README.doc For Traction Force Code V1

Author: Keshav Patel

PI: Dr. Wes Legant

Created: 7/10/2019

This package is meant to take in microscopy cell image data and output a visual representation of the force exerted by a cell on its environment. These forces are usually localized to areas called Focal Adhesions (FA). The computation of forces is given by the following Inverse Problem:

.

Let be the number of beads in the gel and be the number of surfaces a cell could potentially exert a force. is a vector of bead displacements, is a vector of forces applied on the surface of a 3D object, and is the Green's matrix. is a function of the geometry, mechanical properties of the material, and the locations at which forces are applied and beads are initially located. This sequence of steps begins by calculating from imaging data and then from computational models. Afterwards, the inverse problem is solved, and the data is compiled together into a 3D visualization.

This package is organized into folders for each set of computations. At the end of each step, the relevant data is stored in the folder "data" and plots are stored in the folder "plots".

# STEP 0: Imaging

When taking images for this package, be sure to image the entire field of beads in the z-direction (i.e. from the coverslip to the point where beads can no longer be found). You can save the file under any name and any location (so long as you know where it is), but the image should be reformatted to a .TIF file with each channel as a separate file.

# STEP 1: Particle Tracking

The first step is conducted within the folder "particleTracking". This script will localize beads and match beads between time points. From there, the displacement of beads can be calculated. The output of the method will be an vector of unstressed bead locations and an vector of the displacement of that bead.

Let be the location of bead in the coordinate (, , or ) and be the displacement of bead in the coordinate. the bead location vector is given by:

and the bead displacements are given by:

The bead locations and the displacements are saved as a .MAT for future MATLAB scripts.

Before saving the displacement data, a drift correction is applied to the data. The data in each direction is independently fit to a quadratic curve and the residuals are taken as the corrected displacements. Due to swelling and other factors of the gel, the beads move in a non-random fashion over time that the quadratic fit hopes to remove.

## Running the Script

To begin, a set of images corresponding to a timeseries image of the beads in a gel should be stored as a TIF file. These files can be anywhere as long as you know the (absolute or relative) file path. The series of pre-treatment images go into the variable “preTreatmentFile” as a cell array of strings. Each filename should be entered sequentially as separate strings and separated by colons. The path should be entered in the variable “preTreatmentStem”. For the singular post-treatment file, the stem should go into the string “postTreatementStem” and the filename should go into “postTreatmentFile” as a string.

Use the variables “size\_x”, “size\_y”, and “size\_z” to adjust the image by voxel size. Also, change the parameter “bead.intensity” to change the minimum intensity for detecting a bead.

Once these constants are set, run the script. For a set of images with ~1000 beads, the script takes ~ 5 minutes.

# STEP 2: Green's Matrix Computation

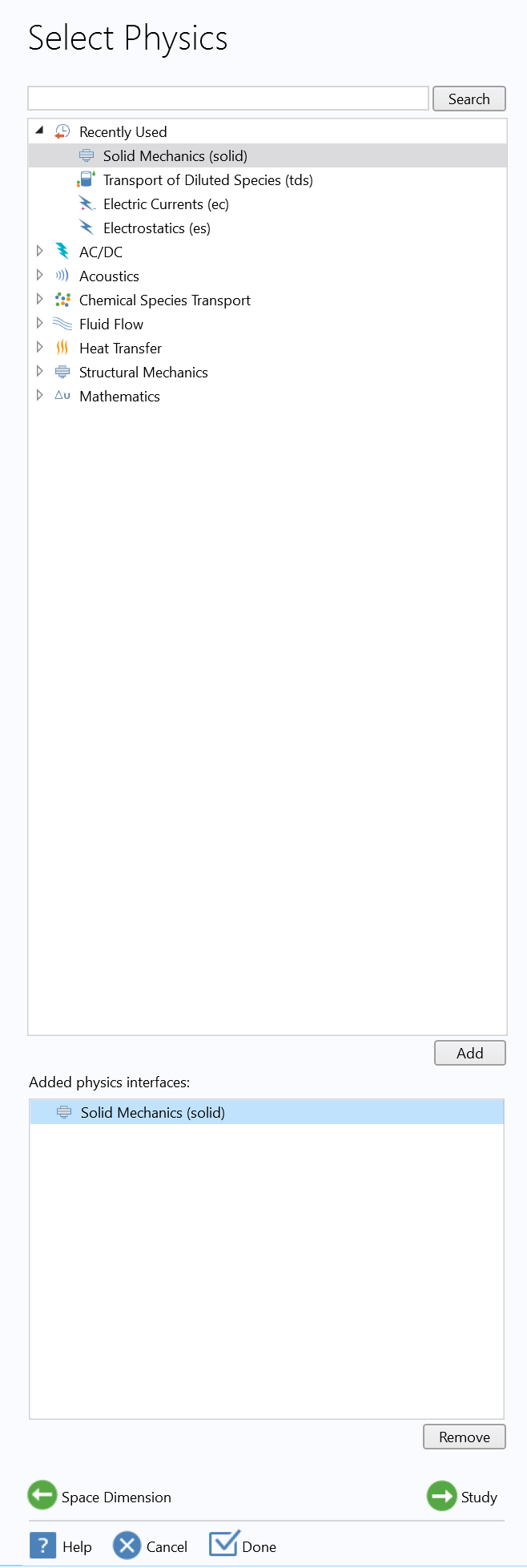
The solid mechanics analysis is conducted in the folder "COMSOL". The COMSOL file performs a sequence of single stress-strain calculations for a particular geometry/mesh and compiles displacement data in a csv file. The mesh is created such that the region where beads are located are finely meshed, while farther regions are coarsely meshed (for computational efficiency). A few preset geometries are available (cells on the top of a gel, cells in a channel, etc.); if a new geometry is needed, we recommend creating a copy of one of the available files and building the new geometry from there (the code and preset values will then be carried over). The next subsection describes how to set everything up from scratch.

This COMSOL file is designed to loop through each mesh element on a desired surface, apply a force along each cartesian coordinate, and measure the absolute displacement at any given location. This data is stored internally in COMSOL, and by querying at specific points (i.e. the initial bead locations found in Step 1), one can determine the strain at each bead given a singular force. The final output of this script is a csv file containing the absolute displacements of each bead along each cartesian direction in response to each applied force. This data is organized in Step 3 to become the Green's matrix necessary for computing the inverse problem solution.

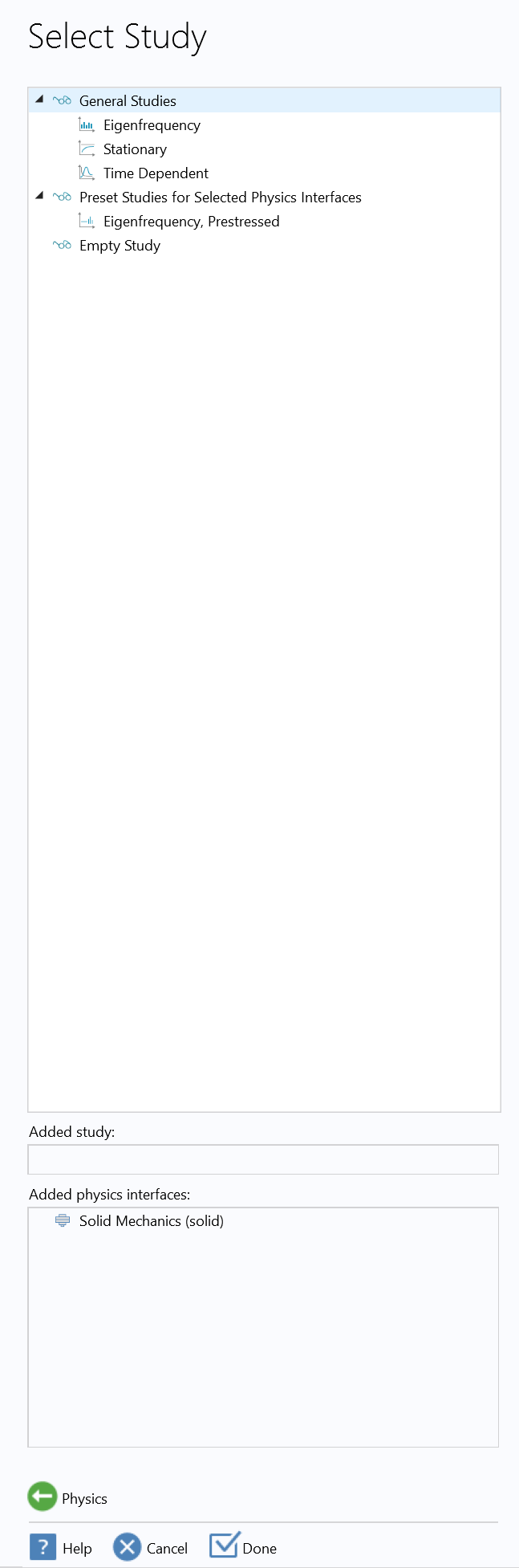
## Setting up a new COMSOL geometry

If you have copied an existing COMSOL file and are interested in creating a new geometry, skip to step 4.

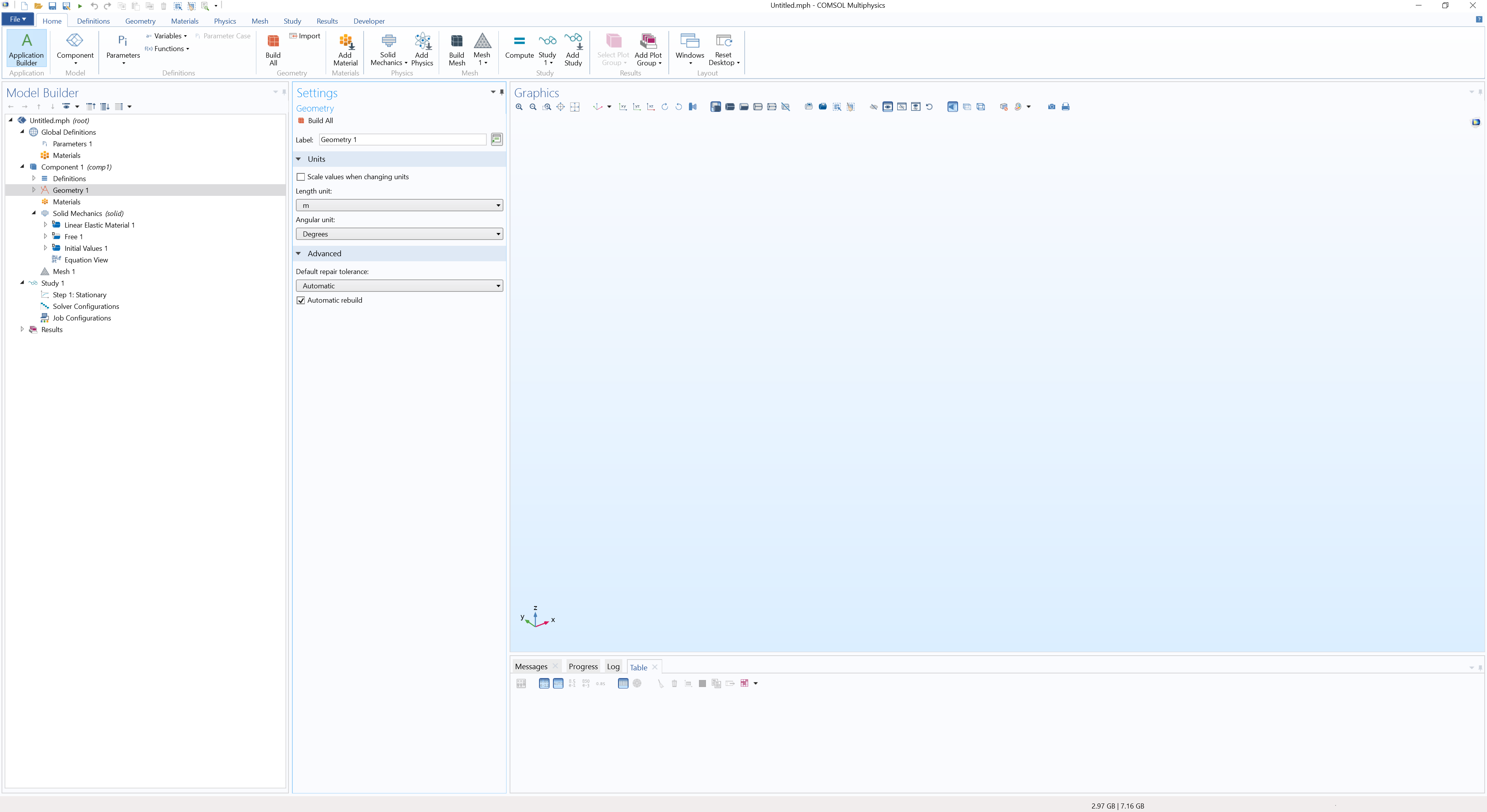
1. Beginning with a brand new file, click on “Model Wizard” > 3D. You will be taken to a new screen with the following LHS:



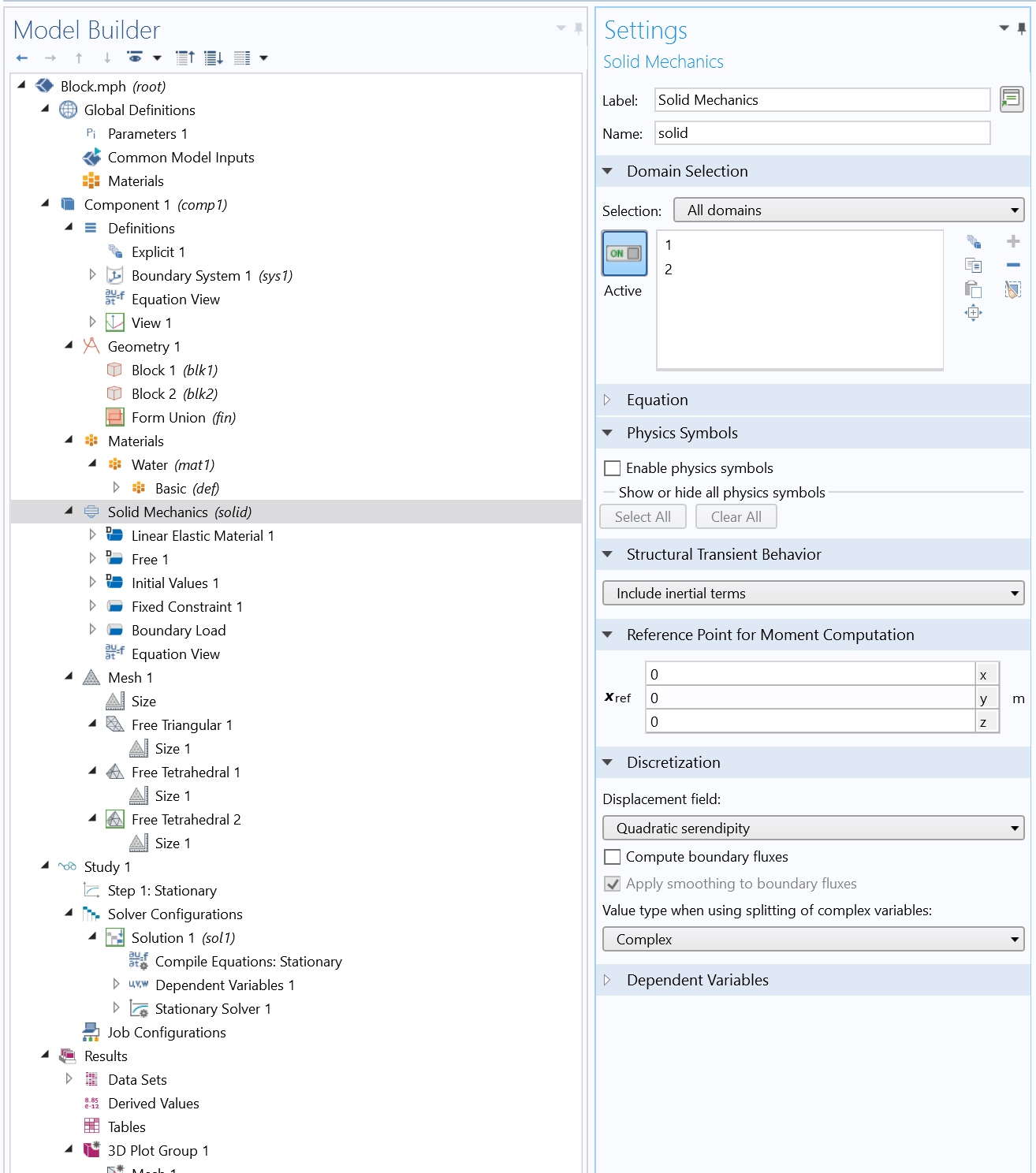
Highlight “Solid Mechanics” then click “Add” to make Solid Mechanics appear in the bottom pane, then click “Study”. In the next pane (see below), click “Stationary” then “Done”



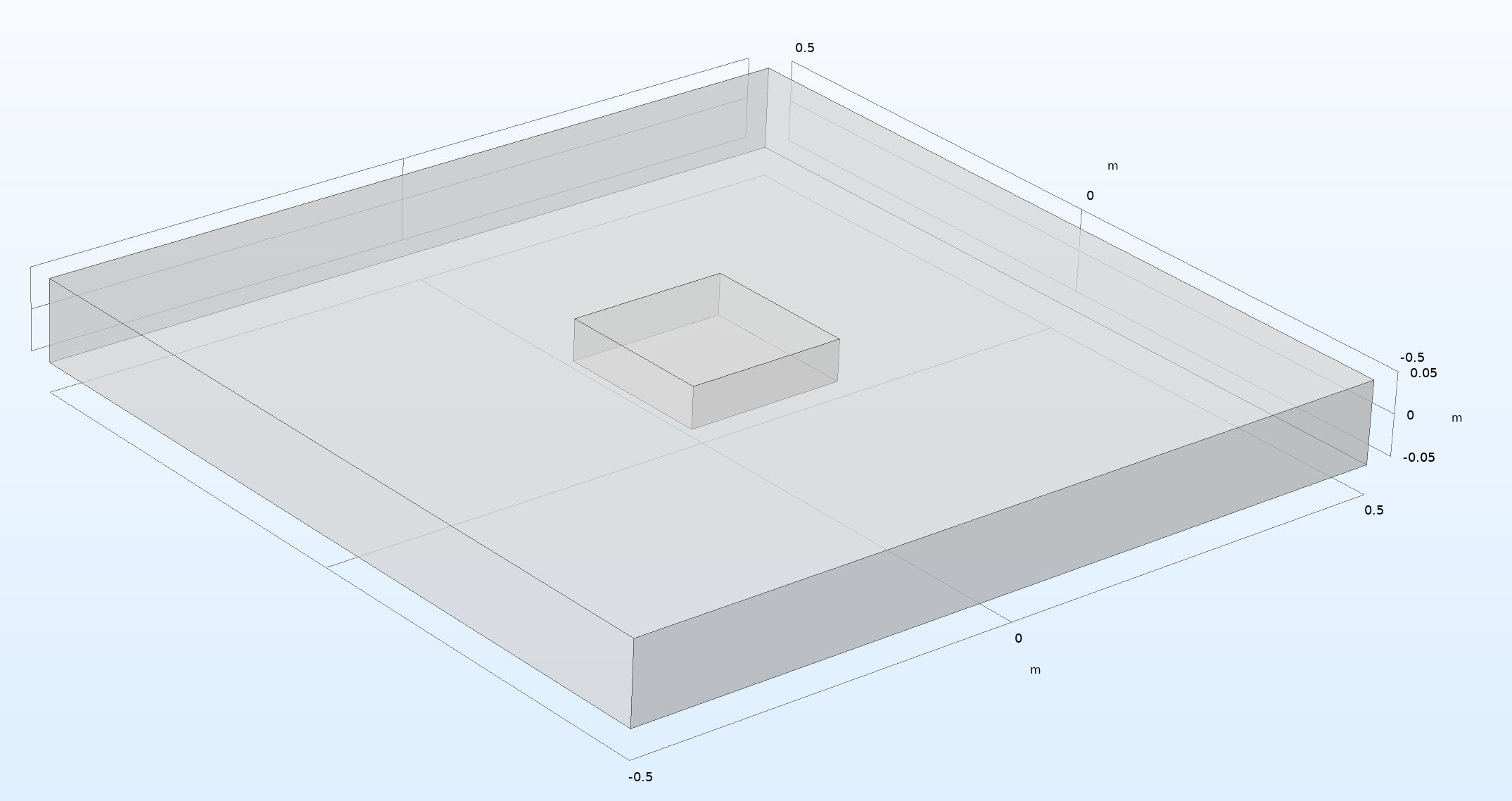
Which brings you to the following window:



1. In the Settings Panel, change “Length unit” from “m” to a unit that makes more sense for your application (i.e. ). In “Solid Mechanics”, you can also change the order of the FEA solver as shown below; this is the order of the polynomials used to compute the strain in each volume.

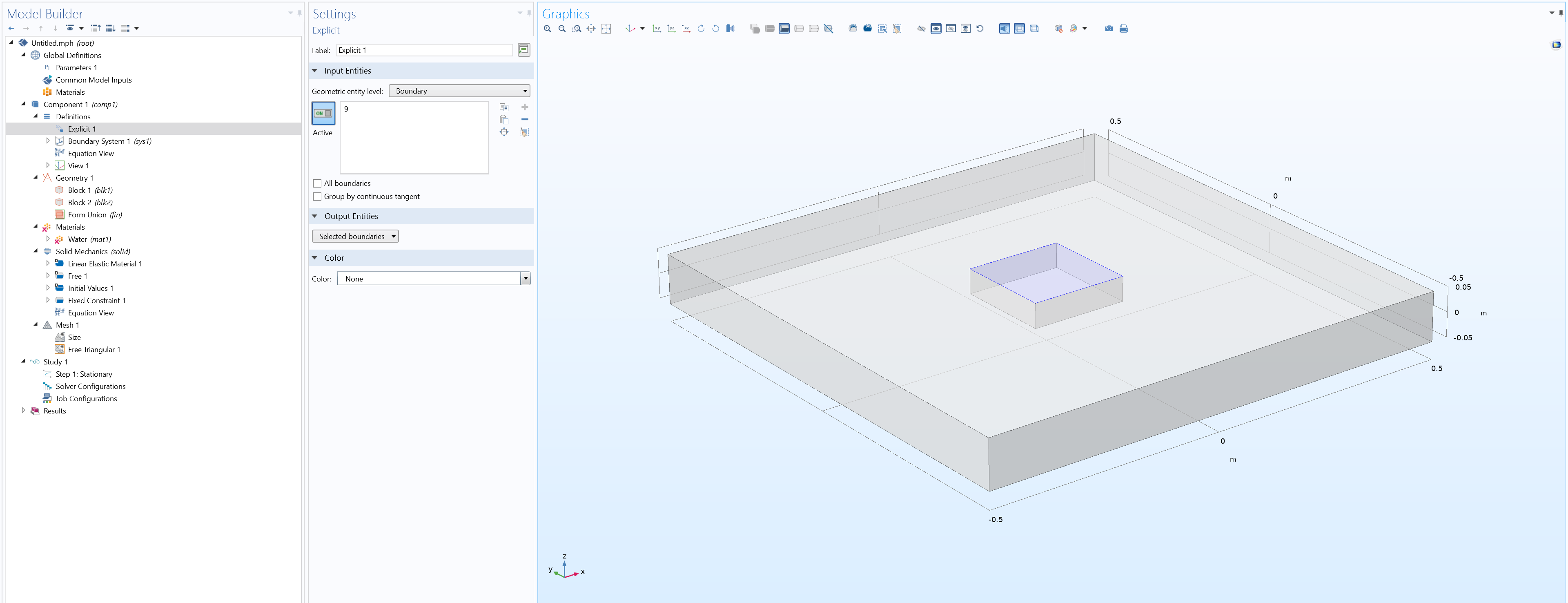


1. From some previously generated COMSOL file, click on “Application Builder” in the top left corner and copy both scripts over to your new file. In the “Method” tab on the top, click “Settings” and on the RHS change the name of the files to “GreensMatDataStudy” and “dataQueryFromCSV” respectively.
2. Use the “Geometry” Tab at the top to create the desired geometry. Also use these tools to enclose the region you want to mesh more finely (see example below):



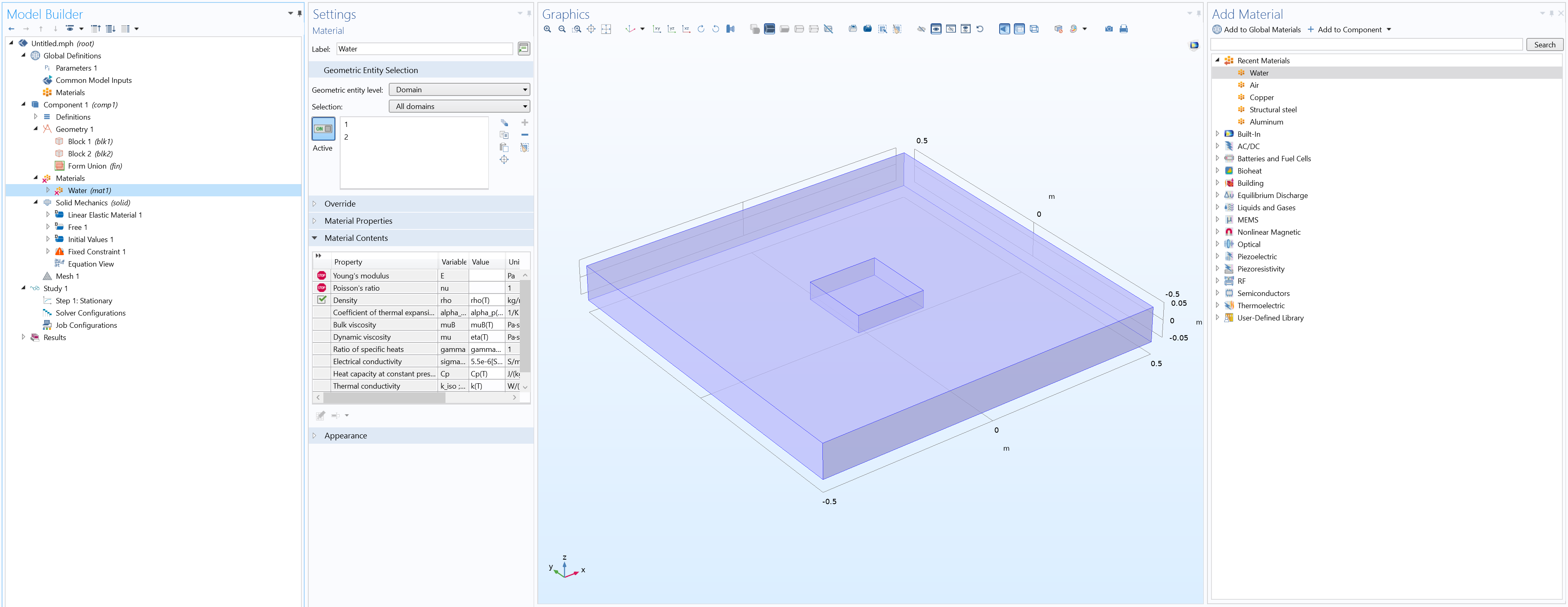
In this example, the cells will live in the smaller block which rests in the center and top half of the large block.

1. In the LHS pane, right click “Definitions”, then under “Selection”, click on “Explicit” (see below).

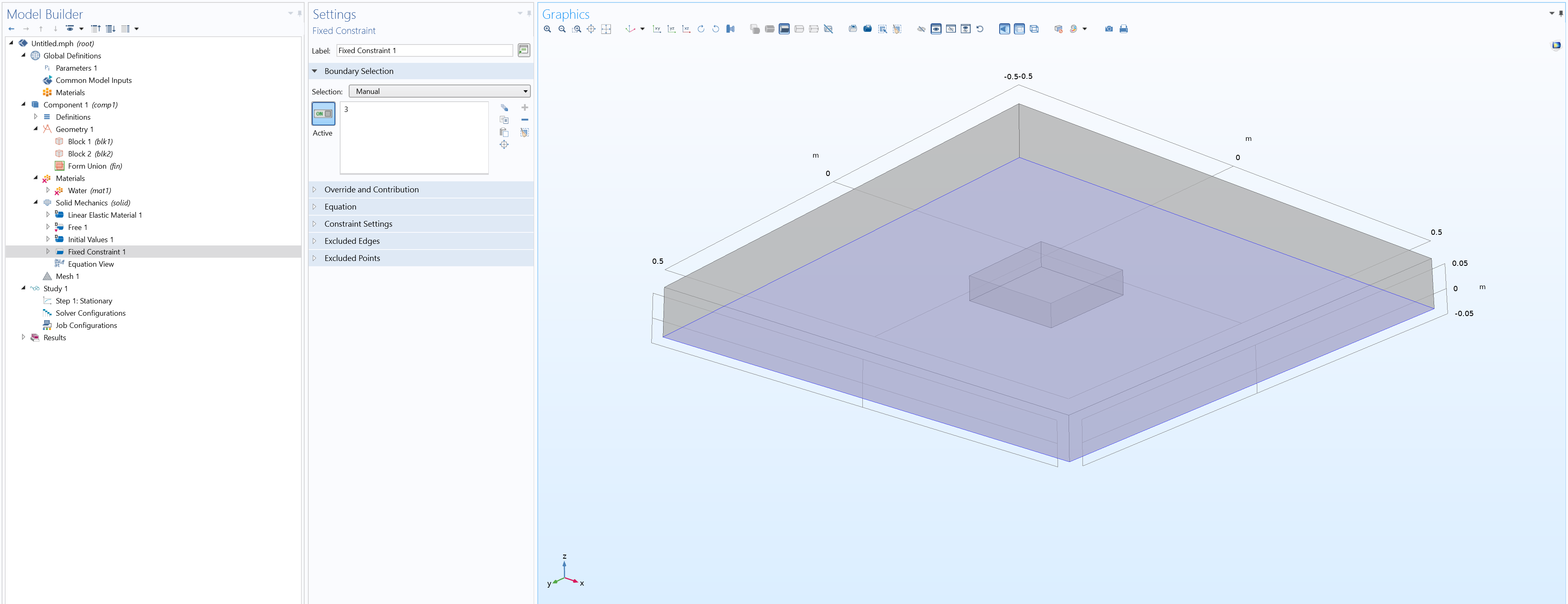


In the adjacent Settings pane, change the input “Geometric entity level” from “Domain” to “Boundary” and select the surface that the cells will rest on, as shown above. This is done to let the script know where to look for surface mesh elements to apply forces on.

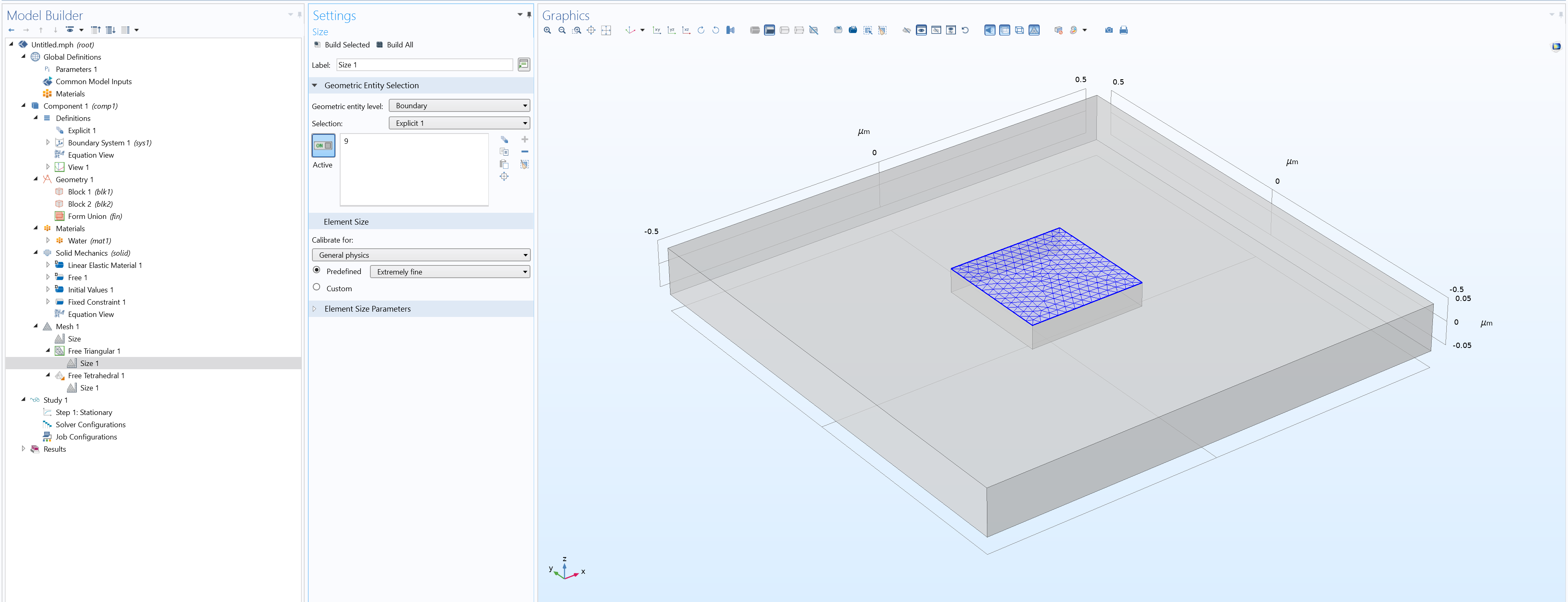
1. Give the geometry a material by right clicking on “Materials” on the LHS pane and select “Add Material from Library”. Highlight any material you want (i.e. water) and select “Add to Component”. In the settings for Water on the LHS, you can change the Young’s Modulus, Poisson’s Ratio, and Density to match the exact desired properties.



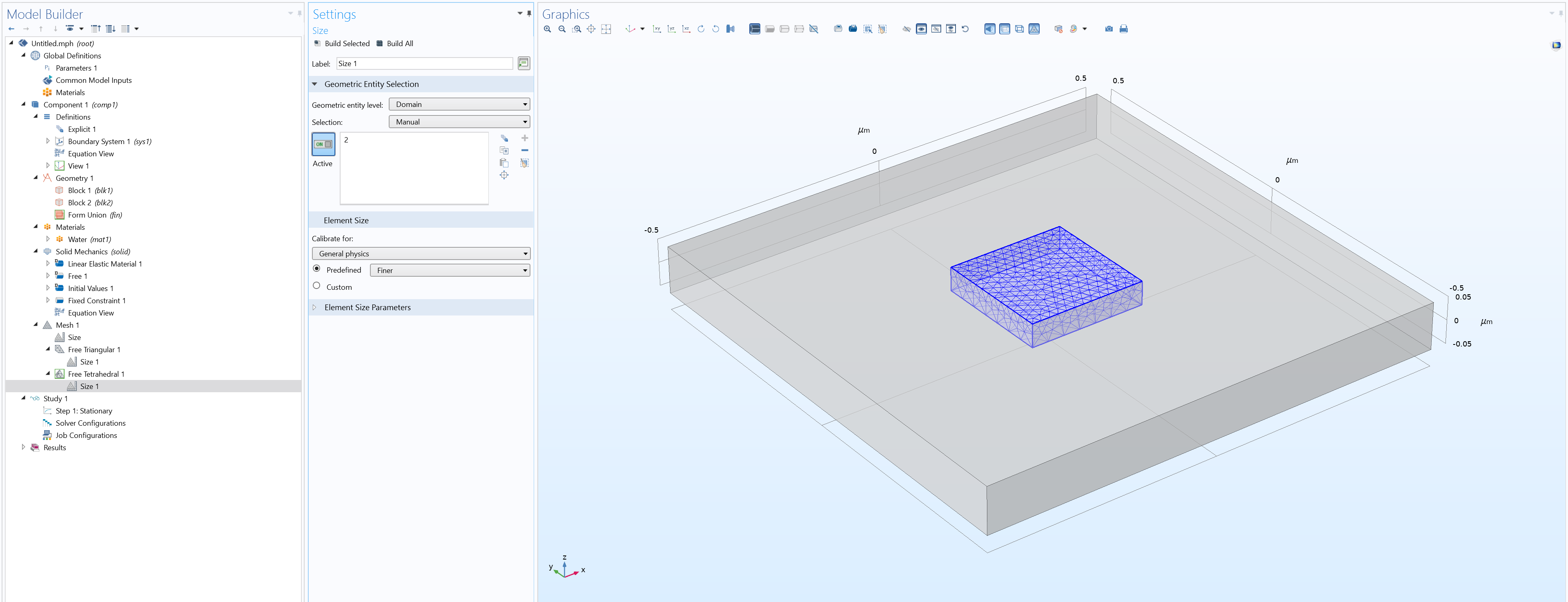
1. Add a fixed boundary condition by right clicking on “Solid Mechanics” on the LHS pane and select “Fixed Constraint”. You will almost certainly need one to represent the bottom of the cover slip. On the geometry you’ve created, select the boundary that the condition should be applied to, and it will light up as seen below



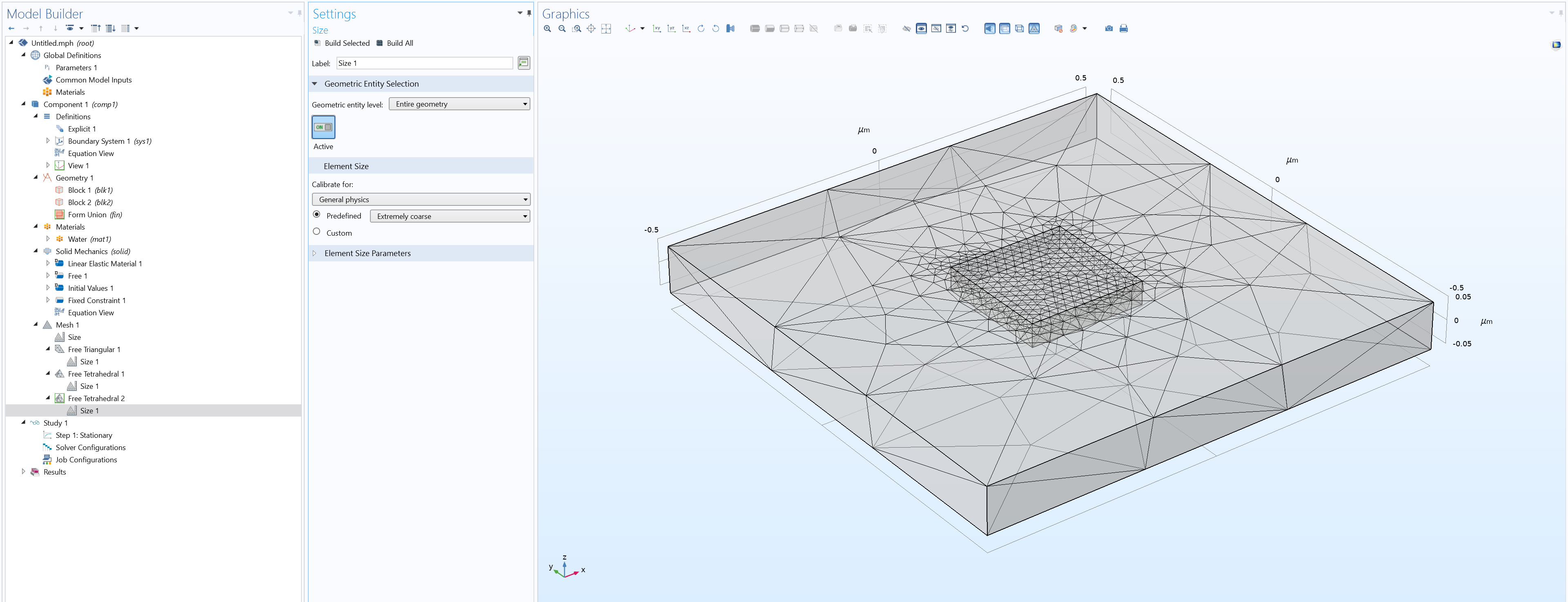
1. Mesh the geometry by right clicking on “Mesh 1”. First, under “More Operations”, select “Free Triangular”. Change the input “Selection” from “Manual” to “Explicit 1”. Right click on “Free Triangular 1” and select “Size”. Here, you can adjust the mesh fineness for just this surface, as shown below:



Right click “Mesh 1” and select “Free Tetrahedral”. In the settings change “remaining” to “domain” and select the volume you would like to be more finely meshed:



Right click “Free Tetrahedral 1” and select “Size”. Right click “Mesh 1” and select “Free Tetrahedral” again. This last mesh will be for the remaining geometry. Right Click “Free Tetrahedral 2” and select “Size”. After setting the mesh sizes and building, your geometry will look like:



## Running the script

Before running any COMSOL methods, go to Developer > Application Builder > GreensMatDataStudy and make sure everything in the top checklist is correct. You will need to input the Afterwards, return to Method > Model Builder, and under Developer > Run Method, select GreensMatDataStudy. This script will set up the computational structure COMSOL will follow. Once this is done (it will only take a few seconds), go to “Study 1” on the LHS and in Settings click “compute”. This program will run for a few minutes before completing, but depends heavily on the mesh size and FEA solver used.

After this program executes, you can verify that the values were collected properly by examining Results > Stress (solid) tab in the model tree on the LHS. Once this is confirmed, go to Application Builder > dataQueryFromCSV and go through this checklist. In the Model Builder, run Developer > Run Method > dataQuery. The data needed for Step 3 will be saved to data/nodalSolutions.csv.

# STEP 3: Inverse Problem Solution and Regularization

After Steps 1 and 2, you now have the data corresponding to u and G in the stress-strain relation. All that is left is solve the inverse problem for .

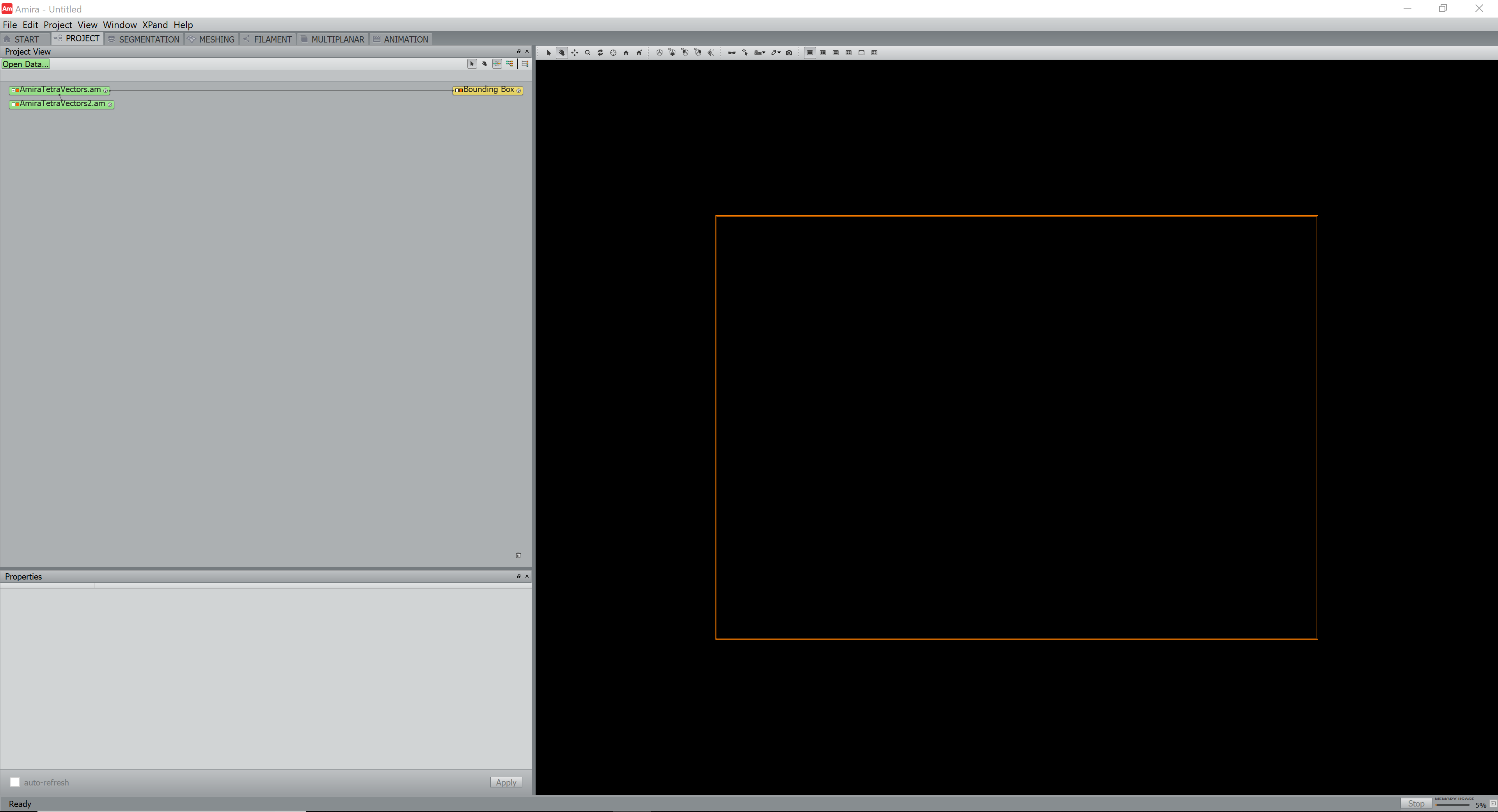
The MATLAB scripts in the "regularization" folder begin by reading in the data from Steps 1 and 2 and formatting it into the correct objects. The script computes the SVD of G before feeding it into the regularization function. Before regularization, the formatted data and SVD are saved into a MAT file so that this need not be repeated for the same set of data (particularly for the SVD, which can take several minutes). Currently, L2 (i.e. Tikhonov) regularization is used, which seeks to find the minimum of where is a regularization parameter. functions from PC Hansen's Regularization Toolkit is used to streamline the computations. After the initial setup, the code moves to l\_curve.m, which generates the L-curve for a wide set of regularization values. Through a maximum curvature algorithm, l\_curve then determines most optimum solution norm and outputs the regularization parameter. The function tikhonov then takes this parameter and fully calculates . The script ends with plots of the input and output data. It also creates an Amira Mesh file that contains two sets of data – first, the locations on the surface where the cell rests and forces are applied (i.e. the center of all the triangles) and second, the force vector computed from the regularization. An Amira Mesh file is created for each timepoint that was used in running the particle tracking code, and they are all saved in the “plots” folder with the files that start with “AmiraTetraVectors”.

## Running the script

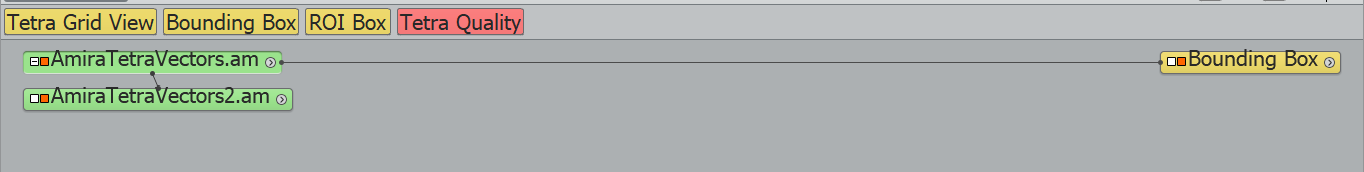
All operations are performed through main.m. All that is necessary for this script is to verify that the constants given in the beginning of the file are defined correctly. See comments above each constant for a description.

# STEP 4: Generating Figures in Amira

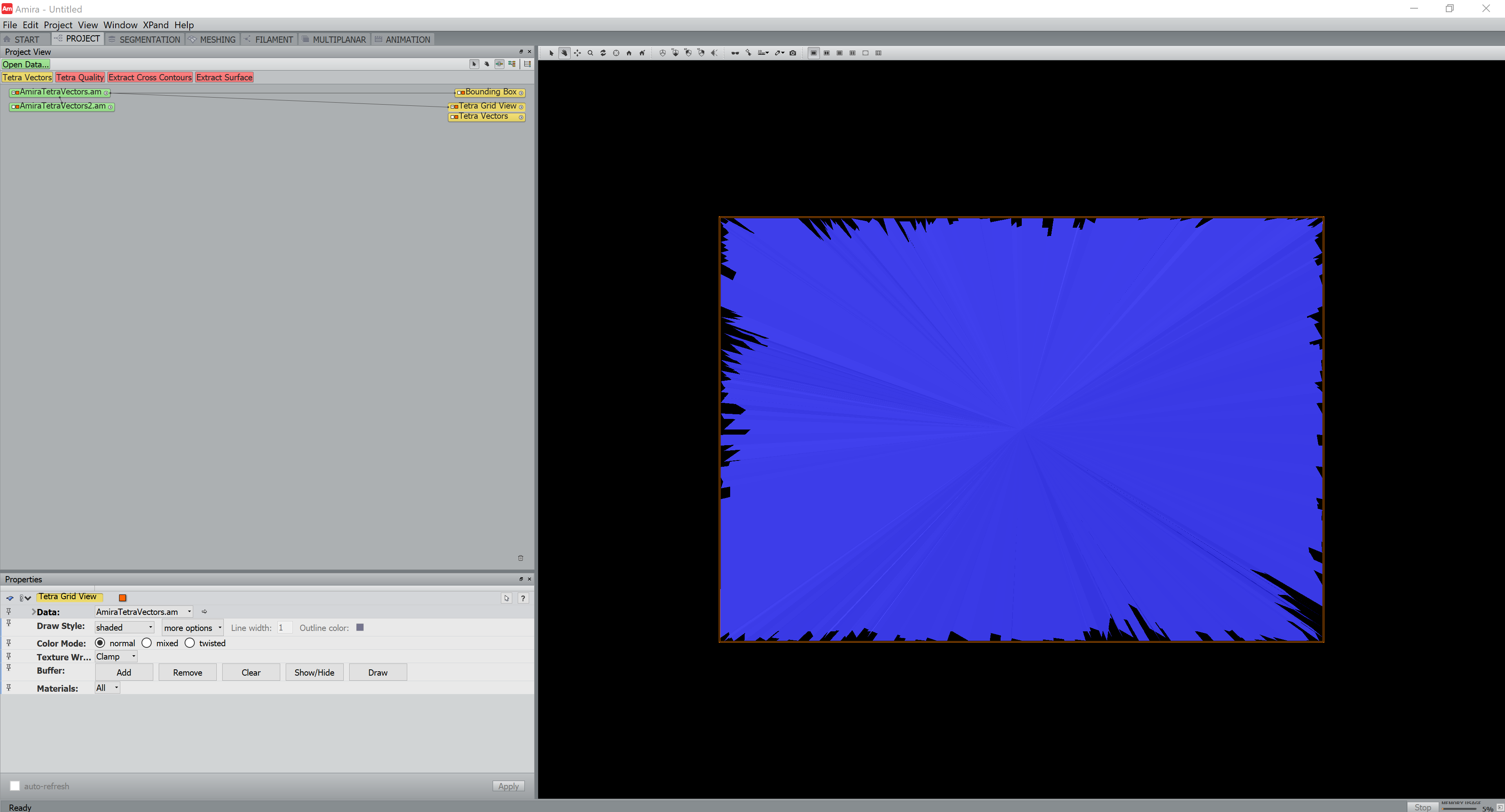
No more math now! The next step is to load them into Amira for visualization purposes. Starting with a blank Amira file, simply drag one of the AmiraTetraVectors files into the left hand side of Amira. You will then have what is below:



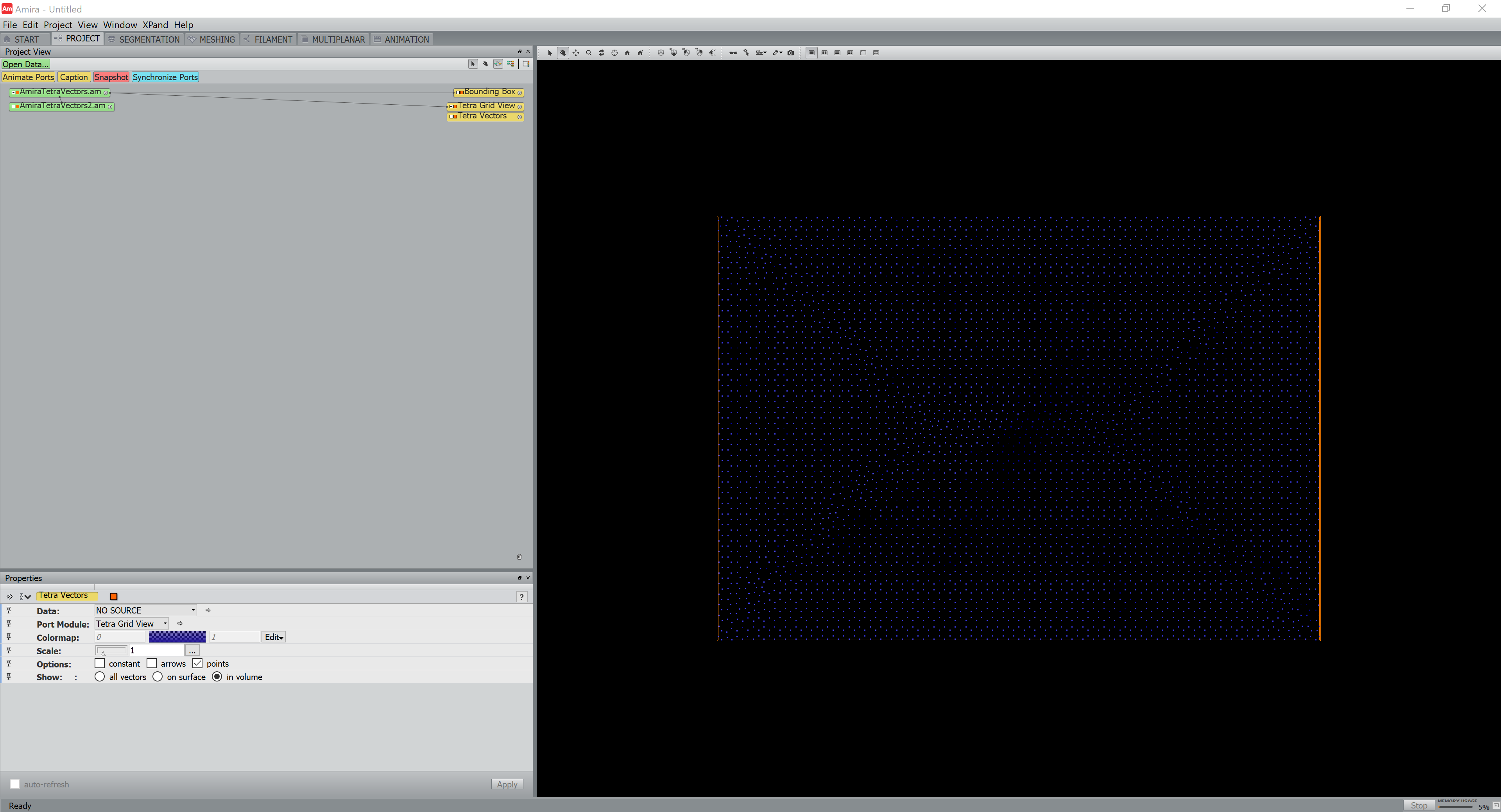
Click on “AmiraTetraVectors.am” (the one on top) and another menubar will appear in the top left:



Click on Tetra Grid View. Then, you will have the option to click on Tetra Vectors. You will then see a big blue mess like below:



In the bottom left, change Draw Style from “shaded” to “points” to see the locations where forces will be displayed. Now, click on Tetra Vectors in the Project View to get the following view:



In the bottom left, change Data from “NO SOURCE” to AmiraTetraVectors2.am (the one on the bottom). You may then be able to see some forces in view, but they are the same color as the points. In the Colormap property, click “edit” to change the color scheme and use the slidebar to change the color range. Then, you can get an image that looks like:

