GM BoltStudio: A Suite of Extensions to Abaqus/CAE for Simulating Bolted Assemblies at General Motors

Cong Wang[1], Max Zhou[1], and Jon Dunn[2]

[1]General Motors, [2]SIMULIA

Abstract: General Motors, in conjunction with SIMULIA Great Lakes, have developed a studio of Abaqus/CAE plug-ins designed to increase the efficiency of their analysts in the definition of their model assemblies. These plug-in are written to bridge the gaps and to enhance the usability of native tools in Abaqus/CAE specifically for bolt joint assembly simulation. The main advantage of the plug-in studio is improved efficiency, both in terms of time saving and modeling accuracy/consistency. Secondary benefits include the ability to enable less experienced users leverage the complex modeling methods encapsulated by each plug-in. This paper presents a few of the developed plug-ins, describing for each the process captured. One of these tools, the 'bolt library' is available to the wider Abaqus user community.

Keywords: Bolt Loading, Scripting, Customization.

1. Introduction

Robust and efficient bolt joint design is critical to vehicle structural integrity and performance. At General Motors, CAE analysis plays an important part in guiding the design and evaluating the performance, applied not only to isolated joints, but also to vehicle subsystems with detailed modeling and bolt-preloading incorporated at the key joints. To consolidate and integrate a number of bolt joint analysis tools/practices from its CAE organizations into a single modeling/simulation environment, General Motors and SIMULIA cooperated in developing a tool suite, named GM BoltStudio, as plug-in in Abaqus/CAE for simulation by Abaqus/Standard. The objectives are to: i) streamline the workflow for best practice and productivity; ii) push these types of analysis upstream in the design process as "standard work" to be performed by less-experienced users; iii) lay a solid basis in terms of modeling quality and consistency for more advanced reliability synthesis.

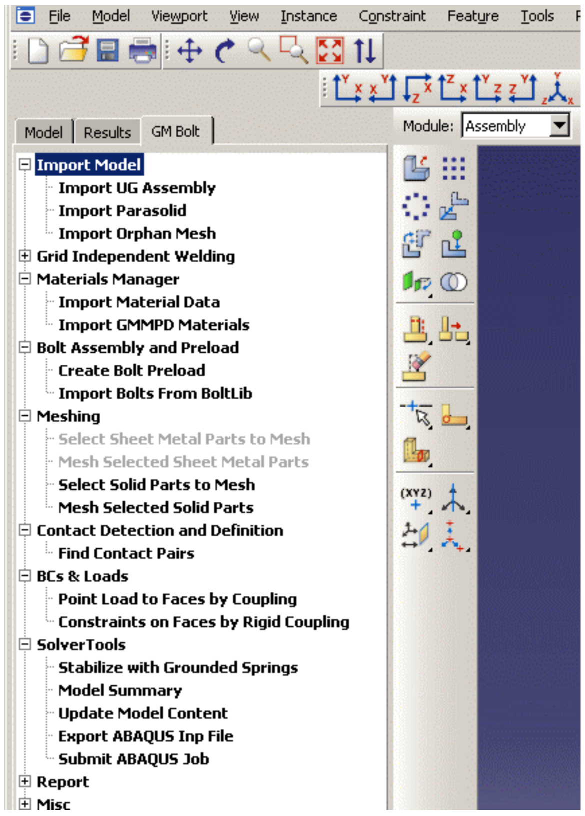
Abaqus/CAE provides general capability of pre/post-processing Abaqus models. For its application to bolt joint analysis in General Motors, Abaqus/CAE still has some functional gaps (for example, lack of direct interface for UG import), and a number of its native tools are somewhat labor-intensive or not intuitive to use. Nevertheless, GM chose Abaqus/CAE as the basis to consolidate the CAE bolt assembly simulation workflow mainly for the following reasons:

- Flexibility that enables modeling practices based on geometry objects (faces/edges/cells, etc.) and/or on meshed entities.
- Capability for seamless incorporation of user developed plug-in modules to bridge the functional gaps and customize for special workflow and analysis tasks.
- Strong support in scripting and GUI toolkit interfaces and the extensive accessibility to kernel functionality for improving the usability.
- Capability to post-process Abaqus output database augmented with additional user defined result variables and to customize plotting style.

This paper presents a few of these plug-ins and, for some, describes the process required to define the same modeling behavior using the native Abaqus/CAE tools.

2. Overview of GM BoltStudio

The GM BoltStudio was structured as a recommended workflow which strings together a sequence of tasks supported by customized tool modules. This whole suite was integrated as a high level plug-in and is shown in Figure 1.



- Import Model
  - Import UG Assembly
  - Import Parasolid
  - Import Orphan Mesh
- Grid Independent Welding
- Materials Manager
  - Import Material Data
  - Import GMMPD Materials
- Bolt Assembly and Preload
  - Create Bolt Preload
  - Import Bolts From BoltLib
- Meshing
  - Select Sheet Metal Parts to Mesh
  - Mesh Selected Sheet Metal Parts
  - Select Solid Parts to Mesh
  - Mesh Selected Solid Parts
- Contact Detection and Definition
  - Find Contact Pairs
- BCs & Loads
  - Point Load to Faces by Coupling
  - Constraints on Faces by Rigid Coupling
- SolverTools
  - Stabilize with Grounded Springs
  - Model Summary
  - Update Model Content
  - Export ABAQUS Inp File
  - Submit ABAQUS Job
- Report
- Misc

Figure 1. The GM BoltStudio plug-in and task tree

When invoked, the GM BoltStudio plug-in will appear as an additional tab to the Model Tree/Results Tree tab book. The workflow and tasks are displayed in a tree structure within this additional tab. The tasks are arranged in a specific order to enable less-experienced users to 'walk through' the process of setting up a simulation by invoking each tool modules in the tree in order. The custom tool modules are written to bridge the gaps and to improve the usability of native tools in Abaqus/CAE specifically for bolt joint assembly simulation. These enhancements include:

1. Provide utilities for rapid downloading and importing parts/assemblies from the GM corporate CAD/PDM (UG NX/TcAE) system, and also for flattening the multi-level CAD assembly and mapping into the simple assembly data structure in Abaqus/CAE.

2. Capture and re-associate inside Abaqus/CAE the key meta data from original CAD/PDM for tracking the pedigree of simulation, and properly sync the name fields of parts, part instances, sections, material properties, …, etc.

3. Automatically create appropriate section properties to root parts and assign sanctioned material properties from GM corporate material database.

4. Provide rapid (native) auto meshing capability for solids.
5. Integrate CAE batch meshing and welding tools existing in GM and provide them as options to native meshing and welding in Abaqus/CAE.
6. Provide pre-defined and parameterized bolt library and positioning tools to expediently create and place bolt/nut/washer into the joints.
7. Enable rapid bolt pre-loading definition to bolt solids that are imported from the original CAD assembly.
8. Apply the "Find contact pairs" tool in A/CAE together with pre-defined interaction properties/options for precise and fast contact/constrain definition.
9. Provide stabilizing measures to take out the undesirable initial "free play" in the model.
10. Automate the loads, BC, output definitions.
11. Provide more interpretative and readable report from the ".msg" file to guide user eliminating the potential modeling quality issue (initial penetration, conflict in constraints, etc.).
12. Enforce appropriate solver settings to ensure robustness and accuracy of the solution.
13. Provide spreadsheet-style model summary report for peer/management review.
14. Augment the Abaqus output database with additional GM customized output variables and re-group output variables to support various plotting styles.
15. Provide customized contour plotting styles for more insightful or intuitive viewing of results.

A number of the modules were developed in-house at General Motors, whereas others were developed by SIMULIA Great Lakes and subsequently extended by General Motors for functionality enhancement and seamless incorporation into the GM BoltStudio workflow.

3. Plug-ins

This section describes in detail some of the developed plug-ins within the toolbox.

3.1 Create bolt pre-loading on solid geometry

General Motors routinely simulate sub-assemblies of components. Typically, these sub-assemblies contain multiple parts which are connected using studded or bolted connections. In some cases detailed CAD representations of bolt/stud are available in the assembly and in others they are not. In cases where bolt geometry is available and appropriate for CAE modeling, a plug-in is provided in the task "Create Bolt Preload" to rapidly partition the bolt shank geometry and specify bolt pretension, as shown in Figure 2 and Figure 3.

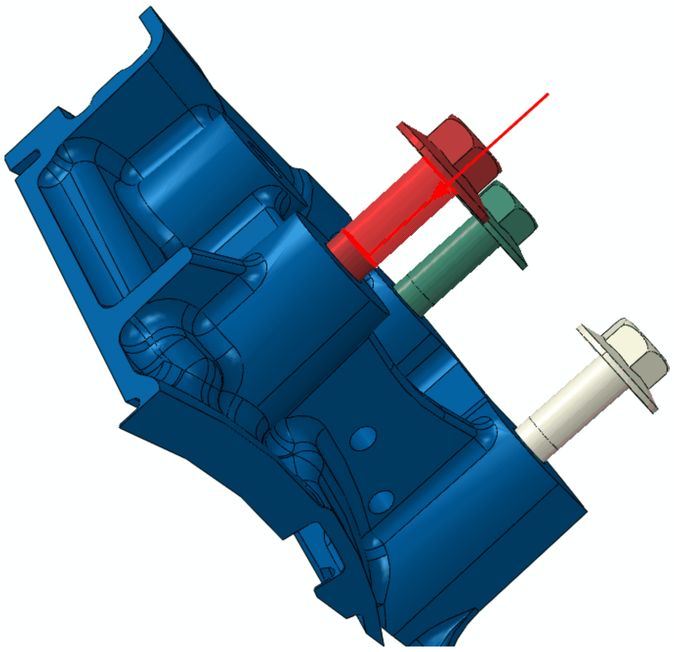
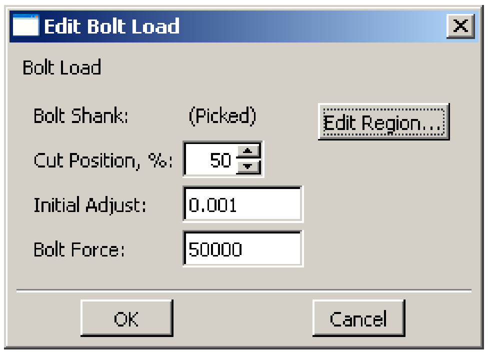


Figure 2. Select the bolt shank face for pre-load



Bolt Shank: (Picked)
Cut Position, %: 50
Initial Adjust: 0.001
Bolt Force: 50000

Figure 3. Define bolt pre-load

Tools exist in native Abaqus/CAE to define bolt loading. However, the process of defining the bolt loading requires a number of steps that span multiple modules. A summary of the required steps is given below:

1. Partition the bolt across the shank to define the loading surface.
2. Define a datum axis to represent the bolt axis.
3. Create step definitions for the initial clamping, full clamping and bolt length fixing steps
4. Specify the bolt loading in each of these steps.

The “Create Bolt Preload” plug-in (see Figure 3) allows the user to specify the bolt loading with significantly less user interaction. First, the user selects a face on the bolt shank. If required, the cut position can be modified from the default value of 50%. The cut position defines the point along the face in the direction represented by the glyph (as seen in Figure 2) at which the bolt will be partitioned across the shank. The initial adjustment and full blot pre-load are also specified.

When a bolt preload is constructed, default analysis steps are created in Abaqus/CAE, if they do not already exist. The created steps are:

Step 1) Initialization	➔ Perform initial preparation to get preloading started
Step 2) BoltPreload	➔ Complete the preloading
Step 3) ServiceLoad	➔ Fix the bolt length and apply the service loads
Step 4 …)	➔ Continued steps for further service loading

The “initial adjust” value is intended for applying low level bolt preloading in Step 1 under displacement control. This step is added in order to remove any initial ‘free play’ in the assembly and resolve initial penetration in the model in order that the analysis can get started robustly. For models such as clevis joints, the initial value should be overwritten by the clearance in the initial geometry.

3.2 Bolt library

In General Motors design practice, the CAD representation of the bolt is usually for packaging/assembly/tooling study. The geometry tends to have very detailed features, overly-complicated for the purpose of the Finite Element (FE) analysis. Rapid auto meshing of such bolt geometry would lead to highly varied element size and distorted element shapes as well as high DOF count. Such mesh can lead to undesirable consequences in CAE solution robustness/performance and results interpretation.

In CAE practice, General Motors prefer to simplify modeling assumptions for the studs/bolts used in these assemblies. These assumptions included modeling the bolt and, where applicable, nut and washer as parametric revolved solids. By making this assumption the bolt could then be meshed using modest sized first-order hexahedral elements, thus dramatically reducing the DOF count for a single bolt. This reduction in DOF count yields a significantly smaller FE model in assemblies that contain many studs or bolts. More importantly, first-order hexahedral mesh is more accommodating and accurate in further reliability synthesis of the joints performance.

The decision was made to create a plug-in that provided the capability to incorporate a library of standard parameterized bolts together with a simplified set of tools to position the bolt/stud in the assembly and to also apply the bolt pre-load. Without automating this process, the following workflow in Abaqus/CAE would be required:

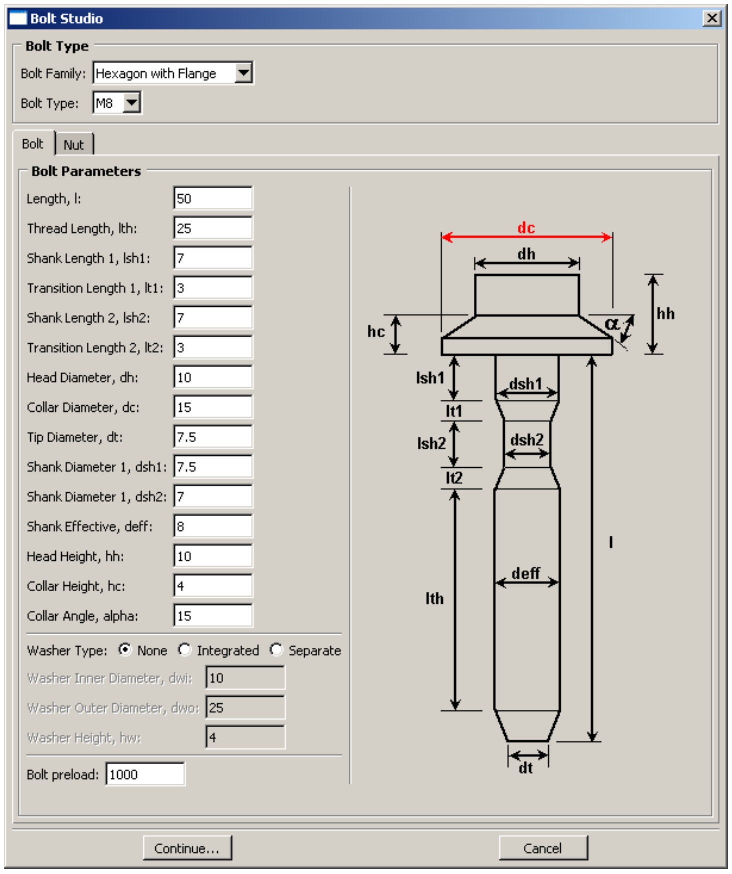
1. Create the bolt/stud part. This would be created as a revolved part.
2. Create, if applicable, the nut part. Again, this would be created as a revolved part.
3. Partition the bolt/stud and nut in order to be able to mesh with hexahedral elements. Also partition the bolt/stud at a point along the shank to apply the pre-tensioning load to.
4. Define a datum axis along the bolt/stud to use during the pre-tensioning load definition.
5. Mesh the parts.
6. Define surfaces – in particular the internal surface require for the pre-tensioning definition.
7. Create instances of the parts in the assembly.
8. Use the native positioning tools to move the bolt/stud and, if applicable, nut to the correct location.
9. Create step definitions (if they do not already exist) for the pre-tensioning loading.
10. Define the bolt loading. This consists of a pre-tensioning force being applied in the first step, followed by the bolt length being fixed in the proceeding step.
11. Repeat steps 7 to 10 for each bolt of this type in the assembly.

Whilst this manual approach to adding bolts/studs to the assembly may be acceptable for occasional model building, the method becomes very time consuming when multiple bolts/studs need to be added to an assembly.

Another drawback is that some of the positioning tools available within Abaqus/CAE only work when native geometric parts are used to form the constraints. In some cases at General Motors, the parts are imported into Abaqus/CAE as orphan meshes that are generated by other FEM tools that General Motors employs or provide by part suppliers. The orphan meshes may not always be closely associated with the native geometry of the parts in the assembly for various reasons. Without the re-association of orphan meshed parts with native geometry, the positioning tools native to Abaqus/CAE are not sufficient to easily position the bolts/studs and nuts in the assembly. It was therefore critical for the plug-in to also allow for the easy positioning of bolts into an orphan mesh assembly.

There are also other drawbacks to the manual method. These include potential mistakes and frequent rework by inexperienced user to perform the modeling.

Figure 4 shows the main dialog box of the plug-in.



Bolt Type

Bolt Family:	Hexagon with Flange
Bolt Type:	M8

Bolt Parameters

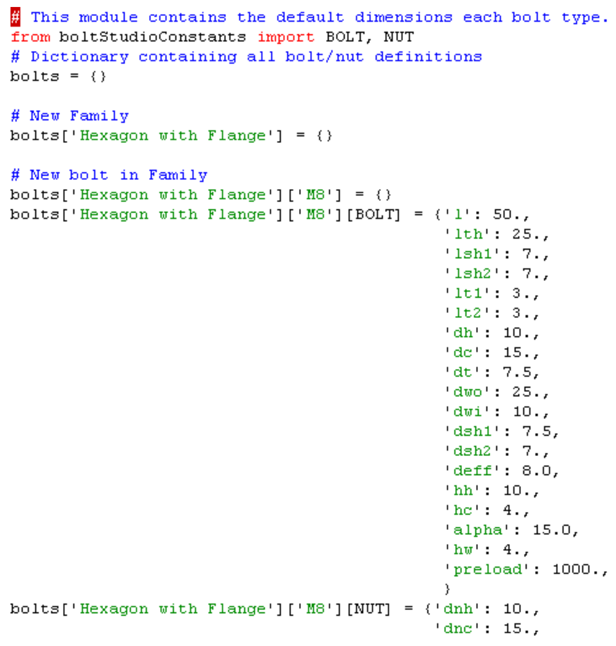
Length, l:	50
Thread Length, lth:	25
Shank Length 1, lsh1:	7
Transition Length 1, lt1:	3
Shank Length 2, lsh2:	7
Transition Length 2, lt2:	3
Head Diameter, dh:	10
Collar Diameter, dc:	15
Tip Diameter, dt:	7.5
Shank Diameter 1, dsh1:	7.5
Shank Diameter 1, dsh2:	7
Shank Effective, deff:	8
Head Height, hh:	10
Collar Height, hc:	4
Collar Angle, alpha:	15

Washer Type: [✓ None] Integrated Separate

Washer Inner Diameter, dwi:	10
Washer Outer Diameter, dwo:	25
Washer Height, hw:	4
Bolt preload:	1000

Figure 4. Bolt library plug-in.

It can be seen from Figure 4 that the main dialog consists of a region where the ‘Bolt Family’ and ‘Bolt Types’ are selected and a region where the geometry and loading is specified. The plug-in was written to incorporate a library of pre-defined bolt specifications. The user can select a ‘Bolt Family’ and then select an existing ‘Bolt Type’ from within that family. In all cases the bolt definition must be consistent with the parameterized bolt shown in Figure 4. The specification of the bolt families and bolt types is controlled by the owner of the plug-in by modifying the data in a specific initialization file. This file uses a standard Python dictionary format for its definitions. A sample can be seen in Figure 5.



```
# This module contains the default dimensions each bolt type.
from boltStudioConstants import BOLT, NUT
# Dictionary containing all bolt/nut definitions
bolts = {}

# New Family
bolts['Hexagon with Flange'] = {}

# New bolt in Family
bolts['Hexagon with Flange']['M8'] = {}
bolts['Hexagon with Flange']['M8'][BOLT] = {'l': 50.,
                                            'lth': 25.,
                                            'lsh1': 7.,
                                            'lsh2': 7.,
                                            'lt1': 3.,
                                            'lt2': 3.,
                                            'dh': 10.,
                                            'dc': 15.,
                                            'dt': 7.5,
                                            'dwo': 25.,
                                            'dwi': 10.,
                                            'dsh1': 7.5,
                                            'dsh2': 7.,
                                            'deff': 8.0,
                                            'hh': 10.,
                                            'hc': 4.,
                                            'alpha': 15.0,
                                            'hw': 4.,
                                            'preload': 1000.,
                                            }
bolts['Hexagon with Flange']['M8'][NUT] = {'dnh': 10.,
                                           'dnc': 15.,
```

Figure 5. Bolt library definition.

Once the bolt type is selected, the geometry region of the dialog is updated to the appropriate dimensions. Note from Figure 4 that there are three types of bolt available: i) with no washer, ii) with an integrated washer and iii) with a separate washer. The diagram updates according the washer type that the user selects. Figure 4 shows the case with no washer and Figure 6 shows the cases with integrated and separated washers.

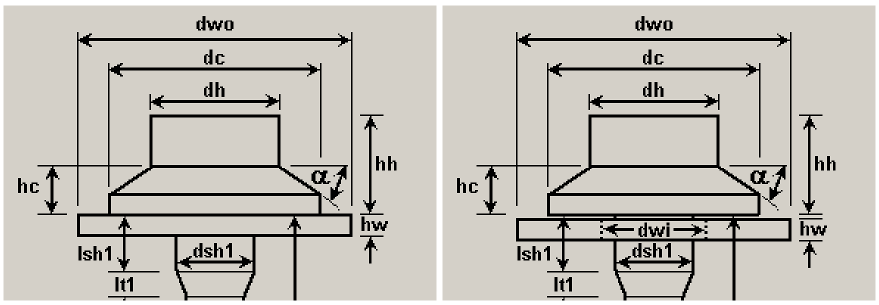
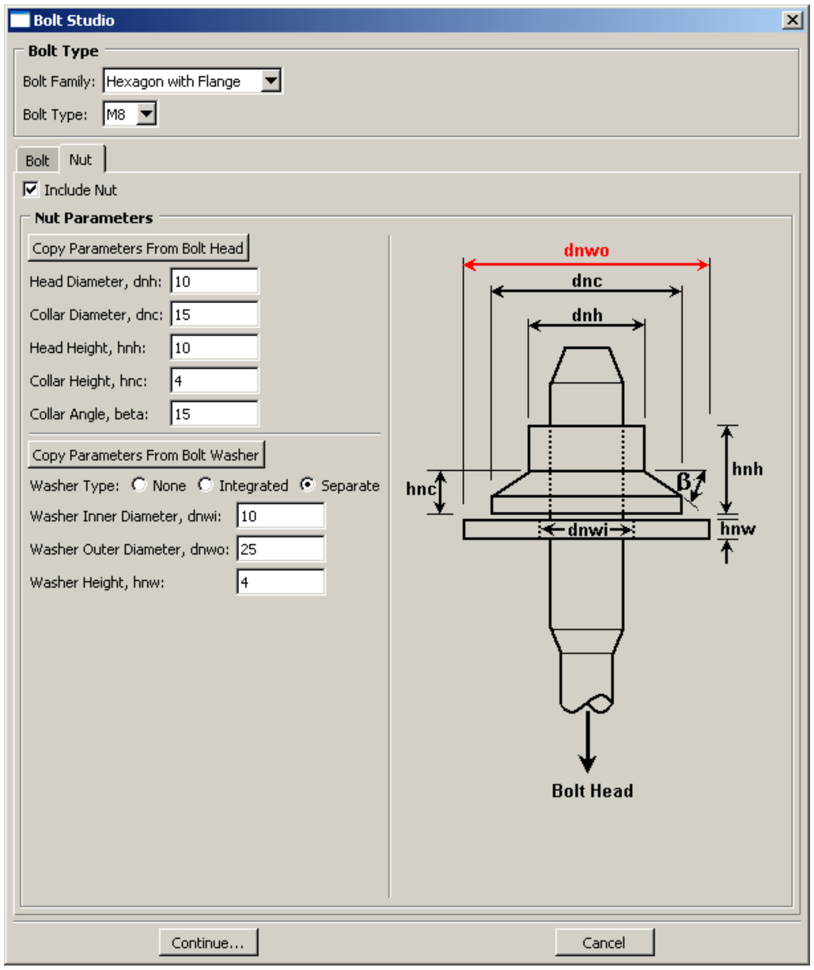


Figure 6. Washer types.

The user can modify any of the dimensions of the bolt to suit their specific needs. As the user selects a dimension to modify, the appropriate dimension is highlighted in the bolt diagram to give a visual indicator of the dimension in question.

Once the bolt dimensions have been defined, the user can switch to the ‘Nut’ tab where they can control the nut definition (if applicable). The nut tab is shown in Figure 7. The nut tab behaves similarly to the bolt tab with a two minor differences. These are i) the user can choose to include the nut or not and ii) the user can copy the nut and washer definition parameters from those defined in the bolt tab.


Bolt Studio

Bolt Type

Bolt Family: Hexagon with Flange

Bolt Type: M8

Bolt | Nut

[✓] Include Nut

Nut Parameters

Copy Parameters From Bolt Head

Head Diameter, dnh:	10
Collar Diameter, dnc:	15
Head Height, hnh:	10
Collar Height, hnc:	4
Collar Angle, beta:	15

Copy Parameters From Bolt Washer

Washer Type: None Integrated [✓ Separate]

Washer Inner Diameter, dnwi:	10
Washer Outer Diameter, dnwo:	25
Washer Height, hnw:	4

Figure 7. Nut definition tab.

Once the user has defined the bolt and nut definitions, the ‘Continue…’ button is pressed to start the process of positioning the bolt/stud and, where applicable nuts and separate washers into the assembly. Once the ‘Continue…’ button has been pressed, the dialog box pops-down and the user is led through the placement process via a series of questions in the prompt area of the Abaqus/CAE main window. Once this process is complete, the bolt/stud, nut and washers are created and positioned in the assembly (the bolt loads are also applied). The series of questions is then repeated from the beginning in order that multiple bolts of the same type can be placed in the assembly. Once the user has placed all of the bolts in the assembly, they can exit the placement process by pressing the cancel button in the prompt area.

The series of placement questions are dynamically modified based on the users answers to previous questions. For example, to specify the axis to which the bolt axis will be aligned, the user is asked to “Select bolt axis, face for concentric constraint, or edge on top of cylinder”. If the user selects the third choice, they would then be prompted to “Select edge on bottom of cylinder” from which the axis would be calculated. In cases where the user is asked to select a face or edge, they can select either geometric faces/edges or element faces/edges. If the user specifies a nut to be defined, an additional question regarding the positioning of the nut face is also asked. A flowchart showing the series of questions is shown in Figure 8.

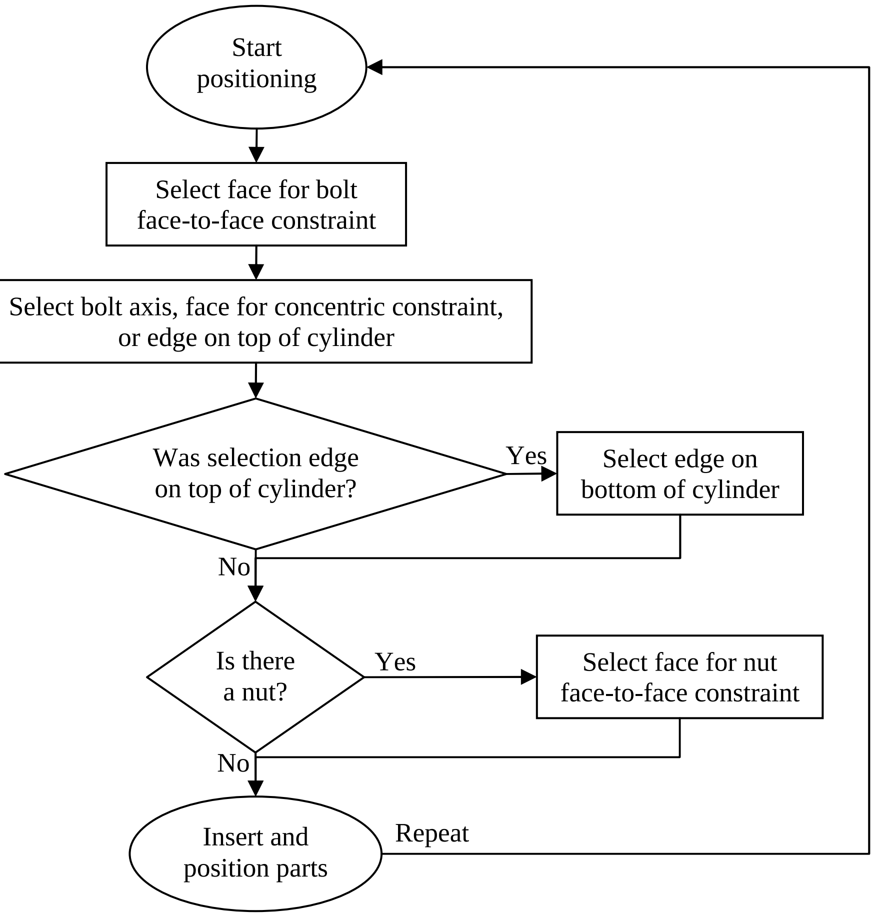

Start positioning
Select face for bolt face-to-face constraint
Select bolt axis, face for concentric constraint, or edge on top of cylinder
Was selection edge on top of cylinder?
Yes
Select edge on bottom of cylinder
No
Is there a nut?
Yes
Select face for nut face-to-face constraint
No
Insert and position parts
Repeat

Figure 8. Positioning flowchart.

Figure 9 shows an example of a nut, bolt and separate washers added to an assembly (half of which is removed for display purposes) using the plug-in. The intention of this analysis is to perform a closing of the bracket from its initially splayed configuration. The plug-in positions the nut washer such that the washer is just touching the oblique face of the bracket.

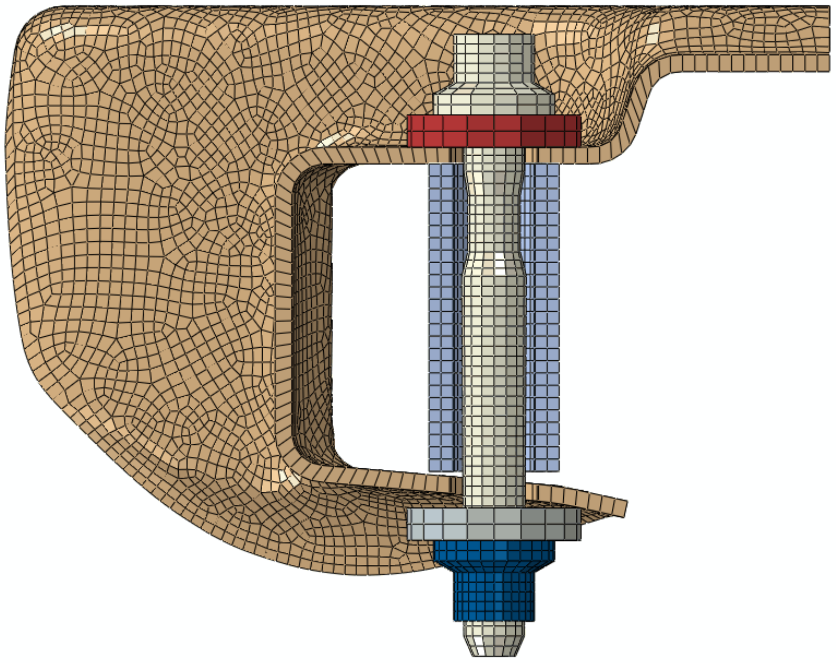


Figure 9. Bolt, nut and washers in assembly.

3.3 Plug-in to facilitate importing orphan meshes

Using this plug-in, the user can select any number of existing Abaqus input files to be imported into the current assembly. For Abaqus simulations, General Motors use ‘flattened’ input files. One drawback with importing flattened input files is that, when the input file is imported into Abaqus/CAE, all of the nodes/elements etc. are added to a single part. There is native functionality to split an orphan mesh into separate parts (this is done through the **Part→Copy** function in the Part module). However, when this operation is performed, no set, surface or section data is copied to the new parts. This plug-in addresses this limitation by separating the parts and maintaining this important data. The plug-in performs the following operations:

1. Imports all of the selected input files into the current model. In each case, the imported assembly is split into discrete parts and the set, surface and section data are copied to these parts.
2. An instance of each part is then generated in the assembly with the same name as the corresponding part.
3. The parts and part instances will subsequently re-named according to their metadata.

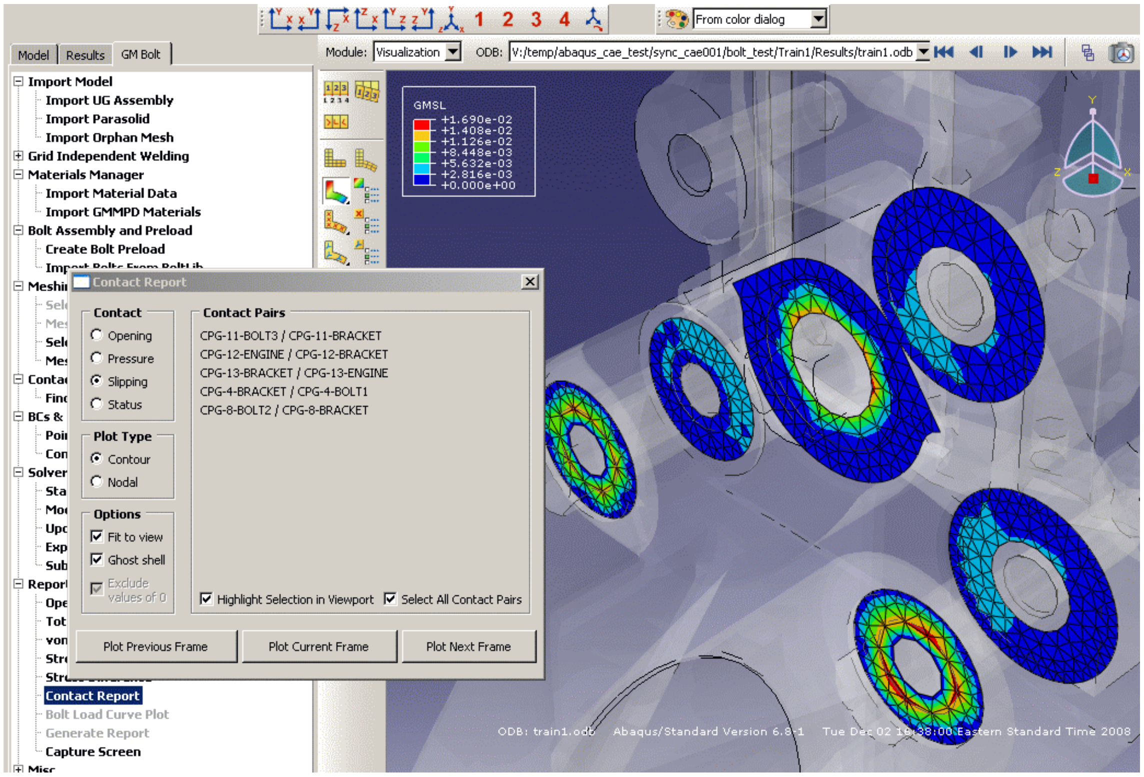
3.4 Plug-in to augment output database and provide new contour plotting style

This plug-in was created to enable customized post-processing functions in the GM BoltStudio. When invoked, the following enhancements are made to augment the visualization capability in Abaqus/CAE:

1. Creation of additional user field variables specifically defined at GM for evaluating bolt-joint performance, and store these back into the output database (the .odb file).
2. Reorganize the contact related results to facilitate the customized contour plotting.
3. Add new contour plotting styles to present results more intuitively.

If the size of Abaqus output databases is large, the augmentation process can be CPU and memory intensive. An option is provided in GM BoltStudio to force the augmentation of the ".odb" file in batch mode as part of the analysis job. The augmentation process can also be evoked interactively in Abaqus/CAE in the task of "Open Results File" in GM BoltStudio. New field output variables are created only if they do not already exist in the target output database.

The contour plotting style in native visualization module of Abaqus/CAE (or Viewer) works well to show contact results in isolated part instance view or in an "exploded" assembly view in which the contact surfaces are completely exposed. In GM BoltStudio, additional contour plotting styles are added to facilitate more interpretative display of contact results. One example is shown in Figure 10. In this example, user can see through the parts in the "as assembled" view and directly look at the contact results on slave surfaces of the whole model or selected contact pair.



Module: Visualization ODB: V:/temp/abaqus_cae_test/sync_cae001/bolt_test/Train1/Results/train1.odb

From color dialog

Contact Report

Contact: Opening [✓ Slipping] Pressure Status

Plot Type: [✓ Contour] Nodal

Options: ☑ Fit to view ☑ Ghost shell ☑ Exclude values of 0

Contact Pairs:
CPG-11-BOLT3 / CPG-11-BRACKET
CPG-12-ENGINE / CPG-12-BRACKET
CPG-13-BRACKET / CPG-13-ENGINE
CPG-4-BRACKET / CPG-4-BOLT1
CPG-8-BOLT2 / CPG-8-BRACKET

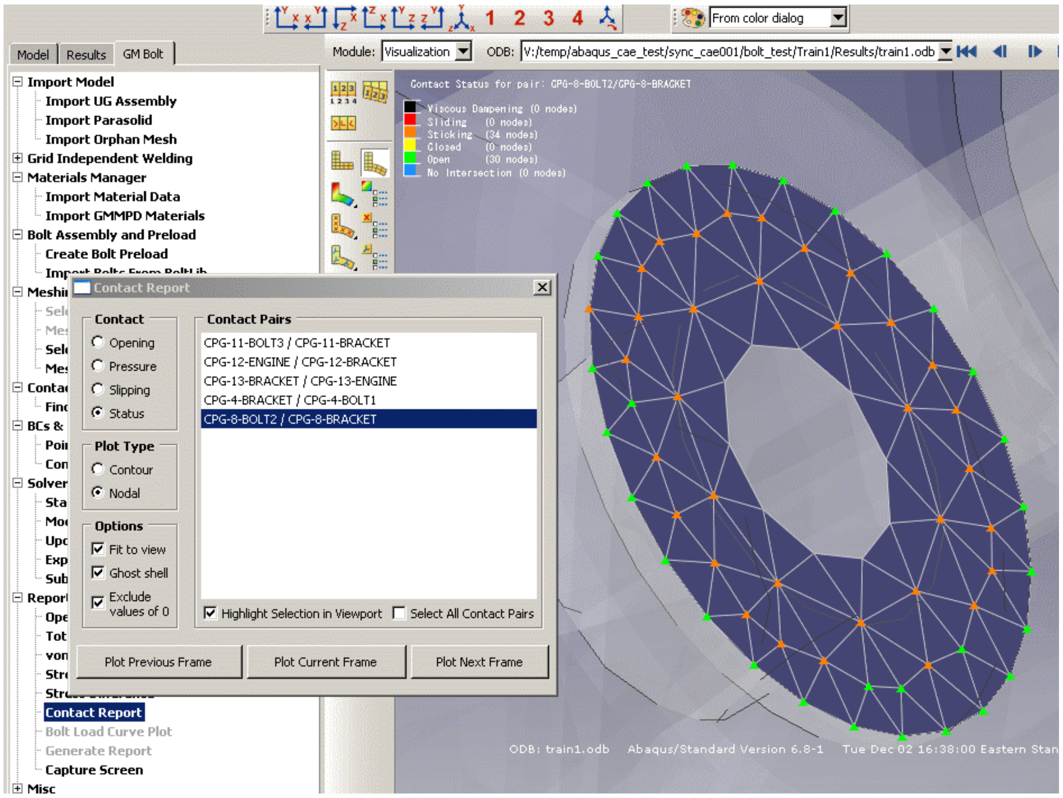
☑ Highlight Selection in Viewport ☑ Select All Contact Pairs

GMSL
🟥 +1.690e-02
🟧 +1.408e-02
🟨 +1.126e-02
🟩 +8.448e-03
🟩 +5.632e-03
🟦 +2.816e-03
🟦 +0.000e+00

ODB: train1.odb Abaqus/Standard Version 6.8-1 Tue Dec 02 16:38:00 Eastern Standard Time 2008

Figure 10. Contour plot of contact results - slipping

Such contact results are plotted either as the regular smoothed contours or discrete "nodal" plot (Figure 11) with color-legend symbols on the slave nodes. The "nodal" plot type is appealing for presenting field view of discrete contact state status (open, sticking, slipping, etc.) in which the custom spectrum indicates the status and report out the number of slave nodes fall in each of the state.



Module: Visualization ODB: V:/temp/abaqus_cae_test/sync_cae001/bolt_test/Train1/Results/train1.odb

```
Contact Status for pair: CPG-8-BOLT2/CPG-8-BRACKET
Viscous Dampening (0 nodes)
Sliding    (0 nodes)
Sticking   (34 nodes)
Closed     (0 nodes)
Open       (30 nodes)
No Intersection (0 nodes)
```

Contact Report

Contact: ( ) Opening ( ) Pressure ( ) Slipping [✓ Status]

Plot Type: ( ) Contour [✓ Nodal]

Options: [✓] Fit to view [✓] Ghost shell [✓] Exclude values of 0

Contact Pairs:
CPG-11-BOLT3 / CPG-11-BRACKET
CPG-12-ENGINE / CPG-12-BRACKET
CPG-13-BRACKET / CPG-13-ENGINE
CPG-4-BRACKET / CPG-4-BOLT1
CPG-8-BOLT2 / CPG-8-BRACKET

[✓] Highlight Selection in Viewport [ ] Select All Contact Pairs

ODB: train1.odb Abaqus/Standard Version 6.8-1 Tue Dec 02 16:38:00 Eastern Stan

Figure 11. “Nodal” plot type of discrete contact status

4. Conclusions

Through a joint effort with SIMULIA, General Motors developed the GM BoltStudio suite to consolidate a number of bolt joint analysis tools/practices from its CAE organizations into a single modeling/simulation environment in Abaqus/CAE and Abaqus/Standard. The strong scripting and GUI toolkit support and the kernel accessibility in Abaqus/CAE allowed rapid development of the plug-in suite to bridge functionality and usability gaps of using Abaqus/CAE for bolt joint modeling inside GM. The significantly improved efficiency and effectiveness of the suite are not only confirmed by GM’s veteran CAE analysts, but also demonstrated by the completion of bolt joint analysis by a few GM’s Fastening Design engineers in their first experience with FE analysis. The ‘bolt library’ has also been written as a standalone plug-in and is available to the general Abaqus population at http://www.simulia.com/products/extensions.html or from SIMULIA Great Lakes.

5. Acknowledgement

The authors wish to acknowledge contribution to the GM BoltStudio development from the other members of the project team: Markus Roth and Sung-ling Twu from General Motors’ Chassis CAE groups. Asif Khan and Christina Feist of SIMULIA also assisted in the project planning. Thanks go to Mark Bohm of SIMULIA for the provision of development resources. The inputs and feedbacks from General Motors’ Fastening Engineering groups are also acknowledged.