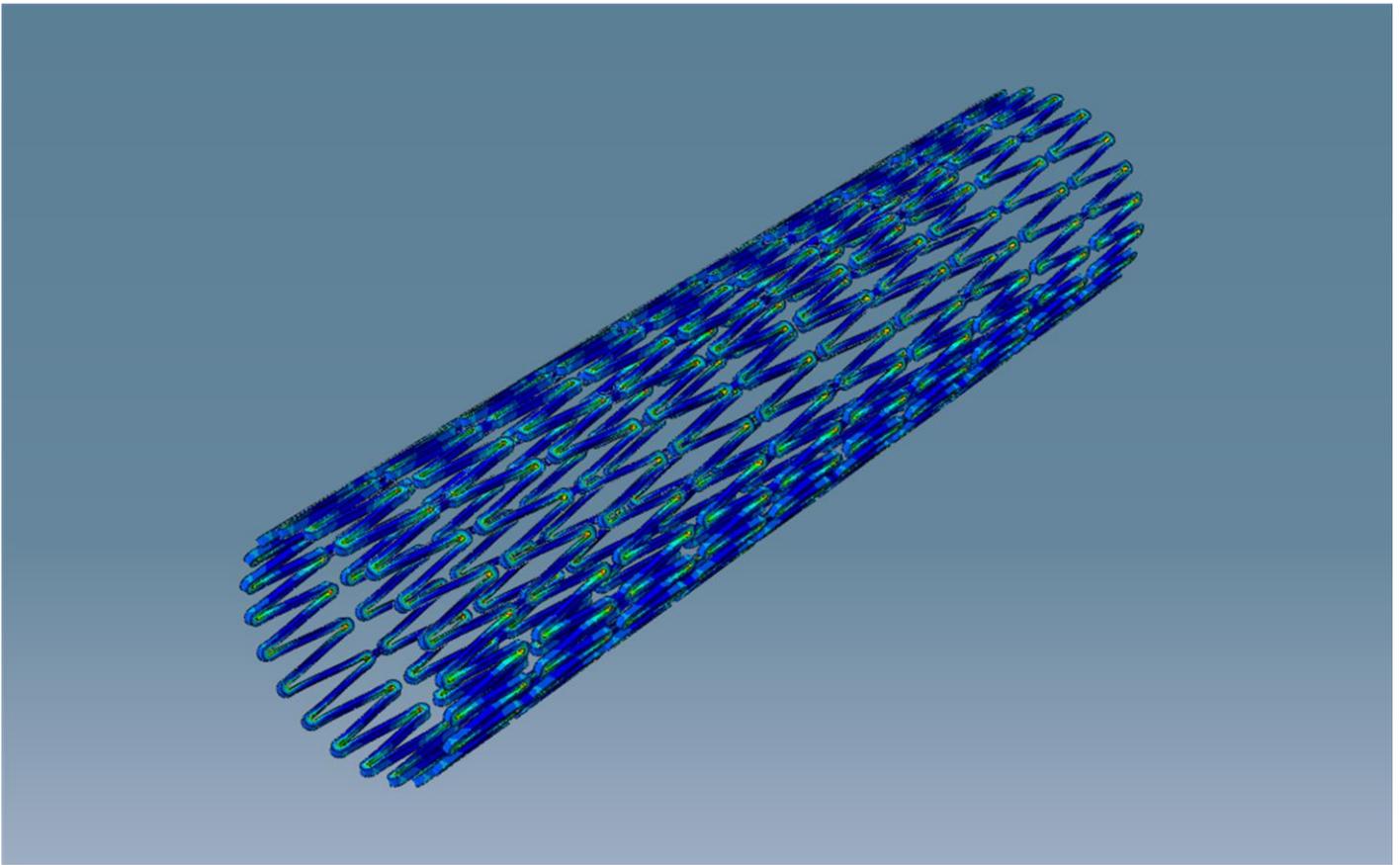




MODELING STENTS

SIMULIA BEST PRACTICES



Confidential Information. © [2016] Dassault Systèmes. All Rights reserved.



3DEXPERIENCE

Version 1.0 - 2/7/2018

Written by: Shashwat Sinha
Validated by: Changjian Wang
Edited by: Arati Desai

SIMULIA Abaqus 2018

Executive Summary

This Best Practices document describes major aspects of stent modeling and simulation, familiarizing the reader with its core concepts and best practices. To take full advantage of this Best Practice, you must be familiar with the SIMULIA product Abaqus.

Stents are medical implants for human bodies and represent one of the most notable areas where bio-medical industry has advanced rapidly. This advancement has been possible in part due to the foray of simulation in this area which has largely complemented or replaced the traditional method of developing stents by prototyping and physical evaluation. Simulation through Finite Element Modeling (FEM) in a computer-simulated virtual environment has allowed the exploration of various stent designs and their failure modes in a very cost-effective and timely manner. Simulation has thus become an integral part in the design cycle of stents, helping the bio-medical industry to attain quicker turn-around times and a faster time to market.

The success of simulation in analyzing stents is demonstrated by the fact that simulation is a required procedure to get regulatory approval for commercialization of the device. Abaqus simulation is being used to predict stress-strain and fatigue response of stents in in-vivo conditions. In light of the importance of stents in the medical industry, it is imperative to tackle modeling of stents in some detail, which is the aim of this document.

The target audience for this document is the Users, Tech Support, Management, and Sales team.

Contents

1. Terminology	6
2. Introduction.....	7
2.1. Types of Stents.....	8
2.1.1. Balloon-expanded Stents.....	8
2.1.2. Self-expanding Stents	8
2.2. Materials Used in Stents	9
2.3. Fabrication Process	9
2.4. Documentation Reference.....	10
3. Geometric Modeling and Meshing	10
3.1. Meshing	11
3.1.1. Abaqus	11
3.1.2. 3DEXPERIENCE	13
3.2. Mesh Refinement	15
3.2.1. Abaqus	15
3.2.2. 3DEXPERIENCE	16
3.3. Mesh Convergence	16
3.3.1. Hex Elements	17
3.3.2. Accuracy	17
3.3.3. Tet Elements	20
3.3.4. Run Times	21
3.4. Documentation Reference.....	21
4. Materials.....	22
4.1. Abaqus 2016 Keyword Interface.....	24
4.1.1. UMAT	24
4.1.2. VUMAT	24

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

4.2.	Abaqus 2016 Output.....	25
4.3.	Abaqus 2017 Keyword Interface.....	25
4.4.	Abaqus 2017 Output.....	26
4.5.	Convergence	26
4.6.	Definition of Nitinol in 3DEXPERIENCE	27
4.7.	Documentation Reference.....	27
5.	Analysis and Steps.....	28
5.1.	Documentation Reference.....	29
6.	Boundary Conditions and Loads.....	30
6.1.	Circumferential	30
6.2.	Axial	30
6.3.	Radial	31
6.3.1.	With a Rigid Cylinder.....	31
6.3.2.	With a Deformable Cylinder	31
6.3.3.	With Contact Interference	31
6.4.	Documentation Reference.....	32
7.	Contact.....	33
7.1.	Abaqus/Standard	33
7.2.	Abaqus/Explicit	33
7.3.	Contact Constraint Enforcement Methods.....	34
7.4.	Contact Convergence.....	35
7.5.	Documentation References.....	36
8.	Analysis Techniques	37
8.1.	Analysis Procedures	37
8.2.	Import Analysis	37
8.3.	Restart Analysis.....	37
8.4.	Annealing.....	38
8.5.	Documentation References	38
9.	Fatigue Simulation.....	39
9.1.	Documentation References	39
10.	Post-Processing.....	40

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

Best Practice

10.1.	Fatigue	40
10.1.1.	Scalar Method	41
10.1.2.	Modified Tensor Method	41
10.1.3.	Fatigue Safety Factor	42
10.1.4.	Using fe-safe	44
10.1.5.	Fatigue Life for Nitinol.....	45
10.2.	Documentation Reference	45
11.	Reports.....	45
12.	Appendix 1	46
12.1.	Model Details	46
12.2.	Hardware.....	48
12.3.	Software.....	48
13.	Appendix 2.....	48
13.1.	Writing Abaqus Input File from 3DEXPERIENCE Platform	48
13.2.	Running Abaqus Job on 3DEXPERIENCE Platform.....	49
14.	References	51
15.	Document History	52

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

1. Terminology

The terminology used in this document is well accepted among experienced users in the field of stent simulation. For the benefit of new users, some basic common terms are briefly explained below:

- ▶ **FEA:** Finite Element Analysis; a numerical solution technique to capture real-life physical behavior and responses.
- ▶ **Element:** Building block of a mesh used in FEA; a geometric topological shape with defined stiffness and physical behavior.
- ▶ **Mesh:** Collection of elements arranged in a specific fashion to approximate geometric structures and shapes, and provide physical behavior.
- ▶ **Hex:** Hexahedral shape
- ▶ **Tet:** Tetrahedral shape
- ▶ **3D:** Three-dimensional
- ▶ **2D:** Two-dimensional
- ▶ **GUI:** Graphical User Interface
- ▶ **Abaqus/CAE:** GUI for Abaqus
- ▶ **Abaqus/Standard:** Implicit solver of Abaqus
- ▶ **Abaqus/Explicit:** Explicit solver of Abaqus
- ▶ **3DEXPERIENCE:** 3DEXPERIENCE Platform (has both Abaqus solvers embedded)
- ▶ **App:** Application on 3DEXPERIENCE platform dedicated to perform a collection of similar functions/activities; the 3DEXPERIENCE platform hosts many apps with different broad functionalities for the different Brands of Dassault Systèmes.
- ▶ **Plugin:** Custom piece of software augmenting some main software (like Abaqus) with some functionality that does not exist in the main software.
- ▶ **KB:** Knowledge Base; which is a searchable online repository of technical articles supplementing the documentation of the products from Dassault Systèmes.
- ▶ **Documentation Reference**
 - ▷ **Abaqus 2016:** Refers to the installed documentation on user's machine
 - ▷ **Abaqus 2017:** Refers to the documentation available on help.3ds.com

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

2. Introduction

Stents are tiny cylindrical metal mesh tubes that are placed into blood vessels, arteries, or other ducts to hold the structure open. They are often deployed to counteract the effects of vascular diseases, such as coronary artery disease, peripheral artery disease, renovascular hypertension, etc. To deploy the stent, it is usually crimped, placed in a delivery system (e.g., catheter), and finally expanded in-vivo which widens the duct. The below Fig 2.1 shows the basic working of a stent.

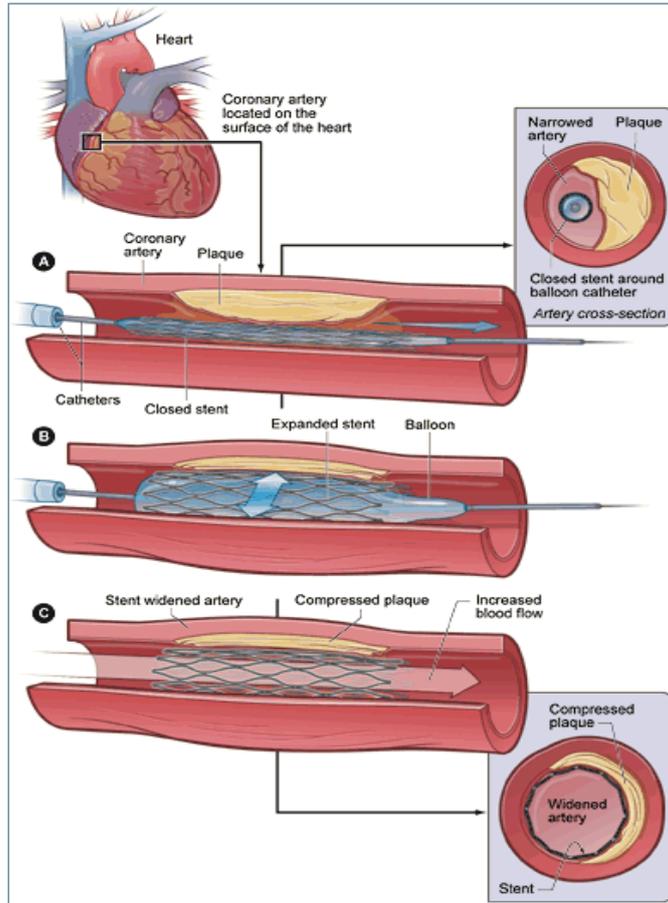


Fig 2.1: Stent Application (Courtesy: nih.gov)

Manufacturing of stents is a growing business driven by human medical needs. Bio-medical companies involved in stent manufacturing are constantly coming up with new and improved designs of stents for different types of diseases which restrict flow in the human body's ducts. It is vital that the design of stents be thoroughly validated by FEA because of their critical function. Simulation of the manufacturing process and fatigue loading conditions for stents can reveal detailed stress-strain distributions that are crucial in predicting fatigue life. Various in-vivo conditions that are hard to reproduce in a lab can be analyzed through realistic simulation and design changes can be made. Simulation avoids costly prototyping and the number of designs that need to be physically tested.

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

2.1. Types of Stents

Typically, stents fall into two major categories described below:

2.1.1. Balloon-expanded Stents

These stents are made from the material that can be plastically deformed, e.g. stainless steel. The expansion in-vivo is achieved through expansion of a balloon as shown in Fig 2.2. The balloons are typically made of hyper elastic materials that get very stiff (almost rigid) during expansion and are able to plastically deform the stent.



Fig 2.2: Balloon-expanded stent (Courtesy: fda.gov)

2.1.2. Self-expanding Stents

These stents as shown in Fig 2.3 typically expand on their own against the artery after deployment. These are typically made of material like Nitinol that can undergo large elastic deformation, and can expand through recovery of elastic deformation after the constraining delivery system has been removed. Nitinol stents are widely used due to their key properties of super elasticity, biocompatibility and fatigue resistance.

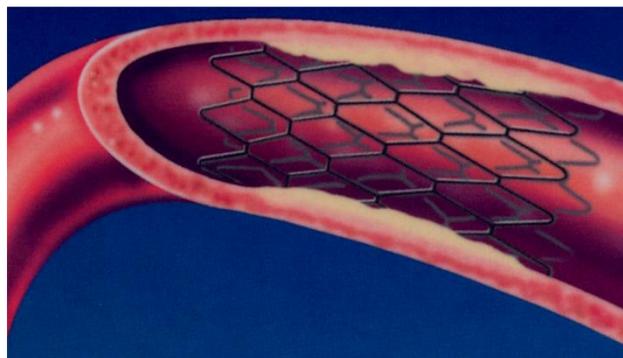


Fig 2.3: Self-expanding Nitinol stent (Courtesy: Journal of Minimally Invasive Therapy and Allied Technologies, Vol. 9 Cover)

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

2.2. Materials Used in Stents

Typical materials used in stents are of the following types:

- ▶ Stainless steel is widely used in balloon-expanded stents due to its corrosion resistance property.
- ▶ Platinum alloys, niobium alloys, cobalt alloys and tantalum are also used for balloon-expanded stents.
- ▶ Nitinol super elastic alloy is used in self-expanding stents due to its recovery and shape memory properties.
- ▶ Polymers are finding their way into the manufacturing of stents – shape memory polymers and biodegradable polymers are being explored for use.
- ▶ Biodegradable metal is also being researched.

2.3. Fabrication Process

Stents are generally intricate in design and the majority of coronary stents are produced by laser cutting. Most stents are cut from metal tubing but some are cut from metal sheets and rolled into a tubular shape. Other fabrication methods used are braiding, knitting, coiling etc. Common examples of stents produced from different fabrication processes are shown below in Table 2.1.

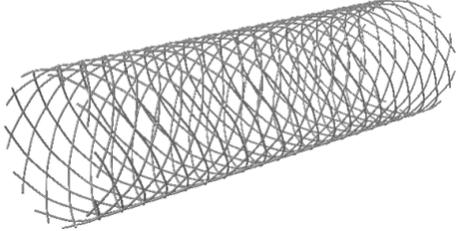
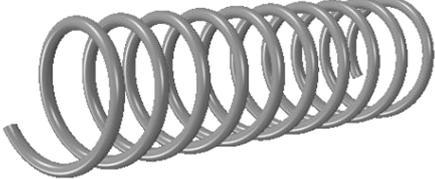
Laser-cut stent	
Braided stent	
Coiled stent	

Table 2.1: Different types of stents

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

2.4. Documentation Reference

The following DS SIMULIA class is recommended as a prerequisite for understanding this Best Practice document:

- ▶ Modeling Stents using Abaqus, Abaqus Advanced Training class

Information about SIMULIA Training classes can be found at

<https://www.3ds.com/products-services/simulia/services/training-courses/schedule-registration/>

3. Geometric Modeling and Meshing

In creating FEA models for stents, its geometry should be carefully studied. The manufacturing process of stents usually starts from a thin tube in which a pattern is micro-machined (laser-cut). The FEA model is built from this machined tube geometry. In most cases, stent geometry has a regular pattern and symmetry. It is highly recommended to take advantage of pattern and symmetry as it reduces the model size and typically results in an overall reduction of simulation cost. Analyzing a full 3D stent model can be prohibitively expensive because it requires enormous compute resources to get reasonable analysis results. It is therefore important to build the correct stent model right from the beginning. Both laser-cut stents and wired or braided stents can be fully represented by simple building blocks that can be repeated sectors of the pattern. Only a symmetrical portion of this building block needs to be considered for modeling. This building block can be meshed appropriately for the analysis as discussed later in this section and is sufficient to get results for the full model.

However, for non-symmetrical loading and deformation patterns (bending, twist, axial, combined) full stent models may be required. The use of regular patterns can be of great benefit when creating a full stent model. Laser-cut stents generally have more complex geometry than braided or coiled stents, and therefore, the idea of using pattern and symmetry will be demonstrated here for a laser-cut stent. It is a common practice for engineers to start with a model that has dimensions of the polished stent, rather than with dimensions of the laser cut stent; polishing stents improves fatigue life. Also, starting with dimensions of the laser-cut stent makes the modeling cumbersome with not much value added. It requires that an outer element layer be created, which has the thickness equal to the expected amount of material removed during the surface finish process. Therefore, in modern simulations, this step is not included and only expansion/shape-set, crimp, anneal and fatigue loading steps are considered [1].

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

The struts of a laser-cut stent usually have large enough cross-sections compared to their lengths that approximation as beams is not sufficiently accurate. Given the complexity of the deformation that the stent undergoes with an ultimate interest in fatigue safety, it is best to model the stent with solid elements that can capture all aspects of deformation and truly represent a state of 3D stress. The example used in this document is provided by Confluent Medical (previously, Nitinol Devices & Components) [8], and will be used throughout the document.

The full stent geometry is shown in Fig 3.2.

3.1. Meshing

It is best to start meshing with a flat piece of pattern, which is generally part of a strut. The boundaries of the flat geometry are laser-paths with which stents are cut. Stents are cut on a thick tube, and the laser is always pointed in the radial direction along the path while cutting. This makes the flat geometry a projection of the cylindrical geometry of the stent at the outer radius. Thus, it is crucial to transform this flat part to cylindrical tubular shape to get the correct stent shape.

3.1.1. Abaqus

To reproduce laser-cut stent shapes as produced by the actual manufacturing process, the flat geometry is first meshed and then wrapped to produce the final cylindrical shape. It is important to adhere to this procedure to get the correct stent shape in the mesh representation. The meshing process starting from the basic flat geometry is shown in Fig 3.1.

The description of the workflow following Fig 3.1 refers to the Abaqus/CAE GUI interface. The modules and the tools used are highlighted in bold. Tool names are italicized.

- ▶ In the **Part** module of Abaqus/CAE, 2D flat geometries of strut and strut & bridge were created (Fig 3.1a and Fig 3.1b).
- ▶ After some necessary face-partitions and local mesh seed assignments in **Mesh** module, the meshes for these two parts were generated (Fig 3.1c and Fig 3.1d).
- ▶ Instancing and linear patterning in the **Assembly** module (Fig 3.1e) were used to get the mesh of the full 2D flat stent sector (Fig 3.1f). The green color represents the 2D struts part while the white represents the 2D strut & bridges part.
- ▶ The ***Merge Instances*** tool in the **Assembly** module was used to merge the meshes and generate a new orphan mesh part 2D flat stent.

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

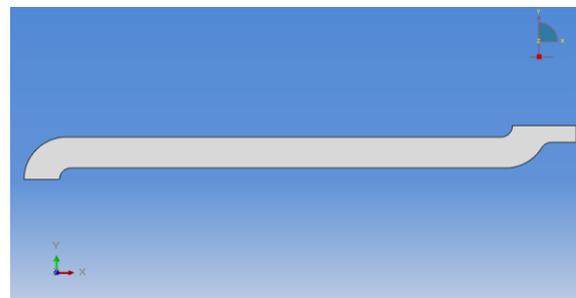
This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

- ▶ In the **Mesh** module of Abaqus/CAE, the 2D flat stent mesh was extruded to a 3D flat stent mesh using the **Edit Mesh → Create Solid Layers** tool. During this process, normals of some regions may need to be flipped to keep all the normal directions the same. The 3D flat mesh is shown in Fig 3.1g.
 - ▶ The 3D flat mesh was wrapped into a tubular shape using the **Wrap Mesh Plugin**. The plugin is available through KB article QA00000008170 at <https://www.3ds.com/support/knowledge-base/>. The final wrapped mesh is shown in Fig 3.1h.
- ▶ (Optional) To verify the accuracy of final wrapped stent mesh, the final stent mesh is superimposed on the full stent geometry in Fig 3.2 below.

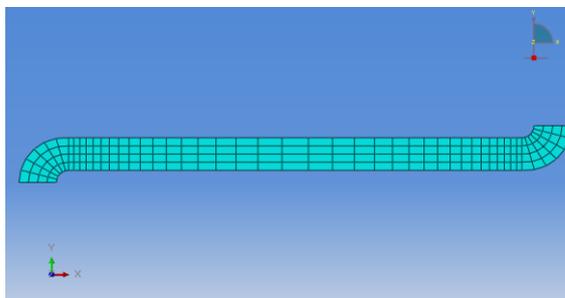
For the sake of computational efficiency, only one meshed sector is used with suitable symmetry boundary conditions for simulation.



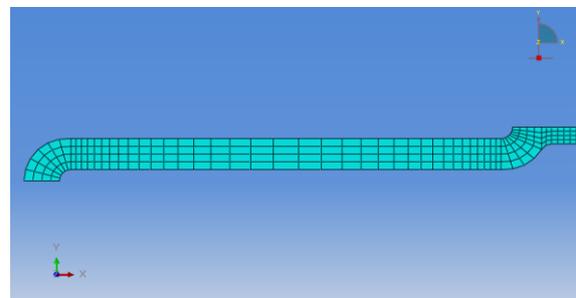
a) Geometry of Part "2D strut"



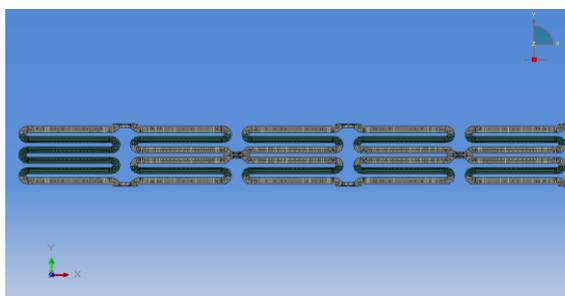
b) Geometry of Part "2D strut & bridge"



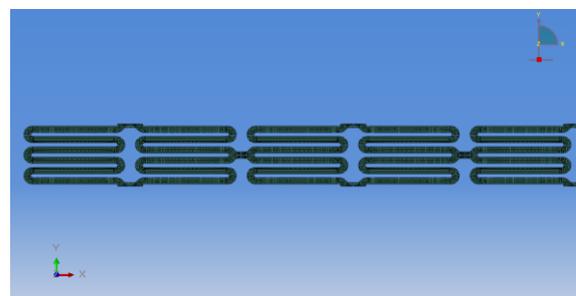
c) Mesh of Part "2D strut"



d) Mesh of Part "2D strut & bridge"



e) Linear patterning & positioning instances



f) Mesh of Part "2D flat stent"

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

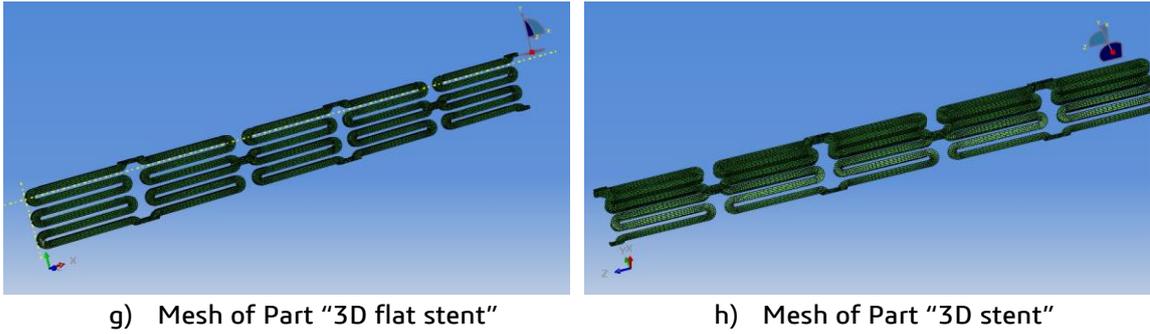


Fig 3.1: Workflow of creating a stent mesh from scratch using Abaqus/CAE.

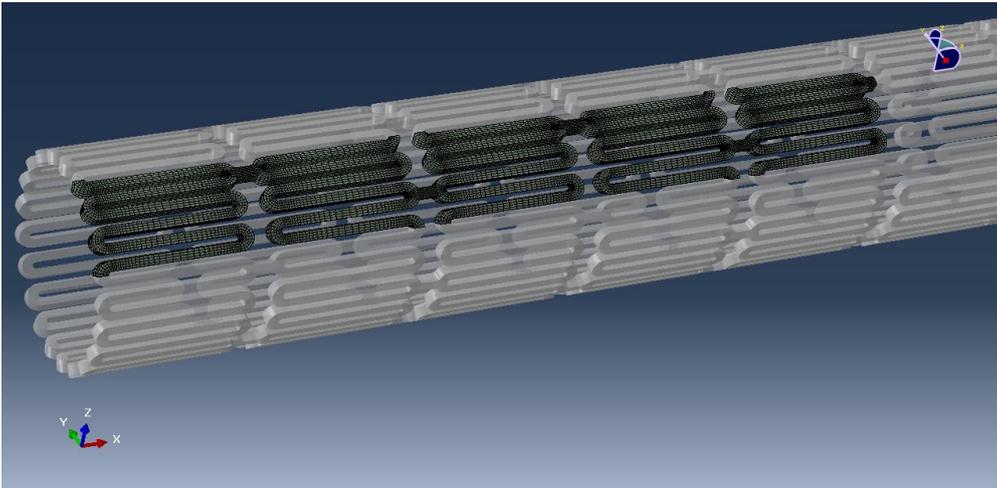


Fig 3.2: Final stent mesh and complete stent geometry.

3.1.2. 3DEXPERIENCE

On the 3DEXPERIENCE platform a tool like the *Wrap Mesh Plugin* is not available as of now, so wrapping of the mesh cannot be performed in the same way as it is done when using Abaqus/CAE. Instead, geometry tools in the 3D modeling apps (west quadrant of the Compass) are used to produce the correct three-dimensional model of the stent starting with a flat planar geometry. Assuming modeling of laser cut stents, once the geometry is created, a hex mesh can be generated on the stent geometry using the 'sweep' meshing technique.

The 3DEXPERIENCE platform allows specification of various meshing rules which can be utilized to do custom meshing in regions of interest, e.g. curved semi-circular portions in the stent.

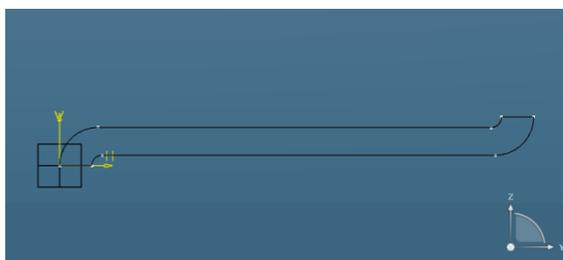
The step-by-step geometry modeling and meshing process in 3DEXPERIENCE is shown below in Fig. 3.3. The description of the workflow following Fig 3.3 refers to

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

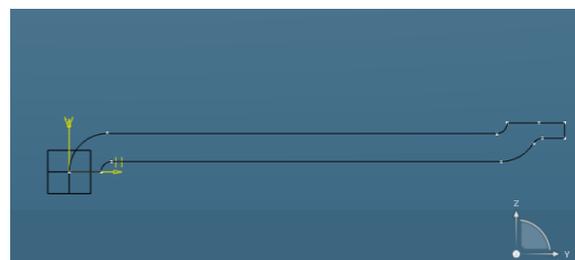
This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

3DEXPERIENCE 2017x. The apps, sections, and the tools used are highlighted in bold. For distinction, app names are in double quotes and Tool names are italicized.

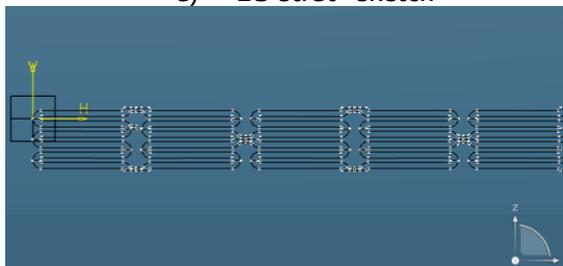
- ▶ In the **Part Design Essentials** app of 3DExperience, a 2D sketch of the strut and the strut & bridge were created via the **Positioned Sketch** section (Fig 3.3a and Fig 3.3b).
- ▶ Under the **Transform** section of **Part Design Essentials** app, the **Regular Pattern**, **Translation** and **Symmetry** tools were used to generate the pattern of 2D Sketches (Fig 3.3c).
- ▶ In the **Generative Shape Develop** app, a cylinder with the same outer radius as the stent was generated via the **Extrude** option under the **Surface** section. It was unfolded at the tear line (straight line on the cylindrical surface in Fig 3.3d) using **Unfold** under **Develop** section. The cylinder and the unfolded cylinder surface (brown) are shown in Fig 3.3d.



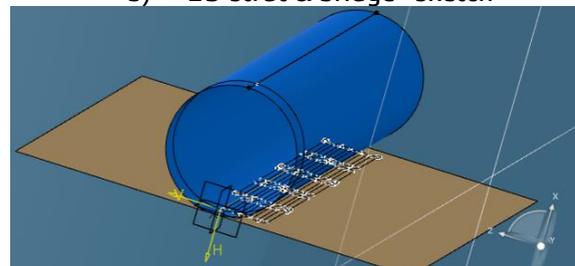
a) "2D strut" sketch



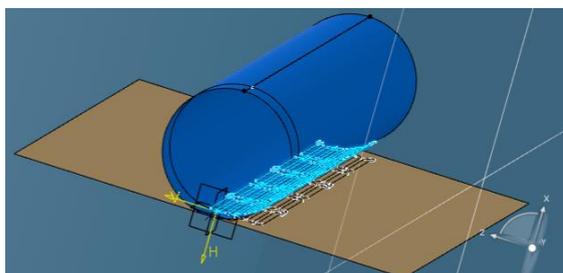
b) "2D strut & bridge" sketch



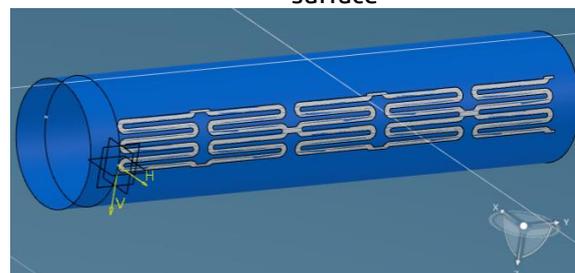
c) Linear pattern & positioning sketches



d) Wrapped cylinder & unfolded cylinder surface



e) Wrapped the flat stent curves to cylinder



f) Wrapped stent surface

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

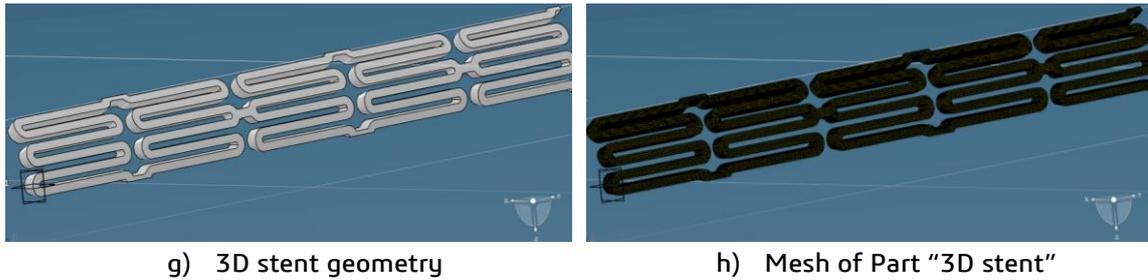


Fig 3.3: Workflow of creating stent geometry and mesh from scratch using 3DX 2017x

- ▶ Using the **Transfer** tool under the **Develop** section, the stent curve is transferred from the unfolded cylinder surface to the cylindrical surface. As a result, all stent curves were wrapped onto the existing cylinder (Fig 3.3e).
- ▶ The Wrapped stent surface (Fig 3.3f) was created from the wrapped stent curve using the **Fill** tool under the **Surface** section. The surface for each strut was generated and then merged together via the **Join** tool under the **Transform** section.
- ▶ The final 3D stent geometry (Fig 3.3g) was created from the wrapped stent surface using the **Thick Surface** tool under the **Volume** section.
- ▶ The **Mesh Creation** app was invoked from the **Structural Model Creation** app by switching to **Mesh** under the **Setup** section. The final mesh (Fig 3.3h) was generated using **Sweep 3D Mesh** under the **Mesh** section.

3.2. Mesh Refinement

3.2.1. Abaqus

One problem faced with refinement of orphan meshes is that they do not have geometry associated with them to change the mesh seeding. With geometry, the mesh seeding on the geometry can be changed to obtain a finer mesh. With orphan meshes, the Abaqus/CAE tool to subdivide the mesh (**Mesh → Edit Mesh → Subdivide Layers**) provides a partial solution to the problem. In many cases of laser-cut stents where the cross-section of the stent is rectangular, the mesh density in the thickness direction, or all directions, can be increased by using this tool.

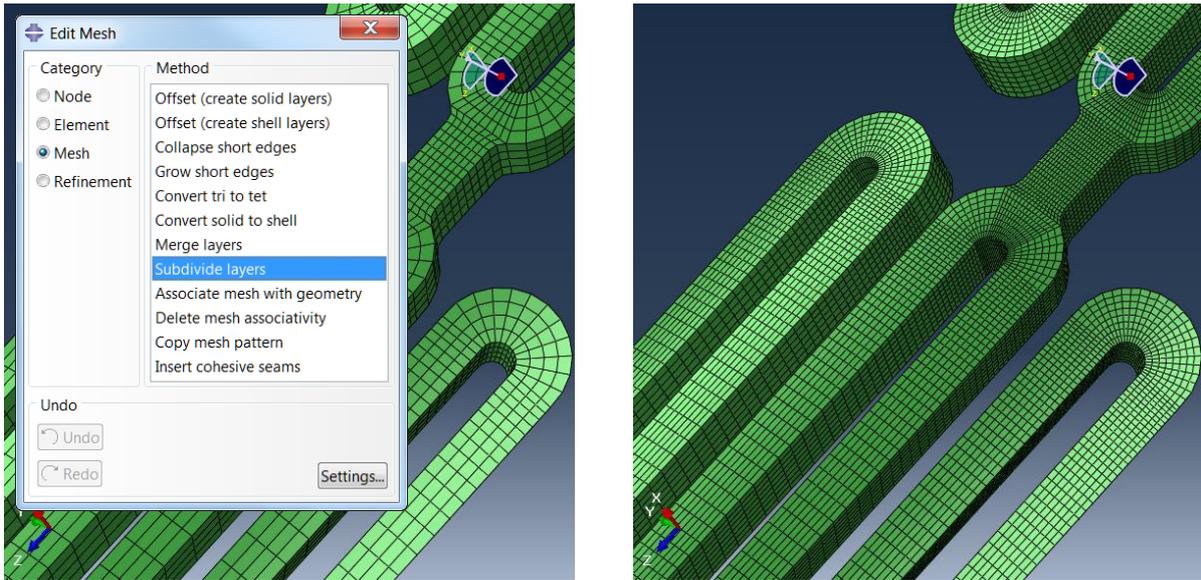


Fig 3.4: Refined mesh (Right) using **Subdivide Layers** with all directions (Left)

However, the user must ensure that after mesh subdivision by the method described above, the mesh is fully consistent; i.e., after revolution or patterning through 360 degrees, the end nodes coincide. In complex geometries, this may not be the case due to tolerance errors, and in that case, it is best to refine the mesh on a flat geometry and then follow the procedure of full mesh generation.

3.2.2. 3DEXPERIENCE

Mesh refinement when using the 3DEXPERIENCE platform can be performed by setting different meshing rules under batch meshing. It is not explored here in detail. The help.3ds.com website can be referred to for the full 3DEXPERIENCE documentation.

3.3. Mesh Convergence

Getting a suitable mesh density for the model to achieve reasonably accurate results is the next important step. Different element types and mesh densities can be used for a given problem, but the fidelity of the mesh should be decided based on the end goals of the analysis. Mesh convergence studies are done to see which mesh gives a sufficiently accurate result before going into further analyses. For doing this, the mesh can be refined, the element type can be changed, or both can be done simultaneously. As the mesh gets finer, a full stent model can easily run into millions of degrees of freedom. In cases where symmetry in the stent cannot be used, analyzing basic stent results helps in deciding which parts of the stent should have a finer mesh with the objective of limiting the overall model size.

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

Best Practice

For completeness, a mesh convergence study was done for this problem. It should be noted that the study provided here is only for reference and the user must conduct their own mesh convergence studies based on their specific problem and goal. Many different element types are compared in this study. The comparative accuracy and cost of the analysis are demonstrated to give users some guidelines for selecting a certain mesh refinement. It gives a general idea to the user in terms of picking the right accuracy vs cost to meet their specific end goals; e.g., design iteration analyses may not need as high a level of accuracy as that for analyses submitted to regulatory bodies.

A symmetric model of this problem turned out to be too expensive for conducting mesh refinement studies, so a strut is chosen instead. It was ensured that the results on the strut on the symmetric model matched with the results on the single strut with appropriate boundary conditions. A mesh is considered to be converged if further refinement does not produce any significant change in the results. The strut is shown in Fig 3.5.



Fig 3.5: Mesh refinement levels

3.3.1. Hex Elements

Reduced integration hex elements and incompatible mode elements are most commonly used in the analysis of stents. The selection of element type and mesh density depends on the ultimate goal of the analysis as discussed above.

3.3.2. Accuracy

Membranes can be conveniently overlaid on solid parts in the **Property** module in Abaqus/CAE through Skin as shown in Fig 3.6. The membrane overlay on the surface of the solid elements is used to obtain higher accuracy in surface results and avoid extrapolation errors. The material properties of the membrane are assumed to be the

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

same as that of the underlying hex elements, and typically the thickness used is less than 1/10th of the underlying hex element size.

The Fig 3.7 below provides a comparison of accuracy in results based on different mesh densities. The hex element (starting with letter 'C') results are compared with and without a membrane element (starting with letter 'M') overlay on the surface.

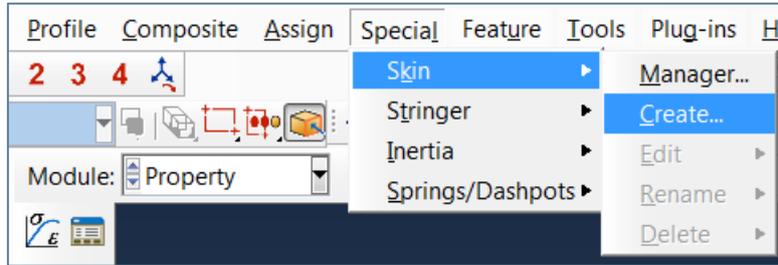


Fig 3.6: Using skin for membrane overlay

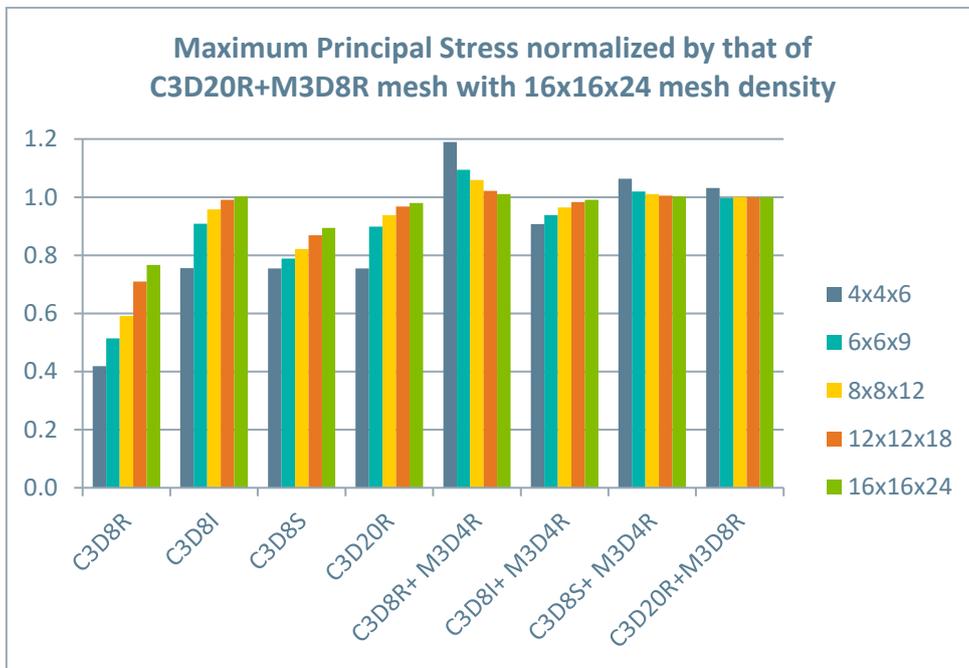


Fig 3.7: Chart comparing different hex elements

The mesh refinement is explained in Fig 3.8 below. An 8 x 8 x 12 mesh refinement is shown, where 8 x 8 represents the 'square' cross section of the stent with 8 elements on each side. The '12' represents the number of elements in the quarter circle in the curved semi-circular portion of the stent. The same scheme is applied for all the mesh refinements used in this study.

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

It should be noted that stent designs continuously evolve in the industry and this is just a simple example using a common shape. For other cross-sections and shapes of stent components, users should conduct their own mesh convergence studies.

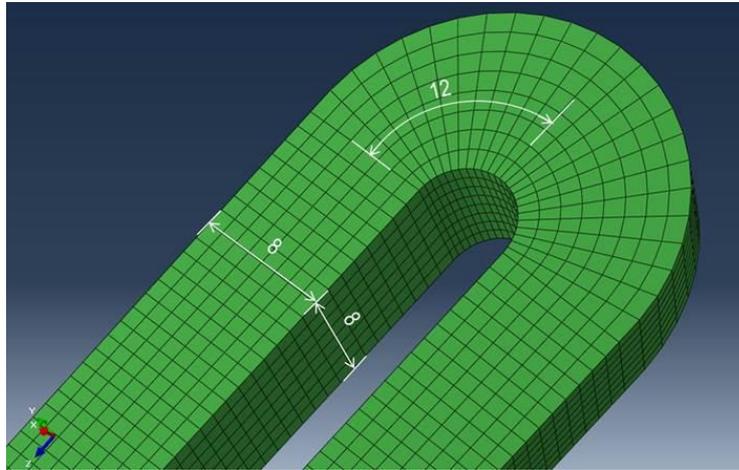
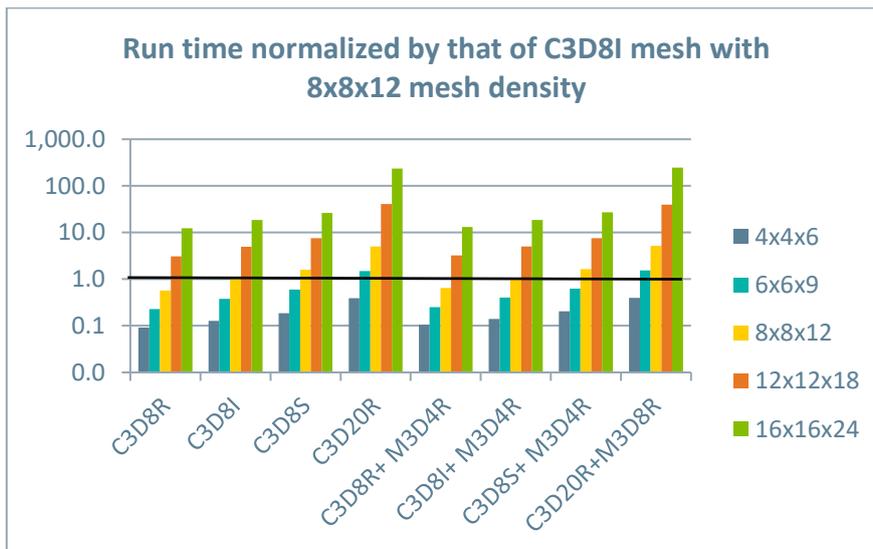


Fig 3.8: 8 x 8 x 12 mesh refinement

3.3.2.1. Run Times

When considering accuracy, it is important to get an idea of run times (cost) for each case in order to judiciously decide on the level of mesh refinement. Run times of the hex meshes considered are shown in Fig 3.9. The chart is normalized by the run time of the C3D8I 8 x 8 x 12 mesh. This mesh is considered to be of reasonable cost and acceptable accuracy (the maximum principal stress is 96% of the result of converged C3D20R+M3D8R mesh). The user needs to decide on the compromise between cost and accuracy based on the ultimate analysis goals.



Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

Fig 3.9: Chart comparing of the run time with different hex elements

For best accuracy, the mesh with C3D20R+M3D8R elements can be used. This mesh converges quickly and even the 6 x 6 x 9 mesh refinement gives reasonably accurate results. Using second order elements is a bit more expensive, but if cost is of not much concern, this combination can be accurate for stent analysis even with a relatively coarse mesh. The accuracy has been confirmed in many stent projects that SIMULIA has undertaken over the years. The consequence of using second order elements is that even a moderately sized model can very quickly run into millions of degrees of freedom, making the model prohibitively expensive to run. So, a judicious engineering judgment is needed before using these elements.

A good compromise between cost and accuracy is achieved with C3D8I elements, which are specifically designed to capture correct bending behavior. Considering accuracy and cost, C3D8I elements appear to be good, economical elements for doing stent analysis with adequate accuracy. An 8 x 8 x 12 mesh refinement of C3D8I elements without use of membranes provides sufficient accuracy for most practical purposes. Care should be exercised to keep the element shape as cuboidal as possible, and to avoid elements shaped like trapezoids or rhombuses. A deviation from cubical shape reduces the accuracy of C3D8I elements. With C3D8I elements, one can do away with membranes and still get reasonably high accuracy. The best accuracy is obtained by using C3D20R elements with M3D8R elements but the cost of analysis is higher.

A note of caution for C3D8I elements is that in some cases with Nitinol material model, convergence may not be achieved. This has been observed sparingly, and in most cases C3D8I elements have been fine with Nitinol. If the convergence is achieved, the results are correct. In case of non-convergence, other element type must be selected.

3.3.3. Tet Elements

Tet elements are usually not considered for stent analysis as they are stiffer and are not suitable for bending response. Major part of loading of stents is bending dominated, and so tet elements can provide inaccurate results affecting subsequent fatigue calculations. Moreover, tet meshes may have a difficult time in converging, especially first order tets. It is due to these reasons that an elaborate mesh convergence study is not conducted with tet elements.

3.3.3.1. Accuracy

A relatively close accuracy compared to hex meshes can be obtained with a much finer tet mesh. Using the same element size as used for the hex mesh will produce a much finer

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

mesh with tets and can be a starting point. Second order tets are more accurate than the linear tets and, therefore, are preferred over linear tets.

3.3.4. Run Times

With finer meshes, the run times using tet meshes can be extremely large compared to hex mesh run times for an equivalent accuracy. This increases the cost considerably compared to the cost of the hex mesh. The run times for tet meshes can be up to 25 times that of hex meshes with equivalent accuracy.

All analyses for mesh refinement studies were run with Abaqus 2016. For details on the version and hardware configuration used, refer to Appendix 1.

3.4. Documentation Reference

- ▶ Abaqus 2016
 - ▷ "Partitioning faces," Section 70.6 of the Abaqus/CAE User's Guide
 - ▷ "Partitioning cells," Section 70.7 of the Abaqus/CAE User's Guide
 - ▷ "Generating layers of solid elements offset from an existing mesh," Section 64.7.1 of the Abaqus/CAE User's Guide
 - ▷ "Merging or cutting part instances," Section 13.10.12 of the Abaqus/CAE User's Guide
 - ▷ "The Mesh Module," Section 17 of the Abaqus/CAE User's Guide
 - ▷ "Element definition," Section 2.2.1 of the Abaqus Analysis User's Guide
 - ▷ "Choosing the appropriate element for an analysis type," Section 27.1.3 of the Abaqus Analysis User's Guide
- ▶ Abaqus 2017
 - ▷ Abaqus | Abaqus/CAE | Using toolsets | The Partition toolset | Partitioning faces
 - ▷ Abaqus | Abaqus/CAE | Using toolsets | The Partition toolset | Partitioning cells
 - ▷ Abaqus | Abaqus/CAE | Using toolsets | The Edit Mesh toolset | Editing the entire mesh | Generating layers of solid elements offset from an existing mesh
 - ▷ Abaqus | Abaqus/CAE | Creating and analyzing a model using the Abaqus/CAE modules | The Assembly module | Creating and manipulating part and model instances | Merging or cutting part instances
 - ▷ Abaqus | Abaqus/CAE | Creating and analyzing a model using the Abaqus/CAE modules | The Mesh module

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

- ▷ Abaqus | Introduction & Spatial Modeling | Spatial Modeling | Element definition | Element definition
- ▷ Abaqus | Elements | Introduction | Choosing the appropriate element for an analysis type

4. Materials

Coronary stents are typically made of Nitinol, stainless steel, or other exotic alloys. Non-coronary stents (typically used in peripheral body parts such as legs, arms, etc.) are typically made of Nitinol. The manufacturing process of laser cut stents needs to be modeled because the final shape is not known a priori, rather it is derived from the laser cutting pattern. It typically involves mandrel expansion and shape-setting (a heat treatment). For analysis purposes, shape-setting is treated the same as annealing, in the sense that stresses, strains and state variables get reset to zero, while the deformed shape is retained.

Most of the materials used and mentioned in section 2.2 can be described by built-in material models in Abaqus, e.g. elastic-plastic material. The Materials section of Analysis User's Guide can be referred to for more details. Nitinol material is described here given its wide use and complexity. Although Nitinol is difficult to process, it is a very attractive alloy for medical implant applications due to its following properties:

- ▶ Superelasticity
- ▶ Biocompatibility
- ▶ Fatigue strength

Nitinol is a shape memory alloy; however most medical implant applications do not use the shape memory effect, but rather its super elastic property is of key interest. Material properties are sensitive to temperature, and the temperature range of interest is quite tight. Its uniaxial stress-strain response and temperature dependence is shown in Fig 4.1 and Fig 4.2 respectively.

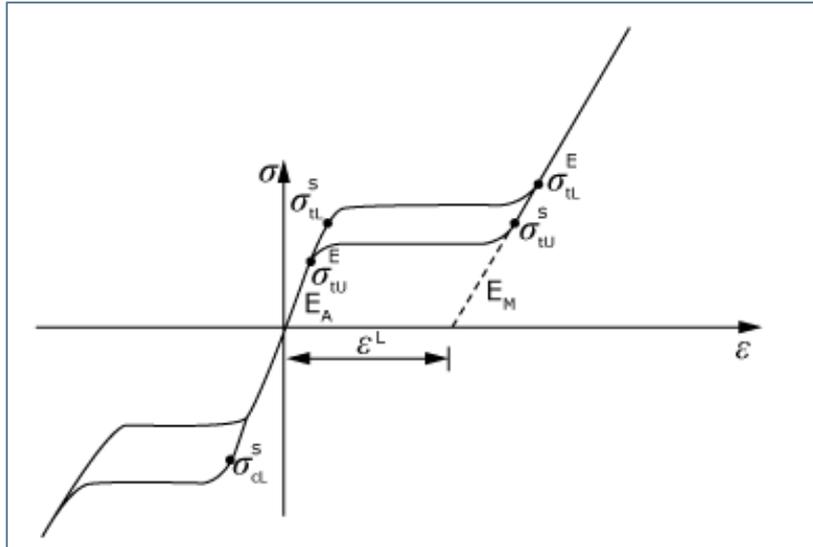


Fig 4.1: Typical stress-strain response of Nitinol

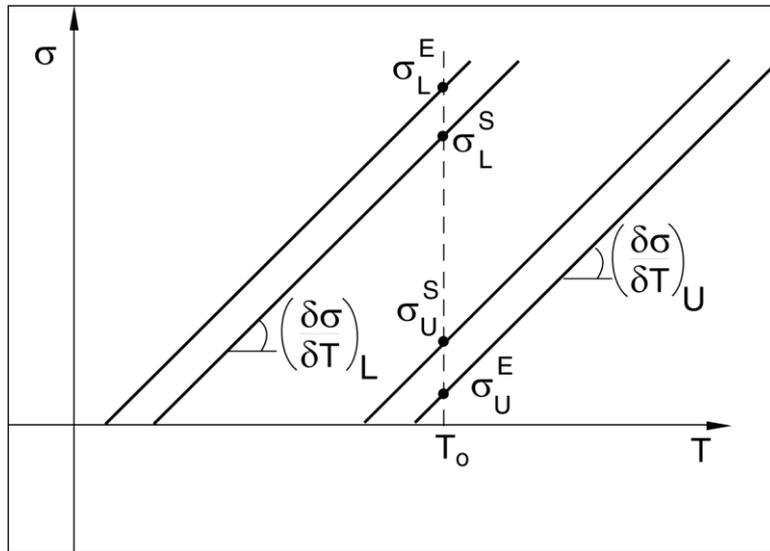


Fig 4.2: Stress-temperature curve for Nitinol

The Nitinol material model is based on the works of Auricchio F. and Taylor R. [2, 3] and uses a Drucker-Prager like formulation to address transformation and plasticity. It is implemented for both Abaqus/Standard and Abaqus/Explicit through user subroutines UMAT and VUMAT [4, 5] respectively. The User Material constants until version Abaqus 2016 can be defined through GUI interface of Abaqus/CAE. The Keyword interface for invoking the material subroutines is shown below.

From Abaqus 2017 and 3DEXPERIENCE 2017x the Nitinol material subroutines have been provided as built-in material models. The GUI interface in Abaqus/CAE is available in Abaqus 2017 under **Super Elasticity** in the **Materials** module; using the **User Material**

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

option is no longer needed. This implementation does not take into account any material behavior changes due to cycling.

Keyword interfaces for Abaqus 2016 and Abaqus 2017 along with the relevant output are described below.

4.1. Abaqus 2016 Keyword Interface

4.1.1. UMAT

* MATERIAL, NAME=name

* USER MATERIAL, CONSTANTS=16 + N_A + 2 N_P

$$E_A, \nu_A, E_M, \nu_M, \varepsilon^L, \left(\frac{\delta\sigma}{\delta T} \right)_L, \sigma_L^S, \sigma_L^E$$

$$T_O, \left(\frac{\delta\sigma}{\delta T} \right)_U, \sigma_U^S, \sigma_U^E, \sigma_{CL}^S, \varepsilon_V^L, N_A, N_{S1}, \dots, N_{SNA}, N_P, \sigma_1^P, \varepsilon_1 \dots \sigma_{NP}^P, \varepsilon_{NP}$$

* DEPVAR

31,

The **name** parameter on * MATERIAL must start with ABQ_SUPER_ELASTIC.

4.1.2. VUMAT

* MATERIAL, NAME=name

* USER MATERIAL, CONSTANTS=15 + 2 N_P

$$E_A, \nu_A, E_M, \nu_M, \varepsilon^L, \left(\frac{\delta\sigma}{\delta T} \right)_L, \sigma_L^S, \sigma_L^E$$

$$T_O, \left(\frac{\delta\sigma}{\delta T} \right)_U, \sigma_U^S, \sigma_U^E, \sigma_{CL}^S, \varepsilon_V^L, N_P, \sigma_1^P, \varepsilon_1 \dots \sigma_{NP}^P, \varepsilon_{NP} \text{ (upto 8 values per line)}$$

* DEPVAR

31,

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

The **name** parameter on * MATERIAL must start with ABQ_SUPER_ELASTIC_N1D (for beam elements), ABQ_SUPER_ELASTIC_N2D (for plane stress, shell or membrane elements) or ABQ_SUPER_ELASTIC_N3D (for 3D, plane strain or axisymmetric solid elements). For information on the material constants and other details, see KB articles QA00000009225 and QA00000009132 at <https://www.3ds.com/support/knowledge-base/>.

The above interface represents super elastic-plastic behavior. For simplicity only the super elastic model is considered in the example presented here.

4.2. Abaqus 2016 Output

Since the material model is available through user subroutines, the output is provided through Solution Dependent Variables (SDVs). The meaning of the different SDVs is described below.

- ▶ SDV1 – 6: Linear elastic strains
- ▶ SDV7 – 12: Transformation strains
- ▶ SDV19: Equivalent transformation strain
- ▶ SDV20: Volumetric transformation strain
- ▶ SDV21: Fraction of Martensite
- ▶ SDV22: Equivalent uniaxial tensile stress
- ▶ SDV23: Equivalent uniaxial tensile transformation strain
- ▶ SDV24: Equivalent uniaxial tensile total strain
- ▶ SDV25: Equivalent plastic strain
- ▶ SDV26 – 31: Plastic strains
- ▶ SENER, ELSE, ALLSE: Linear elastic strain energy variables
- ▶ PENER, ELPD, ALLPD: Transformation strain energy variables

It is observed in Nitinol that the transformation stress levels decrease with an increase in the plastic strain. This variation in the transformation plateau with plastic strain can be specified by using the following user subroutines:

- ▶ USUPERELASHARDMOD for Abaqus/Standard
- ▶ VUSUPERELASHARDMOD for Abaqus/Explicit

4.3. Abaqus 2017 Keyword Interface

*ELASTIC

E_A, ν_A

*SUPERELASTIC, NON-ASSOCIATED= ε_V^L , SHAPE SET FIELD=<FV#>

$E_M, \nu_M, \varepsilon^L, \sigma_L^S, \sigma_L^E, \sigma_U^S, \sigma_U^E, \sigma_{CL}^S$

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

$T_o, \left(\frac{\delta\sigma}{\delta T}\right)_L, \left(\frac{\delta\sigma}{\delta T}\right)_U$, value of field variable # that triggers shape setting

*SUPERELASTIC HARDENING

σ_1^P, ϵ_1

.....

.....

*SUPERELASTIC HARDENING MODIFICATIONS, USER, PROPERTIES

$\sigma_L^S, \sigma_L^E, \sigma_U^S, \sigma_U^E, \epsilon_1^{pl}$

.....

.....

4.4. Abaqus 2017 Output

The output in Abaqus 2017 is not in the form of SDVs but instead use standard output Identifiers. The following describes the correspondence between standard output identifiers and the SDVs for Nitinol in Abaqus 2016:

- ▶ EE: SDV1-6
- ▶ TE: SDV7-12
- ▶ TEVOL: SDV20
- ▶ MVF: SDV21
- ▶ TEEQ: SDV23 PE: SDV26-31

Other related standard output identifiers are:

- ▶ PEEQ: Equivalent plastic strain
- ▶ PENER, ALLPD, ELPD: Transformation Strain Energies

4.5. Convergence

With a little care in defining the material parameters described above, the Nitinol material model gives good convergence behavior. All values used must be reasonable and representative of Nitinol behavior. Care should be taken when defining the slope of the plateaus for the transformations in loading and unloading. The slope of the plateaus under ϵ^{\perp} in Fig 4.1 should be positive for better convergence behavior.

The volumetric transformation strain ϵ_v^{\perp} should be correctly defined. By default, the Nitinol material model uses an associated formulation for the phase transformations. As a consequence, if the behavior in compression is different from the behavior in tension, a natural volumetric transformation strain develops. This associated formulation also exhibits the decrease in transformation strain in compression that is usually observed. Since the experimentally observed volumetric transformation strain is close to zero, the model allows for the volumetric transformation strain ϵ_v^{\perp} to be defined. However, the

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

decrease in transformation strain in compression disappears when ε_V^L is set to zero. In order to recover the default formulation, ε_V^L should be set equal to ε^L . In cases where $\varepsilon_V^L \neq \varepsilon^L$, the **UNSYMM** optional parameter must be used on the ***STEP** keyword for better convergence behavior.

For other materials with yielding behavior, the proper hardening curve should be defined with a positive slope and adequate yield stress – plastic strain data should be provided to cover the entire range of stresses and plastic strains expected in the analysis. More details on convergence are covered in Abaqus Analysis Guide and the training course, “Modeling Contact and Resolving Convergence Issues with Abaqus.”

4.6. Definition of Nitinol in 3DEXPERIENCE

The user interface for Nitinol material is still not available on 3DEXPERIENCE platform in the Material app. From 3DEXPERIENCE 2017x FP1723, the Nitinol material can be used in a 3DEXPERIENCE model by invoking the built-in subroutines for Nitinol as described in section 3. The procedure is similar to using the Abaqus/CAE GUI where the “User Material” option is used to define the material constants of Nitinol. Before 3DEXPERIENCE 2017x FP1723 this option cannot be used, and the only way to define the Nitinol material is writing out the Abaqus input file from the 3DEXPERIENCE platform and manually adding the keywords for the Nitinol material definition. Appendix 2 describes how to write Abaqus input file from the 3DEXPERIENCE platform and how to run that input file on the 3DEXPERIENCE platform.

4.7. Documentation Reference

- ▶ Abaqus 2016
 - ▷ “Metal Plasticity,” Section 23.2 of the Abaqus Analysis User's Guide
 - ▷ “User-defined mechanical material behavior,” Section 26.7.1 of the Abaqus Analysis User's Guide
 - ▷ “UMAT,” Section 1.1.44 of Abaqus User Subroutines Reference Guide
 - ▷ “VUMAT,” Section 1.2.22 of Abaqus User Subroutines Reference Guide
- ▶ Abaqus 2017
 - ▷ Abaqus | Materials | Inelastic Mechanical Properties | Metal plasticity
 - ▷ Abaqus | Materials | Elastic Mechanical Properties | Superelasticity

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

5. Analysis and Steps

Stents are analyzed with the goal of gaining understanding of their durability under *in-vivo* fatigue loadings. The baseline loading in coronary stents is mainly pulsatile, coming from blood flow. Peripheral stents can have other types of loadings too. To get to the correct state of stresses and strains, it is important to go through the whole process of expansion, shape-setting, deployment and fatigue loading in a simulation. Starting from the laser cut shape from the Nitinol tubing, the actual process may involve multiple expansion steps followed by shape-setting at the end of each expansion step. At the end of the expansion steps, the stent will have an “oversized” shape (oversized means having a diameter slightly larger than the diameter of the artery in which it will be deployed). The manufacturing process finally includes an electropolishing step that removes a surface layer of material. We have then a stent in its final “oversized” shape. The stent is then crimped, put into a delivery system and finally deployed in the artery where it is expanded. Since the stent is oversized, it presses against the artery wall keeping the artery “open.” Blood flow causes fluctuations in radial loading which can give rise to cyclic fatigue. It is important to analyze for the safety of the stent under these conditions.

To capture the whole process described above, the stent geometry is created from the electropolished stent dimensions. Simulating the material removal step due to electropolishing adds no value to the final results, and so is not considered [1]. Stent manufacturing usually involves a sequence of expansions followed by shape sets. This is because the laser-cut surface is very rough and the local stress concentrations it creates would be enough to cause fracture if a single expansion is applied. As it is shown in [1], the effect of multiple expansions on the final shape is negligible when compared to a single expansion. Therefore, a single expansion simulation is carried out to the final oversized diameter, followed by a shape set. Crimping, deployment, and fatigue loading simulations are carried out after the expansion and shape set. In Abaqus, each of these steps is an “analysis step,” and all the analysis steps together represent the full simulation. The analysis steps can be defined in the **Step** module in Abaqus/CAE, or through keyword ***STEP**. On the 3DEXPERIENCE platform, steps can be defined in the Mechanical Scenario app under Procedures. The steps can be categorized as follows:

- ▶ Expansion
- ▶ Shape-set or Anneal
- ▶ Crimp
- ▶ Deploy
- ▶ Steps for radial expansion and contraction representing fatigue loading

Each category above can have one or more steps.

Annealing can be performed in one of the following ways:

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

▶ **Abaqus/Standard**

In Abaqus/Standard, there is no analysis option to perform annealing (setting stresses and strains to zero), and therefore annealing is achieved in 2 steps:

- ▷ Removing stent mesh in a step
- ▷ Adding mesh back in the following step in a strain-free condition

The removal and adding back of the mesh can be achieved with the ***MODEL CHANGE** keyword.

▶ **Abaqus/Explicit**

In Abaqus/Explicit the use of ***ANNEAL** keyword in a step is able to achieve the same effect in one step.

▶ **Abaqus/CAE**

An *alternative* approach is that instead of using analysis options of model change or anneal, Abaqus/CAE can be used to import the deformed mesh from the ODB file after the Expansion step directly using **File → Import → Part**; the *.odb filter can be used to select the ODB file with the Expansion results. The final step/increment can be selected to import the deformed mesh. This mesh is imported stress/strain free and can be used for further analyses.

Multiple steps can be defined for expansion and contraction for fatigue, each step representing expansion or contraction. Usually, 3 cycles of expansion and contraction are enough to get a stabilized (same response in two consecutive cycles) fatigue response of Nitinol. Other materials may need more depending on their cyclic response. The steps used in the example used for this document are described in Appendix 1.

Output requests for stresses, strains and other relevant output variables can be made in the **Step** module of Abaqus/CAE or under the ***STEP** keyword. This output can be used for post-processing for fatigue safety.

All analysis techniques required for stent analysis are not yet available on the 3DEXPERIENCE platform (as of 3DEXPERIENCE 2017X), so the analysis in 3DEXPERIENCE will not be discussed.

5.1. Documentation Reference

▶ **Abaqus 2016**

- ▷ "Defining an analysis," Section 6.1.2 of the Abaqus Analysis User's Guide
- ▷ "Element and contact pair removal and reactivation," Section 11.2.1 of the Abaqus Analysis User's Guide
- ▷ "Annealing procedure," Section 6.12.1 of the Abaqus Analysis User's Guide

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

- ▶ Abaqus 2017
 - ▷ Abaqus | Analysis | Analysis Procedures | Introduction | Defining an analysis
 - ▷ Abaqus | Analysis | Analysis Techniques | Special-Purpose Techniques | Element and contact pair removal and reactivation
 - ▷ Abaqus | Analysis | Analysis Procedures | Annealing

6. Boundary Conditions and Loads

The major modes of deformation of stents come from expansion, crimping, annealing and fatigue loading. In some cases, stents are also required to undergo bending. To obtain the correct deformation, physically correct boundary conditions must be specified. Stents are circular in cross-section and cylindrical in shape which leads to the three main boundary conditions described below. These boundary conditions are most easily applied by creating a cylindrical coordinate (R, θ, Z) system placed on the axis of the stent with the Z-axis aligning with the centerline of the circular stent cross section.

6.1. Circumferential

A minimum repeatable sector of the stent cross section that will be able to represent the necessary deformations due to the applied loading should be picked and theta symmetry boundary condition should be applied at the ends of the sector where it is cut. In (R, θ, Z) system, this is DOF 2 which should be constrained on the surfaces of the cut.

If theta symmetry is not feasible, but the stent has repeated sectors that are the same, cyclic symmetry option can be used.

6.2. Axial

In most cases, the stent will also be symmetrical along its length or along its axis. Only the axially symmetric part needs to be considered in the model because in most cases the loading on stents is axially symmetric. Using a cylindrical coordinate system, where the Z-axis is aligned with the axis of the stent, the surface of the stent about which there is axial symmetry can be constrained in the Z-direction. In the (R, θ, Z) coordinate system, this is DOF 3. Even if there is no symmetry along the axis, Z-constraints should still be defined on one end of the stent, picking node(s) appropriately, so that rigid body modes are eliminated. This way, the other end can still deform axially while it is constrained against rigid body motion.

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

6.3. Radial

The expansion of stent is usually achieved by pushing out the stent in the radial (R) direction with a cylindrical surface. In the (R, θ , Z) coordinate system, this is DOF 1. This cylindrical surface can be rigid or deformable. Suitable contact definitions must exist between the cylinder and the stent for the stent to deform according to the expanding cylinder's motion. Separate cylindrical surfaces can be used for crimp and expand. Radial pulsatile loading can also be applied by specifying radial displacements on the cylinder.

6.3.1. With a Rigid Cylinder

Abaqus provides the option of defining an expanding rigid cylinder by using the user subroutine **RSURFU**. The user subroutine is provided through a KB article (QA00000018351) at <https://www.3ds.com/support/knowledge-base/>. The subroutine allows users to crimp, expand, bend, and bifurcate stents. The KB article has detailed examples of how these options can be used.

6.3.2. With a Deformable Cylinder

A deformable cylinder to expand or crimp the stent can be created with surface elements. The geometry meshed with surface elements does not require any section property definition. Radial displacements can be defined on all nodes of this deformable cylinder in order to expand or crimp the stent. In Abaqus/Explicit, mass for surface elements must be specified for contact to occur.

6.3.3. With Contact Interference

This technique is not usually recommended and should only be used by advanced and experienced Abaqus users. Instead of using a deforming rigid cylinder through **RSURFU**, rigid cylindrical surfaces can be created and placed at suitable radial distances as needed for expansion or crimping. With the use of the ***CONTACT INTERFERENCE** keyword, the stent can be crimped or expanded as needed. One may need to use ***MODEL CHANGE** to activate or de-activate contact between stent and cylinders if there are multiple cylinders in the model. This technique has been used successfully in consulting projects that SIMULIA has carried out.

In some cases, pulsatile loading can also be applied through pressure pulses instead of using boundary conditions; e.g., an artificial artery can be constructed with suitable compliance against which the stent is deployed. In this case blood pressure pulses can be applied on the inner surface of the artificial artery to model fatigue loading.

6.4. Documentation Reference

- ▶ Abaqus 2016
 - ▷ “Boundary conditions in Abaqus/Standard and Abaqus/Explicit,” Section 34.3.1 of the Abaqus Analysis User's Guide
 - ▷ “Transformed coordinate systems,” Section 2.1.5 of the Abaqus Analysis User's Guide
- ▶ Abaqus 2017
 - ▷ Abaqus | Prescribed Conditions | Boundary Conditions | Boundary conditions in Abaqus/Standard and Abaqus/Explicit
 - ▷ Abaqus | Introduction & Spatial Modeling | Spatial Modeling | Node definition | Transformed coordinate systems

7. Contact

7.1. Abaqus/Standard

In Abaqus/Standard, using general contact for stent self-contact and contact pairs for contact between the stent and other components is usually recommended. Using contact pairs can be more efficient than using general contact in Abaqus/Standard. When using contact pairs while modeling stents, the finite sliding, surface-to-surface contact discretization with penalty-based enforcement is recommended. Surface-to-surface discretization considers the shape of both the slave and master surfaces in the region of contact constraints. The surface-to-surface formulation enforces contact conditions in an average sense over regions nearby slave nodes rather than only at individual slave nodes (as in node-to-surface) resulting in smoother stresses as shown in Fig 7.1.

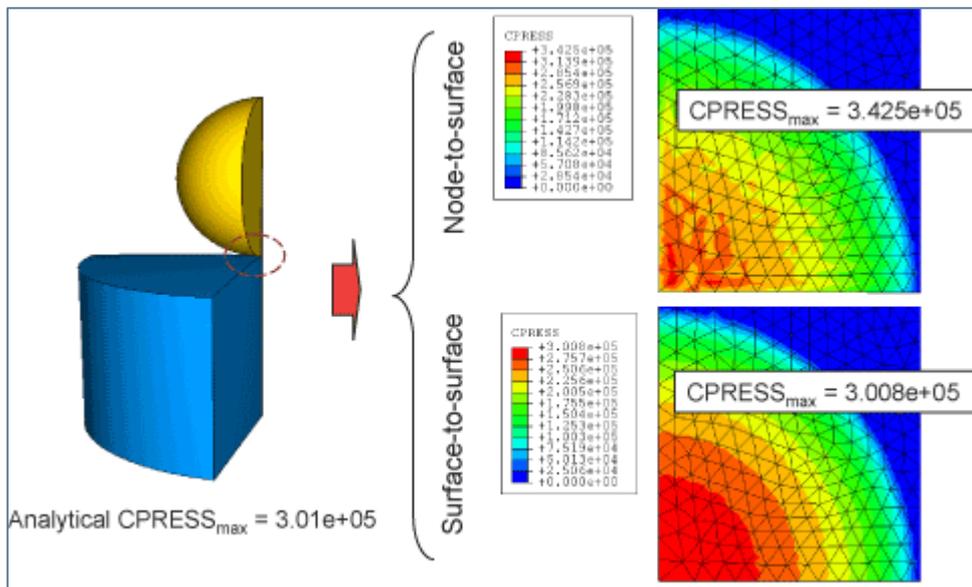


Fig 7.1 Comparison of CPRESS between node-to-surface and surface to surface

However, if edge-to-edge or edge-to-surface contact is expected in some region modeled with contact pairs, the node-to-surface contact formulation is recommended for these areas; for general contact the formulation takes care of such cases when the appropriate formulation is selected as an optional parameter (e.g. edge-to-surface).

7.2. Abaqus/Explicit

In Abaqus/Explicit, general contact is recommended for stent analysis. Contact pairs can be defined within the framework of general contact in Abaqus/Explicit if needed.

7.3. Contact Constraint Enforcement Methods

Contact constraint enforcement methods used for stent analysis typically use the hard contact pressure-overclosure relationship. The hard contact relationship minimizes the penetration of the slave surface into the master surface at the constraint locations and does not allow the transfer of tensile stresses across the interface. In some cases, where difficulty in contact convergence is encountered, the soft contact pressure-overclosure relationship Exponential is also used. For a smooth problem where the contact state is not continuously changing, any contact constraint enforcement method can be used, and the main choice comes in terms of cost of the enforcement method. By default, the linear penalty-based enforcement method is active and provides good results with high accuracy.

A comparative study of different contact constraint enforcement methods using a hard pressure-overclosure relationship is presented here. These are the most commonly used methods. The results are shown in Table 7.1. The finite sliding, surface-to-surface formulation with contact pairs was used for this study. The deformable cylinder for expanding the stent was constructed using both surface elements and a rigid cylinder through RSURFU user subroutine. The results from using surface elements (deformable cylinder) and RSURFU (deformable rigid cylinder) are similar; however, the run time of the analyses using RSURFU are lower than the run time of the analyses using surface elements. The typical element length in the analysis was 1.39E-2, and the thickness of the stent radially was 0.111 mm. As can be seen, the max penetration is relatively small in all methods. Penetration when using the Direct method is the lowest for all methods because it uses Lagrange multipliers to enforce near-zero penetration.

Pressure-overclosure = Hard								
Constraint Enforcement Method	Max Penetration		CPRESS		Max Principal Stress		Run time (normalized)	
	Surf Elements	Rigid Cylinder	Surf Elements	Rigid Cylinder	Surf Elements	Rigid Cylinder	Surf Elements	Rigid Cylinder
Direct	4.13E-16	2.42E-13	60.74	82.85	1626.54	1624.64	1	0.87
Penalty (Linear)	5.56E-06	5.99E-06	60.34	77.97	1626.16	1624.61	0.96	0.82
Penalty (Linear, Sf=0.01)	2.36E-04	2.40E-04	32.23	34.25	1625.35	1624.38	0.81	0.71
Penalty (Nonlinear)	4.73E-05	4.48E-05	52.77	61.82	1626.00	1624.55	0.86	0.75
Penalty (Nonlinear, Sf=0.01)	2.28E-04	2.29E-04	44.70	50.87	1625.64	1624.41	0.77	0.67
Augmented Lagrange	5.59E-08	6.52E-08	60.70	82.78	1626.17	1624.64	0.99	0.87

Table 7.1 Comparison of different constraint enforcement methods

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

7.4. Contact Convergence

Convergence difficulties in contact may arise when using Abaqus/Standard; however, the discussion here can also be used as a guide for Abaqus/Explicit. A common convergence difficulty could arise due to unconstrained rigid body motions before contact is established. Contact stabilization could be used in such situations if there is no better way of controlling the rigid body motion (e.g. applying boundary condition to make the surfaces come in contact). In some cases, initial contact between the stent and the deforming cylinder can suffer from chattering i.e., repeated opening and closing of contact which can cause convergence issues when contact is trying to get established. In such cases, the nonlinear penalty stiffness constraint enforcement method could be used to slowly nudge the components into contact and ease convergence difficulties. This method has small linear penalty stiffness in the initial part of contact and the penalty stiffness increases non-linearly as the contact is firmly established. The default options in Abaqus serve as a good starting point. Scaling down the default penalty stiffness (which is higher than the initial penalty stiffness in the **nonlinear penalty stiffness** option) by a couple of orders of magnitude can also improve convergence without causing significant contact penetration.

In other cases, soft contact using the Exponential enforcement method can help in achieving convergence. When using this method, an initial contact pressure and clearance is needed. This can be obtained by running a simulation with default hard contact and noting the maximum contact pressure (CPRESS) in the first increment when contact is fully established. A setting of 10% of typical element edge length can be considered for clearance.

Abaqus will perform severe discontinuity iterations (SDIs) when open-close changes in contact occur or when stick-slip changes occur due to friction. By default, Abaqus/Standard will continue to iterate until the severe discontinuities are sufficiently small (or no severe discontinuities occur) and the equilibrium (flux) tolerances are satisfied. This is equivalent to using keyword, ***STEP, CONVERT SDI=YES**. This is most preferable. Alternatively, a different approach in which Abaqus/Standard will continue to iterate until no severe discontinuities occur can be chosen with ***STEP, CONVERT SDI=NO**. Forcing the iteration process to continue until no severe discontinuities occur is the more traditional method, but may lead to convergence issues particularly in large problems with many contact points. In such cases, excessive iteration may occur and convergence may not be obtained.

In rare cases, when all other techniques to obtain contact convergence fail, the default penetration tolerance can be relaxed with the fourth entry on ***CONTROLS, PARAMETERS=CONSTRAINTS** keyword.

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

A more comprehensive treatment of the subject is in the Abaqus advanced class of "Contact and Convergence: Successful Strategies."

7.5. Documentation References

- ▶ Abaqus 2016
 - ▷ "Defining general contact interactions in Abaqus/Standard," Section 36.2.1 of the Abaqus Analysis User's Guide
 - ▷ "Defining general contact interactions in Abaqus/Explicit," Section 36.4.1 of the Abaqus Analysis User's Guide
 - ▷ "Defining contact pairs in Abaqus/Standard," Section 36.3.1 of the Abaqus Analysis User's Guide
 - ▷ "Contact pressure-overclosure relationships," Section 37.1.2 of the Abaqus Analysis User's Guide
 - ▷ "Adjusting contact controls in Abaqus/Standard," Section 36.3.6 of the Abaqus Analysis User's Guide
- ▶ Abaqus 2017
 - ▷ Abaqus | Interactions | Defining Contact Interactions | Defining general contact in Abaqus/Standard
 - ▷ Abaqus | Interactions | Defining Contact Interactions | Defining general contact in Abaqus/Explicit
 - ▷ Abaqus | Interactions | Defining Contact Interactions | Defining contact pairs in Abaqus/Standard
 - ▷ Abaqus | Interactions | Contact Property Models | Mechanical contact properties | Contact pressure-overclosure relationships
 - ▷ Abaqus | Interactions | Defining Contact Interactions | Defining contact pairs in Abaqus/Standard | Adjusting contact controls in Abaqus/Standard

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

8. Analysis Techniques

8.1. Analysis Procedures

To solve for static equilibrium, the static procedure denoted by the ***STATIC** keyword in Abaqus/Standard is the first option for stent analysis. Since the stent undergoes large deformation, the NLGEOM option must be turned on (**NLGEOM=YES**) for the step. In some cases, when a converged solution cannot be achieved in Abaqus/Standard due to dynamic instabilities or other non-linearities (e.g., complex contact conditions) Abaqus/Explicit can be used. The dynamic procedure in Abaqus/Standard can also be used, if the non-linearities are mild, by using the 'quasi-static' option. Typically, multiple steps are needed for a full stent analysis starting from shape-setting to fatigue.

8.2. Import Analysis

An import analysis is needed when the analysis needs to be continued using a different solver, say from Abaqus/Explicit to Abaqus/Standard (in some cases the analysis may be continued using the same solver). In cases where the stent geometry is complex with large and discontinuous contact, Abaqus/Standard may not converge or may have extremely slow convergence, e.g. in a shape-setting analysis. In such cases, the shape-setting analysis can be carried out in Abaqus/Explicit while the deployment and fatigue analysis with pulsatile loading can be carried out in Abaqus/Standard. This can be achieved by doing an import analysis from Abaqus/Explicit to Abaqus/Standard. The ***IMPORT** keyword is required in the import analysis input file.

8.3. Restart Analysis

A restart analysis is required when an Abaqus job might fail, or some additional steps are needed in continuation of the original analysis. A common scenario requiring a restart analysis technique in stent analysis is doing additional cycles of fatigue. In some cases, it can be concluded that the number of cycles in the original analysis were not enough, and additional fatigue cycles can be performed in the restart analysis.

The output database (.odb) file created for a restart analysis is different than the original analysis odb. The **abaqus restartjoin** utility appends an output database (.odb) file produced by a restart analysis of a model to the output database produced by the original analysis of that model. Combining the original and restart output database files into a single file enables the user to examine all of the output data for the analysis in Abaqus/CAE.

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

8.4. Annealing

Shape-setting of stents is done with an annealing procedure. In the annealed state, the stent (after being deformed to the appropriate size) is made stress- and strain-free and is ready for deployment. Depending on the solver used, annealing is done in different ways in Abaqus:

- ▶ Abaqus/Standard

The process of annealing is achieved by using the utility ***MODEL CHANGE** through which the elements are removed during the analysis and “added” back in the analysis in a strain-free condition. This requires two analysis steps – one for removal of elements and the other for addition of the elements back into the analysis.

- ▶ Abaqus/Explicit

The process of annealing is simpler when done in Abaqus/Explicit. The use of the keyword ***ANNEAL** in a step transforms the stent into a stress- and strain-free state.

8.5. Documentation References

- ▶ Abaqus 2016
 - ▷ "Transferring results between Abaqus/Explicit and Abaqus/Standard," Section 9.2.2 of the Abaqus Analysis User's Guide
 - ▷ "Restarting an analysis," Section 9.1.1 of the Abaqus Analysis User's Guide
- ▶ Abaqus 2017
 - ▷ Abaqus | Analysis | Analysis Techniques | Analysis Continuation Techniques | Importing and transferring results | Transferring results between Abaqus/Explicit and Abaqus/Standard
 - ▷ Abaqus | Analysis | Analysis Techniques | Analysis Continuation Techniques | Restarting an analysis

9. Fatigue Simulation

Once the desired shape of the stent is obtained, it is deployed in the artery through a delivery system. In the process, the stent is crimped and then expanded to the size of the inner diameter of the artery. Usually, this is less than the outer diameter of the unconstrained stent, and the stent is referred to as 'oversized.' After deployment, the stent usually undergoes pulsatile loading coming from the oscillating pressure of the blood flow. Due to this in-vivo pulsatile loading, estimation of fatigue life for stents becomes crucial, and FEA can be used for prediction of its fatigue life.

If the shape setting was done in Abaqus/Explicit, the model can be imported in Abaqus/Standard to perform the fatigue simulations (after doing crimping and deployment). The fatigue simulations are carried out by applying cyclic displacements in the radial direction from its position of deployment. Usually 3 – 5 fatigue simulation cycles should be carried out so that the fatigue strains can stabilize. Strains are important in characterization of fatigue. One cycle will correspond to 2 analysis steps for both extrema positions. Deformable cylinders as described in Chapter 6 can be used to expand and crimp to the two extrema positions. Separate cylinders are used for expansion and crimping. Based on stabilized fatigue strains, fatigue life prediction can be done.

Consideration should be made for the fact that displacements in Abaqus are always specified as total displacement instead of incremental displacement, and that they refer to the original coordinate system which was specified as part of the model data. More than one coordinate system cannot be used on the same nodes to specify the boundary conditions in one analysis. If due to the previous analyses, it is not possible to accurately define total displacements in the fatigue analysis, velocity boundary conditions may be specified in order to produce the desired incremental displacements. It can be defined as follows:

***BOUNDARY, TYPE=VELOCITY**

The built-in Nitinol model can be used to estimate the fatigue safety factor. For elastic-plastic materials, material models with kinematic hardening should be used.

9.1. Documentation References

- ▶ Abaqus 2016
 - ▷ "Boundary conditions in Abaqus/Standard and Abaqus/Explicit," Section 34.3.1 of the Abaqus Analysis User's Guide
- ▶ Abaqus 2017

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

- ▷ Abaqus | Prescribed Conditions | Boundary Conditions | Boundary conditions in Abaqus/Standard and Abaqus/Explicit

10. Post-Processing

The post-processing is done in the usual way as for any other Abaqus model. Usually, the interest in stent results is the location of maximum principal stresses and strains. If the built-in subroutine of Nitinol is used for simulation all major Nitinol results are reported in terms of SDVs as described in Section 3. Contour plots of all SDVs are available for Visualization. For Abaqus 2017 and later the Nitinol subroutine has been made part of standard Abaqus material library, and therefore, the SDVs are reported as standard output variables as described in section 3.

10.1. Fatigue

The complication in post-processing comes when safety under fatigue loading is to be determined, which is usually the case for stents. Today the Goodman diagram is the most common approach for fatigue analysis of Nitinol stents. In most cases, a point cloud of alternating stress/strain and mean stress/strain is used. Any point above the Goodman curve is an indication of a potential location for fracture. Point cloud needs to be created for all nodes in the model and therefore the mean and alternating stress/strain calculation needs to be performed at all nodes. There needs to be a method to represent triaxial states of stress/strain into a scalar equivalent. These calculations can be performed by 3 methods – Scalar Method, Tensor Method and Modified Tensor Method. The Scalar Method is based on calculation of Maximum Principal stresses/strains at load extrema in fatigue cycles; the Tensor Method is based on first calculating the mean and alternating stress/strain tensors at load extrema in fatigue cycles and then calculating the Maximum Principal stresses/strains; the Modified Tensor Method differs from the Tensor Method due to the fact that it takes a projection of mean stress/strain tensor in the direction of Maximum Principal alternating stress/strain tensor, and so the directions of alternating and mean stress/strain tensors are aligned.

Due to its complex transformation and shape-memory behavior, characterization of the fatigue behavior of Nitinol is not a settled question in the Nitinol community. The most accepted method in the Nitinol community is the Modified Tensor Method which gives predictions closer to experimental observations. The Scalar Method is the more traditional method that works reasonably if the cyclic loading is all tensile and the direction of pre-load and cyclic loads are aligned. It is found not to be suitable for general loading and cyclic loading conditions. To perform the calculations using both methods, a simple set of instructions is provided below. The instructions are provided using strain notation but is equally applicable for stress. The instructions can be automated with a Python script.

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

For metals and common alloys other than Nitinol, fatigue life estimates can be obtained by using the SIMULIA product **fe-safe**, which is standalone fatigue software and offers a comprehensive suite of methods and algorithms for calculating fatigue life. A Low Cycle Fatigue analysis can be run using Abaqus alone, but for estimating high cycle fatigue life it is best to use **fe-safe**.

10.1.1. Scalar Method

This is the traditional method for creating point clouds to compare to the Goodman curve. It is based on calculating maximum principal values at the extremes of the load cycle and obtaining the alternating and mean strains. The method is as follows:

- ▶ Compute the maximum principal strain for the first extreme (p) of the load cycle.
- ▶ Compute the maximum principal strain for the second extreme (q) of the load cycle.
- ▶ Compute the cyclic strain amplitude from the principal values:

$$\varepsilon_a = \frac{(\varepsilon_1)_p - (\varepsilon_1)_q}{2}$$

- ▶ Compute the cyclic mean strain from the principal values:

$$\varepsilon_m = \frac{(\varepsilon_1)_p + (\varepsilon_1)_q}{2}$$

Once the alternating and mean strains are obtained, they are plotted and compared to Goodman diagram for estimating fatigue safety.

10.1.2. Modified Tensor Method

The Modified Tensor Method described below uses **boldface** notation for tensors. It is based on considering the full strain tensors at the extremes ('p' and 'q'); calculating alternating and mean strain tensors from them; calculating max alternating principal strains; and then doing the appropriate projection of mean tensor to align with principal directions of maximum alternating principal strains. The method is as follows:

- ▶ Compute the mean strain tensor by adding the strain tensors from the first and second extremes of the cyclic load and dividing by 2:

$$\boldsymbol{\varepsilon}_m = \frac{\boldsymbol{\varepsilon}_p + \boldsymbol{\varepsilon}_q}{2}$$

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

- ▶ Compute the alternating strain tensor by subtracting the second tensor from the first tensor and dividing by 2:

$$\boldsymbol{\varepsilon}_a = \frac{\boldsymbol{\varepsilon}_p - \boldsymbol{\varepsilon}_q}{2}$$

- ▶ Compute the eigenvalues (principal strains) and eigenvectors (principal directions) of the alternating strain tensor. The magnitudes of the alternating principal strains P_{alt-1} , P_{alt-2} , P_{alt-3} represent cyclic strain amplitudes for performing fatigue evaluations.
- ▶ Use the principal directions (eigenvectors) $\{\mathbf{n}_i\}$ obtained from the alternating strain tensor to transform the mean strain tensor into a coordinate system which is aligned with the alternating tensor principal strain directions. It can be computed as:

$$\boldsymbol{\varepsilon}_{m-ii} = \{\mathbf{n}_i\}^T [\boldsymbol{\varepsilon}_m] \{\mathbf{n}_i\}$$

The 1-1 component of the transformed mean strain tensor is aligned with the alternating tensor P_{alt-1} principal strain; likewise, the transformed mean strain tensor 2-2 and 3-3 components are aligned with the alternating tensor P_{alt-2} and P_{alt-3} .

- ▶ Generate data pairs consisting of the alternating tensor principal strain magnitudes and their associated transformed mean strain $\boldsymbol{\varepsilon}_m$ components.

$$(\boldsymbol{\varepsilon}_{m-11}, |P_{alt-1}|); (\boldsymbol{\varepsilon}_{m-22}, |P_{alt-2}|); (\boldsymbol{\varepsilon}_{m-33}, |P_{alt-3}|)$$

Plot the data pairs as points on a scatter plot where the abscissa represents the mean strain and the ordinate represents the alternating strain. Superimpose the Goodman diagram for the safety assessment.

Note that this procedure must utilize extreme cyclic load strain tensors whose components are output with respect to a set of local material axes that travel with the stent structure while undergoing large displacements and rigid body rotations.

10.1.3. Fatigue Safety Factor

The Fatigue Safety Factor (FSF) can be calculated in different ways. There are different methods prevalent in the industry; two common methods are mentioned here.

- ▶ In the scatter point cloud approach as described above, the FSF can be defined as the ratio of the length of the line from the origin to the Goodman curve

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

passing through the given point to the length of the line from origin to the given point in the scatter plot. Thus for each node FSF can be calculated as follows:

$$FSF = \frac{l_{Origin-ScatterPoint-GoodmanCurve}}{l_{Origin-ScatterPoint}}$$

- Many times, most part of the Goodman curve for Nitinol in strain is flat, parallel to the mean-strain axis. In that case, FSF can be calculated by the following form:

$$FSF = \frac{Fatigue\ strain\ limit}{Alternating\ strain}$$

where,

Fatigue strain limit = Ordinate of the point on the Goodman curve closest to the highest alternating strain point of interest

Alternating strain = Alternating strain value

In case the curve is not parallel to the mean strain axis, the point on the Goodman curve can be determined by dropping a perpendicular from the alternating strain point of interest on to the Goodman curve.

- ▶ If the calculations are done with stress instead of strain, FSF can also be calculated by the following form:

$$FSF = \frac{\sigma_E \sigma_U}{\sigma_{alt} \sigma_U + \sigma_E \sigma_{mean}}$$

where, σ_E is the endurance limit and σ_U is the ultimate strength.

It should be noted that the above definition is valid only if the Goodman curve is a straight line from σ_E to σ_U .

A sample strain point cloud plot and Goodman curve [6] is shown in Fig 10.1 below. This is only for demonstration purpose and is not related to any real data. It shows that all points considered (red scatter) are calculated are safe. The Goodman curve shown in Fig 10.1 is the most widely accepted curve in the Nitinol community. The curve is also referred to as Fatigue Limit Curve in the community.

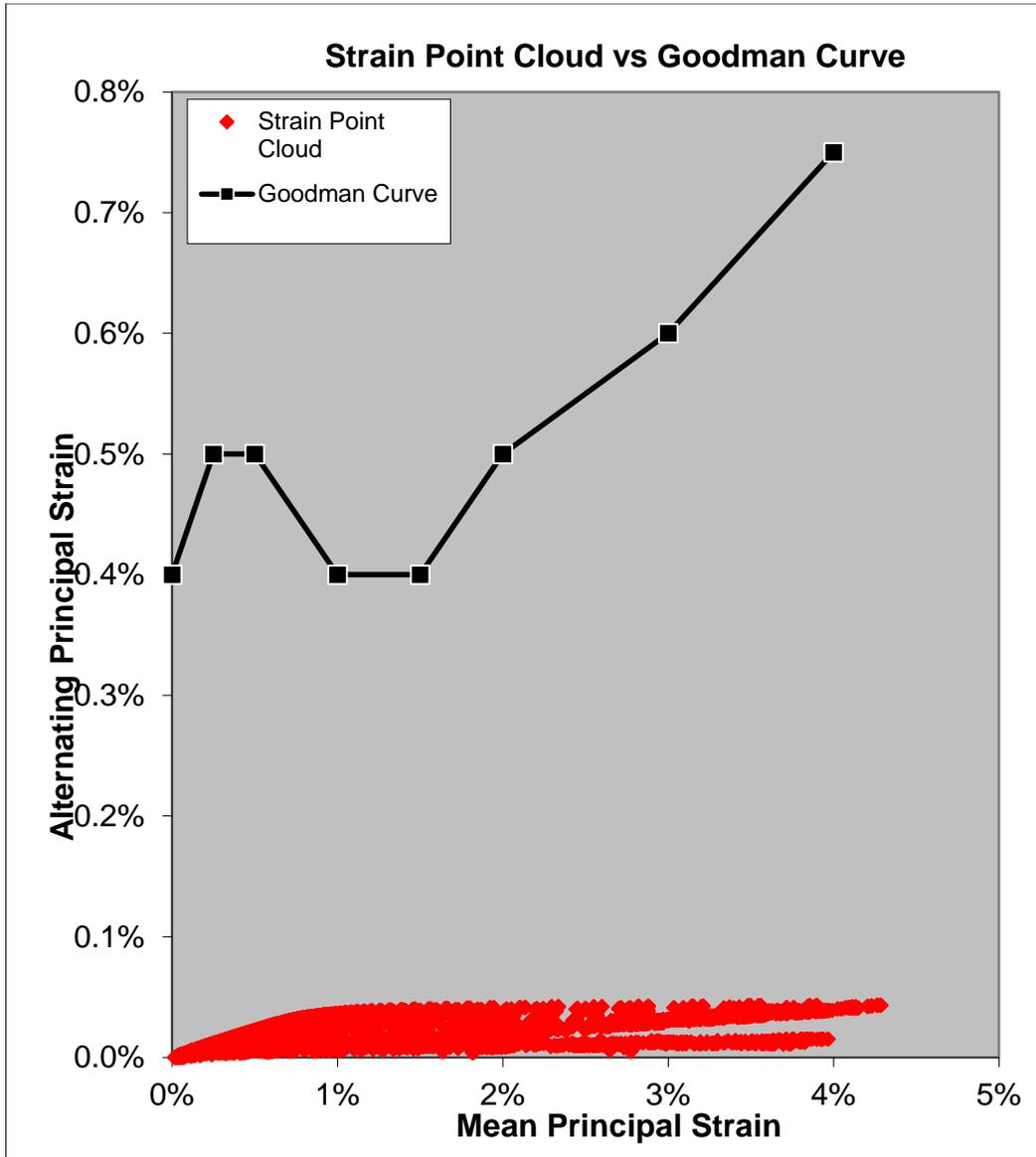


Fig 10.1: Sample strain point cloud plot with Goodman curve [6]

10.1.4. Using fe-safe

Fatigue *life* can be estimated with **fe-safe** for common metals, alloys and rubber under various cyclic loading service conditions after FEA analysis is done. It can read Abaqus, Ansys, Nastran output results files and post-process them for fatigue life calculations providing the worst location for fatigue life. Some common methods fe-safe offers are Brown-Miller, Smith-Watson-Topper (SWT), Dang Van, Haigh diagram, Buch etc. With some restrictions, the methods are available for both elastic and elastic-plastic FEA results. Both stress and strain fatigue lives can be obtained, along with traditional Miner's rule, Goodman and Gerber diagrams etc. It also provides a contour plot of fatigue life on the model for easy visualization.

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

It should be noted that most of these methods apply to metals and common alloys. Nitinol is distinct in the sense that due to martensite to austenite transformation, the fatigue life will depend on which phase is active due to loading. Caution should be exercised in calculating fatigue life for Nitinol as martensitic phase and austenitic phase can have different lives. There is no settled standard form for estimating life of Nitinol.

10.1.5. Fatigue Life for Nitinol

Recently, SIMULIA (Abaqus and fe-safe) has been working on developing a method to predict life for Nitinol fatigue based on published experimental results, e.g. [7] and getting inputs from the medical device industry. For the ease of the user, a plugin will be provided that will perform the life calculations based on the method. The plugin is not yet available but will be made available in a year.

10.2. Documentation Reference

- ▶ Abaqus 2016
 - ▷ Abaqus Scripting User's Guide
 - ▷ Abaqus Scripting Reference Guide
 - ▷ fe-safe User Guide
- ▶ Abaqus 2017
 - ▷ Abaqus | Scripting
 - ▷ Abaqus | Scripting Reference
 - ▷ fe-safe User Guide

11. Reports

As to how FEA modeling should be used in the overall stent design, and how to use the FEA results in the FDA submission process, please consult the guidelines provided by concerned regulatory agencies for example:

- ▶ *"Reporting of Computational Modeling Studies in Medical Device Submissions"*, FDA, Office of Science and Engineering Laboratory, September 21, 2016
- ▶ *"Standard Guide for Finite Element Analysis (FEA) of Metallic Vascular Stents Subjected to Uniform Radial Loading"*, ASTM F2514-08 (2014)

12. Appendix 1

12.1. Model Details

The details of the example model used in this Best Practices document are described below. The example is for demonstration purposes only and should not be considered as suitable for validation purposes.

- ▶ Parts
 - ▷ Stent: 3D deformable orphan mesh
 - ▷ Crimper: 3D deformable shell geometry by revolution
 - ▷ Expander: 3D deformable shell geometry by revolution
- ▶ Materials
 - ▷ ABQ_SUPER_ELASTIC_Stent
 - *User Material
- ▶ Sections
 - ▷ Stent:
 - Type: Solid, Homogeneous
 - Material: ABQ_SUPER_ELASTIC_Stent
 - ▷ Crimper_Expander:
 - Type: Surface
 - Material: (None)
- ▶ Assembly
 - ▷ Instances
 - Stent
 - Crimper
 - Expander
- ▶ Steps
 - ▷ Expansion
 - ▷ Annealing_RemoveMesh
 - ▷ Annealing_AddMesh
 - ▷ Crimp
 - ▷ Deployment
 - ▷ Diastolic1
 - ▷ Systolic1
 - ▷ Diastolic2
 - ▷ Systolic2
 - ▷ Diastolic3
 - ▷ Systolic3

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

► Loads

- ▷ Symm-Radial: restrained the circumferential displacements of the stent nodes at the cyclic symmetrical plane.
- ▷ Symm-Z: restrained z-displacement of the stent nodes at the X-Y symmetrical plane.
- ▷ Expand: Apply the radial displacement for all expander nodes in "Expansion" step.
- ▷ Crimp: Apply the radial displacement for all crimper nodes to simulate stent crimping, deployment, diastolic and systolic for three cycles.

Boundary Condition Manager													
Name	Initial	Expansion	Annealing	Annealing	Crimp	Deployment	Diastolic1	Systolic1	Diastolic2	Systolic2	Diastolic3	Systolic3	
✓ Crimp					Created	Modified							
✓ Expand		Created	Propagatec	Propagatec	Inactive								
✓ Symm-Radial	Created	Propagatec											
✓ Symm-Z	Created	Propagatec											

Fig 12.1: Boundary condition manager

► Interactions

- ▷ Stent2Expander_Contact: Surface-to-surface contact between stent and expander in "Expansion" step only.
- ▷ Stent_Annealing: Deactivate stent part in "Annealing_RemoveMesh" step with ***Model Change** and reactivate it with zero stress and strain in "Annealing_AddMesh" step.
- ▷ Expander_Removal: Deactivate the expander part in "Annealing_AddMesh" with ***Model Change**.
- ▷ Stent2Crimper_Contact: Surface-to-surface contact between stent and crimper from "Crimp" to the end.
- ▷ Stent-SelfCont: Stent self-contact beginning from "Crimp" to the end.

Interaction Manager													
Name	Initial	Expansion	Annealing_Rem	Annealing_Add	Crimp	Deployment	Diastolic1	Systolic1	Diastolic2	Systolic2	Diastolic3	Systolic3	
✓ Expander_Removal					Created	Propagati	Propagated	Propagate	Propagatec	Propagatec	Propagatec	Propagatec	
✓ Stent-SelfCont					Created	Propagated	Propagate	Propagatec	Propagatec	Propagatec	Propagatec	Propagatec	
✓ Stent2Crimper_Contact					Created	Propagated	Propagate	Propagatec	Propagatec	Propagatec	Propagatec	Propagatec	
✓ Stent2Expander_Contact	Created	Propagatec	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	
✓ Stent_Annealing			Created	Modified	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	Inactive	

Fig 12.2: Interaction manager

► Mesh

- ▷ Refinement
- ▷ Type of element:
 - Crimper: SFM3D4R
 - Expander: SFM3D4R
 - Stent: C3D8I

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

Best Practice

- ▶ Input file
 - ▷ Name: "Stent-Fatigue.inp"
- ▶ CAE file
 - ▷ Name: "Stent-Fatigue.cae"
 - ▷ Model Name: "Stent-Fatigue"

The example input and CAE files are available for reference in the Knowledge Base Q&A QA00000053987 at <https://www.3ds.com/support/knowledge-base/>.

The mesh refinement study was done using the expansion step of the stent in a separate analysis.

12.2. Hardware

- ▶ Processor: Nehalem X5650@2.67GHz
- ▶ Nodes: 4 with 24 GB memory per node
- ▶ Cores: 48
- ▶ OS Version: Redhat 6.3

12.3. Software

- ▶ Abaqus Version: Abaqus 2016.FP1613
- ▶ 3DEXPERIENCE Version: 3DEXPERIENCE 2017x.FP1705

13. Appendix 2

13.1. Writing Abaqus Input File from 3DEXPERIENCE Platform

Once the FEA model is ready on the 3DEXPERIENCE platform, the Abaqus input file can be written out from it following the steps below:

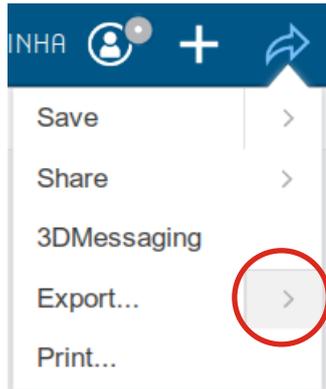
- ▶ Go to the top right hand corner and select the arrow icon:



- ▶ Click on the arrow in front of **Export** menu option, be careful not to click on **Export**:

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.



- ▶ Click on **Export Simulation** menu option.
- ▶ In the **Export Simulation** window, click on File Selector and choose format **Abaqus input file**.

After successful export of the Abaqus input file, the input file can be modified to include keywords for inclusion of Nitinol material. Other section property keyword also needs to be updated.

Note:

- ▶ If there are errors in the simulation set up, the export will not work and will give an error.
- ▶ If there is a problem in scenario definition, then instead of Export Simulation, **Export FEM** option will appear after clicking on the arrow in front of **Export** menu option. Selecting the **Export FEM** option will only export nodes and elements in the Abaqus input file and all other keywords will not be exported.

13.2. Running Abaqus Job on 3DEXPERIENCE Platform

Abaqus input file generated by the procedure described above can be run on the 3DEXPERIENCE platform by using **Abaqus Study** app. The app resides in the south quadrant (**V+R**) of the Compass on top left as shown in the blue highlight in Fig 14.1.

- ▶ Click on the south quadrant.
- ▶ Search **Abaqus Study** app by clicking on the magnifying glass.
- ▶ Click on **Abaqus Study** app icon that appears under 3DEXPERIENCE Roles & Apps.
- ▶ Now, you can create a new Study or drop a Study in the work area on the right as shown in the figure below:
 - In either case, you will be guided step by step to use Abaqus input file and select Abaqus solver (Fig 14.2)

- 'Add Job' is used for adding the solver

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

- Abaqus input files can be added to Managed Data under Content

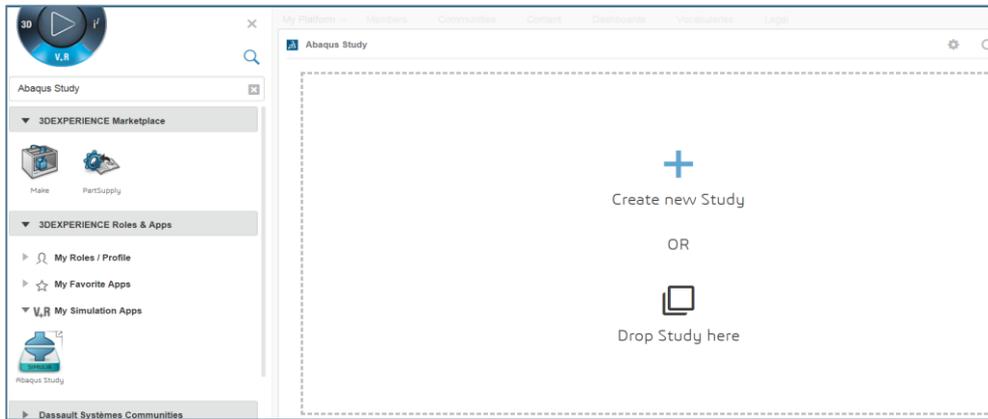


Fig 13.1: Abaqus Study app interface

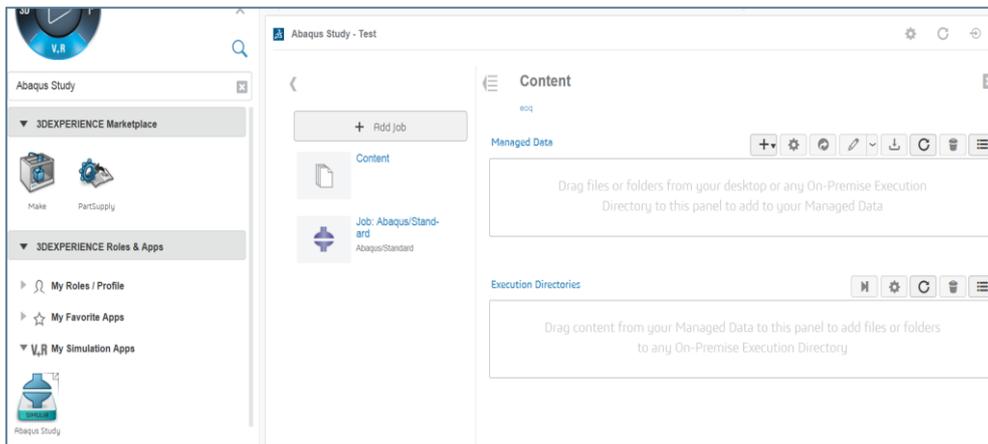


Fig 13.2: Options in Abaqus Study app

Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.

14. References

- [1] Rebelo N., Radford R., Zipse A., Schlun M., Dreher G., 2011, "On Modeling Assumptions in Finite Element Analysis of Stents," *Journal of Medical Devices*, Vol 5, pp. 031007-1 – 031007-7.
- [2] Auricchio F., and Taylor R., 1996, "Shape Memory Alloys: Modeling and Numerical Simulations of the Finite-Strain Superelastic Behavior," *Comput. Methods Appl. Mech. Eng.*, 143, pp. 175 – 194.
- [3] Auricchio F., and Taylor R., 1997, "Shape Memory Alloys: Macromodeling and Numerical Simulations of the Superelastic Behavior," *Comput. Methods Appl. Mech. Eng.*, 146, pp. 281 – 312.
- [4] Rebelo N., Gong X.-Y., and Connally M., 2004, "Finite Element Analysis of Plastic Behavior in Nitinol," *SMST – 2003 Proc. International Conference on Shape Memory and Superelastic Technologies*, Pelton A., and Duerig T., eds., SMST Society, Menlo Park, CA, pp. 501 – 507.
- [5] Rebelo N., Zipse A., Schlun, and Dreher G., 2011, "A Material Model for the Cyclic Behavior of Nitinol," *J. Mater. Eng. Perform.*, 20 (4 – 5), pp. 605 – 612.
- [6] A. R. Pelton, V. Schroeder, M. R. Mitchell, X.-Y. Gong, M., Barney and S. W. Robertson: 'Fatigue and durability of Nitinol stents', *J. Mech. Behav. Biomed. Mater.*, 2008, 1, 153–164.
- [7] Pelton, A.R., et al, 2013, "Rotary bending fatigue characteristics of medical-grade Nitinol wire", *Journal of the Mechanical Behavior of Biomedical Materials*, 27 pp. 19 – 32.
- [8] "Derived from Open Stent Design, Craig Bonsignore, Nitinol Devices & Components (NDC), <http://nitinol.com>"

15. Document History

Document Revision	Date	Revised By	Changes/Notes
1.0	2/7/2018	Shashwat Sinha, Changjian Wang	Original

Our **3DEXPERIENCE®** platform powers our brand applications, serving 12 industries, and provides a rich portfolio of industry solution experiences.

Dassault Systèmes, the **3DEXPERIENCE®** Company, provides business and people with virtual universes to imagine sustainable innovations. Its world-leading solutions transform the way products are designed, produced, and supported. Dassault Systèmes’ collaborative solutions foster social innovation, expanding possibilities for the virtual world to improve the real world. The group brings value to over 210,000 customers of all sizes in all industries in more than 140 countries. For more information, visit www.3ds.com.



Confidential Information. © [2017] Dassault Systèmes. All Rights reserved.

This document is intended for internal use only and is provided for information purpose. Any other use without the written prior authorization from Dassault Systèmes is strictly prohibited, except as may be permitted by law.