLID property, which is a property dedicated exclusively for nonlinear hyperelastic materials.
- Hyperelastic materials are valid for solid elements: CHEXA, CPENTA, CTETRA
- Since it is a large displacement analysis, LGDISP card should be activated (together with HASHASSM,YES)
- Thermal parameters can be defined for MATHE material

- For materials models: order greater than 2 recommended only for partial polynomials
- Use first order element to avoid element distortion

3.2.4 Tutorial: Compression of a Hyperelastic Part

In this exercise, users will walk through the set-up process for creating the material, element properties, output requests, and load steps for a hyperelastic part impacted by a plunger in a loading and unloading sequence.

The hyperelastic part will be described by a two-term Mooney-Rivlin material model. The model has already been outfitted with contacts and load collectors appropriate to the solution. The outputs of stress, strain, plastic stress, and plastic strain will be requested for post-processing.



Problem Setup

You should copy the file: rubber_original.fem

Step 1: Import the model: rubber_original.fem into HyperMesh Desktop



Step 2: Create a new MATHE material with Mooney-Rivlin modeling

1. Create a new material of type MATHE named MATHE_30 with parameters as shown below:

Name	Value
Solver Keyword	MATHE
Name	MATHE_30
ID	11
Color	
Include File	[Master Model]
Defined	
Card Image	MATHE
User Comments	Hide In Menu/Export
MODEL	MOONEY
NU	
RHO	
TEXP	
TREF	
C10	400.0
C01	50.0
D1	0.001
TAB1	

Step 3: Create a new PLSOLID property and assign to the part

- 1. Create a new property of type PLSOLID named PLSOLID_3.
- 2. Set the **Material** for this property to MATHE_30.
- 3. Assign this property to the Seal component.

Step 4: Create a NLPARM load collector to set the load increments

- 1. Create a new load collector of type NLPARM named NLPARM_SolverSettings.
- 2. Set the **NINC** for this load collector card to 10.

Step 5: Create the output requests, turn on large displacement analysis, and enable continuing nonlinear subcases

- 1. In the **control cards** panel, enter the GLOBAL_OUTPUT_REQUEST section.
- 2. Ensure that the following outputs are requested:

Output Type	Format	Option
DISPLACEMENT	H3D	ALL
STRAIN	H3D	ALL

- 3. In the GLOBAL_OUTPUT_REQUEST section, request **STRESS** in H3D format for **LOCATION(1)** CORNER with **OPTION(1)** set to ALL.
- In the GLOBAL_OUTPUT_REQUEST section, select CONTF and change CONTF_NUM to 2. This requests that the solver duplicate the output results for contact forces in multiple formats.
- 5. For the first **CONTF**, set **FORMAT(1)** to H3D format and **OPTION(1)** set to YES.
- 6. For the second CONTF, set FORMAT(2) to OPTI format and OPTION(2) set to YES.
- 7. In the **control cards** panel, under the PARAM section, set **LGDISP** to 1.
- 8. In the **control cards** panel, under the OUTPUT section, select H3D for **KEYWORD** and set **FREQ** to ALL.
- 9. In the **control cards**, under the GLOBAL_CASE_CONTROL, check the box to activate CNTNLSUB.

Step 6: Create the loading and unloading subcases

- Create a new Non-linear quasi-static load step named Load. Set the SPC to SPC_Constraints, set LOAD to scaled_load and set NLPARM(LGDISP) to NLPARM_SolverSettings.
- Duplicate the previous load step and name it Unload. Clear the LOAD specification from the load step, leaving all other settings the same.

Step 7: Run the analysis in OptiStruct

Step 8: Review the results in HyperView

Name	Value
Solver Keyword	SUBCASE
Name	Unload
ID	2
Include File	[Master Model]
User Comments	Hide In Menu/Export
Subcase Definition	
 Analysis type 	Non-linear quasi-static
SPC	SPC_Constraints (12)
LOAD	<unspecified></unspecified>
NLPARM	<unspecified></unspecified>
NLPARM(LGDISP)	NLPARM_SolverSettings (78)
SUPORT1	<unspecified></unspecified>
DEFORM	<unspecified></unspecified>
PRETENSION	<unspecified></unspecified>
MPC	<unspecified></unspecified>
STATSUB (PRE	<unspecified></unspecified>
NLADAPT	<unspecified></unspecified>
NLOUT	<unspecified></unspecified>
CNTSTB	<unspecified></unspecified>
DLOAD	<unspecified></unspecified>

🛆 Altair | University



A plot of the vonMises Element Strains (2D & 3D) (t) in HyperView for subcase 1

🛆 Altair | University

4 Geometric Nonlinearity

Recognizing and understanding in which cases our FEM model might produce geometric nonlinearities is crucial for properly conducted simulation. There is no one consistent explanation what actually is geometric nonlinearity, it seems that the best way to help you understand it, is to present examples and explain the difference in results with or without nonlinear geometry activated.

The idea behind geometric nonlinearity generally is an update of equillibrium equations to new element locations, i.e. without nonlinearity solver would calculate stress and strains basing on the initial model shape and location, even though the structure may already be deformed or relocated. Geometric nonlinearity is then considered in following cases:

4.1 Large Deflection

Large deflection usually consists of three main phenomena: large strain, displacement and rotation. The best example would be to consider a solid block, which is twisted due to applied moment. Notice that initially the corner points will displace in the direction perpendicular to the "radius" connecting moment center and corners.



But engineer's intuition is telling us that the direction of U vectors will change as the deformation progresses - it will at all times be perpendicular to the "radius" (black lines on the picture above). Since in linear static analysis calculation of equilibrium



equations is conducted for initial shape, because the solver assumes the displacements are so small there is no need to update to the current shape, it will cause a situation that our solution will end up with the corners located somewhere on the line, which is basically an extension of initial displacement vector **U** (picture below, on the left). On the other hand, a solution which considered geometric nonlinearity, i.e. equilibrium equations were instantly updated to current geometry shape, managed to predict real behavior of the solid block – the corners were displacing on the circular path. (Picture below)



Right from that point we can deduce that the solver now requires incremental procedures, so that the stiffness matrix can be updated to the new locations. Increments should be then smaller if the nonlinearity of deflection is larger, so that convergence is easier to obtain. Notice that here geometric nonlinearity does not depend on material nonlinearity, presented example shows linear elastic material being twisted and therefore enhances relevance of geometric nonlinearity consideration as separate phenomenon.

• Specific case: large rotation and displacement without strains

Large rotation and displacement individually occur as a specific case of large deflection, when there is no strain present. Since there is a possibility that a model



will rotate in a rigid body mode, displacements may be significant, while the model does not contain any stress or strain. In this case, there is a need for equilibrium equations update to predict proper displacements, hence, nonlinear geometry needs to be considered.



Example Tutorial on Large Strain

4.2 Large Strain

Since large deflection is the main source of incorrect shapes generated in the analyses without geometric nonlinearity, one should also remember about the effects on stress calculation due to wrong reference geometry. Let's look further in the example that was presented before and investigate the stress results.

🛆 Altair | University



Notice that the stress results are almost the same, but the strain is actually at great difference, just take a look at the element size of the most deflected corners. Different material stiffness? No, the answer lies in the equilibrium equations, which were derived from initial, undeformed shape.

• Engineering vs True Stress: Variable Cross-Section

The most basic example depicting large strain effects of geometric nonlinearity is a simple tensile test, during which a test coupon is subjected to pure tensile loading. Although described already in the nonlinear material chapter, this is actually a geometric nonlinearity case, since even in the elastic region there is a nonlinear force to displacement response due to change in sample cross-section.

True stress is a measure, which derives from geometric nonlinearity, because it refers to current cross-section, whereas engineering stress does not respect geometric nonlinearity, since it is obtained with respect to initial cross-section.

4.2.1 Large Strain in Shells

From OptiStruct 2018.0 Large strain formulation for shells is now supported in large displacement nonlinear analysis. No PARAM/option needed to activate a large strain formulation. Until OptiStruct 2017.2.3, large displacements and large rotations for shells were accounted for in LGDISP analysis. However, large strains for shells were



not supported and you could be getting warning like below for shells undergoing large

strains.



4.3 Non-Uniqueness of the Solution: Snap-

8 10 Displacement (Grids)

Through

As already presented in the nonlinear analysis chapter, there might be cases which can present a few different solutions for just one boundary conditions setup. A snapthrough model is just the perfect example.





Notice that together with an axial compression of the beams, the load required to compress them further is not linearly increasing. Moreover, loading magnitude is repeating itself, since this assembly contains a few metastable configurations.

4.4 Follower Forces

Another case when geometric nonlinearity is significant are follower forces and pressure. They are basically geometry dependent loading conditions, which means that its location and orientation remains the same with respect to deflecting geometry, regardless if it needs to move in global coordinate system of not. The simplest example is inflation pressure. When a balloon is being inflated, the pressure orientation is always perpendicular to the balloon walls.



The case on the left is inflated with pressure of constant orientation, on the other hand the case on the right is subjected to follower pressure, which is always perpendicular to the walls – this is a real situation.

4.4.1 Follower Forces Setup

Follower forces are used for large displacement analysis. In order to define a created force/pressure as follower, one should perform one of the two following actions:

- a. Global control subcase information entry: define a global parameter, which refers to all loads present in the model. FLLWER card should be activated then through PARAM card: PARAM,FLLWER,x where x stands for the options for follower force/pressure calculation
- b. Specific load control Bulk Data entry: Indicate certain Load Collector as follower force. The chosen load collector should have a card image FLLWER assigned. The OPT parameter should be defined OPT = x, where x stands for an option for follower force/pressure calculation.

ΟΡΤ	Options for the calculation for Follower Loads.
	Default = 1 <-1, 0, 1, 2, or 3>
	= -1, 0: Follower force calculation is not activated.
	= 1: Follower effect is activated. For pressure load, both element surface area
	and load direction are involved to calculate follower force. For concentrated
	force, only the force direction is involved.
	= 2: Follower effect is activated. For pressure load, only element surface area
	is involved. For concentrated force, only the force direction is involved (same
	effect as FLLWER = 1).
	= 3: Follower effect is activated. For pressure load, only load direction is
	involved. For concentrated force, only the force direction is involved (same
	effect as FLLWER = 1).

🛆 Altair | University



When using follower force formulation in OptiStruct, please note the following:

• Element Support

Only elements supported for large displacement analysis are supported for follower load

• Recommendations

It is best to use hash assembly for subcases with follower loads (PARAM, HASHASSM, YES)

OptiStruct supports both symmetric solvers and un-symmetric solvers for follower loadcases

BCS solver should not be used with hash assembly and un-symmetric solver for follower load subcases

MUMPS solver should not be used with PARAM, HASHASSM, NO

• Limitations

If a load set is applied in both continuation subcases(CNTNLSUB), the FLLWER option of the load set must be exactly the same in the two subcases PLOAD2, PLOAD4, FORCE1 and FORCE2 loads are currently supported



4.4.2 Tutorial: Follower Forces on a Curved Rod

This exercise demonstrates the differences between non-follower and follower forces for a curved bar under a load directed into the bar parallel to the axis at one end. The material models, including plasticity effects, the property, the applied force, and the load steps will be created by the user.



Problem Setup

You should copy this file: cvrod.fem

Step 1: Import the file cvrod.fem into HyperMesh Desktop

Step 2: Create an elastic material with the following values:

- Name: Material
- Type: MAT1
- E: 2.1e5
- NU: 0.3
- **RHO**: 7.85e-9

Step 3: Create a new PSOLID property named PSOLID

Step 4: Assign the newly created material card to the PSOLID property card


Step 5: Assign the PSOLID property to the PSOLID_1 component



Step 6: In a new load collector named SPC, constrain node 85 in DOFs 1-6

Step 7: In a new load collector named Load, create a force on Node 86 with the following components: {70.253,0.0,-71.165}



Step 8: Create an NLPARM load collector named NLPARM with an NINC of 10



Step 9: Enable hash assembly, large displacement, and the unsymmetric solver using the control cards PARAM, HASHASSM (YES) as well as PARAM, LGDISP (1) and PARAM, UNSYMSLV (YES).

Step 10: Create a new Non-linear quasi-static analysis loadstep with SPC set to SPC, LOAD set to Load, and NLPARM set to NLPARM.

Step 11: Run the analysis as cvrod_nf.fem



Step 12: Review the results in HyperView

Step 13: Return to HyperMesh Desktop to replace the force definition and

enable follower forces

1. Delete the existing load in the Load load collector.

2. Make the Load load collector the current load collector.

3. In the **forces** panel, set the **load types=** to FORCE1 and use the nodes entity selector to select node 86.



4. Enter a **magnitude=** of 100 and select node 82 as **N1** and node 78 as **N2** as shown below:



5. Click **create** to create the FORCE1 definition.

Tip: This definition is identical to the original force in vector and magnitude, and is defined for load collector 2 which is the Load load collector, but unlike the previous analysis using the FORCE card, this can make use of the follower force settings.
6. In the control cards, set PARAM, FLLWER to 1.



Step 14: Save the FE model file as cvrod_ff.fem and rerun in OptiStruct. Review the results in HyperView

5 Contact Nonlinearity

The initial question might be: why contact definition introduces nonlinearity to the model?

The explanation for that occurs to be simplest when we think about contact as just another boundary condition/constraint applied to elements: The basic idea of contact interface is to prevent bodies penetration by generating reaction forces on the opposing surfaces for e.g. additionally shear to simulate friction. Since, for instance, SPC is used to apply restrictions to body movement, contact interface in fact does the same – it does not allow movement in the direction of penetration.

What about nonlinearity then? Since whether the contact status is closed or open (contact takes place or not) depends on the distance between bodies, this status can change during analysis time, which in fact is a change of boundary condition – this is already a nonlinear phenomenon, which needs equilibrium equations update during analysis time. Another source of nonlinearity associated with contact is nonlinear contact stiffness, which is an optional solution.

5.1 Contact Discretization

As in Finite Element Analysis a model needs to be discretized in order to create finite number of equilibrium equations, the same process must be applied to contact interface. Discretization of contact interface leads to creation of contact elements. There are two main contact discretization approaches: N2S (node-to-surface) and S2S (surface-to-surface) and a choice between them has a great influence on the solution computation.

(1) (2) (3) (4) (5) (6) (7) (8) (9) (10) CONTACT CTID PID/TYPE/MU1 SSID MSID MORIENT SRCHDIS ADJUST CLEARANCE TRACK DISCRET SMOOTH SMSIDE SMREG +

Contact discretization is set through **DISCRET** (N2S or S2S) option in **CONTACT** card.



5.1.1 Node-to-Surface Contact

The Node-to-Surface Discretization approach involves the creation of contact elements linking the Master Surfaces to Slave Nodes. The contact interface is constructed by searching, for each slave node, a respective facet of the master surface, which contains the normal projection of the slave and is within the Search Distance (SRCHDIS field on the CONTACT entry) distance from the slave node.



Slave SURF (or SET of GRIDs)

Master SURF

Structure of a N2S Contact Element:

- One slave node
- A few master nodes •
- One master face including the master nodes
- One projection from the slave node to the master face

For each slave node, a respective master segment will be looked for, which contains the normal projection of the slave and is within SRCHDIS (CONTACT or TIE entry) distance from the slave node.

If a master segment with normal projection is not found, the nearest segment is selected, if the direction from the slave to master is within a certain angle (30 degrees) relative to the normal to the master segment.

Note:

- 1. For small displacement analysis, N2S contact elements are created with CGAPG as the core.
- For Large displacement analysis, N2S contact elements consist of non-CGAPG cores.

- 3. Each N2S contact element consists of a slave node, a few master nodes, and one master face.
- Within a contact interface, slave nodes should be unique to each N2S contact element. For instance, two different N2S contact elements cannot consist of the same slave node.
- 5. Within a contact interface, master faces can be shared among N2S contact elements.

General recommendation:

Referring to notes 4 & 5 above, it is always recommended that in case of contact between bodies with different mesh sizes, the slave component should always be the one with finer mesh. In that case all master faces will be associated to a slave node. In the opposite case (master entity with finer mesh), due to bigger number of master faces than slave nodes, some master faces would be included in the contact interface - because slave nodes cannot be shared among them.

5.1.2 Surface-to-Surface Contact

The Surface-to-Surface Discretization approach involves the creation of contact elements linking the Master Surfaces to Slave Surfaces. For each slave facet, a respective master facet will first be searched which contains the normal projection of the sample points on the slave facet and is within the SRCHDIS distance from the sample points. Then, for each slave node, the contact element is created with the surrounding slave facets and the master facets found by projection of the sample points on the slave facets.



Slave SURF

Master SURF



The structure of Surface-to-Surface contact elements is different when compared to that of the Node-to-Surface (N2S) CGAP(G) elements.

- One "core" slave node
- Several slave nodes surrounding the "core" slave node 0
- One or more slave faces including the slave nodes and sharing the "core" slave node
- Several master nodes
- One or more master faces including the master nodes
- One or more projections from the slave faces to the master faces

Note:

- 1. Multiple S2S contact elements have different "core" slave nodes (that is, each S2S contact element has a unique "core" slave node).
- 2. Slave face(s) may be shared among multiple S2S contact elements.
- 3. Multiple S2S contact elements may share one or more master face(s)

5.1.3 Which One to Choose - N2S or S2S?

Choosing N2S vs S2S:

- 1. For N2S, it is recommended to use the finer mesh as the slave.
- 2. S2S is typically consumes higher computational time and memory resources compared to N2S; however, it typically can generate smoother contact pressure in many cases.
- 3. If the slave is a SET of GRID's, then N2S should be used.
- 4. N2S is recommended, if the slave is a SET of solid elements. Alternately, the surfaces of solid elements can be used as the slave for S2S.
- 5. N2S is recommended if the slave or master surface includes a corner. Alternately, the surfaces can be split into smaller smooth surfaces and used as masters or slaves for S2S contact.



5.1.4 Self-Contact with Surface-to-Surface

Now, it is possible to define self-contact with S2S CONSLI contacts. To activate selfcontact, the master surface ID needs to be kept blank or defined same as slave surface ID. Self-contact using N2S CONSLI was already available from 2017.2.

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)				
CONTACT	CTID	PID/TYPE/MU1	SSID	MSID	MORIENT	SRCHDIS	ADJUST	CLEARANCE				
	Master surface to be left blank or defined using same id as slave											

Example:



5.1.5 Continuous, Finite vs Small Sliding Contact

In OptiStruct there are three possibilities regarding contact elements updation throughout the simulation. Since by default contact elements are defined for the whole duration of the analysis, i.e. each slave node has the same master entity projection, it might lead to incorrect results when the relative displacement of slave and master entities exceeds the fraction of the element size, In which case, Continuous or Finite Sliding Contact must be defined.



	Small Sliding	Finite Sliding Contact	Continuous Sliding
	Contact		Contact
Activation (see:	By default (TRACK	Must be defined	Must be defined
Contact	=SMALL)	(TRACK = FINITE)	(TRACK = CONSLI)
Utilities)			
Contact elem.	No update	After each load	After each iteration
update	throughout	increment	
frequency	simulation		
Analysis time	Computationally	Computationally	Even more
	inexpensive	expensive	computationally
			expensive
Types of NL	Small and Large	Effective only for	Effective only for
analysis	Displacement	Large Displacement	Large Displacement
	Nonlinear Analysis	Nonlinear Analysis	Nonlinear Analysis
Application	Only contacts with	Contacts with relative	Contacts with relative
	relative small	large displacement	large displacement
	displacement	(sliding)	(sliding)
	(sliding)		
Valid types of	All	Slide and Frictional	Slide and Frictional
contact (see:		contact	Contact
Contact Types)			
Comments	1.Continuous Sliding	g Contact, although more	computationally
	expensive than Finit	te SC, provides sometimes	s more accurate results
	and leads to quicker	r convergence – therefore	should be used for
	relatively very large	sliding.	
	2.Finite and Contine	ouous Sliding is not suppo	rted for self-contact.
	3.Finite and Continu	uous Sliding works only wi	ith Hash Assembly and
	the MUMPS solver (used for frictional contac	ts).



5.1.6 Contact Entities

Here is a short list about what kind of entities can be defined for contact interface.

Master & slave surface:

- Contact surface (shell elements or solid faces)
- Set of shell elements
- Set of solid elements

Slave nodes:

- Set of nodes
- Contact surface (shell elements or solid faces)

5.1.7 Tutorial: N2S and S2S Contact

This exercise demonstrates the differences between node-to-surface (N2S) and surface-to-surface (S2S) contact modelling for the same analysis setup performed on a pair of rectangular blocks. The material models including the property, the applied loading, and the load steps are already created, allowing the user to focus on creating the contact definitions.



Problem Setup

You should copy this file: blocks.fem

Step 1: Import the file blocks.fem into HyperMesh Desktop



Step 2: Hide everything but the top component in the model

1. Click on the **Shaded elements and mesh lines** button to shade the model

and click the **XZ Plane** view button in the visualization toolbar to orient the model as shown.



- 2. In the **Model Browser**, expand the **Component** subtree to show the components in the model. Right-click on top to bring up the context-sensitive menu.
- 3. Click Isolate Only to hide everything but that component in the model.

Step 3: Create a contact surface for the top component

- Right-click in the Model Browser and select Create > Contact Surface. A new contact surface definition is created and loaded into the Entity Editor.
- 2. Name the new contact surface Top and ensure that Card Image is set to SURF.
- 3. Click on the **Elements** selector to pick elements to form the contact surfaces in the model.

▼ add shell elements	add
elems II	review
	return

This brings up the contact surface creation panel.



- 4. Change the first drop-down selector to add solid faces.
- 5. Change the second drop-down selector to the elems entity selector, select elems
 > by collector, and select the top component.
- Click on the **nodes** entity selector and select three nodes on the corner of one element on the bottom of the top block. This defines the face for creating the contact surface.



7. Click add to add that face to the contact surface. Depending on which order the nodes were selected in, the solid faces may have inverted normals. The normals are represented as pyramidal elements on top of the solid faces and should point toward the surface to be contacted. The image below shows an example of contact faces which are pointed into the solid face. These normal must be inverted to correct the contact.



8. [Optional] To correct inverted normals for a contact surface, change the dropdown for the elements selector to adjust normals.



 [Optional] Because there is only one contact surface created in the model so far, leave all elements selected and click **reverse**. The normals for the contact surface now point in the correct direction.



10. Click return to exit the contact surface element selector.

Step 4: Isolate Only and create a contact surface for the bottom component





If the normals need to be corrected when there is more than one contact surface in the model, uncheck the all elements option and simply select the elements **by collector** prior to clicking **reverse**.

Step 5: Create a new contact definition

- Right-click in the Model Browser and select Create > Contact. A new contact card is created and loaded into the Entity Editor.
- 2. Name the new contact "Contact" and enter the values as shown below:

News	Mala -
Name	Value
Solver Keyword	CUNTACT
Name	Contact
ID	1
Color	
Include File	[Master Model]
Card Image	CONTACT
User Comments	Hide In Menu/Export
Property Option	Property Type
TYPE	SLIDE
SSID	Bottom (2)
MSID	Top (1)
MORIENT	
SRCHDIS	
Adjust Option	String Value
ADJUST	
CLEARANCE	
DISCRET	\$2\$
TRACK	
CONTACT_NUM_SMOO	0

Tip: To select the Bottom and Top contact surfaces, switch the Slave and Master selectors from Set to Contactsurf. The S2S option for the **DISCRET** parameter indicates that this will be a surface-to-surface contact in the solver.

Step 6: Create the control cards to output contact forces to both H3D and ASCII format

- In the control cards for GLOBAL_OUTPUT_REQUEST, activate the CONTF card and set CONTF_NUM to 2.
- 2. For the first card, set **FORMAT(1)** to H3D and **OPTION(1)** to YES.



3. For the second card, set **FORMAT(2)** to OPTI and **OPTION(2)** to ALL.

Step 7: Create a new Non-linear quasi-static analysis loadstep with SPC set to SPC and NLPARM set to NLPARM.

Step 8: Run the analysis as blocks_s2s.fem





A contour plot of the Surface-to-Surface (S2S) Contact Force (v) for the bottom component

Step 10: Return to the model and change the contact definition to enable node-tosurface analysis: set the DISCRET parameter for the contact definition Contact to N2S.

Step 11: Re-run the analysis as blocks_n2s.fem

Step 12: Review the N2S results in HyperView and compare against the S2S Formulation





A side-by-side contour plot of the Surface-to-Surface (S2S) [left] and Node-to-Surface (N2S) [right] Contact Force (v) for the bottom component



5.2 Contact Types & Properties

OptiStruct supports a number of different contact types depending on their behavior, whether the status is open or closed, or if there is friction or not etc. Contact interface is generally defined through either CONTACT or TIE card image, created through **Interfaces** option available in **analysis** panel or **BCs** in the menu bar. In the model tree, available in **Groups** folder.

Generally, contact definition in OptiStruct is made of a following structure:



5.2.1 Contact Types

- Freeze contact (linear)
 - o It cannot change its status to open (closed all the time)
 - o Zero relative translations and rotations between contact surfaces
 - o Contact gap opening remains fixed at the original value
 - Activated in a **CONTACT** card image as **Type**

• Stick contact (nonlinear)

 \circ $\;$ Active only during closed status, can enter open status $\;$



- No sliding
- Allows relative displacement within contact stiffness, which equals 0.1*STIFF
- Activated in a **CONTACT** card image as **Type**

• Slide contact (nonlinear)

- Applies to both closed and open status
- Has only normal contact stiffness
- No friction
- Valid for small sliding only
- May cause divergence problems (little friction or Stick contact may help)
- o Activated in a CONTACT card image as Type

• Frictional contact (highly nonlinear)

- Frictional properties defined either through PCONT property or Static
 Friction Coefficient in CONTACT card image under Type
- Further discussed in this chapter

• Tie contact (linear)

- The same as Freeze, except:
- Defined by **TIE** card image (not **CONTACT**)
- User can choose which DOFs are freezed (Freeze has always 6 DOFs matched)
- Can be a MPC-based constraint (instead of default penalty-based, discussed later)

• Fast contact (nonlinear)

- o Alternative, simplified contact formulation
- \circ $\,$ Can be used when the only nonlinearity in the model is due to contact
- Two versions are possible: SUPPORT-based and CONTACT/CGAP(G)based
- Further discussed in this chapter



• Thermal contact (nonlinear)

o used for thermal analysis, not discussed in this chapter

5.2.2 Contact Stiffness (Penalty)

The majority of contact implementation within OptiStruct involves the penalty-based process. This can be activated using the CONTACT (PCONT, if necessary) and TIE Bulk Data Entries. (...) The discretized contact elements are assigned variable stiffness values (penalties) depending on the proximity of the contact surfaces to each other and the type of contact forces on the element. The primary purpose of the variable stiffness values is to minimize penetration of the contact surfaces.

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
PCONT	PID	GPAD	STIFF	MU1	MU2	CLEARANCE	SEPARATION		
	FRICESL								

Penalties are assessed by using spring elements to model the effective restitutive force to eliminate the resulting penetration

- As soon as a slave node penetrates the master, an elastic spring is activated between slave node and master segment. This resistant or contact force will tend to reject the slave node.
- Contact force is proportional to penetration distance
- Some degree of penetration (very small) may happen. Convergence is usually improved with this method in return for allowing slight amounts of penetration



F = k' δ

• Linear penalty curve (linear analysis)



In linear static analysis, since no boundary nonlinearity is allowed, contact can be modelled so that it does not change its status throughout the simulation. In which case, the stiffness figure looks like this:



• Linear penalty curve (nonlinear analysis)

To allow a contact status change during analysis, nonlinear subcase must be defined. Linear penalty curve provides constant stiffness value for closed contact, which is defined through PCONT property - STIFF parameter. The stiffness for open contact status is automatically prescribed with very small value, small enough to be neglected.



The AUTO, SOFT, and HARD options are available for automatic predefined stiffness setting (STIFF). Additionally, a positive real value can be specified to directly prescribe



the contact stiffness value. If a negative real value is specified, then it scales the STIFF=AUTO value by the specified scaling factor.

• Nonlinear penalty curve

In the previous section, Linear penalty curve was discussed, wherein, the contact stiffness remains constant when either in the open or closed configuration, and only changes when the contact status is updated. This may, in some small number of cases, lead to convergence difficulties due to the sudden and highly discontinuous nature of the stiffness change. To help smooth the stiffness change at the contact open/close location, nonlinear penalty curves can be used via the STFEXP and STFQDR continuation lines on the PCONT entry. Nonlinear penalty curves are currently only applicable to S2S contact discretizations.



For more information regarding nonlinear penalty curve setup, please refer to Help Documentation, Contact chapter.

5.2.3 Friction Modelling

Friction definition in OptiStruct uses Isotropic Coulomb model, which means that there is initially a static friction, which prevents sliding until a certain tangential force is reached and the sliding begins against constant kinetic friction value. To model a sliding resistance and avoid convergence problems, static friction allows elastic relative displacement, which is opposed by a spring element (with KT stiffness). There are two ways to define friction in the contact interface definition (see the contact definition structure chart):

• When Property option is set to **Static Friction Coefficient** in the **CONTACT** card, this allows **MU1** static friction definition.



(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
CONTACT	CTID	PID/TYPE/MU1	SSID	MSID	MORIENT	SRCHDIS	ADJUST	CLEARANCE	
	DISCRET	TRACK							
+	SMOOTH	SMSIDE	SMREG						

PCONT entry allows setup for both static (MU1) and kinetic (MU2) friction values.
 Providing different values for MU1 and MU2 may lead to divergence due to force discontinuity – see the chart below.



With constant Kt values it provides a situation when initial sliding distance (in the static friction range) is dependent on normal force (Fx), which provides unstable behavior during static-kinetic friction transition. Therefore, by default Elastic Slip Distance model is active, which provides constant initial sliding distance, but the spring stiffness is adaptive (evaluated from this initial slip distance and **MU1**). This is the generally recommended solution for nonlinear problems.





ESL friction model is defined through **FRICESL** entry in **PCONT** card:

- **FRICESL = AUTO** (default), which corresponds to 0.5% of average element size from available master surfaces.
- FRICESL = <non zero value>, elastic slip distance specified in model distance units
- **FRICESL = 0**, activates constant KT model (previously presented)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
PCONT	PID	GPAD	STIFF	MU1	MU2	CLEARANCE	SEPARATION		
	FRICESL								

OptiStruct has an unsymmetric Solver for use with frictional contacts

- Friction constraints produce unsymmetric terms when surfaces are sliding relative to each other
- These terms might have strong influence on convergence specially if frictional stresses have strong influence in overall displacement field
- Unsymmetric solver (MUMPS) will be used by default when friction is specified
- This formulation is more solver intensive, so the solution will be slower, but can speed up the overall convergence while producing more accurate results.
- To turn off unsymmetric solver, use PARAM, UNSYMSLV, NO, if frictional effects are not anticipated to be significant



Example Tutorial on Contacts with Friction



5.2.4 Alternative Approach – MPC-based TIE Contact

MPC based TIE uses multi-point constraints to define a tied contact between master and slave surfaces: each node of the slave surface has the same translations as the master surface. It is activated using CONTPRM, TIE, MPC and once specified all the TIE contacts in the model are defined using MPC based method. This does not apply to contacts defined using CONTACT, FREEZE.



Now the user can specify the DOF to be used by the MPC equations generated for TIE contact for the slave nodes. Any combination of DOF can be used. For MPC based contact you must use CONTPRM,TIE,MPC.

The DOF to be used are specified in field 6 of the TIE data

- The specification is any combination of the numbers 123456
- The Default value is 123

Note that only DOF 1, 2, and 3 can be used if the slave surface is solid elements.

- Including DOF 4, 5, and 6 will also transfer the moment from the slave to the master.
- There is a slight expense for this as the number of MPC equations is doubled.

There are limitations to using MPC-based TIE contacts in an analysis

- The results when using MPC based TIE will not be accurate if there is a physical gap between slave and master. The 6 rigid body modes may not be clean in such case
- MPC based TIE is only available for small displacement analysis



5.2.5 Fast Contact

As mentioned in the Contact Types chapter, Fast Contact, which is a simplified approach for contact modelling, applicable when the only nonlinearity is due to contact presence (no nonlinear material, Small Displacement Nonlinear Analysis), can be set up in two different ways, as a:

- CONTACT/CGAP(G)-based Fast Contact
- SUPPORT-based Fast Contact, which will not be discussed in this chapter as the first one is simpler and sufficient.

To setup a CONTACT-based Fast Contact, user first needs to define Contact Interface (<u>must be N2S discretization</u>) and additionally add **PARAM,FASTCONT,YES** control. NOTE: Fast Contact also supports Pretension.

Example: Wheel Rail Interface using Fast Contact





5.3 Contact Utilities

Additional options are available through **TIE**, **CONTACT**, **PCONT** and **CNTSTB** entry to help adjust contact interface, so that it is feasibly defined. These parameters are highly useful, since in case of divergence, they may help fix the issues without going for geometry edit. Therefore, one should be aware of them, understand their function and be able to implement them when contact is present in the model.

(1)	(2)	(3)		(4)	(5)		(6)		(7)	(8)	(9)	(10)
CONTACT	CTID	PID/TYPE/N	/IU1	SSID	MSI	D	MORIE	INT	SRCHDIS	ADJUST	CLEARANCE	
	DISCRET	TRACK										
+	SMOOTH	SMSIDE		SMREC								
(1)	(2)	(3)		(4)	(5)		(6)		(7)	(8)	(9)	(10)
PCONT	PID	GPAD	S	TIFF	MU1		MU2	CLE	EARANCE	SEPARATION		
	FRICESL											

MORIENT - orientation of contact "pushout" force

It applies only to masters that consist of shell elements or patches of grids. Masters defined on solid elements always push outwards irrespective of this flag. MORIENT is the direction of contact force that master surface exerts on slave nodes.

- In most cases leaving this blank provides correct contact behavior
- Needed only when master surface is defined on shell elements or patches of grids in combination with initial pre-penetration
- Ignored for solid elements master surfaces always push outwards in that case
- Can be set to OPENGAP (default), OVERLAP, NORM, REVNORM
 - a) **OPENGAP** the pushout direction is defined using the assumption that the gap between slave and master is initially open, and the contact condition should prevent their contact





b) OVERLAP - assumes the reverse, namely that the slave and master bodies are already overlapping and the contact condition should push them apart (this is useful in pre-penetrating models when the entire slave set is prepenetrating into the master object)



c) NORM - the pushout force is oriented along the normal vector to the master surface.



d) **REVNORM** - creates pushout force reversed relative to the NORM option.

EXAMPLE: OVERLAP option can be used when **press fit analysis** need to be conducted, in that case contacting cylindrical surfaces are penetrating initially, since the inner element has slightly bigger diameter than the out ring.





SRCHDIS – search distance

The **SRCHDIS** field on the **CONTACT** and **TIE** entries is the search distance criterion for creating the contact interface. When specified, only slave nodes that are within the **SRCHDIS** distance from the master surface will have the contact condition checked. The default value is equal to twice the average edge length of the master surface. For **FREEZE** contact, it is equal to half the average edge length.

ADJUST – contact adjustment (highly useful utility)

Option ADJUST (available in **TIE** and **CONTACT** cards) is used to reposition certain nodes so that they are lying ideally on the master surface. This applies for nodes that are initially penetrating the master body and can also adjust nodes within a specified distance away from the master surface. The repositioning is basically a change in the geometry (for N2S) or change in initial contact opening/penetration (for S2S), so that means no stress and strain is imposed. It can be set to following options:

- a) NO no adjustment (default)
- b) **AUTO** value of 5% of the average edge length on the master surface is the depth criterion
- c) Real value in X.X format (> 0.0) depth of the search zone for adjustment (in <u>the</u> <u>outward direction</u> from the master surface, not into the material penetrating nodes will be adjusted with this option <u>regardless of penetration depth</u>). In case search depth is an integer, it must be specified in X.0 format.
- d) Integer number X (>0) identification number of a SET entry with TYPE = GRID (not a search depth). Only the nodes on the slave entity which also belong to this SET will be selected for adjustment





GPAD – Contact interface padding

Padding (**PCONT entry**) is used for considering the real thicknesses of shell elements, i.e. contact occurs between slave and master on the offset depth equal to real thickness of the master/slave surface.

- By default, set to THICK respects the shell thickness on both sides (slave and master)
- Can be set to NONE

CLEARANCE cannot be set in conjunction with (non-zero) GPAD. Blank GPAD field in presence of CLEARANCE is interpreted as NONE.

GPAD + ADJUST

Notice that when PCONT property is used in contact interface definition and GPAD is then active, ADJUST option repositions the slave nodes with respect to newly specified target contact layer. The result is as follows:





If the MORIENT field is OPENGAP or OVERLAP while the GPAD field in the referred PCONT entry is NONE or zero, the nodal adjustment will be skipped, since for OPENGAP or OVERLAP there is no way to decide the master pushout direction if slave nodes are adjusted to be exactly on the master face.

CLEARANCE – contact clearance

Specific about this parameter is that the clearance value does not consider the actual distance between master and slave entities.

- Can be defined through **CONTACT, PCONT** or via **U0** in the **PGAP** entry.
- Positive **CLEARANCE** is equal to the distance that slave and master surfaces have to move towards each other in order to close the contact. Every slave node can move about this specific **CLEARANCE** value towards master.
- That means if the **CLEARANCE** is set to 0, the contact is initially closed and the slave nodes (even if they are irregularly distant from master surface) cannot move further towards master surface.
- Using **CLEARANCE** overrides the default contact behavior or calculating the initial gap opening from the actual distance between slave and master.
- It is important to pick the right SEARCHDIS so that only desired slave master pairs are involved – else all contact elements even though whose slaves are geometrically distant from the respective master, will be considered
- Negative CLEARANCE indicates contacting bodies have initial penetrations and is equal to resolving interference-fit / press-fit problems

CLEARANCE on the CONTACT entry cannot be used in conjunction with PID of the PCONT entry. In this case, clearance must be specified on the PCONT entry.





- For a CLEARANCE value of 0.0, contact becomes active immediately, irrespective of actual gap between slave and master surfaces.
- For a positive CLEARANCE of 0.07, contact becomes active once the relative displacement between slave and master becomes 0.07.

SMOOTH – contact smoothing

Smoothing improves the approximation of smooth surface behavior (continuous normal) from the FE discretized surface (discontinuous normal)

- Curved surfaces in contact are approximated as group of faceted element faces.
 The use of faceted surface rather than true surface can contribute to inaccuracy in contact stress and contact pressure distributions.
- This only applies to S2S contact surfaces, can be applied to the master/slave surfaces as a whole or can be applied over certain regions of the surfaces.
- Contact stress and contact pressure distributions are usually improved with smoothing.





• Smoothing is defined under CONTACT card and does not apply to FREEZE contacts, or when ADJUST or CLEARANCE is specified.



More uniform distribution of stresses (1st picture) and contact pressures (2nd picture) when contact smoothing is used.

Finite Sliding Contact (TRACK)

For further information, see: **Continuous, Finite vs Small Sliding Contact** chapter.

- TRACK = SMALL (default): activates small sliding contact
- **TRACK = FINITE**: activates finite sliding contact
- TRACK = CONSLI: activates continuous sliding contact

For non-solid elements, the **MORIENT** field should not be set to **OPENGAP** or **OVERLAP** for Continuous, Finite Sliding. If **CONTPRM,CORIENT,ONALL** is active, **MORIENT** is applied to solid elements. In such cases, **MORIENT** should not be set to **OPENGAP** or **OVERLAP** for solid elements also.

SEPARATION – disabling separation of closed contacts



Flag indicating whether the master and slave can separate after the contact has been closed. Applied only to frictional, SLIDE, and STICK contacts with S2S or large-displacement N2S (does not support N2S in Small Displacement Analysis):

- YES (default): Separation could happen even after the contact is closed.
- **NO**: No separation happens once the contact is closed.

CNTSTB – contact stabilization

In some cases, during analysis a rigid body motion may occur due to separation of bodies in contact or sliding without friction (insufficient constraints). Such phenomena often lead to divergence, since no static equilibrium can be achieved. To prevent such behavior, contact stabilization can be switched on. This can be done in two ways:

- As an analysis Expert option: PARAM, EXPERTNL, CNTSTB weak springs are applied on all the contacts and then they are released to zero
- Or through CTNSTB entry (load collector) <u>damping</u> stabilization parameters in normal and tangential direction, which are based on contact penalty stiffness

	Model with Small Dis	splacement Analysis	Model with Large Di	isplacement Analysis
Contact	N2S	S2S	N2S	S2S
Discretization	(CGAPG core)			
PARAM, EXPERNL,	Effective	Effective*	Effective*	Effective*
CNTSTB				
CNTSTB (bulk card	Ineffective	Effective	Effective	Effective
referenced by				
subcase entry)				
PARAM,EXPERNL,	Stabilization with the	For the subcase that	For the subcase that	For the subcase that
CNTSTB	PARAM is effective.	references BULK card	references BULK card	references BULK card
+		CNTSTB, stabilization	CNTSTB, stabilization	CNTSTB, stabilization
CNTSTB (BULK card		with the BULK card is	with the BULK card is	with the BULK card is
referenced by		effective;	effective;	effective;
subcase entry)		For other subcase,	For other subcase,	For other subcase,
		stabilization with the	stabilization with the	stabilization with the
		PARAM is effective.	PARAM is effective.	PARAM is effective.

See below a quick guide for applying contact stabilization:



*For these contacts, the stabilization activated with the PARAM is consistent with the stabilization activated with default settings of the BULK card CNTSTB.

CNTSTB load collector consists of following parameters:

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
CNTSTB	ID		APSTB	LMTGAP		S 0	S1		
	SCALE	TFRAC							

Where **S0,S1,SCALE,TFRAC** deal as scaling factors for stabilization stiffness (with respect to contact stiffness).

- **SCALE** scale for normal stabilization stiffness (default = 1)
- **TFRAC** scaling factor for tangential stabilization stiffness with respect to normal stabilization stiffness (default = 0.1)
- S0 scales the stabilization stiffness at the beginning of the analysis (default = 1, full stabilization at the beginning)
- S1 scales the stabilization stiffness at the end of the analysis (default = 0, no stabilization at the end)

Relationships between these parameters are as follows:

Normal Stabilization for CNTSTB, opening < LMTGAP:

$$K_{CSTB}^{normal} = SCALE \cdot f(t^*) \cdot K_{CREF}$$

Normal Stabilization for CNTSTB, opening \geq LMTGAP:

$$K_{CSTB}^{normal} = 0$$

Tangential Stabilization for CNTSTB:

$$K_{CSTB}^{tangent} = TRFAC \cdot K_{CSTB}^{normal}$$

where, KCREF is the automatically calculated reference stiffness (10-5 times the contact penalty stiffness) for contact stabilization.

• For time, t in a subcase, t^* is calculated as:

$$t^* = (t - t_0) / (t_1 - t_0)$$

- where, t₀ is the time at the start of the subcase and t₁ is the time at the end of the subcase.
- f (t*) is calculated using SO and S1 as:



f(t*) = SO*(1 - t*) + S1 * t*

where SO and S1 are scale factor for stabilization at start and end of subcase respectively

- SCALE is scale factor for normal stabilization (1.0 by default) and TRFAC is the factor for scaling tangential stabilization wrt normal stabilization and is 0.1 is by default
- For temporary instabilities that might occur at the end of analysis, you can specify S1 (scale factor for stabilization at end of subcase) to be a non-zero value
- For convergence difficulties, user may also change the default tangential stabilization factor TRFAC (default = 0.1)

5.3.1 Tutorial: Using ADJUST Option in Contact Analysis

This exercise demonstrates the various ways in which modifying the ADJUST parameter can alter contact modelling issues. The model is a pair of blocks with sliding contact modeled with a node-to-surface formulation and features inconsistent master-slave search depth between the surfaces including a number of initial penetrations. The material models including the property, the applied loading, contact definition, and the load steps are already created, allowing the user to focus on altering the contact definitions.



Problem Setup

You should copy this file: adjust.fem


Step 1: Import the file adjust.fem into HyperMesh Desktop

Step 2: Run the initial model setup and review the results and adjust.out file



A plot of the Contact Force (v) on the bottom component for subcase 1

This model has a high amount of deformation, but OptiStruct was also able to detect that there may have been initial penetrations in the model due to the state of some of the gap elements created for the run. This information can be seen in the adjust.out file:

```
*** WARNING # 312
In static load case 1, the compliance is negative or large -5.56615e+008.
Optimization/buckling analysis cannot be performed.
due to possible rigid body mode.
**** INFORMATION # 942
Apparently some contributions to negative compliance come from
gap/contact elements. This can happen in case of initial interference (U0<0)
or significant attractive force (KB>0). To ignore gap/contact contributions
to compliance, set GAPCMPL to NO on the GAPPRM card.
(Running in-core solution)
Volume = 2.00055E+003 Mass = 0.00000E+000
Subcase Compliance
1 -5.566151E+08
```

Step 3: Alter the contact definition to set an adjustment distance

1. In the HyperMesh client, expand the Group folder in the Model Browser and click

on the top_to_bottom contact definition to load it into the Entity Editor.



 Change the Adjust Option value to Real Value and enter 0.5 in the ADJUST field. This sets the maximum possible adjustment for detected initial penetrations to 0.5 units.

Name	Value
Solver Keyword	CONTACT
Name	top_to_bottom
ID	1
Color	
Include File	[Master Model]
Card Image	CONTACT
User Comments	Hide In Menu/Export
Property Option	Property Type
TYPE	SLIDE
SSID	bottom (1)
MSID	top (2)
MORIENT	
SRCHDIS	
Adjust Option	Real Value
ADJUST	0.5
CLEARANCE	
DISCRET	N2S



Step 4: Run this model as adjust_real_0.5.fem and compare in HyperView

Using an ADJUST of 0.5 searched for both gaps and penetrations within the requested distance and moved all slave nodes in that found space to precisely contact the master surface, resulting in a more uniform distribution.



Step 5: Alter the contact definition to set the adjustment distance to AUTO

- In the HyperMesh client, click on the top_to_bottom contact definition in the Model Browser to load it into the Entity Editor.
- With the Adjust Option value set to String Value, enter AUTO in the ADJUST field. This sets OS to perform an adjustment for any gap or initial penetrations within 5% of the average master element edge length.

Name	Value
Solver Keyword	CONTACT
Name	top_to_bottom
ID	1
Color	
Include File	[Master Model]
Card Image	CONTACT
User Comments	Hide In Menu/Export
Property Option	Property Type
TYPE	SLIDE
SSID	bottom (1)
MSID	top (2)
MORIENT	
SRCHDIS	
Adjust Option	String Value
ADJUST	AUTO
CLEARANCE	
DISCRET	N2S

Step 6: Run this model as adjust_auto.fem and compare in HyperView





Using an **ADJUST** of AUTO searched for both gaps and penetrations within 5% of the average master edge length value and moved all slave nodes in that found space to precisely contact the master surface.

5.3.2 Tutorial: Utilizing CLEARANCE with a Variable Gap

This exercise demonstrates using the CLEARANCE contact parameter with a variable gap part set. The model is a pair of blocks with sliding contact modeled with pressure loading on the outer face. The material modelling including the property, the applied loading, and the load step are already created. The user must create the contact definition, alter the clearance value, and review the analysis results.



Problem Setup

You should copy this file: variablegap.fem

Step 1: Import the file variablegap.fem into HyperMesh Desktop

Step 2: Define the contact surfaces and contact using Auto Contact tools

 In the drop-down menu, click on View > Browsers > HyperMesh > Contact to bring up the Contact browser.

Session	Mask	Model	Contact >	×		
Enter Sear	ch String					\bigcirc \checkmark $>$
۰ 🖷	*	¢.	🧇 •	• 🖶	ନ୍ମ 📄	() +/- 1
Entities			ID 💊 📦	State Per	netration E	xport Status
🕀 🍋 Co	mponents	(2)				
🖶 🛜 Se	ts (6)					
🗄 🝋 Co	ntact Prop	erties (1)				
						-
Enter Sear	ch String					Q, ~ ×
• 🖷	9	\$ *	- 💞 -	• 🖶	k 📄	() +/- 1
Entities ID	State Pen	etration Ex	port Status	Type Sla	ve Master	Property Frict

2. Right-click in the Contact Browser area and select Auto Contact.

💪 Create AutoContact	×
Name	Value
Pick application region	0 Components
Parameters	
Contact tolerance type	Vicinity tolerance
Vicinity tolerance	1.0
Reverse angle	15.0
Consolidate contact pairs	
Contact type	Touch
Master entity type	Set of elements
Slave entity type	Set of nodes
Contact property	
Property option	Property Type
Property type	SLIDE
Contact Detector Options	
Run penetration check before contact creation and auto adjust penetrating elements	
	Create Close

This brings up the Create AutoContact dialog.

3. Click on the **Components** entity selector to bring up the **Select Components** dialog. Select the TOP and BOT components.

Δ	Select C	omponents	×
Ente	er Search S	String	Q, ~
	Name TOP BOT	ID 1 2	Color
F			2 selected.
		OK	Cancel

- 4. In the **Create AutoContact** browser, set the **Vicinity tolerance** to 0.5.
- Set the Property Option to Use Existing Property Id and set Select existing contact property to NONE (2).

💪 Create AutoContact	×
Name	Value
Pick application region	2 Components
Parameters	
Contact tolerance type	Vicinity tolerance
Vicinity tolerance	0.5
Reverse angle	15.0
Consolidate contact pairs	
Contact type	Touch
Master entity type	Set of elements
Slave entity type	Set of nodes
Contact property	
Property option	Use Existing Property Id
Select existing contact property	NONE (2)
Contact Detector Options	
Run penetration check before contact creation and auto adjust penetrating elements	
	Create Close

- 6. **Create** the Auto Contact.
- In the Model Browser, expand the Properties section of the model tree and change the name of the contact property NONE to AutoContact. Edit the CLEARANCE on the contact property to 0.1.

Name	Value
Solver Keyword	PCONT
Name	AutoContact
ID	2
Color	
Include File	[Master Model]
Defined	
Card Image	PCONT
User Comments	Do Not Export
GPAD_OPT	
GPAD	
STIFF_REAL_VAL	
STIFF	AUTO
MU1 Options	Real Value
MU1	
MU2	
CLEARANCE	0.1
SEPARATION	
FRICESL_opts	
FRICESL	
STFEXP	
STFQDR	
PCONTX	
PCONTHT	

Step 4: Run this model as variablegap_0.1.fem and review the results in





A contour plot of the Contact Force (v) on the elements of the contact interface for subcase 1



Step 5: Return to HyperMesh, reset the CLEARANCE to 0.07, run this model as variablegap_0.07.fem and review the results in HyperView

Step 6: Return to HyperMesh, reset the CLEARANCE to 0.0, run this model as variablegap_0.0.fem and review the results in HyperView



A contour plot showing Contact Force, Element Stresses (2D & 3D) (Pressure), and Displacement (Mag) for each of the clearance levels: 0.1 (first column), 0.07 (second column), and 0.00 (final column)



5.4 Contact Setup – General Steps

Contact definition can be performed through either the **Model Browser** or the browser for contacts the **Contact Browser**. Generally, Contact Browser is a very well sorted utility that might enhance your contact setup process, but when there is, for instance, just one contact to define in the model, then there is no need for opening the Contact Browser - all actions can be performed in **Model Browser** or **analysis** panel. However, for large assemblies it is way more convenient to follow the **Contact Browser** structure or even apply **AutoContact** which is capable of generating multiple contact interfaces at once.

Additional features of Contact Browser:

- Swap Master-Slave option available by right clicking in second pane on chosen contact interface.
- Swap Cp-Tie option available in the same way



5.4.1 Via Contact Browser – Manually

Contact browser can be found in **Tools** menu or **View** menu. The Contact browser consists of three panes; the first pane displays the entities needed for contact definition, the second pane displays the contact information, and the third pane displays the Entity Editor.



In the Contact Browser you can manually create contacts.

 In the first pane of the Contact browser, right-click and select Create from the context menu to create all of the entities needed for contact definition, such contact surfaces.

You can create:

- **Contact surfaces**, which is the most common approach to define contact entities, for detailed information, see chapter below about setting up contact surfaces.
- **Contact properties,** in case contact needs additional parameters set up, like friction or nonlinear contact stiffness
- Sets, instead of contact surfaces, sets of elements (SET_ELEM) or nodes (SET_GRID) can be created. Elements set is used for both master and slave in S2S contact and for master in N2S contact. Grids set is used for slave in N2S contact.
- 2. In the second pane of the Contact browser, right-click and select Create from the context menu to create contacts. The Entity Editor opens and displays the new contact.

You can create:

• **CONTACT** card image for all contact types except TIE contact



• TIE card image

Note: The entities available in the context menu will vary depending on the user profile loaded.

- 3. Assign Master and Slave entities when creating contact pairs and contact ties.
- 4. Define additional entity parameters and properties accordingly.

5.4.1.1 Contact Surface Setup

Contact surface, as visible in **Entity Editor**, has a **SURF** card image. It requires elements to be added, based on which the contact surface will be evaluated.

 User needs to click on **0 Elements**, then again click on **Elements** in the yellow bracket.

Card Image	SURF
Elements	0 Elements
User Comments	Hide In Menu/Export

Card Image	SURF	
Elements	Elements	I
User Comments	Hide In Menu/Export	

 Now a specific menu in the panel area pops up. The most common options are Add Shell Faces and Add Solid Faces. With these the user can pick a face by clicking in the Graphics Area



- 3. The other way to choose a face is to change the second option in the panel area from **faces** to **elems**.
 - a) Pick all elements for which face search will be performed
 - b) Switch to **nodes** selection. Now define three nodes on the face, which is desired for contact surface. Contact surface will then be created only for the face which contains the three nodes.
- 4. Click Add.

5.4.2 In Model Browser – Manually

Generally the same steps as done via the Contact Browser are valid for contact setup in Model Browser. However, user needs to access all the controls through different options:

- Contact surfaces are available in Model Browser by right clicking
 -> Create > Contact Surface or via analysis panel -> contactsurfs.
- CONTACT interface is available either through Model Browser by right clicking -> Create ->



Contact or via analysis panel -> interfaces. Type should be set as CONTACT.

- TIE interface is available through analysis panel -> interfaces too. The Type should be chosen as TIE. Through Model Browser user can create a TIE interface by creating CONTACT interface and changing the Card Image to TIE.
- Sets can be created independently in Model Browser by right clicking -> Create > Set or through analysis panel -> entity sets.

vectors	load types		interfaces	control cards	0	Geom
systems	constraints	accels	rigid walls	output block	0	1D
preserve node	equations	temperatures	entity sets	loadsteps	C	2D
	forces	flux	blocks		C	3D
	moments	load on geom	contactsurfs	optimization	•	Analysis
	pressures		bodies		C	Tool
			nsm	OptiStruct	C	Post



5.4.3 Via Contact Browser – AutoContact

A very different way on how to create contacts – especially for large models with many different contacts – is to employ the so called AutoContact.

The Auto Contact automatically searches for possible contacts across selected components based on vicinity tolerance. For components within the vicinity tolerance interfaces of type CONTACT are created. As in the manual process described above, the type Slide, Stick, Freeze, Friction (MU1) may be specified.

Create AutoContact		×
Name	Value	
Pick application region	0 Components	
Parameters		
Contact tolerance type	Vicinity tolerance	
Vicinity tolerance	1.0	
Reverse angle	15.0	
Consolidate contact pairs		
Contact type	Touch	
Master entity type	Set of elements	
Slave entity type	Set of nodes	
Contact property		
Property option	Property Type	
Property type	SLIDE	
Contact Detector Options		
Create contact entities with confirm state		
Run penetration check before contact creation and auto adjust penetrating elements		
	Create	Close

Auto contact generation works independent of whether the model is FE or geometric. When you elect to generate contacts automatically, contact bodies are detected and contact pairs and contact sets are created.

 In the Contact browser, right-click and select AutoContact from the context menu. The Create AutoContact dialog opens.

Tip: To select all of the components in the model, right-click on the **Components** folder.

2. In the **Pick application region** field, select components to create contact interfaces and contact surfaces between.

Note: You must select a minimum of two components.

- 3. Define contact parameters and properties accordingly.
 - a. Select a Contact tolerance type.



- b. If Vicinity tolerance is selected, you must enter a vicinity tolerance value.
- c. Enter a **Reverse angle limit**. This determines the maximum curve angle to identify contacts.
- d. Enable **Consolidate contact pairs** to group multiple isolated surfaces into one surface.
- e. Select a Contact type to specify the type of contact to be generated.
 Select Touch to generate "Contacts", and Gap/Tie to generate "Tie".
- f. Create and assign, or pick an existing property to attach to the contact pair. **Property Option** field defined as in entity editor
- 4. Click Create. Based on a vicinity tolerance, the AutoContact tool searches throughout all the selected components and automatically creates new Contact interfaces and contact surfaces between them. All the objects created are populated in the Contact browser into their respective folders.
- 5. Review your contacts by right clicking on the contact interfaces in second pane.

IMPORTANT NOTES for AutoContact:

Contact entities are created only for directly adjacent elements, which instead of using full faces as contact surfaces, creates entities like this:





This is a case of Finite or Continuous Sliding Contact and is not a correct solution, therefore, such contacts need either reassignment of contact entities or a manual setup.

- In case of multiple contacts creation, all of them will have the same interface type. Therefore, if, for instance, friction is required in just a part of them, they should be redefined manually.
- AutoContact does not assign a part with finer mesh as slave, it is a random process and all the interfaces with improper master/slave relation, should be edited with Swap Master-Slave option.

5.5 Analysis Controls for Contacts

Some of the controls available through NLADAPT load collector were not discussed before as their usage is specifically for contact-dependent convergence. Since in nonlinear analysis it is allowed to change contact status (open <-> closed, stick <-> slip), solver achieves convergence, besides the UPW criteria, when contact status is stable, which means, there is no contact status change between iterations.

ID	Unique identification number
NCUTS	Number of cutbacks allowed to reduce the time increment.
DTMAX	Maximum time increment allowed.
DTMIN	Minimum time increment allowed.
NOPCL	Number of grids allowed to have open-close contact status change.
NSTSL	Number of grids allowed to have stick-slip contact status change when the current time step converged.
FXTRΔ	Determines whether to use linear extrapolation for load increments.



NLADAPT card:

- NOPCL when the solution has converged regarding UPW criteria, but the contact status is still changing for a certain number of grids, user can specify an integer value in NOPCL entry to allow that number of grids change their status in the converged iteration solver will complete analysis successfully when number of grids with contact status change is the same or smaller than the NOPCL value.
- NSTSL the same as above, but applied to stick <-> slip status change.
 Supported only for LGDISP.

5.6 Contact Output

As for models, results for contact interfaces can also be requested for HyperView. Contact forces are activated through **GLOBAL OUTPUT REQUEST/CONTF** card. Set **TYPE(1) = ALL** to request all the following values (**PCONT** – pressure type results + **FORCE** – force results + **FRICT** – frictional results):

- Contact Pressure **CONTF(PCONT)**
- Open-closed status information CONTF(PCONT)

Contour results will be available, which say assign 0.0 for Open status and 1.0 for closed status. Intermediate values indicate transition between Open and Closed.

- Contact Gap Opening and Penetration CONTF(PCONT)
- Contact Forces CONTF(FORCE)
- Frictional Traction CONTF(FRICT)

Frictional traction is output in vector form and scalar form respectively for H3D format. Contact Traction/Tangent Vector (v) contains the components of the friction in the basic coordinate system, while Contact Traction/Tangent(s) contains the components of the friction in the local coordinate system of the contact surface. Friction results are always tangential to the surface, and Contact Normal pressure results are always perpendicular to the surface.



• Sliding Distance – CONTF(FRICT)

Sliding Distance represents total sliding distance accumulated while the surfaces are in contact. This may be different than just the difference in displacements between the starting and final position.

• Stick-slip status information – CONTF(FRICT)

Slip/Stick Status is represented by 0.0 for Open, 1.0 for Slip and 2.0 for Stick. On graphical display, intermediate values may appear due to transition of status across individual elements.

5.6.1 Print Contact info Prior to Analysis

Summary of initial conditions for contact interfaces is now available in an ascii format file named *.CPR. This information could be helpful while debugging complex contacts. As this information is printed in as a separate table for each contact interface in the model.

*.CPR file is generated, if CONTPRM, PREPRT, YES is specified in any input file which contains a contact condition. & this file can be generated even with a check run.

This file contains the following data for contact.

 Contact interface ID, Slave grid ID, Number of connected master grids, Number of connected slave grids (only for surface to surface) ,Contact Status (OPEN/CLOSED/FROZEN) ,Gap Opening, Raw gap opening based on mesh (adjusted mesh coordinate with Node to surface ADJUST),Gap padding (GPAD, S2S ADJUST, CLEARANCE and so on) and Nodal contact area on the core slave node (only for surface to surface)

This is currently supported for Linear, NLSTAT or NLTRANS analysis. Supported contacts types are: SLIDE, FREEZE, CONSLI, & FINITE.

SUMM	SUMMARY OF INITIAL CONDITIONS OF CONTACT INTERFACES					
****	********	***************************************				
Type	Name	Content				
* * * * *	********	***********************				
(I)	CONTID	Contact interface ID				
(I)	SLVGID	Core slave grid ID of the gap				
(I)	NCMSTG	Number of connected master grids of the gap	(I): Integer			
(I)	NCSLVG	Number of connected slave grids beside the core slave	number			
		grid of the gap (only for S2S contact)	(R): Real number			
(C)	STATUS	Gap status (OPEN/CLOSED/FROZEN)	(C): Characters			
(R)	OPENING	Gap opening (negative value means penetration)				
(R)	RAWGAP	Raw gap opening based on mesh coordinates				
		(adjusted mesh coordinates with N2S ADJUST)				
(R)	PADDING	Gap padding due to GPAD, S2S ADJUST, CLEARANCE, etc.				
(R)	SAREA	Nodal contact area on the core slave grid				
		(only for S2S contact)				

Example:



Contact In S Ma	terface (S2 lave SURF/S ster SURF/S	2S): SET: SET:	2 "sh 3 "sl 4 "ma	ell_conta ave" ster"	act"				
CONTID	SLVGID	NCMSTG	NCSLVG	STATUS	OPENING	RAWGAP	I	PADDING	SAREA
2	2081	16	8	OPEN	0.50000E+0	0.55000E+	-01	0.50000E+01	0.51840E+02
2	2082	12	8	OPEN	0.50000E+0	0.55000E+	-01	0.50000E+01	0.51840E+02
2	2083	9	8	OPEN	0.50000E+0	0.55000E+	-01	0.50000E+01	0.51840E+02
2	2084	12	8	OPEN	0.50000E+0	0.55000E+	-01	0.50000E+01	0.51840E+02
2	2112	9	8	OPEN	0.50000E+0	0.55000E+	-01	0.50000E+01	0.51840E+02
2	2113	12	8	OPEN	0.50000E+0	0.55000E+	-01	0.50000E+01	0.51840E+02
2	2133	12	8	OPEN	0.50000E+0	0 0.55000E+	-01	0.50000E+01	0.51840E+02
2	2134	9	8	OPEN	0.50000E+0	0 0.55000E+	-01	0.50000E+01	0.51840E+02
2	2151	9	8	OPEN	0.50000E+0	0.55000E+	-01	0.50000E+01	0.51840E+02
2	2166	12	8	OPEN	0.50000E+0	0.55000E+	-01	0.50000E+01	0.51840E+02
2	2167	9	8	OPEN	0.50000E+0	0.55000E+	-01	0.50000E+01	0.51840E+02
2	2184	9	8	OPEN	0.50000E+0	0.55000E+	-01	0.50000E+01	0.51840E+02
2	2204	12	5	OPEN	0.49999E+0	0.55000E+	-01	0.50000E+01	0.25920E+02
2	2205	9	5	OPEN	0.50000E+0	0.55000E+	-01	0.50000E+01	0.25920E+02
2	2206	9	8	OPEN	0.50000E+0	0.55000E+	-01	0.50000E+01	0.51840E+02

Same contact with CLEARANCE = 0.0

Contact In S Ma	terface (S2 lave SURF/S ster SURF/S	2S): BET: BET:	2 "sh 3 "sl 4 "ma	ell_contac ave" ster"	t"			
CONTID	SLVGID	NCMSTG	NCSLVG	STATUS	OPENING	RAWGAP	PADDING	SAREA
2	2081	16	8	CLOSED	0.00000E+00	0.55000E+01	0.55000E+01	0.51840E+02
2	2082	12	8	CLOSED	0.00000E+00	0.55000E+01	0.55000E+01	0.51840E+02
2	2083	9	8	CLOSED	0.00000E+00	0.55000E+01	0.55000E+01	0.51840E+02
2	2084	12	8	CLOSED	0.00000E+00	0.55000E+01	0.55000E+01	0.51840E+02
2	2112	9	8	CLOSED	0.00000E+00	0.55000E+01	0.55000E+01	0.51840E+02
2	2113	12	8	CLOSED	0.00000E+00	0.55000E+01	0.55000E+01	0.51840E+02
2	2133	12	8	CLOSED	0.00000E+00	0.55000E+01	0.55000E+01	0.51840E+02



5.7 General Tips for Contact Setup

Which entity should be chosen as slave and which as master?

- Smaller mesh component should be used as slave
- More curved surface should be chosen as slave
- Stiffer part should be chosen as master
- Generally smaller components should be used as slave

What is the influence of contact parameters on simulation time?

- Simulation time is influenced by the size of the slave entity and almost does not depend on the master entity size
- Presence of friction increases simulation time by 2 times, therefore it should be used only when it is crucial

When there are a number of contact interfaces that need to share a common part, what is the solution for choosing master or slave?

• Slave entities cannot be shared amongst two or more contact interfaces, therefore the common part must be the master entity

5.8 Tutorial : Nonlinear Analysis of an Interference

Fit

This tutorial demonstrates how to carry out non-linear quasi static analysis in OptiStruct of a 1 mm interference/press fit as well as non-linear quasi static analysis of the interference fit along with an enforced displacement of 5 mm.

Download the model file first.

Model Description

Figure 1 illustrates the structural model used for this tutorial, containing two solid cylinders. The dimensions of the cylinders can be found in the table below. The narrow cylinder is being press fit into the wide cylinder, with an interference of 1 mm. The



two solid cylinders are meshed using a tetramesh with an average element size of 15 mm.

Units	Length: mm; Time: s; Mass: Mgg; Force: N; Stress: MPa
Wide Cylinder	OD: 250 mm; ID: 60 mm; Height: 100 mm
Narrow Cylinder	OD: 61 mm; Height: 500 mm



Step 1: Import Model

1. Import the model into OptiStruct.

Step 2: Define Materials

Define the Non-Linear Block Material

1. Create a MAT1 card with the following properties:

Name	Value
Solver Keyword	MAT1
Name	Steel-Block
ID	40
Color	
Include	[Master Model]
Defined	
Card Image	MAT1
User Comments	Do Not Export
E	195000.0
G	
NU	0.29
RHO	8e-009



2. Select the box next to MATS1. This will allow us to set the non-linear properties of the material. Ensure the following fields are filled:

🖻 MATS1	
TID	<unspecified></unspecified>
TYPE	PLASTIC
Н	
YF	
HR_REAL	
HR	
LIMIT1	
TYPSTRN	1

3. To create the Stress vs Strain table, create a new TABLES1 card with the following properties:

Value
TABLES1
StressVsStrain
54
[Master Model]
TABLES1
Do Not Export
6

4. Fill in the table with the following data:

TABLES1_NUM =		×	
	x	у	
1	0.0	275.0	
2	0.03	340.0	
3	0.06	372.0	
4	0.1	393.0	
5	0.15	408.0	
6	0.3	412.0	



5. To link the Stress vs Strain table to the material, set the TID of the MATS1 card

to the Stress vs Strain table as such:

MATS1	
TID	(54) StressVsStrain
TYPE	PLASTIC
Н	
YF	
HR_REAL	
HR	
LIMIT1	
TYPSTRN	1

Define the Rigid Shaft Material

1. Create a MAT1 card with the following properties:

Name	Value
Solver Keyword	MAT1
Name	Steel-RigidShaft
ID	44
Color	
Include	[Master Model]
Defined	
Card Image	MAT1
User Comments	Do Not Export
E	400000.0
G	
NU	0.29
RHO	8e-009

Step 3: Define Properties

1. Create two PSOLID Card one for the Block and one for the Shaft:

Name	Value	Name	Value
Solver Keyword	PSOLID	Solver Keyword	PSOLID
Name	PBlock	Name	PShaft
ID	40	ID	44
Color		Color	
Include	[Master Model]	Include	[Master Model]
Defined		Defined	
Card Image	PSOLID	Card Image	PSOLID
Material	(40) Steel Block	Material	(44) Steel_Shaft
User Comments	Do Not Export	User Comments	Do Not Export
CORDM options	BLANK	CORDM options	BLANK
ISOP		ISOP	
FCTN		FCTN	
PSOLIDX		PSOLIDX	



Name	Value
Solver Keyword	PCONT
Name	PCONT
ID	1
Color	
Include	[Master Model]
Defined	
Card Image	PCONT
User Comments	Do Not Export
GPAD_OPT	
GPAD	
STIFF_REAL_VAL	
STIFF	AUTO
MU1 Options	Real Value
MU1	0.2
MU2	
CLEARANCE	
SEPARATION	YES
FRICESL_opts	

2. Create a PCONT Card for the Contact Surfaces:

3. Assign the Block & shaft component to the PBlock Property:

Name	Value	Name		Value
Name	Block	Nan	ne	Shaft
ID	40	ID		44
Color		Colo	or	
Include	[Master Model]	Incl	ude	[Master Model]
Property	(40) PBlock	Prop	perty	(44) PShaft
Material	(40) Steel_Block	Mat	erial	(44) Steel_Shaft

Step 4: Define Contact

1. Create two contact surfaces:

🖨 🙀 Contact Surfaces (2)		
🗾 Block	1 🛄	0
🔤 📶 Shaft	2 📘	0

2. To select the faces for the Block Contact Surface, click Elements.

Name	Value
Solver Keyword	SURF
Name	Block
ID	1
Color	
Include	[Master Model]
Card Image	SURF
Elements	0 Elements
User Comments	Hide In Menu/Export
FACE_FORMAT	

Ensure the panel area looks like this:

-	add solid faces	add reject
-	faces I	review review direction flip
		return

Select the inner faces of the Block and click add. The selected faces will turn white.



3. To select the faces for the Shaft Contact Surface, click Elements.

Name	Value
Solver Keyword	SURF
Name	Shaft
ID	2
Color	
Include	[Master Model]
Card Image	SURF
Elements	Elements
User Comments	Hide In Menu/Export
FACE_FORMAT	



Ensure the panel area looks like this:

.	add solid faces			add
.	faces II	face angle =	30.000	review
				review direction flip
				return

Select the outer faces of the Shaft and click add. The selected faces will turn white.



4. Create a CONTACT Card with the following properties:

Name	Value
Solver Keyword	CONTACT
Name	ContactBlockShaft
ID	1
Color	
Include	[Master Model]
Card Image	CONTACT
User Comments	Hide In Menu/Export
Property Option	Property Id
PID	(1) PCONT
SSID	(1) Block
MSID	(2) Shaft
MORIENT	
SRCHDIS	
Adjust Option	String Value
ADJUST	NO
DISCRET	S2S
TRACK	FINITE
CONTACT_NUM_SMOOTH =	0

Step 5: Define Loading

Define SPCs

- 1. Create three load collectors named SPC_Shaft, SPC_Block, and SPC_Sum
- Make SPC_Shaft the current collector and constrain one end of the shaft with DOFs 1, 2, and 3 fixed. The entire face of the shaft should be constrained. Ensure the Load Type is set to SPC:



3. Make SPC_Block the current collector and constrain the outside of the block with DOFs 1, 2, and 3 fixed. Additionally. Constrain the other end of the shaft with DOF 1 fixed. The entire face of the shaft should be constrained. Ensure the Load Type is set to SPC:





4. Make SPC_Sum the current collector and set the Card Image to SPCADD. Fill in the table as follows:

Name	Value
Solver Keyword	SPCADD
Name	SPC_Sum
ID	59
Color	
Include	[Master Model]
Card Image	SPCADD
User Comments	Do Not Export
SPCADD_Num_Set =	2
Data: S	

	SPCADD_Num_Set =	Х
	S	
1	(56) SPC_Block	
2	(58) SPC_Shaft	

Define Enforced Displacement

 Create a load collector named EnforcedDisplacement. On the end of the shaft opposite to that constrained in SPC_Shaft, constrain the shaft with an enforced displacement of 5 mm in the x-direction. The entire face of the shaft should be constrained. Ensure the Load Type is set to SPCD. The panel area should be as follows:



create C update	Image: size and size a	Image: style styl
	load types =	SPCD

Define Non-Linear Cards

1. Create a NLOut Card with the following properties:

Name	Value
Solver Keyword	NLOUT
Name	NLOut
ID	55
Color	
Include	[Master Model]
Card Image	NLOUT
User Comments	Do Not Export
🗉 NINT	
VALUE	10
SVNONCNV	
VALUE	YES

2. Create a NLMon Card with the following properties:

Name	Value
Solver Keyword	NLMON
Name	NLMon
ID	53
Color	
Include	[Master Model]
Card Image	NLMON
User Comments	Do Not Export
ITEM	DISP
INT	ITER

3. Create a NLParm Card with the following properties.

Name	Value
Solver Keyword	NLPARM
Name	NLPARM
ID	1
Color	
Include	[Master Model]
Card Image	NLPARM
User Comments	Do Not Export
NINC	10
DT	0.0
KSTEP	
MAXITER	100



Step 6: Set up 1st Subcase

1. Create a Load Step with the following properties:

Name	Value
Solver Keyword	SUBCASE
Name	InterferenceFit
ID	1
Include	[Master Model]
User Comments	Hide In Menu/Export
Subcase Definition	
Analysis type	Non-linear quasi-static
SPC	(56) SPC_Block
LOAD	<unspecified></unspecified>
NLPARM	(1) NLParm
NLPARM(LGDISP)	 <unspecified></unspecified>
SUPORT1	<unspecified></unspecified>
DEFORM	<unspecified></unspecified>
PRETENSION	<unspecified></unspecified>
MPC	<unspecified></unspecified>
STATSUB (PRETENS)	<unspecified></unspecified>
NLADAPT	<unspecified></unspecified>
NLOUT	(55) NLOut
CNTSTB	<unspecified></unspecified>
DLOAD	<unspecified></unspecified>
MOTNJG	<unspecified></unspecified>
SUBCASE OPTIONS	
LABEL	
label	Load Case 1
SUBTITLE	
ANALYSIS	
TYPE	NLSTAT
CNTNLSUB	
EIGVRETRIEVE	
EIGVSAVE	
ENDLOAD	
FLLWER	
MODCHG	
NLMON	
ID	(53) NLMon



Step 7: Set up 2nd Subcase

1. Create a Load Step with the following properties:

Name	Value
Solver Keyword	SUBCASE
Name	EnforcedDispLoading
ID	2
Include	[Master Model]
User Comments	Do Not Export
Subcase Definition	· · · · · · · · · · · · · · · · · · ·
🖃 Analysis type	Non-linear quasi-static
SPC	(59) SPC_Sum
LOAD	(57) EnforcedDisplacement
NLPARM	(1) NLParm
NLPARM(LGDISP)	<unspecified></unspecified>
SUPORT1	<unspecified></unspecified>
DEFORM	<unspecified></unspecified>
PRETENSION	<unspecified></unspecified>
MPC	<unspecified></unspecified>
STATSUB (PRETENS)	<unspecified></unspecified>
NLADAPT	<unspecified></unspecified>
NLOUT	(55) NLOut
CNTSTB	<unspecified></unspecified>
DLOAD	<unspecified></unspecified>
MOTNJG	<unspecified></unspecified>
SUBCASE OPTIONS	
🗆 LABEL	
label	Loading
SUBTITLE	
ANALYSIS	
TYPE	NLSTAT
CNTNLSUB	
OPTION	YES
EIGVRETRIEVE	
EIGVSAVE	
ENDLOAD	
FLLWER	
MODCHG	
NLMON	
POST	
RADSND	
RESVEC	
SOLVTYP	
TEMP	
TEMP_LUAD	

Step 8: Submit Job and View Results for loadcase 1 and 2

 Submit the job to OptiStruct and open the results in HyperView. To view Contour Plots in HyperView, load the model. After loading the model, click



the Contour Panel Button to open the Contour Panel: . Then, select which Contour Plot you would like to view and click "Apply" to display the plot on the model. In the images below, Displacement, von Mises Stress, von Mises Strains, and Plastic Strains are utilized.

Result type: Displacement (v)	Selection:	Averaging method:	Value filter:	Display Legend Result
Mag 💌	Resolved in:	□ Variation < 10 (%)		Clear Contour
Layers: 💽 🛒 - 🎯	Analysis System 👻	Averaging Options		Create Plot Style
🔲 Use corner data	Use tracking system	Envelope trace plot	Cache	Projection Bule
	M Show midside node results	None 💌	Apply	Query Results

2. Ensure Load Case 1 results are as follows



5.9 Capstone Tutorial Part 1: Setting Up Frictional Contact in a Bolted Pipe Flange Model

This exercise requires the user to set up contact surfaces, contact properties, and contacts for a bolted pipe flange modeled with symmetrical constraints in a 120-degree arc. The material model details including the property, the symmetry constraints, and the tied contacts for some surfaces are already created, allowing the user to focus on creating the frictional contact definitions.



Problem Setup

You should copy this file: bpf_ex6a.fem

Step 1: Import the file bpf_ex6a.fem into HyperMesh Desktop

Step 2: Create contact surfaces for the Bolt and Pipe_Flange_upper components in the regions where contact is expected



The amount of upper pipe flange to set up as a contact surface for the bolt-flange contact



The contact surface on the bolt head (the Bolt_Flange_upper has been masked)

Step 3: Setup a frictional PCONT definition and use that as the property for a new contact definition between the Bolt and Pipe_Flange_upper contact surfaces as shown



Name	Value	Name	Value
Solver Keyword	CONTACT	Solver Keyword	PCONT
Name	Bolt to PipeFlangeUpper	Name	Frictional Contact
ID	3	ID	6
Color		Color	
000		Include File	[Master Model]
Include File	[Master Model]	Card Image	PCONT
Card Image	CONTACT	User Comments	Hide In Menu/Export
User Comments	Hide In Menu/Export	GPAD_OPT	
Property Option	Property Id	GPAD	
PID	Frictional Contact (6)	STIFF_REAL_VAL	
CCID	Polt BiseElangel IsperSide (11)	STIFF	AUTO
5510	Boil_riperiarigeopperside (11)	MU1 Options	Real Value
MSID	Pipe_Flange_Upper_BoltSide (10)	MU1	0.2
MORIENT	NORM	MU2	
SRCHDIS		CLEARANCE	
Adjust Option	String Value	FRICESL_opts	
		FRICESL	
DICCDET		PCONTX	
DISCHEI		PCONTHT	

Tip: The slave and master identifiers can be selected under contactsurfs not set.

Step 4: Create contact surfaces for the Bolt and Pipe_Flange_lower components in the regions where contact is expected

Step 5: Setup a contact definition between the Bolt and Pipe_Flange_lower contact surfaces using the frictional contact property definition created in

Step 6: Create contact surfaces for the complete facing surfaces of the Pipe_Flange_upper and Pipe_Flange_lower components





Note: The Pipe_Flange_lower contact surface may require multiple surface selections in order to add elements on either side of the gasket channel.



Step 7: Setup a contact definition between the Pipe_Flange_upper and Pipe_Flange_lower contact surfaces using the frictional contact property definition from Step 2

Step 8: Save the model file and export the FE model as bpf_ex6b.fem
6 Bolt Pretension

Assemblies of parts, which are subjected to static structural analyses usually besides contact definition, include some connections, amongst which the most common are welds, bolts & nuts, keys etc. on the other hand such a "connection" can also be a close fit. Some of them, due to assembly conditions, introduce additional loading, which on one level can be necessary for proper connection reliability (bolts) or on another occurs as unwanted side-effects (welding stresses). Such an internal loading may in some cases be the crucial factor for proper assembly work, therefore it needs to be included in strength verification.

A typical nonlinear problem in mechanical assemblies is presence of bolt & nut assembly. Nonlinearity in that case originates for instance from multiple contacts between mounted parts. Therefore, it seems to be a perfect example of Nonlinear Quasi-Static Analysis.

6.1 Pretensioning Process

- In the bolt & nut assembly, first step usually includes tightening up the nut with force F, so that it locks up the mounted parts. A particular minimum compression force is required in most cases, so that eventual tensile forces F' will not detach the assembly.
- 2. Such nut tightening introduces compressive stresses to mounted parts simultaneously stretching the bolt and therefore inducing tension. The bolt is stretched by distance: ΔL . This depends on applied force and stiffness of bolt and mounted parts:



3. External forces are applied to the assembly (working condition):



6.2 Bolt Pretension in OptiStruct

In analysis of structures, the FEA model is usually defined in material reference frame and the amount of material is assumed to remain fixed, while the structure undergoes stretching and deformation. However, in the case of pretensioning, the actual working part of the model has some material removed by being driven through the nut (usually the protruding part of the bolt is not included in the working FEA model, since it does not participate in the balance of forces on the structure).

The simulation of this phenomenon in Optistruct follows the approach described below:

First, it is recognized that for straight bolts, from the viewpoint of balance of forces it does not matter at what location the removal of the material happens in the bolt. Therefore, instead of simulating the precise interaction between the nut and the bolt,



the pretensioning is handled within the length of the bolt. Pretensioning in OptiStruct is implemented with the help of Multiple Point Constraints (MPC's) via two different processes:

- 1D Bolt Pretensioning
- 3D Bolt Pretensioning

Multiple point constraints (MPC's) are used in both 1D and 3D pretensioning, the difference between the two implementations is the number of duplicate grid points created and controlled via MPC's.

6.2.1 1D Bolt Pretensioning

Many bolts are represented as 1D entities with pretension applied to 1D non-rigid elements (rod, bar, beam).

- 1. For each bolt, a 'cut' or pretension section is defined and subject the section to tensile loads
- 2. The length of bolt at pretension section will change by amount necessary to carry prescribed load, while accounting for the compliance of system
- 3. In the next steps, the prescribed bolt loads are replaced by fixed DI
- 4. The rest of bolt is free to deform under assembly loads





6.2.2 3D Bolt Pretensioning

In 3D bolt pretension, pretension loads are applied to contacting surfaces.

- The 3D pretension section is a surface inside the bolt that "cuts" it into two parts
- 2. A row of elements would need to be identified that describe the pretension section
- 3. Either the top or bottom surface of this row of elements is designated as the cutting surface

NOTE: Choosing a different pretension section (row of elements) would not change magnitude of the results, just cause a shift the contour



6.2.3 Bolt Pretension Setup

The basic analysis setup including pretension definition usually contains two loadsteps. The first one is special for Bolt Pretension definition (**PRETENSION** entry) and the second one contains both external loads (**LOAD**) and a reference to the pretensioning subcase (**STATSUB(PRETENS)**). Additionally, subcase sequence is activated. This way solver can initially run the pretensioning simulation (1st loadstep), creating a starting condition for the main loading analysis (2nd loadstep). During second subcase pretension is 'locked', because the subcase containing pretension is also referred there. Besides that, main loading is incrementally applied.



Pretensioning can be defined in two ways, by applying:

- Pretension Force (PTFORCE) pretension is defined by a force value and respective ΔL distance is calculated during initial loadstep
- Pretension adjustment (PTADJST) a certain amount of bolt shortening is applied
 ΔL, forces and stresses are resultant from this shortening.

Both **PTFORCE** and **PTADJST** are assigned to a 1D beam in case of 1D Bolt Pretensioning or to contact surface in case of 3D pretensioning, defined exclusively for cut section indication. The basic steps for setting up pretension analysis are as follows:

 Pretension in the bolt is defined through Pretension Manager, which is available in either Utility Browser or in Tools menu.

Note: Initially, if we want to define 1D Bolt Pretensioning, we need to setup 1D Bolt assembly in a fragment of Bolt, which consists of a **CBEAM** element attached by Rigid (**RBE2**) spiders.

- We always start with option Add 1D Bolts or Add 3D Bolts. For 3D Pretensioning, we may define contact surface inside the bolt beforehand, or we can do that through the Pretension Manager (Add 3D Bolts -> Create New Surface).
- 3. We then define Force Type (Force (PTFORCE) or Adjustment (PTADJST)) and enter Load Magnitude in respective units.

	Pretension N	Manager								×
View:	By Bolt		~							
	Bolt Type	Bolt Id	EID/SURFID	Load Type	Loadcol	Load Id	Load Magnitude	Orientation	Out	Add 1D Bolts
	3D	1	0	Force	PRETENS_1(5)	633	30000.0	0	0	Add 3D Bolts Add Ioad Delete Review
<									>	
										0 selected.
							OK	Арр	ly .	Cancel



- By clicking Apply, Load Collector PRETENS_1 is created, which will contain all pretension loads or adjustments we just defined (you will see that in Pretension Manager under Loadcol).
- If we add more pretension loads and want to split them into different collectors. After creation of that loads (again Add 1D/3D Bolt... etc.), we can click on the assigned Load Collector in Loadcol column, extend options and choose Create New – this will create PRETENS_2 collector automatically. This is applicable in case of pretensioning sequence.
- Create a Pretensioning loadstep. This consists basically of Analysis type = Nonlinear Quasi-Static, SPC, NLPARM, PRETENSION (load collector with Pretension loads) definition.
- Create a main loadstep. This usually needs following definition: Analysis type = Nonlinear Quasi-Static, SPC, LOAD, NLPARM, STATSUB(PRETENS) (pretension loadstep created in 6)).

Subcase Definition	
🖃 Analysis type	Non-linear quasi-static
SPC	SPC (2)
LOAD	<unspecified></unspecified>
NLPARM	Niparm (1)
NLPARM(LGDISP)	<unspecified></unspecified>
SUPORT1	<unspecified></unspecified>
DEFORM	<unspecified></unspecified>
PRETENSION	PRETENS_1 (6)
MPC	<unspecified></unspecified>
STATSUB (PRETENS)	<unspecified></unspecified>
NLADAPT	<unspecified></unspecified>
NLOUT	<unspecified></unspecified>
CNTSTB	<unspecified></unspecified>
DLOAD	<unspecified></unspecified>



6.3 Capstone Tutorial Part 2: Creating Pretensioned 3D Bolts in the Bolted Pipe Flange Model

This exercise begins from the previous model, which was set up with frictional contact definitions. Within the next steps, the model will be set up with 3-dimensional bolt pretensioning.



Problem Setup

You should copy this file: bpf_ex6b.fem

Step 1: Import the file bpf_ex6b.fem into HyperMesh Desktop

Step 2: Isolate the Bolt component using Isolate Only and orient the view using the XY Plane view

Step 3: Mask a few rows of elements off of the bottom of the bolt



The isolated bolt, partially masked, and ready for 3D pretensioning

Step 4: Use the Pretension Manager to set up the bolt in 3D pretension

 In the drop-down menu, select Tools > Pretension Manager to launch the Pretension Manager.

4	Pretension N	lanager								×
View:	By Bolt		~							
	Bolt Type	Bolt Id	EID/SURFID	Load Type	Loadcol	Load Id	Load Magnitude	Orientation	Output D	Add 1D Bolts
										Add 3D Bolts
										Add load
										Delete
										Review
<									>	
										0 selected.
								OK	Apply	Cancel

- Click on Add 3D Bolts to begin the pretension setup. Select Create New Surface to make a new pretension surface within the bolt.
- Change the drop-down selector to entities to activate the elems entity selector.
 Select the bottom visible row of the partially masked bolt as the solid elements.
 Select the nodes entity selector and click three nodes on a single element face.



- 4. Click **add** to create the pretension surface from the selected element face.
- Click return to exit the surface creation panel and return to the Pretension Manager dialog box.
- 6. Set the **Load Type** for the newly created pretension surface to Force.
- 7. For Loadcol, select Create New.
- 8. Set Load Magnitude to 30000.

ΔI	Pretension N	/lanager								×
View:	By Bolt		~							
	Bolt Type	Bolt Id	EID/SURFID	Load Type	Loadcol	Load Id	Load Magnitude	Orientation	(Add 1D Bolts
	30	1	PRETENS_1(16)	Force	PRETENS_1(5)	633	30000.0	()		Add 3D Bolts
										Add load Delete
										Review
1										
L.										0 selected.
							OK	Apply		Cancel

- 9. Click **Apply** to create the pretension loading information.
- 10. Click OK to exit the Pretension Manager.



Name	Value
Solver Keyword	SUBCASE
Name	Pretension
ID	1
Include File	[Master Model]
User Comments	Hide In Menu/Export
Subcase Definition	
 Analysis type 	Non-linear quasi-static
SPC	SPCADD (1)
LOAD	<unspecified></unspecified>
NLPARM	nlparm (2)
NLPARM(LGDISP)	<unspecified></unspecified>
SUPORT1	<unspecified></unspecified>
DEFORM	<unspecified></unspecified>
PRETENSION	PRETENS_1 (5)
MPC	<unspecified></unspecified>
STATSUB (PRETENS)	<unspecified></unspecified>
NLADAPT	<unspecified></unspecified>
NLOUT	<unspecified></unspecified>
CNTSTB	<unspecified></unspecified>
DLOAD	<unspecified></unspecified>

Step 5: Create the load step for solving the pretension loading as shown

Step 6: Run the model in OptiStruct and review the pretension results in HyperView



Element Stresses (2D & 3D) (vonMises) & Contact Force (exploded view), Pretension subcase

Step 7: Save this model as bpf_ex7a.hm

7 Analyzing Gaskets

It happens frequently that together with pretensioned bolts, a gasket is present between mounted components, which provides closure. Gasket material presents specific material behavior, which in OptiStruct consists of following properties and modelling structure:

- Anisotropy in the thickness direction. These parameters (thickness direction tensile and transverse shear) are modelled within **MGASK** material card:
 - Elastic-plastic behavior with possibly nonlinear elastic character (BEHAV
 = 0) or nonlinear elastic behavior with damage (BEHAV = 1)
 - Elastic behavior differs depending on previous plastic strain
 - Material stiffness parameters are modelled within Pressure-Closure relations in **TABLES1** entries (closure is a compressive displacement).
- Membrane properties of gasket material are modelled within MAT1 card
- Both MGASK and MAT1 cards are material data referred by PGASK property
- CHEXA elements must be changed into CGASK elements

Note regarding Large Displacement Analysis: Gasket elements can exist in the model, but they will be resolved using small displacement nonlinear theory (equivalent to ANALYSIS = NLSTAT with plasticity). If large translations or rotations are expected on these elements, the results may not be accurate.

7.1 Loading/Unloading Data Setup

Tables (**TABLES1**) are used to provide the unloading/reloading path of the gasket material (pressure vs. closure)







- All the data points in tables **TABLDi** and **TABLUi** are specified in the first quadrant
- Points on loading (TABLDi) and unloading/reloading (TABLUi) paths must be defined in order of increasing pressure and distance

When creating elasto-plastic gasket material:

- The initial yield pressure should match a point in table **TABLD**. If the initial yield pressure is not specified, it will be determined automatically by finding the first point on the TABLD curve where the slope changes by more than 10%.
- The loading path starts from the origin to initial yield pressure (nonlinear elastic range) and continues with strain hardening slope into the plastic region.
- All unloading/reloading paths must start with zero pressure and positive closure distance, and end at the loading path in the plastic region. Subsequent unloading/reloading curves must start with larger closure distances (when pressure is zero) and end with larger closure distances than previous unloading/reloading curves.

When creating elastic with damage gasket material:

- All loading and unloading/reloading curves must start at the origin of the coordinate system (0, 0).
- All unloading/reloading paths must end at the loading path. Subsequent unloading/reloading curves must end with larger closure distances than previous unloading/reloading curves.

Closure Pressure (Y) (X) 0.0 0.0 0.005 20.0 0.05 45.0 0.135 70.0 82.0 0.22 0.287 83.0

Closure (X)	Pressure (Y)
0.08	0.0
0.12	14.0
0.135	70.0

Closure (X)	Pressure (Y)	Closure (X)	Pressure (Y)
0.17	0.0	0.23	0.0
0.2	25.0	0.265	36.0
0.22	82.0	0.287	83.0

Exemplary loading/unloading data:



7.2 Gasket Material & Property Cards

MGASK card contains following parameters:

MID	Material identification number
BEHAV	Behavior type
YPRS	Initial yield pressure for the thickness direction of the gasket material (Applies only to elasto-plastic behavior)
EPL	Tension stabilization coefficient or direct tensile modulus (based on the value of the EPLTYPE field) for the thickness direction of the gasket material
GPL	Transverse shear modulus of the gasket material (the unit of measure depends on the value of the GPLUNIT field)
ALPHA	Coefficient of thermal expansion in the normal direction
EPLTYPE	Type of definition of EPL
GPLUNIT	Type of the unit of measure of GPL
TABLD	Identifier of a TABLES1 table providing loading path of the gasket material (pressure vs. closure)
TABLUi	Identifier of TABLES1 table providing unloading/reloading path of the gasket material (pressure vs. closure)

- BEHAV behavior type can be defined as BEHAV = 0 elastic-plastic material or
 BEHAV = 1 elastic material with damage
- **YPRS** for initial yield pressure see comments **Loading/unloading data setup** chapter.



PGASK card contains following parameters:

PID	Unique gasket element property identification number
MIDG	A MGASK entry identification number that defines thickness-direction and
MIDO	transverse shear behaviors
MID1	A MAT1 entry identification number that defines membrane properties and
WIDI	mass density
CORDM	Material coordinate system identification number
HO	Initial thickness of the gasket (Includes initial open gap and initial void)
GAP	Initial open gap of the gasket
VOID	Initial void of the gasket
LEAKP	Leakage pressure
STABMT	Membrane and transverse shear stabilization flag

CORDM – by default CORDM = -1, which corresponds to element local coordinate system – thickness direction feasible with element normals (see chapter Gasket Elements Setup). When set to 0, basic coordinate system is used.

• STABMT:

- If gasket membrane or transverse shear stiffness is not provided (e.g. zero), then some stabilization is recommended for gaskets in contact
- This is especially useful for cut-out regions or overhang regions
- It can be set to 1 in PGASK card to provide membrane and transverse shear stiffness



7.3 Gasket Elements Setup

In order to have a correctly defined gasket component, user must perform these two steps:

a) Change element type

Gasket parts are modelled with CHEXA, CPENTA or CTETRA elements, however their type must be changed to CGASK8, CGASK6 or CGASK4 formulation respectively. This is done through **elem types** option available in the **2D** panel.



b) Check elements normals

Since gasket material definition is anisotropic and MGASK material card defines its properties in thickness direction, which corresponds to element normals direction, user <u>must validate</u> whether they are correctly defined and if not, update must be done. This can be done by using **normals** option available in the **Tool** panel.



7.4 Capstone Tutorial Part 3: Creating Gaskets for the Bolted Pipe Flange Model

This exercise is the third step in the bolted pipe flange setup. Now that the model has pretensioned bolts, the next step is to set the gasket material and property up. The final HyperMesh model from the previous exercise may be used to begin this exercise setup or the model may be imported from the exercise directory.



Problem Setup

You should copy this file: bpf_ex7a.hm

Step 1: Open the file bpf_ex7a.hm in HyperMesh Desktop

Step 2: Define the gasket loading curve in a new TABLES1 card

- In the drop-down menu, click on View > Browsers > HyperMesh > Utility to bring up the Utility tab.
- Click on the TABLE Create button to bring up the Create Table dialog box. Set the Options: to Create/Edit Table and the Tables: type to TABLES1. Click Next to bring up the table creation dialog box.

Create Table		×				
Options:						
C Import Table						
 Create/Edit Table 						
Tables:						
C TABDMP1	C TABLED3	C TABLEM3				
C TABFAT	C TABLED4	C TABLEM4				
C TABLED1	C TABLEM1	TABLES1				
C TABLED2	C TABLEM2	C TABRND1				
	Next	Cancel				

- 3. Enter LOADCURVE for the name and enter the following data.
- Click Apply to create the table. Close the dialog box to return to HyperMesh Desktop.

Step 3: Create a MGASK gasket material card to set the gasket nonlinear material information including the loading curve

- 1. Create a new material named MGASK of type MGASK.
- 2. Edit the values as shown below:

Create / Edit TABLES1		23
× Y 0 0 .2 2 .5 5 1.35 20 2.2 45 2.87 80	Options: Create New Table Name : LOADCURVE C Edit Existing Table Select:	Ρ
	Apply Can	cel

Name	Value
Solver Keyword	MGASK
Name	MGASK
ID	3
Color	
Include File	[Master Model]
Card Image	MGASK
User Comments	Hide In Menu/Export
BEHAV	
YPRS	5.0
EPL	0.001
GPL	20.0
ALPHA	
EPLTYPE	
GPLUNIT	
TABLD	LOADCURVE (6)
MGASK_TABLU_NUM =	0
MAT4	
MAT5	

This sets the gasket behavior to be elasto-plastic with an initial yield pressure of 5.0, with a tensile modulus of 1e-3 in the shear direction, a transverse shear modulus of 20.0 and the loading curve determined by the TABLES1 created in Step 2.

Step 4: Edit the PSOLID property named PGASK to be a PGASK gasket property definition with the parameters shown below:

Name	Value
Solver Keyword	PGASK
Name	PGASK
ID	3
Color	
Include File	[Master Model]
Card Image	PGASK
Material	MGASK (3)
User Comments	Hide In Menu/Export
MID1	GASKET_MAT1 (2)
CORDM options	BLANK
HO	
GAP	
VOID	
LEAKP	
STABMT	1

This uses the MGASK material entry created as the normal properties for the gasket and the existing PSOLID elastic definition as the membrane properties for the gasket. A stabilization flag is also set to stabilize the solver convergence. This is mostly used for computing overhanging gaskets but may be useful to have set in this model.



Step 5: Assign the PGASK property card to the Gasket component

Step 6: Change the element type for the elements in the Gasket component to CGASK8

- 1. On the **2D** page enter the **elem types** panel.
- 2. Enter the **2D & 3D** subpanel.
- 3. Using the **elems** entity selector on the right-hand side of the panel, select the elements in the Gasket collector.
- 4. Click on the hex8 entry and select CGASK8 as the element type.
- 5. Click **update** to update the element type and click **return** to exit the panel.

Step 7: Review the Gasket element normals

- 1. On the **Tools** page enter the **normals** panel.
- 2. Use the **comps** entity selector to select the Gasket collector.
- 3. Set the toggle selector to color display normals and click **display normals** to show the normals orientation.



For gasket elements it is critical that the normal direction be identical to the thickness direction, as the two sets of gasket material behaviors are completely dependent on the element normal orientation.



Step 8: Run the model as bpf_ex7a.fem in OptiStruct and review the pretension results with gasket behavior in HyperView



The Gasket Thickness-direction Pressure (Pressure) results, showing the Gasket and Pipe_Flange_lower components for the Pretension subcase

Step 9: Save this model as bpf_ex7b.hm

8 Nonlinear Direct Transient Analysis

When the static assumption for a specific event is not valid anymore, Nonlinear Direct Transient Analysis can be used, which additionally includes inertial behavior in the equilibrium equations. Main idea behind this solution is to include dynamic features in the calculation, because they are crucial for the correctness of the results.



Nonlinear Transient is a time-based analysis. Loading is always defined as TLOAD and referred in DLOAD entry, which consists of a time-dependent load path. Although in Nonlinear Quasi-Static Analysis it is also possible to define 'time' dependent TLOAD, inertial features are still neglected there and TLOAD simply represents load incrementing path defined in virtual x-axis, which can be interpreted as, let's say, 'progress' of the analysis (0 = beginning, 1 = end of analysis). In Transient analysis x-



axis is a real, time unit axis, which unit needs to be consistent with the overall unit system used in the model.



Video: Nonlinear Transient Analysis

8.1 Solution Method

- Nonlinear Transient solver uses automatic time stepping.
- Equations formulated in a time basis are solved using Newton's method.
- Two time stepping schemes are available in OptiStruct:
- Generalized Alpha Method used in most cases (default)
- Backward Euler Method.
- Rayleigh Damping is used (subcase dependent)



8.2 Nonlinear Transient Analysis Setup

Nonlinear Transient loadstep setup is similar to Nonlinear Quasi-Static, besides following differences:

- For analysis controls, besides **NLPARM** card, **TSTEP** load collector should be created and referred in the NL Transient subcase.
- Or: instead of NLPARM + TSTEP: TSTEPNL card can be defined individually, which is basically a modification of NLPARM card with additional time step controls.
 - Automatic time stepping is not supported
 - Not recommended

TSTEP card setup:

TSTEP card is devoted to time stepping control in terms of their quantity and sizes. User can define his own time steps in the following way:

- 1. Create a load collector and set the card image to **TSTEP**.
- TSTEP_NUM = entry allows input of how many different stepping definitions are needed (must be Integer > 0), where each so called stepping definition consists of time step size definition and a number of such steps.
- If TSTEP_NUM = 3 for example, following data needs to be in Data,N,... table (exemplary values):

N	DT	NO	W3	W4
10	0.01	5		
4	0.05	2		
2	0.1	1		

That input data mean:

Initially solver performs 10 time steps (N1) of size 0.01 (DT1), where every 5th is saved for output (NO1) results. Next 4 time steps (N2) are done with a size of 0.05 (DT2), every 2nd step is saved for output (NO2). At the end two time steps (N3) of size 0.1 (DT3) are done, which are saved to output (NO3) results.



Parameters marked with orange color (**NO,W3,W4**) need to be defined through card edit, since they are inactive for entity editor setup.

Overall, the event time in this analysis will be:

Event time = $N1 * DT1 + N2 * DT2 + \dots + Ni * DTi$

So, in our case it is **0.5** (seconds).

4. Refer the **TSTEP** load collector in **TSTEP** subcase entry in Nonlinear Transient loadstep.

Since TSTEP goes together with NLPARM card in NLTRAN loadstep definion. Several remarks need to be done regarding the time stepping control (reminder: **NLPARM** also provides increments control, which in case of transient analysis is actually time step control):

- Time stepping schemes, automatic time stepping can be defined using **TSTEP** bulk data card
- DT on NLPARM overrides DT on TSTEP
- **DT** and **TTERM** on **NLPARM**, if specified together, override the value defined on the NINC field
- If **DT** on **NLPARM** is not specified, the default initial load increment is equal to **TTERM/NINC**
- If DT and N are defined on TSTEP card and no TTERM is defined on NLPARM, then TTERM = DT*N
- If TTERM on NLPARM is defined, it overwrites TTERM calculated from DT and NDT on TSTEP
- If **DT, TTERM**, and **NINC** on **NLPARM** and **DT** and **N** on **TSTEP** are not specified, the default initial load increment is equal to 1



8.3 Tutorial: Nonlinear Transient Analysis of a Plate with a Hole

This tutorial is just a brief demonstration on how to setup a simple nonlinear transient analysis. It is aimed to show just a general procedure and the values used there are demonstrative too. User may modify some values (like loading and material plasticity) to check different results and understand the characteristic of transient analysis. You should load the NL_Transient_Plate.hm file.

1. Review the model and notice that basic setup has already been done, user now needs to define all the relevant data for nonlinear transient analysis.



2. Update material properties in existing Steel material – add plasticity.

- View the Steel material in Entity Editor
- Check the box next to **MATS1**.
- Choose **TYPE** as **PLASTIC**
- Set Limit1 to 290



3. Create tabular data for load history

- Create new load collector named TABD1
- Set card image to TABLED1. Leave XAXIS and YAXIS as LINEAR
- In TABLED1_NUM = field, enter 3
- Click the table icon next to Data: x,... and provide values like on the image below

Name	Value			
Solver Keyword	TABLED1	Г	_	
Name	TABD1			TABLED1_NUM
ID	4			x
Color		1		0.0
Include File	[Master Model]	2		0.5
Card Image	TABLED1	3		1.0
User Comments	Do Not Export			
XAXIS	LINEAR			
YAXIS	LINEAR			
TABLED1_NUM =	3			
Data: x,				

4. Create a TLOAD to associate tabular data with an existing load

- Create a new load collector named TLOAD
- Set card image to **TLOAD1**
- For **EXCITEID**, choose existing **Load** load collector
- For **TYPE**, select **LOAD**
- In TID entry, select TABD1, which was created before

Name	Value
Solver Keyword	TLOAD1
Name	TLOAD
ID	5
Color	
Include File	[Master Model]
Card Image	TLOAD1
User Comments	Do Not Export
EXCITEID	Load (1)
DELAY_OPTION	
DELAY	
TYPE	LOAD
TID	TABD1 (4)

- 5. Create a DLOAD load collector to refer TLOAD (DLOAD must be used when multiple TLOADs are available, in this case it is not mandatory and TLOAD can be referred directly in DLOAD subcase entry)
 - Create a new load collector named **DLOAD**
 - Set card image to **DLOAD**
 - The **S** scale should be **1.0**
 - In the **DLOAD_NUM** = field, enter 1
 - Click on the table icon next to **Data: S,...,** and provide following input:

Name	Value		
Solver Keyword	DLOAD		
Name	DLOAD		
ID	6		
Color			
Include File	[Master Model]		
Card Image	DLOAD		
User Comments	Do Not Export		
S	1.0		
DLOAD_NUM =	1		
Data: S,	**		
DLOAD_NUM =			
S		L	
1 1.0		TLOAD (5)	



- 6. Define the time steps for transient analysis within TSTEP load collector
 - Create a new load collector named TSTEP, set the card image to TSTEP
 - Set the **TSTEP_NUM** = field to **1**
 - Click on the table icon next to **Data: N,..** and provide following values:

Name	Valu	e		
Solver Keyword	TST	TSTEP		
Name	TST	EP		
ID	7			
Color				
Include File	[Mas	ter Model]		
Card Image	TST	TSTEP		
User Comments	DoN	Do Not Export		
TSTEP_NUM =	1	1		
Data: N,	1			
TSTEP_NU	JM =			
N	DT	NO	~	W3
1 10	0.1			

- 7. Set up nonlinear analysis parameters within NLPARM load collector
 - Create a new load collector called NLPARAM
 - Set card image to NLPARM
 - Set **NINC** to **10**
 - Set **DT** to **0.0**

Name	Value
Solver Keyword	NLPARM
Name	NLPARAM
ID	3
Color	
Include File	[Master Model]
Card Image	NLPARM
User Comments	Do Not Export
NINC	10
DT	0.0
KSTEP	
MAXITER	
CONV	



8. Request for intermediate results and results in case of non-convergence with NLOUT card

- Create new load collector called NLOUT and set card image to NLOUT
- Set NINT to 10
- Set SVNONCNV to YES

Value
NLOUT
NLOUT
8
[Master Model]
NLOUT
Do Not Export
10
YES

- 9. Create a load step which will contain created cards.
 - Create a load step with **Analysis Type Generic**, name it NLTRANS.
 - In SPC entry refer BCs
 - In **DLOAD** entry refer **DLOAD** collector
 - In TSTEP (TIME) entry refer TSTEP
 - In NLPARM(LGDISP) entry refer NLPARAM collector
 - In TTERM entry type in 0.0
 - In **NLOUT** entry refer **NLOUT** load collector
 - In SUBCASE OPTIONS, check the box next to ANALYSIS and choose TYPE as DTRAN

10. Edit the GLOBAL_OUTPUT_REQUEST card to add plastic strains output.

- View the GLOBAL_OUTPUT_REQUEST card in Entity Editor
- Search for STRAIN output and select EXTRA as PLASTIC in the STRAIN options



V
1
6T 1
PLASTIC
ALL

11. Run the analysis and review the results

Y

Х

- Set the animation mode to transient
- View the contours for stresses and play the animation



9 Tips and Tricks

This section provides quick responses to typical and frequently asked questions regarding OptiStruct nonlinear analysis.

9.1 Essential Steps To Start With Nonlinear FEA

- Learn first how the software works on a simple model before you use a nonlinear feature which you haven't used. Also guess how your structural component will behave, i.e. check for available studies, reports and benchmarks .
- Try to understand the software's supporting documentation, its output and warnings.
- Know what you are looking for. Prepare a list of questions you think your analysis should be able to answer. Design the analysis, including the model, material model, and boundary conditions, in order to answer the questions you have in mind.
- Keep the final model as simple as possible. A linear analysis done first can provide a lot of information such as where are the high stresses in the model, where the initial contact may occur, and what level of load will introduce plasticity in the model. The results of the linear analysis may even point out that there is no need for a nonlinear analysis. Examples of such a situation include the yield limit not being reached, there is no contact, and the displacements are small.
- Verify and validate the results of the nonlinear FEA solution. Verification means that "the model is computed correctly" from the numerical point of view. Wrong discretization with respect to the mesh size and time stepping are common errors. Validation asks the questions if "the correct model" is computed e.g. the geometry, material, boundary conditions, interactions etc coincide with the one acting in reality.
- Try to look into the assumptions made with respect to the structural component, its geometry behavior with respect to large strain (On/Off), look into different



material models if the earlier model is unable to give you a result you expect (sometimes you might look into a possibility of changing the element formulation)

9.2 Debugging

First experience with nonlinear analysis in OptiStruct may lead to frequent divergences during analysis run. This is generally a tricky task to obtain proper analysis controls and you shouldn't get discouraged by that. As already presented in this chapter, OptiStruct can assist analysis control with some really powerful functionalities, or, on the other hand, user may try to apply some generally helpful modifications. Below, there are a few most useful tips that may help you succeed with any diverging analysis.

• Review carefully your setup

First thing that should be done when divergence has occured is to review the model setup step by step. View each collector in Entity Editor and verify if: The units are correct

Loadstep is fully defined (all appropriate load collectors are referred) Tabular data is correctly defined, if you are not sure, refer to Help Documentation Material is sufficiently defined – maybe stresses during analysis exceed the values defined material data, like plasticity curve

• Consider using Large displacement analysis and follower forces

User needs to be especially careful about potential geometric nonlinearities. If such should occur, but **LGDISP** is not active, divergence is likely to occur due to that (or at least invalid results).

• Decreasing increment size

Large nonlinearities in the model usually bring quite a singificant divergence during first iterations within one load increment. Since the first iteration presents a linear solution and the consecutive iterations approach a real, nonlinear solution, large



increments may sometimes require a great number of iterations to converge or may never achieve convergence. Decreasing load increment makes the first assumption (iteration) already closer to the correct solution and therefore the chances, that the simulation run will converge, increase.

Increment size is controlled by the **NINC** entry in **NLPARM** card or user can also specify **DTMAX** in **NLADAPT** card to limit the load increment size to be not greater than **DTMAX** value.

• Activate analysis expert system (only small displacement NL analysis)

EXPERTNL (set to **YES** in **PARAM** control card) is a very powerful OptiStruct functionality that controls analysis parameters in a very smart way. The tasks that are performed by analysis expert involve: performing additional iterations, under-relaxation, automatic adjustment of the load increment, backing off to the last convergent solution and retrying.

• Use frictional contact only when it's really crucial

Presence of friction in an analysis with contacts lead to very complicated computing algorithms and requires use of unsymmetrical solver (**MUMPS**). Sometimes the solution for diverging analysis can be to assume a STICK (in case sliding is minimal) or SLIDE contact.

• Rigid Body mode? Try applying Contact Stabilization

Since contact is a kind of boundary condition, sometimes it may constrain a model insufficiently and rigid body mode conditions may occur (free movement of a part). In that case **PARAM,EXPERTNL,CNTSTB** may help solve the problem of rigid body mode

• See what happens with the model just before divergence

Although the simulation run may diverge, user has the possibility to see what happens with the model just before divergence, i.e. we can view the results just for converged load increments. To write h3d file containing non-convergent solution:

1. Create a load collector with card image NLOUT


2. Activate SVNONCNV,YES

3. Refer **NLOUT** load collector in load step **NLOUT** entry.

Sometimes seeing the behavior just before divergence may give helpful information regarding, e.g. proper unit system used for plasticity data, time-dependent loading, lack of follower force etc...

• Nothing helps? Try Nonlinear Transient Analysis

In some cases assumption of static behavior of a system maybe a little bit too far and dynamic effects need to be considered, in that case transient analysis may help.

9.3 Useful Tutorials and Industry Examples

The layout of online help has been changed from HyperWorks Solvers help 2017.2.2 **2018.0 OptiStruct Solver Online Help**





List of useful tutorials for Nonlinear Analysis

Advanced Small Displacement Finite Element Analysis OS-T: 1320 Nonlinear Gap Analysis of an Airplane Wing Rib OS-T: 1360 NLSTAT Analysis of Gasket Materials in Contact OS-T: 1365 NLSTAT Analysis of Solid Blocks in Contact OS-T: 1390 Pretentioned Bolt Analysis of an IC Engine Cylinder Head, Gasket and Engine Block System OS-T: 1392 Node-to-Surface vs Surface-to-Surface Contact OS-T: 1393 Basics of Contact Properties and Debugging

Large Displacement Finite Element Analysis

OS-T: 1510 Follower Loads, Nonlinear Adaptive Criteria, and Nonlinear Intermediate Results OS-T: 1520 Finite Sliding of Rack and Pinion Gear Model

List of useful examples for Nonlinear Analysis

Nonlinear Small Displacement Quasi-static Analysis

OS-E: 0010 Wheel Rail Interface using Fast Contact

OS-E: 0015 Pretension Analysis using Gasket Material

OS-E: 0020 Inline-4 Engine Head

Nonlinear Large Displacement Quasi-static Analysis

OS-E: 0100 Nonlinear Analysis of Cantilever Beam OS-E: 0105 Nonlinear Analysis of Cantilever Beams with Follower Forces OS-E: 0110 Elastic-Plastic Large Displacement Analysis OS-E: 0115 Finite Sliding Contact OS-E: 0120 Nonlinear Static Analysis with Hyperelastic Material Model, under Compression Loading and Unloading



- OS-E: 0125 Diagnose Non-converged Solution using NLMON
- OS-E: 0130 Plate with Central Hole using NLOUT
- OS-E: 0135 Contacts and Pretensioning of Bolts
- OS-E: 0140 Pretension Analysis of Piston
- OS-E: 0145 Nonlinear Analysis of Rack and Pinion using Restart
- OS-E: 0150 Large Displacement Analysis of Chassis
- OS-E: 0155 Snap-Fit Analysis (CONSLI vs FINITE)
- OS-E: 0160 Nonlinear Static Analysis of Snap Fit Assembly Using Continuous Sliding
- OS-E: 0165 Contact Smoothing With Two Concentric Rings
- OS-E: 0170 Revolute Joint Using JOINTG and MOTNJG
- OS-E: 0175 Layered Solid-Shell Composites Using PCOMPLS
- OS-E: 0180 3-Point Bending using RBODY
- OS-E: 0185 Rubber Ring: Crush and Slide Using Self-Contact

10 Appendix

Contact Modeling in Structural Simulation – Approaches, Problems and Chances The following article was kindly provided by Professor R. Steinbuch, Department of Engineering, Reutlingen University

Abstract

Nearly all structural problems deal to some extend with interacting bodies [1]. In consequence, contact between these bodies is one of the most often occurring problems in modelling. On the other hand, many contacts are either neglected in modelling or assumed to be represented by appropriate boundary conditions. Today most of the simulation codes used in structural mechanics allow to model contact. In consequence there is a need to qualify more users to apply contact models in everyday simulation practice. Modelling contact implies to consider the appropriate setting of the available contact parameters. Using the standard values proposed by the codes or unqualified selection of parameter combinations may produce simulation results which are either poor or even misleading. To stimulate the discussion about contact modelling some of the aspects to be considered and their nonunique consequences are discussed.

Choosing qualified element types surely will influence significantly the simulation velocity and quality. Adapting meshes to the current state of the problem may improve speed and quality essentially, but mapping the analysis results from old meshes to the new ones may cause loss of local quality. Reduced integration helps to speed up the computing time and avoid numerical stiffness, but the stress results are less qualified. Artificial stiffness introduced to prevent misleading effects like hourglasing produces potential sources of numerical errors. The ways to detect and release contact in time and space influence significantly the course of the analysis. Simple approaches yield fast responses at the price of incorrect penetrations and incomplete definitions. Qualified algorithms may converge only after large numbers of recycles and time step reductions, so acceptable response times may not occur.



Friction models suffer from the uncertainty of data and the large scatter of their input values, especially in transitions of stick to slip, high speed sliding and lubrication. The decision whether to use explicit or implicit time integration is not easy to be made in many cases.

A qualified discussion of the interaction of the parameters in contact analysis will certainly not yield a simple scheme applicable in all cases required to be considered in structural simulation. Users will be faced with problems where the ideas they have been applying successfully in many studies fail, do not converge in appropriate time or even come up with misleading results without indicating the potential danger. So the most important parameter in engineering contact analysis is the experience of the people running the jobs. Some application examples underline the problems mentioned and sketch solution proposals for some parameter configurations. Critical regions are defined, workarounds proposed and experiences presented. Small parameter changes may cause large changes in system response. In consequence for contact problems robustness studies, indicating the sensitivity of the physical and numerical parameters should be performed. In any case the qualification and care of the analyst applying contact methods in engineering remain the most important tools to solve contact problems successfully and to learn about the interaction of structures.

Contact In Structural Problems

Nearly all coherent structures like metal parts, stones, plants or furniture, to mention some, interact with their surrounding via surface contact. A table is standing on the floor, the ends of its legs contacting the surface of the floor of the room. The only exceptions of bodies not in structural contact with some neighbours are structural units flying through the air or the empty space like meteorites. So contact and the necessity to handle contact is not an exotic task, but one of the basic questions in structural mechanics. But as contact problems are often related to nonlinear phenomena many engineering approaches try to avoid contact modelling by the use of linear surrogates.



• Boundary Conditions Or Contact?

In mechanics the idea of free body models, replacing all interaction with the contacting surrounding by application of corresponding forces or moments is crucial to start the studies of static problems. Introducing contact models at that early state of education would increase the confusion in the novices' heads, who are struggling hard enough to enter the world of mechanics. So models like the one shown in Figure 1a are supposed to represent real support loading situations indicated in Figure 1b. Effects like local notches or sharp corners between the interacting bodies are ignored and believed to be of none or little influence. But when dealing with real components and their failure modes, we realise, that a good part of the problems are caused by these local disturbances we neglected before.







(a) beam with boundary condition

(b) build in beam under real loading

Fig. 1 Idealised and real loading situations of beam

• Contact As Part Of Manufacturing Processes

Contact, being present as a static boundary condition in most structural systems, is additionally defining some of the processes in mechanics. All the shaping of structural but deformable material including unwanted forming like crash is specified by some contact history. Segments of parts surfaces come into contact, the acting forces cause deformation of the surrounding material [2]. New contacts are possible due to the fact that material preventing the contact has been removed by the contact process.



Fig. 2.a demonstrates some steps in an axisymmetric deep drawing process, using different tools and finally covering the newly manufactured can by a seal.



During two steps of forming, the shape of the top of the can is manufactured (subfig. 1 and 2). Then the tampering head is introduced (subfig. 3). Finally, some thermal load steps caused by temperature changes of the whole system are applied (subfig. 4). The analysis aimed to check if the sealing by the tampering head would be sufficient during the lifetime of the assembly. The contact study performed checks whether the manufacturing is possible, if problems may occur, e.g. large strains may indicate the failure of the sheet metal and other related quantities like the spring-back after the removal of the die. In addition, including thedeep drawing into the system's analysis helps to come up with a better evaluation of the structure. Figure 2b plots the sealing forces between the can and the tamping due to temperature changes. Obviously, the study not including the deep drawing steps (w/o manufacturing) predicts a failure of the seal during the thermal loading (radial pressure = 0) while the more elaborate simulation (with manufact) predicts a sufficient tightness (radial pressure > 0).



Fig. 2 Some steps of a deep drawing of the top of a can (axisymmetric) and sealing

• Numerical Contact Studies

If two bodies come into contact there is a small region of first encounter, which will be increasing, if the bodies are continuing to move in the closing direction. Perhaps some of the early contact regions are released from contacting, while others support the load to be transmitted via the contact areas. A continuous process of establishing and releasing contacts will evolve. When studying more complex problems of continuum mechanics it is well known that there are no analytical or classical solutions available as soon as the problems exceed a certain level of simplicity. Discrete numerical strategies like FEM or related approaches are used to analyse these nontrivial problems. But these discrete systems have the disadvantage that their discrete schemes including discrete discovery and release of contact will fail to reproduce the soft initiation of continuous contact. Figure 3 compares the smooth rolling of a circular wheel with the inevitable rumbling of a discrete representation of this wheel.



Fig. 3 Rolling of a cylinder and a FE-model of the cylinder on an inclined plane

While Figure 4a depicts a pipe bending study, Figure 4b presents the local contact forces during this pipe bending study at a given time step. There are sharp changes in the contact state and forces at the Finite Element nodes, which is quite the opposite of the real smooth process in time.



(a) Deformation of pipe during bending (b) Local contact forces on pipe Fig. 4 Pipe bending study



Terms And Definitions

Having mentioned some of the basic terms of contact modelling the need to clarify terms used in contact studies arises.

Real Contact

The following short definitions for contact between real parts hold and are generally accepted (see Figure 5a):



Fig. 5 Main contact parameters

Contact: Two bodies meet, are in structural interaction, the contact normal force is positive.

Release: The contact ends, no more structural interaction, the contact normal force is zero.

Contact force: Force normal to the contact area acting between the contacting bodies (CF).

Contact pressure: Contact force/contact area (p).

Friction: Force transverse to contact closing direction, opposing the sliding tendency (FR).

Stick: No relative transverse movement, often due to friction.

Slip: Sliding along contact surface overpowering the sticking friction forces.



Numerical Contact

The same terms used for real contact may be used for the discrete numerical modelling of contact. We are taking into account that no longer areas are interacting, but discrete nodes or points are checked, if they are penetrating opposite surfaces. All contact properties, especially normal and friction forces, stick or slip responses have to be transferred to the discrete nodes like indicated in Figure 5b.

Parameters Used In Contact Studies

Going deeper into the problem of modelling, the process so simple to describe experimentally exhibits a large number of parameters to be covered. Table 1 tries to identify some of them, including the most commonly ways used to handle them.

Parameter	Values			
elements adaptivity element integration deformation	linear refine reduced elasto-plastic	quadratic coarsen full rigid-plastic	higher order remesh 	
contact type contact partners contact modelling shell contact corner handling catching release	master-slave flexible vs. stiff penalty include thickness one side only range immediately	both partners both flexible Lagrange multipliers neglect thickness slip around after penetration slowly	contact range merging 1 or 2 sided	
friction friction definition friction onset	Coulomb velocity stick-slip	shear displacement continuous	glue 	
time integration time modelling	explicit static	implicit dynamic	- large masses	

Table.1 Some of the major contact modelling parameters

Neither the completeness of the parameters list nor the proposed handling must be accepted by other contact modelling engineers or systems. Nevertheless Table 1 gives some idea that contact modelling is not a simple task. Furthermore, it tries to motivate potential users not to trust their codes without counterchecking the default values proposed by the different simulation systems. The users should analyze their problems, compare the specific requirements with the possibilities the codes offer



and search for reasonable solutions. There is no commonly accepted agreement about the preferences to be made, the parameter combinations to be used or unwanted interactions of parameter choices to be avoided. In many cases extensive studies have to be performed before a reliable contact analysis may converge to a satisfactory result.

Two examples may enlighten this warning. Linear elements are simple to be used, but require small contact edges, while quadratic elements may yield faster results using essentially smaller node and element numbers. On the other hand, the local forces of quadratic elements are different at the corner and mid-side nodes. Larger local forces occur at the midside nodes than at the edge nodes. As a consequence, the forces transmitted locally between contacting structures may not be correctly placed. Explicit time integration is known to produce fast results at relatively small time and storage requirement, while implicit time integration needs large storage and lots of time for the matrix inversion.

Some Problems and Some Proposals

Some of the contact parameters and related problems should be looked at, to come up with an understanding of the necessity of sensible handling contact simulation.

• Element Types and Contact Definition

Contact is (see Figure 5) the interaction between two bodies. The result of the modelled contact should not depend too much on the modelling approach. Figure 6 however indicates that local effects may vary significantly due to the element size and element type used. Only for very fine meshes, the desired symmetric and element type independent result occurs. This leads to the proposal to use very fine meshes or sensitive adaptive algorithms



• Contact State

The definition of the contact closing and the removing of penetrations may be done by many strategies, Lagrange Multipliers and Penalty Methods being the most popular ones. Some codes propose locally conforming meshes, merging the respective nodes or degrees of freedom. Others apply forces instead of displacement constraints to remove overlapping of the contacting parts.

Even the definition when contact happens is not unique, master-slave and symmetric checking strategies show acceptable results, but sometimes yield different contact response.



Fig. 6 Contact convergence, grey tones indicate contact normal stresses



• Friction

Friction, the force acting normal to the contact closure direction and parallel to the contacting surfaces is one of the most difficultand little understood phenomena in contact studies. In the microscopic scale local structures interact, build bridges, hooks, joints, are removed by sliding of the surfaces and rebuild at a new position.

Molinari [3] presents impressive studies of local interaction yielding local forces and the resulting stresses. Simulation codes generally use coarse simplification of the friction effects by introducing friction coefficients, sometimes dependent on the local pressure and sliding velocity. Nevertheless evenmore qualified ideas are hardly able to represent the friction of real surfaces. Figure 7 demonstrates for two simple 2Dsurfaces, how the local friction depends on the meshing parameters, even the refinement of the meshes produces no tendency of convergence



Fig. 7 Friction as function of element type and size, outside arrows indicate transverse displacement, inner arrows indicate local friction force.

• Time History Integration



Contact is the result of the motion of different bodies, which initially had some distance or little common surface. During the contact process the area of interaction, the contact surfaces vary as well as the local and global forces. Regions being initially free are detected by free surfaces of the other contact partners, contacting regions are released or at least carry a reduced load. There are some promising approaches to use the final relative position of the bodies for the analysis [4, 5]. In most cases however it is a good idea to start with free bodies. Then one defines a motion, bringing into contact the bodies like indicated in Figure 8. From the initial independent state contact surfaces are evolving, showing a time dependent local response, again different for linear and quadratic elements.



Fig. 8 Time dependent contact surfaces and local forces

• Time Integration Schemes

There are many discussions about the procedures of time integration. Since the "early days" scientists discuss whether to use explicit or implicit time integration schemes. From the classical point of view there is no doubt, that the implicit solution

$$K_n u_n = F_n \tag{1}$$

or its dynamic version

$$\mathbf{M}_n \ddot{\mathbf{u}}_n + \mathbf{C}_n \mathbf{u}_n + \mathbf{K}_n \mathbf{u}_n = \mathbf{F}_n \tag{2}$$

where the index n indicates the new time step, should be preferred, as it implies the equilibrium of forces at the new time step. In consequence relatively large time steps may still yield sufficient quality of the numerical prediction of the time history. Unfortunately for both static and dynamic problems the need to solve large linear equation systems requires large computation time and storage requirements. Especially in the case of metal forming or crash one needs to recycle the analysis at every time step until satisfactory convergence in terms of equilibrium of internal and external forces in equations (1) or (2) is achieved. This makes implicit time integration an expensive, in many cases an inacceptable tool. So the upcoming of explicit integration schemes marked an essential step towards the feasibility of large contact studies. Assuming that in small time increments, things will not change essentially, the equation to be solved is

$$K_a\Delta u = \Delta F$$

respectively,

$$\mathbf{M}_{o} + \ddot{\mathbf{u}}_{o} + \mathbf{C}_{o}\dot{\mathbf{u}}_{o} + \mathbf{K}_{o}\mathbf{u}_{o} = \mathbf{F}_{o}$$
(4)

the index o now indicating the old stiffness, mass and damping matrix, displacement and force. Substituting the finite difference approximations for the time derivates and using diagonal mass and damping matrices, equation (4) yields

(3)

 u_n denoting the new displacement again, u_{o-1} standing for the past time step's displacement.

$$\mathbf{M}_{o} \frac{\mathbf{u}_{a} - 2\mathbf{u}_{o} - \mathbf{o}_{o-1}}{\Delta t^{2}} + \mathbf{C}_{o} \frac{\mathbf{u}_{a} - \mathbf{u}_{o-1}}{2\Delta t} + \mathbf{K}_{o} \mathbf{u}_{o} = \mathbf{F}_{o}$$
(5)

From equation (5) we find the new displacement un. Using equations (4) and (5) has the big advantage that the stiffness matrix has never to be assembled or inverted. So the CPU-time and storage requirements to perform one time step are reduced by large factors. On the other hand the time steps have to be reduced essentially to avoid to large an accumulation of errors due to the fact that the equilibrium at the old time



step is used to predict the new displacements. The numerical limit to the time step enforces even smaller time increments.

To give some ideas about the impact of the time integration scheme we remember the pipe bending example demonstrated in Figure 4. The CPU-time and the storage requirements are compared in Table 2. Obviously, the explicit code has large advantages. But on the other hand, the fact that explicit codes always yield results is not without drawbacks. When implicit codes show little stability or fail to converge, this may be indicate that physical problems arise. So, the deficit in numerical performance may be an advantage, motivating the qualified analyst to improve his ideas, to check assumptions, to go deeper into the details of the problem posed. Table 2 Comparison of implicit and explicit resources used for pipe bending

Integration	Processors	RAM [GByte]	CPU-Time [h]	Rel. velocity
implicit	16	24	8	1
explicit [9]	2	2	2	32

Reliability of Codes

One of the most dangerous obstacles in simulation of contact is the reliability of the commercial or private codes. Nonlinearity and contact are included in most today modelling systems, even if sometimes doubtful contact options are offered to the user. Unfortunately, many of the options and parameters are neither clearly understood nor defined. In addition, the difficulty of contact modelling makes the software provider proposing many different parameter schemes. Even high-level quality assurance does not inhibit erroneous codes as indicated in Figure 9 [4]. The thermal contact between two bodies is modelled with conforming and nonconforming meshes using linear (HEX8) and quadratic (HEX20) elements. Obviously, the contact definition of the higher order elements is wrong, at least unsatisfactory. This is even more surprising, as the HEX20-element of this code shows very good performance when used in structural contact problems as shown in Figure 4. So, any contact used in commercial codes needs to be checked at some benchmark problems before using it in engineering or scientific application.



Fig. 9 Failing contact quality using HEX20 [11]

• Scatter and Robustness

The increasing computing power improved the simulation of parts from single jobs of few variants to the study of the response of large sets of models. This may be necessary when we need to vary some or many of the parts parameters. A typical application is the modelling of the scatter of manufactured components in large series production. The term of robustness entered the world of virtual development opening new sights, new possibilities and new problems to the simulation teams. At the onset of robustness studies due to the large number of parameter combinations it was considered to be available only for few elaborated problems. The introduction of statistically based approaches like the Latin Hypercube and the Response Surface allowed accelerations of the studies. So today we may handle many problems with scattering parameters in acceptable and affordable time. Nevertheless, robustness studies, especially in the nonlinear regime need essentially more time and computing power. This is due to the fact that many variants have to be analyzed over a certain time of the contact process.



• Scatter in Contact Studies

Table 1 offered a selection of parameters and solution strategies used in contact modelling. The selected values of the parameters and their values influence the result of the contact analysis. In consequence robustness or sensitivity studies seem inevitable. Table 3 lists some of the most dangerous and scattering components of the physical contact and its numerical modelling. Once more neither completeness nor uniqueness is assumed. Other analysts with other experiences may add or remove entries to or from Table 3. In any case care should be taken that the range of possible states is covered by the range of numerical parameters.

Table 3 Most sensitive and influencing parameters in contact modelling

Physical data	Numerical data		
 friction surface quality surface definition 	 gap, catch range mesh quality element definition 		
- loads	- initial contact		
 supports flexibility 	 penetration local stiffness friction model 		

• Parameter Sensitivity

Figures 6 to 9 indicate some of the possible uncertainties encountered using specific contact models. The range of input definitions is hardly understood by the majority of the engineers working with the different numerical codes. Figure 10 gives an idea how small input changes result in large differences in the results predicted by the simulation. As a consequence, automated or semiautomated systems should provide the possibility to do a sensitivity study to understand whether physical or numerical effects influence the results of the nonlinear analysis.



Fig. 10 Scatter and robustness

Some typical steps may include:

- Check which input parameter changes cause significant changes in results.
- Cover possible parameter range.
- Find critical parameter combinations.
- Check reliability of parameters used.
- Compare results with experimental data, if available and reliable.

The outcome of these studies could result in an increased critical interpretation of the numerical results. This is always a good idea in nonlinear studies, especially in large and complex contact problems like crash and metal forming.

Conclusions

- As a resume, some statements may help to improve the awareness of problems, obstacles and traps in contact simulation.
- Contact studies are possible with commercial and private simulation codes and without deep understanding of the underlying principles. Regarding the questions mentioned in this article, we doubt that this is always positive. Sometimes there should be some fences preventing inexperienced users to enter fields where their simulation is in great danger of being at least not very meaningful. The definition of these fences requires large experience.
- As many parameters for contact modelling are available in commercial simulation systems, many aspects have to be taken into account, many of the sometimes-

contradicting possibilities checked, interactions looked at, inconsistencies excluded.

- Often the meaning of parameters is not properly defined. Therefore, a reproducible strategy of benchmarking and quality assurance needs to be established, improved and updated
- The tendency of using the default values provided by the code includes the danger to run totally inappropriate studies. All default values have to be checked, questionable entries removed, large warning signs posted to avoid following nonsense roads.
- The large number of physical and numerical parameters in contact studies makes a consequent system of robustness and sensitivity studies inevitable. It should be included in the problem-solving rules and never be neglected due to lack of time or money.
- The interpretation of nonlinear simulations in most cases is a difficult task. So experience, comparison with experiments, preceding studies, comparable results have to be included in the evaluation of simulation results. This holds not only but even more thoroughly when dealing with contact problems.
- Qualified users should not and cannot avoid doing contact studies. They should always be aware of the many possibilities of failing by ignoring or misinterpreting physical or numerical aspects.

References

[1] R. Steinbuch. Remarks on the engineering treatment of nonlinear simulation problems. Mathtool03, St. Petersburg, Russia, June 2003.

[2] G. Kammler, G. Mauch, and R. Steinbuch. Automatische Neuvernetzung stark deformierter Strukturen. Berichtsband des MARC-Benutzertreffens 1992, M⁻⁻unchen, 1992 [in German].

[3] J.F. Molinari. Multiscale modelling of nanotribology: Challenges and opportunities. Presented at ICCCM09, Lecce, 2009.



[4] G. Zavarise and L. De Lorenzis. A strategy for contact problems with large initial penetrations. Presented at ICCCM09, Lecce, 2009.

[5]

http://resources.altair.com/hyperworks/pdfs/product_brochures/HW13_HyperFor m.pdf

[6] Private communication with Michael Lautsch, Lautsch Finite Elemente, Esslingen, Germany.