IMPLEMENTATION DOCUMENTATION

This project was about making an excel worksheet and series of existing functions user friendly. The layout of the worksheet made the professor worry about user experience, especially for those who were not too familiar with the inner workings of elliptic integrals. I created a user form that takes up the whole screen when the excel file is opened, see Figure 1.

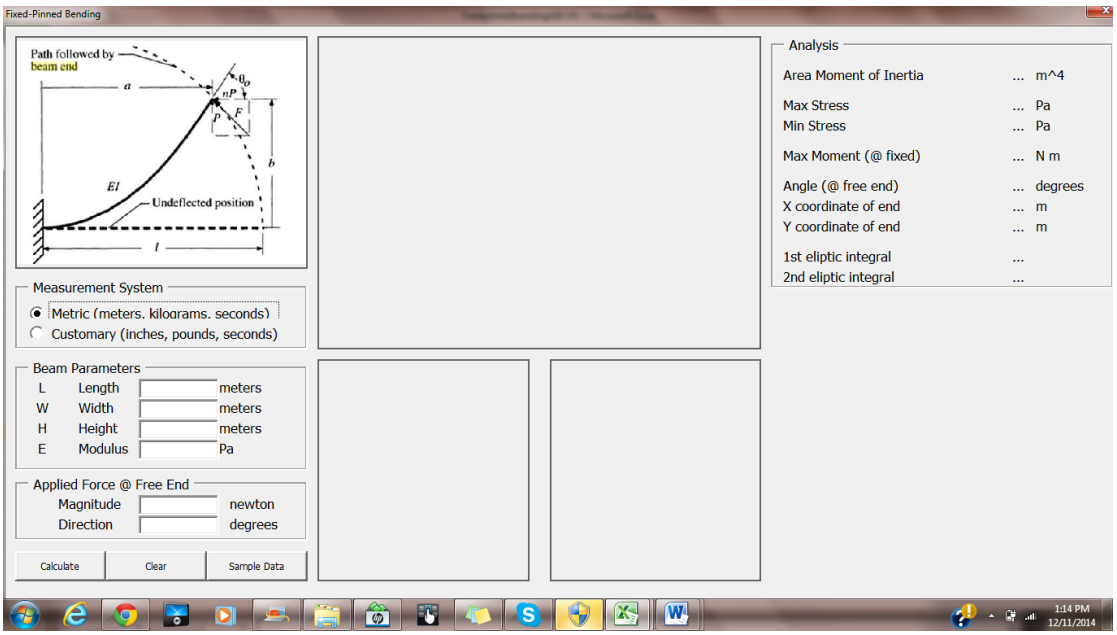

Figure 1: The user form automatically appears as the excel file is opened. Additionally the user form takes up the entire monitor as it has some scaling built into the code.

The user then fills in the required parameters through text boxes. A sample data set can also be input through clicking on the “Sample Data” command button. The results are shown in Figure 2.

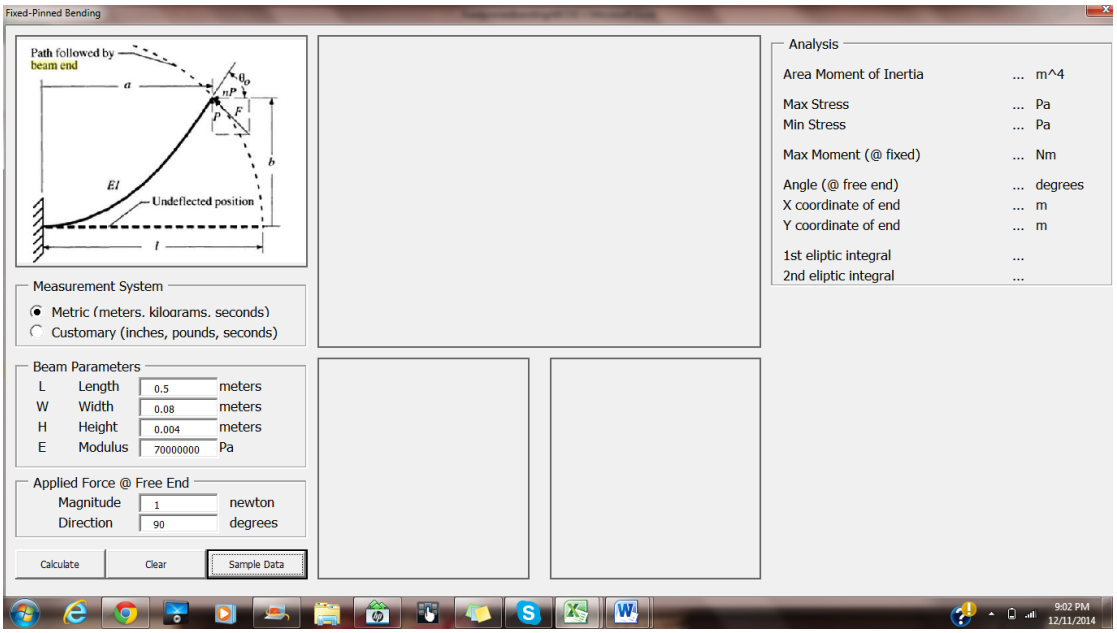

Figure 2: The user inputs the parameters for the design into the text boxes. Sample parameters can be uploaded by choosing the "Sample Data" button.

Once the parameters have been entered the user clicks on the "Calculate" command button to perform the analysis. In the background an optimization is taking place via solver working with two elliptic integral functions and their outputs to iterate to a solution. The solution comes in the form of 3 charts and various outputs of interest as shown in Figure 3.

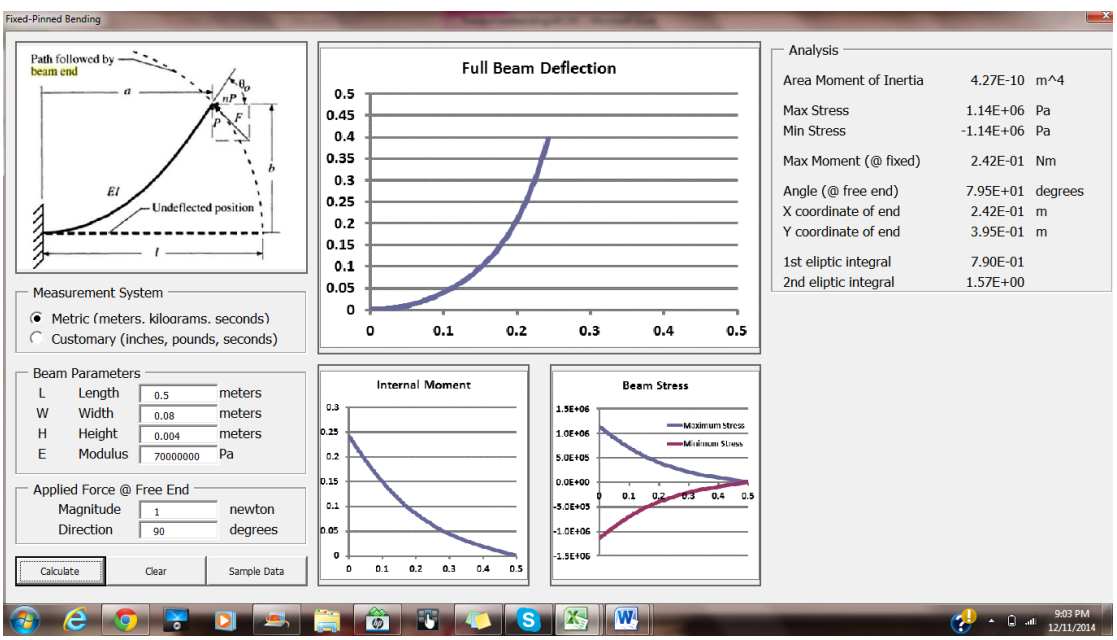

Figure 3: When the "Calculate" button is selected the program runs. The results are written to the appropriate analysis areas, both graphically (in the charts) and numerically (in the analysis frame).

Further the user can supply parameters in either the metric or the customary measurement system via the selection of the appropriate radio button. The units labels adjacent to the text boxes as well as the labels of the analysis outputs are updated to the appropriate measurement system, see Figure 4.

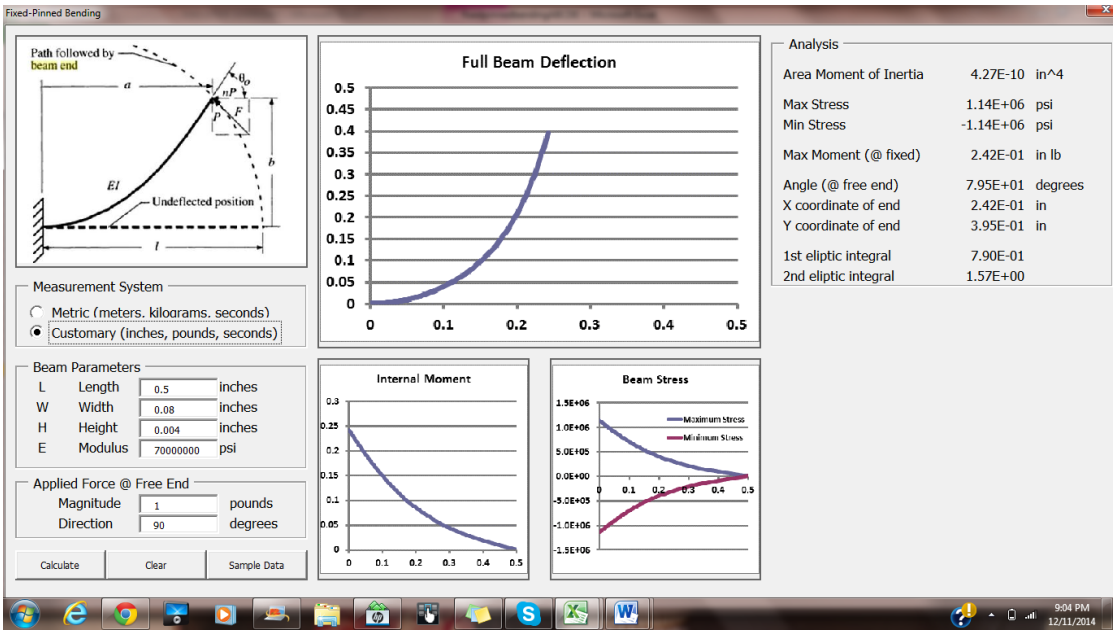

Figure 4: The user can input units in metric (meters, kilograms, seconds) or customary (inches, pounds, seconds) via the radial selection button.

As the length of the beam and other parameters are changed the code selects the appropriate scaling of the charts, see Figure 5.

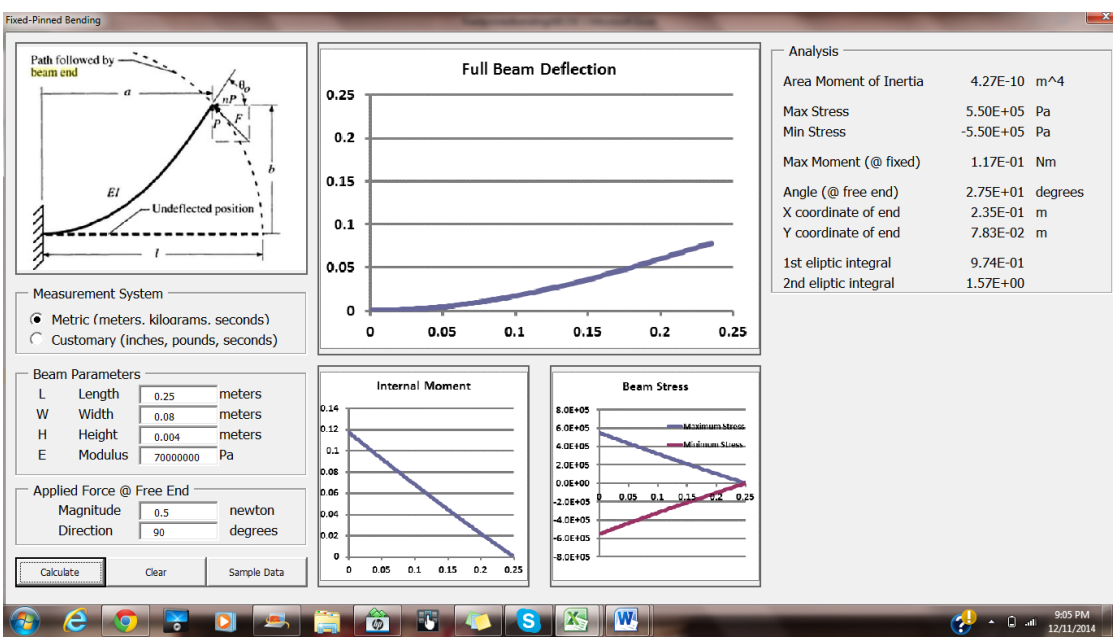

Figure 5: The parameters can be altered and the calculate button selected to analyze different design scenarios.

If the user puts in non-numeric values an error will be given.

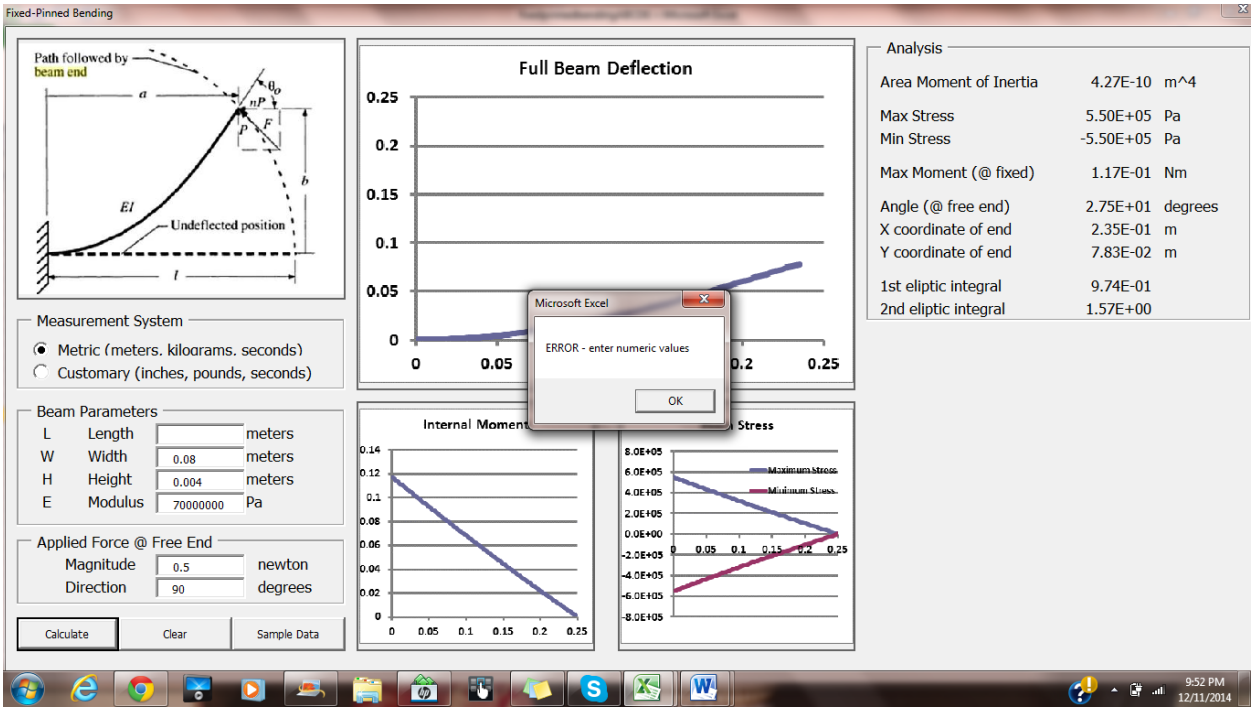

Figure 6: Warning error given if non-numeric data entered and the "Calculate" command button selected. The analysis is exited, the text box with the invalid entry cleared and the user is able to enter a numeric value.

Detail on Components:

In creating the user experience I wanted to make the interaction with this model more bulletproof. One way I did this was to have the user form scaled to the size of the screen, so it completely blocked off the spreadsheet. As a result the user is led to naturally interact with the user form, and nothing else.

The user then inputs data and clicks on the "Calculate" command button. This is where the magic happens. This triggers a series of events that goes as follows:

1. The inputs (text in the text boxes) are checked to ensure they are numeric values. If they are non-numeric values (characters, symbols, blanks, etc.) the specific input that is non-numeric is cleared, a message box instructs the user to input numeric values, and the program exits the execution phase and the user can enter proper numeric values. If all are numbers the program continues on.
2. The inputs are written to the excel worksheet and a solver optimization is carried out. This solver routine uses function evaluations for two elliptic integrals (elliptic integral code was written by a different author and was not in the scope of the project) and iterates till the stopping conditions are met.
3. A separate worksheet provides detailed intermediate evaluations of the properties along the beam. The values on this worksheet are referenced and worksheet functions select critical data to be reported in the analysis section. Values reported in the analysis section are formatted into

scientific notation as they vary greatly in orders of magnitude (and reading that many zeros can be challenging so the engineering convention is to use scientific notation).

4. Charts are generated in the Worksheet from the intermediate evaluations. These charts are then put into the picture frames in the user form.

Also included in the form is when the sample data is uploaded it changes the measurement system to metric as the measurements are meant to be in metric.

The graphs are optimized to the parameters of the beam. The scaling of the x-axis was scaled to be the length of the beam (as the x-axis is the original beam with no force applied). Likewise in the full beam deflection the y-axis is scaled to be the length of the beam to provide a closer pictorial representation of the situation.

I did not include any ribbon functionality as I wanted to direct the user's attention completely on the form. Also the form and the functionality through the command buttons is in the style of some engineering user interfaces.

The blank space in the bottom right corner of the input box is for further development. I wanted to compare this method to a simplified, but relatively precise method Dr. Howell has created, the pseudo-rigid body model. This will be done at a future date as it requires me to code in many tables (and V-look-ups would be used off of those tables).

LEARNINGS

This project was tough as it seemed like every component required me to learn new things. However through my pattern of having to learn via the other projects this semester I was able to find the assistance I needed through online posts.

One of the first things I learned about was the limitations of dealing with charts in user forms. I read that the simplest way to deal with them was to create them in the workbook and then export them as a picture and save them in the same file path. Once saved outside the workbook they could be inserted into the form. I also learned about auto-sizing them for the best result. Further dealing with the charts in a loop has provided additional challenges. Sometimes when I open the workbook and run the program, an error occurs. However when I add a breakpoint in the loop and increment through it the first time subsequent times work well. Don't know why. I also thought it would be cool not to use the automatic scale assignments for these charts but to use the user-defined parameters to give a better special meaning to the deflected beam profile. With a few tries I was able to get this working.

One of the things I was interested in was having the user form take up the entire window. This would keep the user from even seeing the worksheet side and without seeing it they would have no desire to change it. However screen resolution could be different every time this program is opened so I needed to base it off of screen size. I looked up some methods online and settled on a fairly simple that I implemented at the opening of the workbook (or rather at the initialization of the user form).

Another lesson I learned was to create a graphical map of the user form before I went to coding. The first attempt at the user form went well. After building 95% of it and going back to add some final labels I found that it would not let me add more labels as apparently I had gone wrong in a label and duplicated a label name. It was time for a fresh start. The second time went a lot more smoothly and out of fear I did not name any of the labels as I did not want to run the same risk of accidental duplication. Further it enabled me to better map out the labels and put them in series where I could reference them in simple loops when I needed to change a series of labels.

ASSISTANCE

I went into the TA to get help with the label error, but the TA said it would probably be easier to start over. This enabled me to clean up a lot of the things that were not quite right about the first iteration anyways. This was the extent of the help I received from other people. I did look at a lot of online code and postings to things I was incorporating in my project, but never posted for help on forms.