

**Coreform User-Interaction for DIW-Pad**  
**Project ID: CEEEn\_CPST\_016**

**by**

**Royal Engineering**  
**Jordan McInelly**  
**Joseph Stout**  
**Israel Rivera**

**A Capstone Project Final Report**

**Submitted to**

**Greg Vernon**  
**Coreform LLC**

**Department of Civil and Construction Engineering**  
**Brigham Young University**

**April 11, 2022**

**Executive Summary**

**PROJECT TITLE: COREFORM USER-INTERACTION FOR DIW-PAD**  
**PROJECT ID: CEEEn\_CPST\_016**  
**PROJECT SPONSOR: Coreform LLC**  
**TEAM NAME: Royal Engineering**

Coreform IGA is an isogeometric analysis nonlinear structural solver that delivers dramatically faster solutions with improved accuracy. Coreform's IGA technology allows calibration of simulation speed and accuracy to fit any stage of the product development process. Using Coreform's IGA software, Cubit, our team was assigned to help Coreform's clients in developing a DIW (Direct-Ink Write) model. Along with developing a DIW model, we were tasked with identifying improvements for Cubit that would improve the client's success for these models, supporting the prospective sale of Coreform's software to the client.

The purpose is to identify improvements in the user experience of Coreform's software that increase its usefulness for DIW pads to drive the sale of software to the prospective client. Key differences are that traditional FEA uses body fitted meshes which require manually cutting geometry into pieces that can be meshed. Traditional FEA programs typically take a long time to mesh a geometry because of their method. However, with Coreform's IGA, we can immerse a geometry in a bounding box/enclosure to mesh our model in a matter of minutes.

**Table of Contents**

Executive Summary..... 2  
List of Figures ..... 4  
Introduction ..... 5  
Schedule ..... 6  
Assumptions & Limitations..... 7  
Design, Analysis & Results..... 8  
Related Issues..... 13  
Lessons Learned..... 13  
Conclusions ..... 14  
Recommendations..... 15  
Appendix ..... 16

**List of Figures**

Figure 1: Abaqus Option Tabs ..... 16  
Figure 2: Set Manager ..... 16  
Figure 3: Abaqus Interface ..... 17  
Figure 4: Export Option ..... 17  
Figure 5: Unit Converter ..... 18  
Figure 6: Ribbon Interface ..... 18  
Figure 7: Material Properties ..... 19

## Introduction

In mechanical assemblies, the accumulation of tolerances in manufacturing results in small gaps between subassemblies. During operation, the fully assembled product may encounter a wide variety of temperature, shock, and vibration conditions. Historically, cellular silicone pads have been used to fill-in the gaps between these subassemblies. These pads, also called "cushions", are designed such that they will be in a compressed state after assembly to ensure that they don't disengage. Other applications of these cushions include thermal insulation, weight reduction, vibration damping, mitigating shock, and acoustic attenuation. Essentially, the manufacturing of these cushions is a stochastic process that can result in a varying density of pores, pore-sizes, and wall widths. These manufacturing irregularities result in a material with performance characteristics that are difficult to predict.

Direct ink-write foam pads are a recent development that aim to address many of the challenges associated with traditional foams. DIW pads are a high-priority technology particularly at the US Dept. of Energy's National Nuclear Security Administration (NNSA) and were initially developed at the DOE's Lawrence Livermore National Laboratory (LLNL). Initially, we were tasked to work closely with Kansas City Engineers to improve upon a previous direct ink-write pad by decreasing the length of time to print each pad, produce a softer pad and decrease variability from pad to pad. We saw this entire project as an issue due to our lack of knowledge in this subject. We felt overwhelmed that we would have to solve these complex issues.

However, the scope of our capstone project changed to working on the DIW pad using both Abaqus and Coreform's FEA software called Cubit. Coreform's IGA is an isogeometric analysis non-linear structural solver that delivers dramatically faster solutions with improved accuracy. Coreform's IGA technology allows calibration of simulation speed and accuracy to fit any stage of the product development process. As we went about trying different practice models on Abaqus and then Cubit, we were tasked with identifying improvements for Cubit that would improve the client's success for these models, supporting the prospective sale of Coreform's software to the client.

## Schedule

- Assigned Capstone Project by Dr. Mitchell and Dr. Hotchkiss (September 2021)
- Finite Element Analysis labs from Dr. Scott (October 2021-December 2021)
- Literature about model prep/design (Assigned to read January 2022)
- Define project definition
  - o Call customer (February 2022)
  - o The Phenomena Identification and Ranking Table (PIRT) (February 2022)
  - o Revise project document (February 2022)
- Create a practice model (puck) in Abaqus (March 2022)
- Recreate practice model (puck) in Cubit (March 2022)
- Tasked with identifying improvements for Cubit (March 2022-April 2022)
- Set up and run initial DIW pad model in Abaqus (March 2022) [Done by Greg]
- Set up and run initial DIW pad model in Cubit (March 2022) [Done by Greg]
- User Interaction Report (Deliverable to Coreform April 2022)
- Meet with Greg (Sponsor) and present our results (April 2022)
- Complete Capstone Report and present to undergraduate seminar (April 2022)

## Assumptions & Limitations

The first limitation we ran into happened to be an assumption by our sponsor. Greg Vernon from Coreform LLC, was under the assumption that we, the capstone group, had completed the CE Finite Element Analysis class at BYU. Immediately, the expectations he had for us were lowered because this project was our first exposure to Finite Element Analysis and with Abaqus which is a program that performs this type of analysis. This resulted in a misalignment of our capacity to use FEA programs comfortably.

We ran into a handful of other challenges early on for example, not knowing the scope of our project. We didn't start the bulk of our project until the end of January. From the time of assignment to December, we were asked to work on FEA labs, however we frequently found ourselves lost because of how foreign the subject was to us. Additionally, the initial scope of our project was to help Honeywell Engineers in Kansas City develop a better DIW-pad product. To complete this, we met with Honeywell Engineers over Zoom and asked about the current issues they were running into with their product. Soon after, the scope of our project changed from helping Honeywell Engineers with their DIW-pad to helping Coreform LLC develop ways to improve their FEA program to be competitive with other traditional FEA programs.

Because of this, our time frame was pushed back, and we got put in a tight position with getting our DIW-pad model into both Abaqus and Cubit. Other challenges included having to learn how to use Abaqus and Cubit from scratch. Greg was very helpful and gave us a crash course 1-2 times a week in the month of February and parts of March. Once we got the hang of Abaqus, we started with creating a model of a puck between two platens as practice. Then we created the same model, but this time in Cubit. We gathered information along the way to improve Cubit and make it more user friendly. Due to time, Greg stepped in and created the model of the Direct-Ink write pad in both Abaqus and Cubit. He also showed us the time it took to model the DIW-pad in Cubit vs Abaqus. We compared these times with the amount it took to mesh the two and found drastic differences. At the end of this capstone project, we were able to create 9 different user-interaction suggestions for Coreform to implement into their Cubit software.

## Design, Analysis & Results

To help our client clearly see the suggested improvements, our “Design, Analysis & Results” portion of our report has broken down into suggested desired components. The title of the suggested improvement will be in **bold** font to help with identifying the suggestion. In these results, we will narrate the wanted function through the user’s eyes. Additionally, we will be explaining the new capabilities the improvements would entail and how the function would be implemented. All figures referenced can be found in the Appendix.

The first suggestion we thought of was the **simplification of set creation in Cubit**. As users, we want a graphical interface for creating sets. So that we can more quickly and efficiently create sets for use in our simulations. We’ll know when this is done when we can assign all the sets in our model without writing code. This would look like a graphical user interface environment for creating and assigning sets within Cubit to be able to use on designs.

The user will be able to:

- Click on objects in the render window. As in Figure 1: Abaqus Option Tabs
- Have a pop-up window to assign properties to sets.
- Have a set manager that allows you to create/delete/rename sets. As shown in Figure 2: Set Manager
  - o Within the set manager, create a function that highlights the set that is being selected in the set manager. As we can see in Figure 3: Abaqus Interface

The next suggestion we thought of was being able to **toggle between unit and unitless systems**. As users, we want to be able to toggle between a system that uses units and a system that is unitless. We also want to choose our unit system to ensure we are being consistent and to help avoid errors. We’ll know when this is done when we are able to hit a button and units are populated behind the numbers and that we can hit that same button to remove units. This would look like having a toggle functionality permitting users to navigate problems in a unitless interface or in a chosen unit interface.

The user will have:

- On Cubit’s table of icons, a unit selection option. In that selection, a checkbox under options to toggle between units and unitless should be added.
  - o If units are checked, table un-grays and allows user to choose which unit system to use by checking the circle.

- Values being displayed with no units behind them or units behind them depending on selected setting.

The next suggestion we thought of was **optimizing data exportation**. As users, we want to export our data into a tabulated excel sheet. So that we can quickly create graphs and other visuals with the data from Cubit. We'll know when this is done when we have a button that exports data into an excel sheet. This would like an export button on the file tab that creates a table from test data in an excel sheet.

The user will be able to:

- Under the basic operations tab, be able to click file and have one of the options be export. An example of this is seen in Figure 4: Export Option
- When export shows up, we'd like there to be multiple options like xlsx, pdf, doc, and other types of files rather than just csv.
- If we're doing an excel export, have a nicely formatted table with the possibility of a graph auto populated to match data produce by Cubit.

The next suggestion we thought of was to **include a unit converter**. As users, we want to enter in a value no matter what units the data is given and have Cubit convert the data to the desired unit system. So that we can save time not converting units elsewhere. We'll know when this is done when we can input data from any unit base to any other unit base. This would look like an integrated unit convertor built into the software that allows users to enter data from any unit and select which unit they want it converted and displayed to.

The user will be able to:

- When inputting data for a geometry, have a drop-down box next to the entry box where users can choose which units the input data is assigned to. Of course, Coreform LLC will have to implement new code to support different unit systems.
- If an existing geometry is unitless or is in units different than what the user desires, the user should be able to enter values in any units to have them converted.
  - o The idea described above should be found in the same window, to allow the user to select which unit they want their values converted to. As shown in Figure 5: Unit Converter
  - o The converted unit data will be what is displayed when you click on an element of hover over them depending on what other users want. I would rather click to find out their units.
  - o It would also be great to be able to edit the units of an element/shape just in case the units need to be changed in the future.

The next suggestion we thought of was having the **ability to put in the bulk modulus and shear modulus if you don't have the elastic modulus**. As users, we want to be able to input the bulk modulus or shear modulus if we're not given elastic modulus and have Cubit calculate it. So that we can save time by not having to calculate it elsewhere. We'll know when this is done when we are asked for the bulk modulus and shear modulus, or the elastic modulus. This would like an integrated calculator that populates elastic modulus by taking bulk and shear modulus inputs and computing the elastic modulus.

The user will be able to:

- When prompted for the elastic modulus, have both bulk and shear modulus boxes there.
- Fill out the bulk and shear modulus and have the software solve the elastic modulus which can be used and applied.

The next suggestion we thought of was having a **new ribbon interface for Cubit**. As users, we want a large and easy-to-work ribbon under the basic operations. So that we can navigate quickly through the different element operations. We'll know when this is done when a ribbon exists like that of AutoCAD, Solidworks, or Excel. This would look like a large, user-friendly ribbon that allows us to navigate through different operations like meshing, unit-converter, shape creation, etc.

The user will be able to:

- Upon opening the program, have a large ribbon that spans left to right at the very top, just under the basic operations. An example of this is seen in Figure 6: Ribbon Interface from Solidworks.
  - o This ribbon should have all the different operations that are commonly used in Cubit FEA. Along with the operations, a small description of each operation would be helpful for new users. That way the operator knows what they are doing before mis-clicking an operation.
  - o Icons on the ribbon should have a toggle option that allows you to see a brief definition. Ideally, we would like this option to be found in the File tab under the basic operations at the very top.
  - o As an initial phase for the ribbon, Coreform can start with the traditional ribbon that has light lines to separate/ group similar operations. The ribbon in Solidworks has a sub-ribbon (more tabs) under it. We don't think we need this, but if in the future more users want this, Coreform can add it.

The next suggestion we thought of was the **implementation of new sketching methods**. As users, we want to have more options to sketch like point-click and parametric sketching methods. So that we can more easily create our geometry/model. We'll know when this is done when we can sketch a model using parametric or point-click sketching. This would look like an interface to sketch any geometry by either using a point-click method or parametric method.

The user will be able to:

- Have the option to sketch a geometry using a point-click method or a parametric sketching method with the mouse's cursor. We also want the option to adjust the next line by having options to arc, chamfer, spline, among other options.
  - o The options should appear as a small list of icons when you are sketching. The option to sketch can be initiated by clicking an icon if other users find it to be necessary.
  - o Parametric modeling should also have options to toggle between 2D and 3D by having other windows. These windows could have options to view the model from different faces like a top, front, and side view. Additionally, it should have a dimensional isometric projection that can be rotated. This last one already exists in Cubit, but it would be great to quickly get to certain views.

The next suggestion we thought of was having **the ability to have syntax in both script and command lines be corrected when a mistake is detected**. As users, we want our script and command line to be fixed when my syntax is input incorrectly. So that we can run our script properly without any bugs. We'll know when this is done when we have auto-correcting scripts that can be executed smoothly. This will look like having the syntax of the coding language used by Cubit to be more user friendly.

The user will have:

- The command line to auto-fix our lines when common words are spelled incorrectly. If the program is unsure about a word, we would like some suggestions in a small pop-up window that lets us know that a possible mistake has been detected and if we would like the word to be changed to a different one.
  - o As for the code syntax, we would like suggestions, in the same way we have described above, to pop-up when the program detects an invalid script. We want the program to recognize invalid scripts and correct them so they can be running

correctly. This would require a smart code recognition from the program to know what kind of statement is trying to be scripted.

The last suggestion we thought of was **a list of pre-installed materials to choose from**. As users, we want a palette of common materials with all of their properties saved in Cubit. So that we can quickly assign material properties to a geometry. We'll know when this is done when we have a palette of materials saved in Cubit. This will look like having the option to assign materials to geometries that already exist inside of Cubit.

The user will have:

- The option to choose from various materials that are pre-built into Cubit to assign to geometries. Below we have pre-installed materials shown in SOLIDWORKS.
- The option to save material properties that they have created to be added to the materials library that is to be built. Figure 7: Material Properties shows a lot of the properties we would like included.
  - o We also want to be able to edit existing material properties by doing a "save as" and renaming it to have the original existing along with the modified material property.

## Related Issues

Our project has many impacts on a local and global scale including the following issues:

- Safety – The DIW pads have practical usage in military grade missiles that prevents multiple metal components from clanging together and setting off an accidental detonation. Without these pads there would be a major safety risk in using these projectiles.
- Global factors – Traditional foam pads are used globally but have a great range of variance so implementation of the new DIW pads will have a massive effect on a global scale in several different private sector companies and military assets in many different countries.
- Cultural factors – The team had to learn new company cultures while working for Coreform and the teams in Kansas City. After learning the cultures of different individuals, it helped the team to work closely and effectively.
- Environmental factors – The DIW pads are more environmentally friendly than the old traditional foam pads.
- Economic factors – Improving Cubit will help Coreform save money since it is their own software, and they won't have to use department funds to pay for a third-party software to complete their desired outcomes.

## Lessons Learned

- Importance of defining scope at the beginning of a project.
- Ability to stay on schedule and not get behind on a project.
- Expectations must be set at the beginning of a project so that each team member and leader understand their assigned role and responsibilities.
- The need to communicate effectively over email to successfully complete a project.
- The importance of being able to do your own research when you lack knowledge on a subject.
- Good relationships between members are vital to successfully complete a project.
- Things will go wrong, there will be delays.

## Conclusions

We learned that Coreform is doing great work with their software and that the hard part of creating a mesh is something Cubit does well. The bulk of our suggestions are to help with the modeling part of the program. It still lacks behind in creating a complex geometry the way Abaqus quickly can. For this reason, the user interactions we have suggested will make the modeling process much faster and competitive. These suggestions can be modified to fit the needs of other users was Cubit gains more popularity in the FEA world.

These suggestions will benefit Coreform because Cubit will be able to perform the same functions as other, more popular FEA programs, but with faster meshing. This will result in companies using Cubit as their primary FEA program rather than a secondary program. Right now, Cubit is often used to create preliminary models at a cheaper price and having 2 subscriptions of other more popular FEA programs when the models are ready to be transferred.

## Recommendations

For the Capstone Experience:

- Define the scope of the project clearly and earlier so that teams can work on the project sooner.
- Ideally, as civil engineers we wanted a project that had to do with civil engineering. The project we got, albeit interesting, was a mechanical engineering project.
- Continue to have food for meetings because what Greg from Coreform LLC was doing really helped with encouraging us to stay late and work on the project.

For Coreform LLC:

- Suggested Improvements to Cubit
- Simplification of set creation
- Toggle between unit and unitless systems
- Optimizing data exportation
- Include unit converter
- Include bulk and shear modulus without elastic modulus
- New ribbon interface
- Implement new sketching methods
- Correct both script and command lines when a mistake is detected
- A list of pre-installed materials to choose from

Appendix

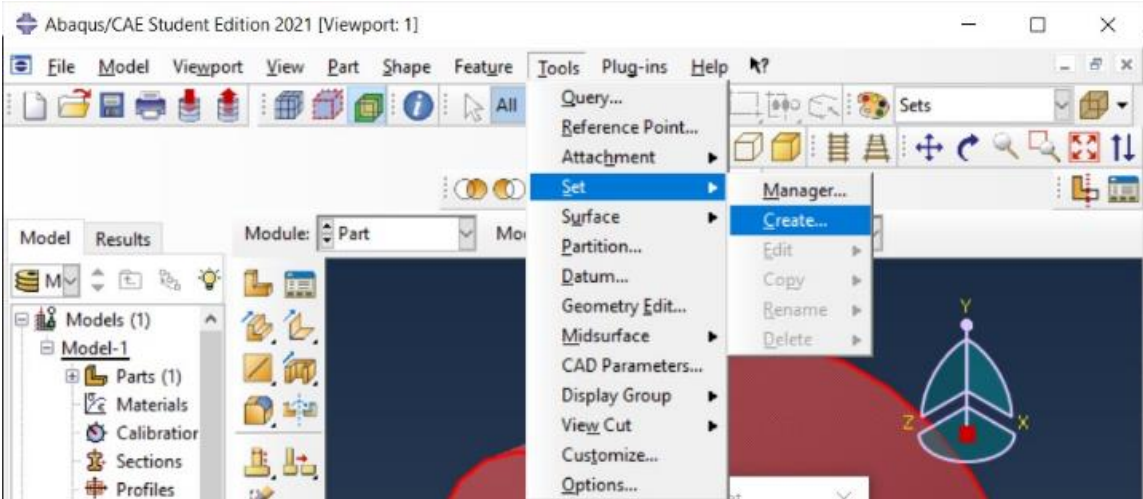


Figure 1: Abaqus Option Tabs

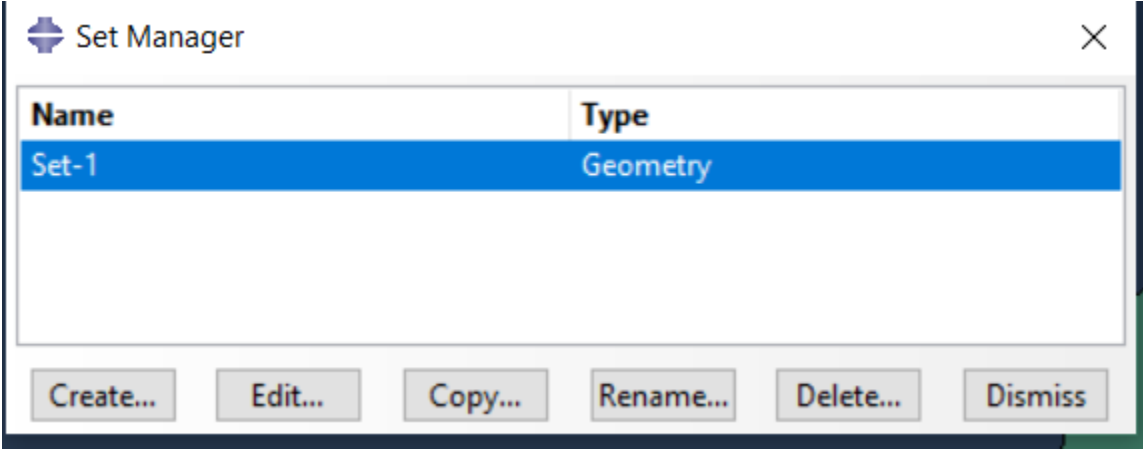


Figure 2: Set Manager

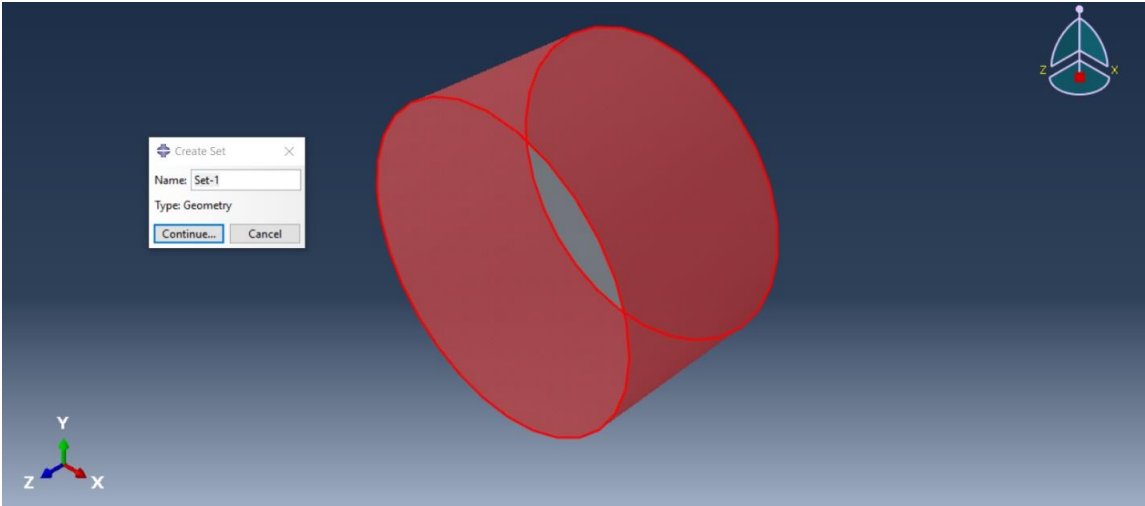


Figure 3: Abaqus Interface

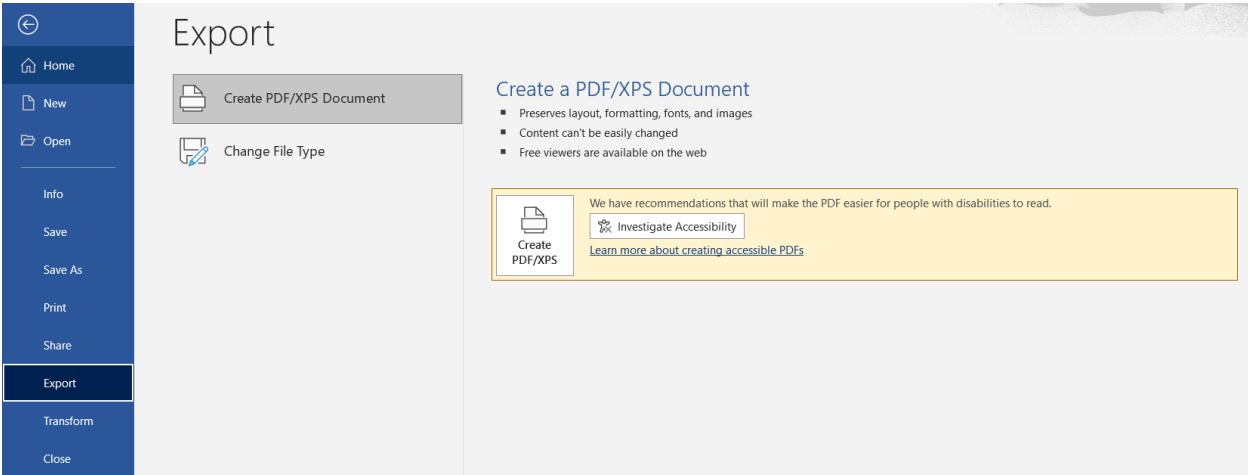


Figure 4: Export Option

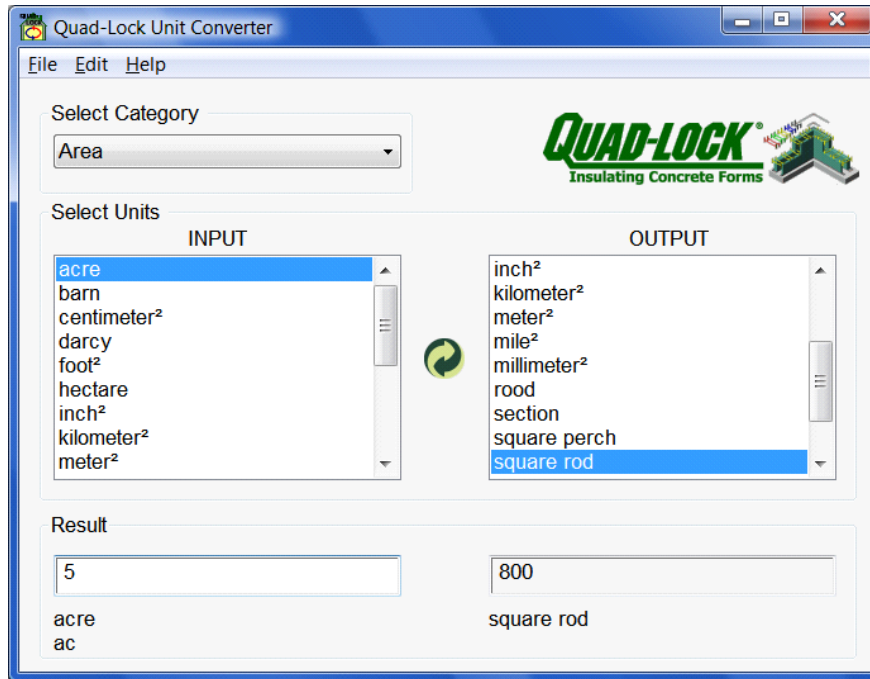


Figure 5: Unit Converter

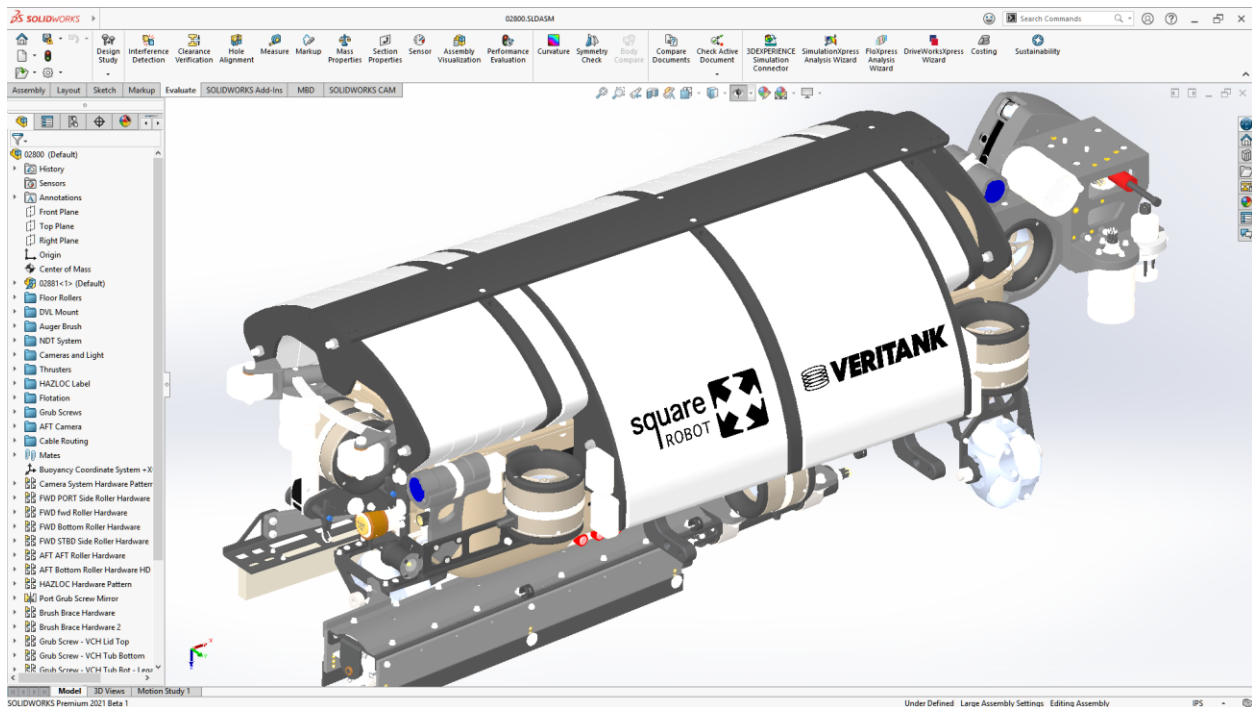


Figure 6: Ribbon Interface

Property	Value	Units
Elastic Modulus	2.01e+011	N/m <sup>2</sup>
Poisson's Ratio	0.31	N/A
Shear Modulus	7.7e+010	N/m <sup>2</sup>
Mass Density	7920	kg/m <sup>3</sup>
Tensile Strength	620000000	N/m <sup>2</sup>
Compressive Strength		N/m <sup>2</sup>
Yield Strength	275000000	N/m <sup>2</sup>
Thermal Expansion Coefficient	1.65e-005	/K
Thermal Conductivity	15.1	W/(m·K)
Specific Heat	420	J/(kg·K)
Material Damping Ratio		N/A

Figure 7: Material Properties