# StaticsCalc

Tej Modi & Eric Noordam

E177 Spring 2015

**Abstract**

Sizing and choosing materials for regular shapes, such as tubes and prisms can be very time consuming. StaticsCalc aims to improve this process by allowing the user to quickly calculate factors of safety against multiple loads for tubes of any material. The results can be saved and compared against alternative tubes. In addition, it also compares the tubes' weight. The material database is easily expandable and includes both uniform materials and composite materials.

## Introduction

This project was born from a frustration of continually looking up calculations and material properties for hollow tubes while designing our Formula SAE car. Berkeley Formula SAE designs, builds, and races a Formula style car every year. Many of our components, including suspension members and our drive shafts can be modeled as simple beams and shafts. It is important that these components are both as light as possible and have high enough factors of safety to withstand their loads. StaticsCalc's GUI allows the user to enter in types of loads, materials, and tubes.

## Theory

StaticsCalc uses material properties to calculate factors of safety to loads. Isotropic material data is straightforward; it is readily available data from tables. However, StaticsCalc can not only utilize uniform materials, but also use many composite materials as well. Composite materials are made by layering sheets of fiber, made of materials like graphite and glass, in a matrix of a bonding agent. The fibers are usually incredibly strong in tension, but less so in all other directions.

Laminar plate theory combines the matrix and fiber properties into properties of the newly formed composite material. These equations use the material properties of both the matrix and the fibers to calculate a strength matrix for each ply of material. These matrices are then combined into a stiffness matrix for the entire material. From these matrices material properties of the combined structure can be estimated. When designing with composites it becomes very clear how important the fiber orientations are. When most fibers are oriented perpendicular from the load, the part will not be stiff and will fail quickly.

Once the material properties are determined simple beam equations are used to find the failure points for each load with the given geometry. These calculations are compared to the failure points of the materials. For isotropic materials these are either yield, ultimate, or arbitrary fatigue stresses determined from an SN curve. For composites the strain of the fibers is more closely considered as well. The calculations consider simple loading scenarios, such as being fixed at one end of the tube and torqued on the other, or a cantilevered beam loaded on its end are solved for using equations learned in ME C85. A more complex torsion failure, torsional buckling is also considered. For the compressive buckling load, the users is prompted to provide the effective beam length factor from a diagram we provide.

## Class Descriptions

**StaticsCalc** - Is a handle class that initializes our GUI. It's starts by asking users to open the calculator or results viewer. The *LoadGUI* method opens a GUI that allows the selection of which types of loads users wish to input and their values. These are stored in **Loads**. *MaterialGUI* allows the selection of material, uniform or composite, and type of tube, round or circular. The data in the GUI is used to create **Tube** and **Material** classes. The *ResultsGUI* and *GraphGUI* methods create GUI to open **Results** classes and view their properties.

**Material** - Material is an abstract class that contains the material properties used by other classes. The material properties are used by **Tube** to calculate factors of safety to various loads. The material properties required are the Elastic Modulus (in 2 directions), Shear Modulus, Poisson Ratio, and density. A failure criteria is also used.

**Uniform** - This class creates isotropic materials. It inherits and substantiates properties from **Material**. It has a method *openReference* that will open a web link to more information on that material. Uniform takes its data from text files stored in the uniform directory. The failure property is either the yield stress, ultimate stress, or an arbitrary stress. This stress may have been set using our fatigue curve feature. Materials also have names.

**Composite** - This class inherits from the abstract class **Material**. Its inputs are a **Fiber** class, **Matrix** class, name of a layup, as well as a few properties of a composite layup. These properties also have default values. The layup name is used in *setLayup* to read the fiber orientations from a text file. The *laminaProp* and *densVol* methods are used to perform the laminate plate theory calculations to determine the structures material properties.

**Fiber** - Fiber contains the material properties of the fiber used in a **Composite**. Its only input is a fiber. It uses *importProperties* to read these properties from a text file. The fiber data contains many properties that vary depending on the direction of loading. For example, 2 elastic and shear moduli are required. An ultimate strength of the fibers is also used. Fiber Aerial weight is a measure of the weight of the fibers over a given area. This is used in **Composite** to calculate a density and ply thickness.

**Matrix** - Matrix contains the material properties of the fiber used in a **Composite**. Its only input is a Matrix. It uses *importProperties* to read these properties from a text file.

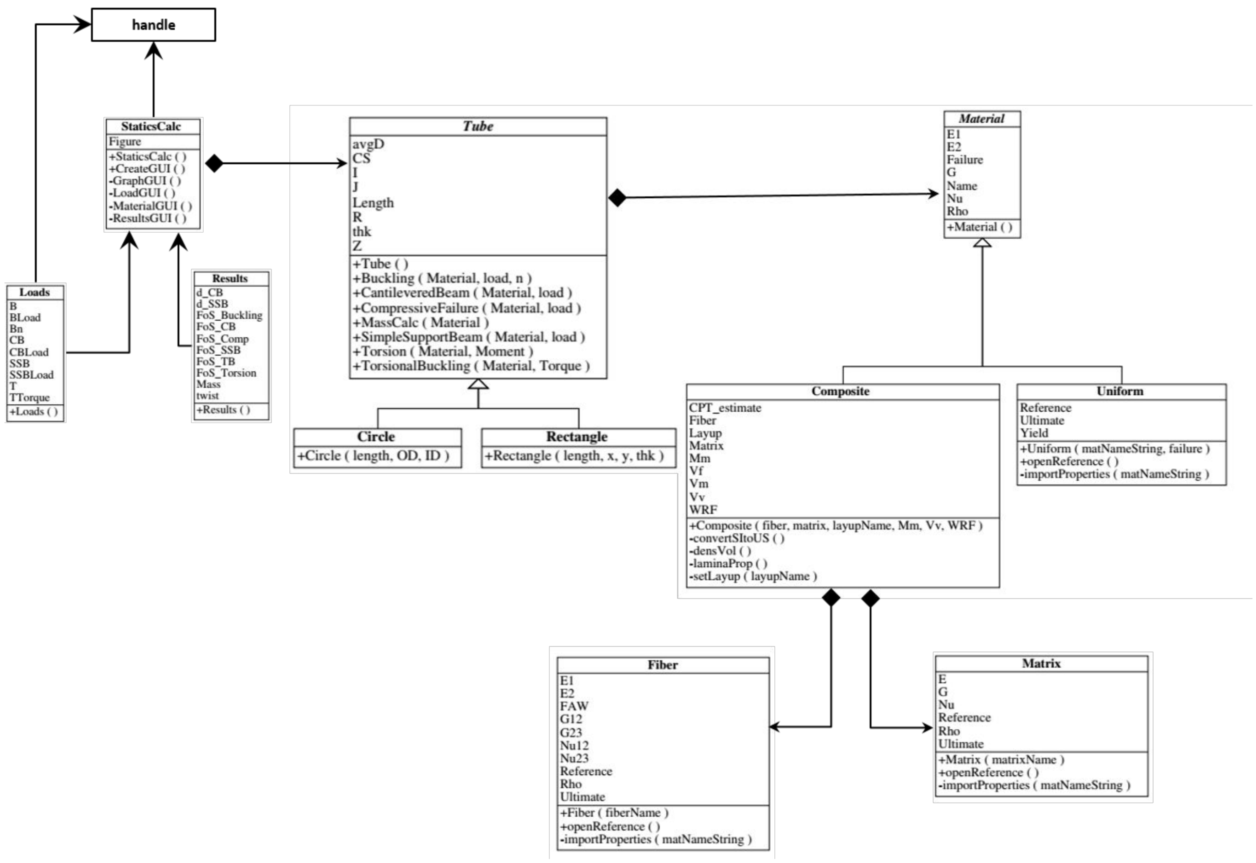
**Tube** - Tube is an abstract class. It contains properties of the tube geometry such as length, cross sectional area, and moment of inertia. Tube contains methods, such as *CatileveredBeam*, that use the Tube's properties, a **Material**, and a force to calculate a factor of safety and displacement. These values are saved into Result class by **StaticCalc**.

**Circle** - This class represents a tube with a circular profile. It inherits from **Tube**. It's constructor substantiates the **Tube's** properties from the length, outer diameter, and inner diameter of the tube.The circle class is created by the **StaticsCalc** class.

**Rectangle** - This class represents a tube with a rectangular profile. It inherits from **Tube**. It's constructor substantiates the **Tube's** properties from the length, x width, y height, and wall thickness of the tube.The circle class is created by the **StaticsCalc** class.

**Loads** - The loads class stores which loads the user wishes to perform. It's properties contain booleans for which of our 6 loads are active and values determining the forces. The loads are compression, buckling, torsion, torsional buckling, cantilevered beam bending, and simple beam bending. It is created by **StaticsCalc.**

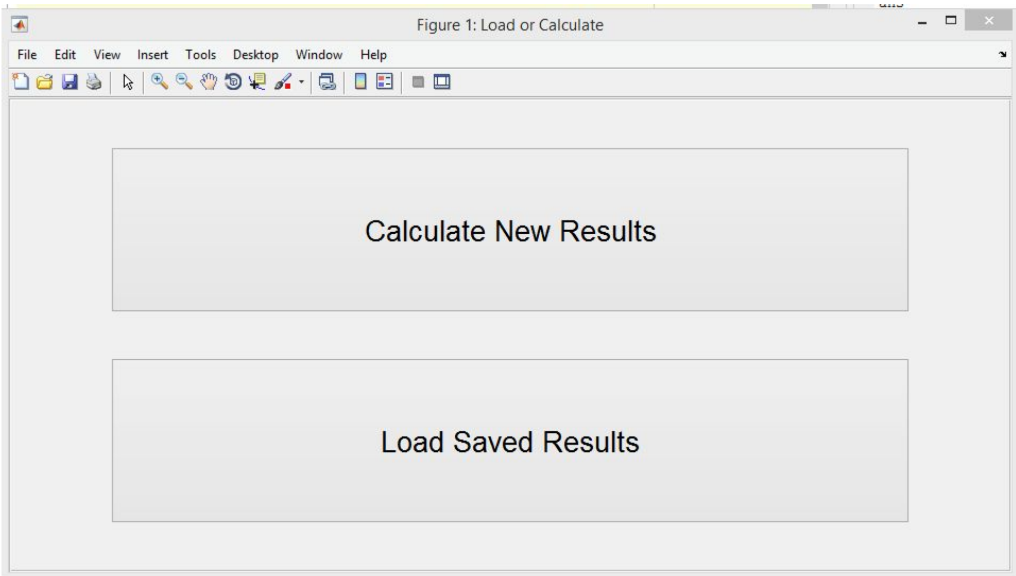
**Results -** The results class is a storage container for the factors of safety calculated by our calculator. The Factors of Safety are outputted by Tube classes. The **StaticsCalc** class reads Results classes and compares them on a bar graph.



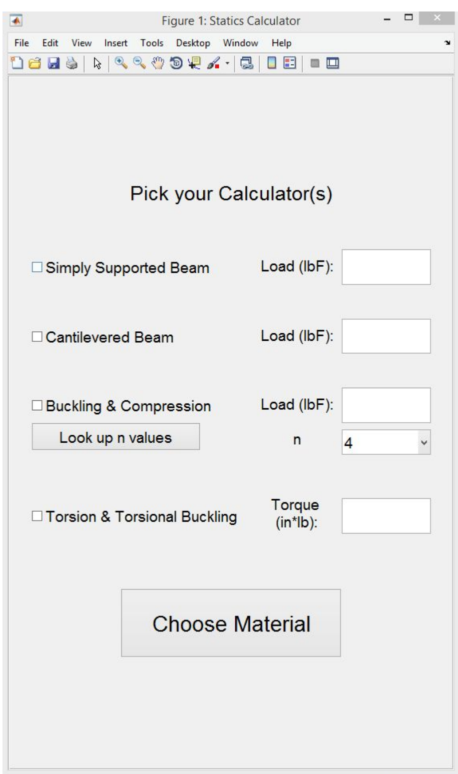

**Using StaticsCalc**

**Procedure:**

1. To run the program, first set MATLAB's working directory as the code folder or add it to the path. Next type in StaticsCalc; . This will open the first window:

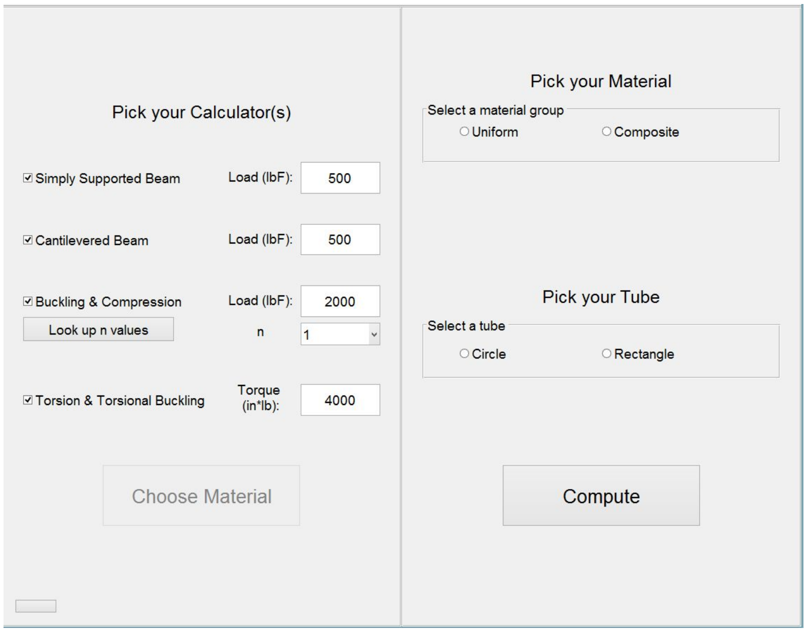



2. Here the user has the option of either creating new results files or loading saved results files.
3. If the user clicks Load saved results, skip to step 17. If the user clicks the Calculate New Results button, the next GUI window pops up. This is where the user will pick the calculations he or she wishes to perform as well as the loads he or she wishes to apply.



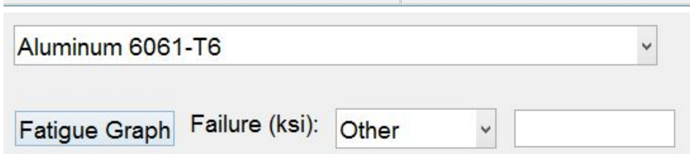

4. To pick which calculators to run, simply check the box next to the calculator name. Then fill out the load at the end of the row. If a calculator is picked but no load set, the code will error.
5. To complete the buckling calculations, the user must also pick an n value from the drop down menu. If the user is unsure what n value to pick, click on the “Look up n values” button. The button will pop up an image to explain what every n value means. Then close the pop up window and set the value.
6. Once all the calculators are selected and the appropriate loads filled out, click on “choose material”.

7. Next the user has the option of either picking an Uniform material (ie metals, plastics) or Composite material (ie Carbon Fiber, Fiberglass, etc). Simply click on the

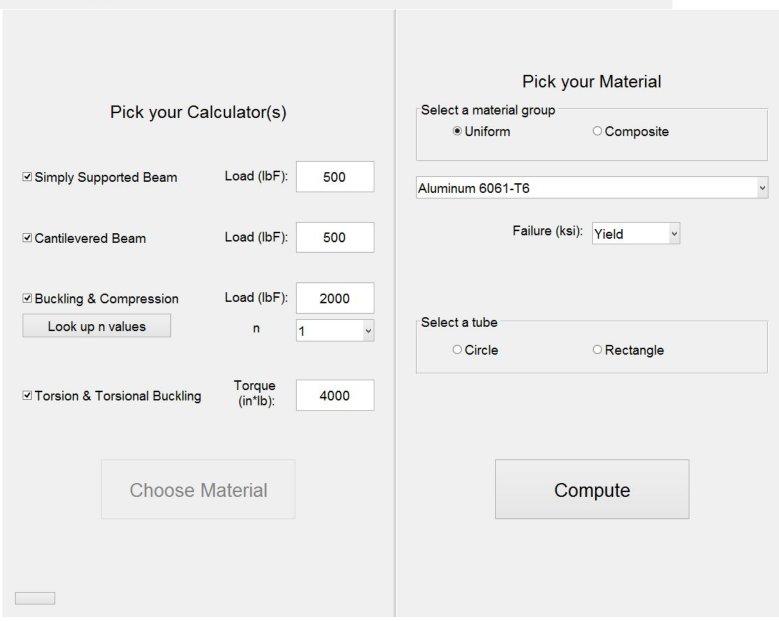



appropriate radio button for the application to set the material.

8. If the user picks Uniform, he or she can pick a material from the drop down menu of pre populated materials. Next, pick the failure point for which a FoS should be compared against, the choices are: Yield, Ultimate, and Other. Move to step 11 if the user picks Ultimate or Yield.



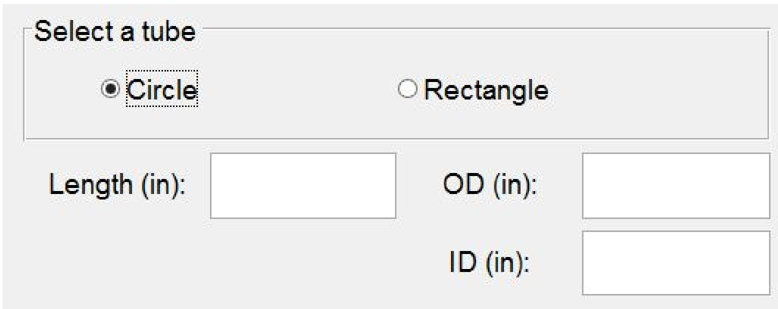

9. If the user picks "Other", the user will need to manually set a failure point by typing in a failure point in ksi in the new input field. If the material selected has a known SN curve, a button saying "Fatigue Graph" will also show up. If the user clicks on Fatigue Graph, a new window will open with the SN curve. Once the user knows what to set the failure point at, close the SN curve window and type in the failure point in the appropriate field. If however, the material does not have a known SN curve, no button will show up.



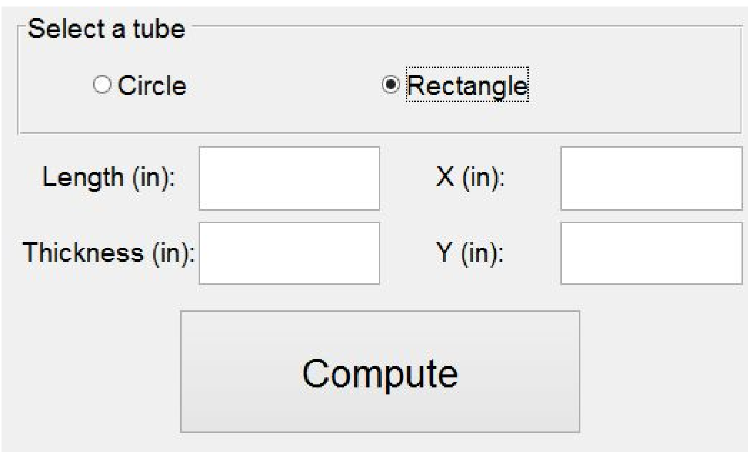

10. If originally, the user picked Composite as the material, the user needs to pick the which fiber, matrix, and

layup from the appropriate drop down menus. These are read from stored data files.

11. Next, the user needs to pick the shape of the object. If it is a circular tube click on the Circle radio button. If it is a square and rectangular tube, pick the rectangle radio button.
12. If the user picked, Circle, the user then needs to fill in the length of the tube, the outer diameter (OD) and the inner diameter (ID) in the appropriate boxes. If the ID is greater than the OD, the code will error. If it is a solid tube leave ID blank or set it to 0.



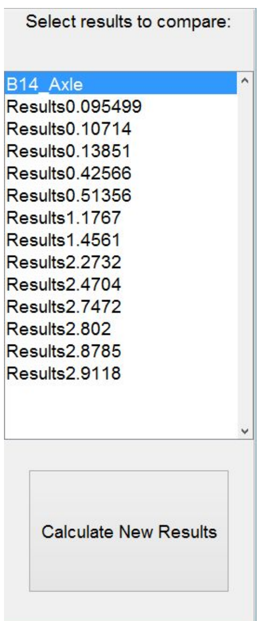

13. If the user picked Rectangle as the tube shape, the user then needs to fill in the length of the tube, the base length (X), the height (Y), and the wall thickness of tube. The code will error if the Thickness is too large. If it is a solid tube leave Thickness blank or set it to 0.



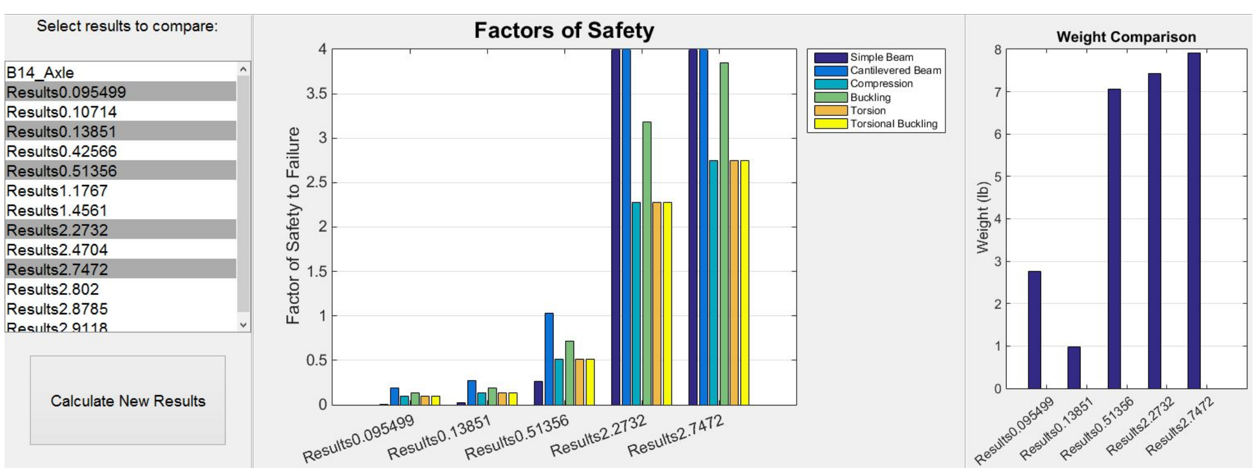


14. Finally, the user can select Compute to calculate the FoS of all the selected calculators.

15. The result is a FoS graph of all the calculations performed. In addition, it also displays weight. If the user wanted to save this data to compare later, type in the name of the file where it says Enter File Name, and then click Save Results. The results will now to be viewable in the Saved Results window. If the user wishes to perform another calculation, simply change the appropriate values and buttons, and click Compute once again. This will update the results graph.
16. To compare the results against other saved results, click on Load Saved Results (this will take the user to the same window that would have popped up had the user picked Load Saved Results at the very first UI window). This will open a list of result files:




17. By clicking on any of these files, the UI will update to show the results of that one run. In addition, if the user holds down the keyboard button ‘ctrl’ and then picks multiple tubes, multiple tubes will be displayed on the right.



18. The tubes will be grouped by their file name. The colors represent the different calculators used. Here the tubes can be compared side by side against other alternatives. Furthermore, on the far right, the tubes are also compared by weight.
19. Now if the user wishes to do another calculation, the user can click on the Calculate New Results button in the bottom left. It will take the user back to step 3.

Contributions:

Eric - 50% - Wrote the material classes, property importing code, carbon fiber analysis, laminate plate theory code, and results comparison GUI. Wrote half the report.
Tej - 50% - Wrote the most of the GUI and the tube, circle, and rectangle classes. Wrote half the report.