ANALYSIS PROGRAM

Version: 0.1

MANUAL

by Fatih Yaman

E-mail:
fatih.yaman@metu.edu.tr

Manual Last Edited:
12.04.2022

Contents

Thanks to everyone whom programs or softwares I used...

And thanks to anyone who contributes or even pays attention and becomes a companion in this fantastic journey...

1 Preface

I am actually very exited, even when writing this preface. I believe that it will be a long one but it is hard to foresee since I am writing without planning it :D This project is a project that I am very eager to see its outcome. This is a dream of mine that I nurture. Indeed, it is a far dream, yet I believe one day I can accomplish it.

The main motive of this project is that there are no free licensed professional analysis programs to my knowledge. There are fantastic programs such as Ansys or Nastran-Patran; however, they are all licensed and require colossal amount of money to use. This amount is such big that it can even sometimes put companies into hardships as well.

My compassion has started and is growing thanks to the 3D computer graphics software Blender. It was a small program back when I first discovered it, yet it was a great program. It worked smoothly, it was updated frequently, it had a terrific community which was full of people with endeavour. Today it is a very capable program which provides a free-to-use powerful tool to students and smaller companies. Even a very good Netflix animation called "Next Gen" was done using Blender. This is my hope for this program as well. One day it may have a very good community who supports the program, and it becomes a very powerful tool. Even better than I dare to imagine.

Today of course, I will start with baby steps. I will not write any part of the program if I can manage not to. I will use already made open resources to build the program. It will most probably won't even be close to being a very efficient program. But this is the starting step that every dream needs.

Thank you dear reader if you are here and valuing my work enough to read this.

2 Introduction

In this section I want to talk about how I want this program to be.

Firstly, and most importantly, it must be fully open source. Any program or software that is a part of this project must an open source one. For this purpose (and also because of curiosity) I have chosen Julia to be the coding language. It is an open source, promising language. Qualities that I also wish for my project.

Secondly, I will use already proposed solutions if there are any, at least at the starting phase. Because it would be a waste of time to build everything from scratch. For example, I will use already made Julia packages for meshing, or if I add wing structure analysis at some point, I will use a program such as xfoil or xflr5 to make the aerodynamic analysis since I think that replacing an already smoothly working program is just a waste of time. However, my main concern will be that program being a free one or not.

3 Required Packages

You need to download the following required packages in the folder you are keeping Space Vortex files. THe packages are as follows,

1. GLMakie
2. PrettyTables

4 User's Manual

Here I will tell how the program is used for the ones who are just interested in using the program instead of its theoretical or coding background.

4.1 Main Menu

In the menu there is a list to choose from. If you click the list you can select the analysis type. For now, there is only 2d Truss analysis available, hence the choice makes no difference.

After making a selection, you can see the selection you have made and also there is a button that you should click to proceed to the meshing screen.

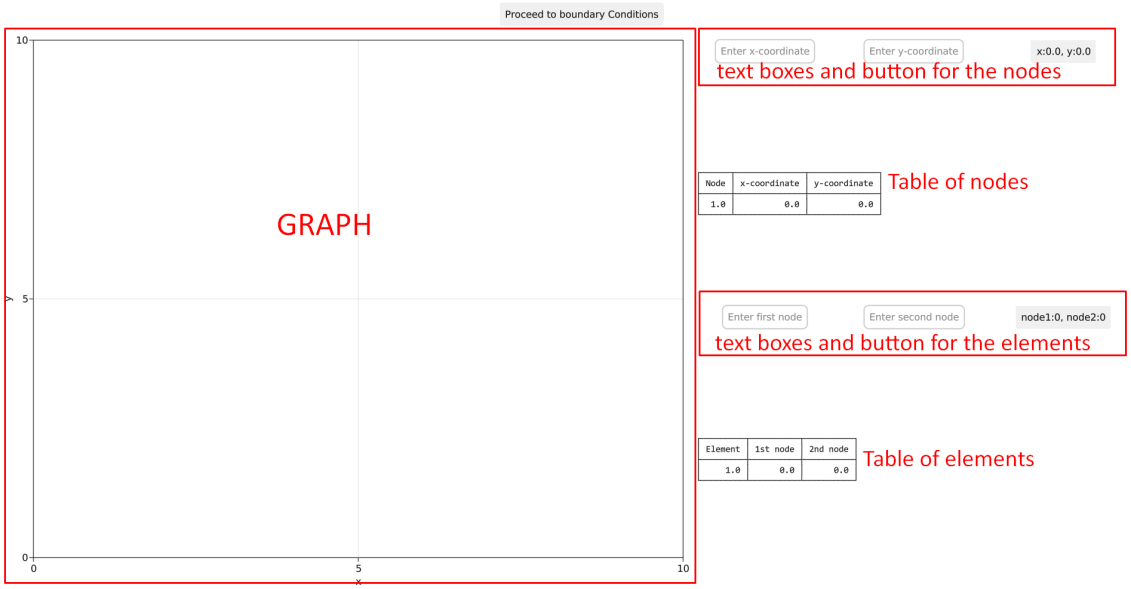
4.2 2D Truss Analysis

4.2.1 Adding Nodes and Elements

In this screen, you see text boxes to the right of the screen and a button besides them. Upper one is for adding the nodes with the inputted coordinates, the lower one is to add the elements with the inputted coordinates.

To make an input to a text box, all one needs to do is to write the appropriate number and click enter. If the selection is accepted, the number on the button will update and change to the inputted number.

After adding each element the graph on the left will be updated to show. Also adding each node or element, node or element table will be updated.



Currently the upper menu does nothing.

5 Display

Visualization is not my main concern at the moment, therefore although the code might be clumsy or inefficient in terms of visualization, I will use it as long as it works. At least until the analysis part starts working properly and I have time to make the program better instead of trying to make it work.

At the moment I will use the package GLMakie for making the visualization.

Since I have not found any better solutions at the moment, every menu will just be a distinct figure that I will show using display() function.

Here, I will not explain how to use makie layouts. For such unexplained topics, one can refer to: https://makie.juliaplots.org/stable/.

5.1 Main Menu

For now main menu will just be a list from which we choose the appropriate analysis type. For the main menu I have set up a figure with just a list. I have defined the first seen version (just a list and a text over it) of the main menu globally in the SVDisplay module. But then, I want the main menu to change accordingly to display the selected analysis type, and a button to continue with the analysis. For this purpose, I couldn't find any better solutions but to add the second version of the main menu inside a function. This function is called whenever a selection is made from the list. For the texts that show a particular information, such as the text that is showing the selected analysis type, I have used Observables. Observables are a makie type that allow us to do dynamically changing visuals.

To add the text over the list, the only solution that I could come up with was to add an axis, hide its decorations and spines using,

```
hidespines!(axText)
hidedecorations!(axText)
```

And adding the text using the command:

```
text!(axText, "Please select the analysis type...", position = (0,0))
```

In the current version (0.1) there are no other analysis types hence the selection of the analysis type does not make a difference.

5.2 2D Truss Analysis

5.2.1 Adding Nodes and Elements

Here I first had to design a GUI such that it allowed us to add nodes and elements. For this purpose I have divided the screen into 3 grids as follows,

In the upper grid, there will be menu buttons which is yet to be added. In the lower left grid there is the graph that shows our current mesh structure. Or in this case, truss structure. The lower right grid is for adding new nodes and elements.

In the lower right grid I have added two text boxes and a button for node inputs and element inputs. This was the most efficient way that I have found. To update the numbers on the buttons, I store the variables as observables as well as the tables. The package PrettyTables forms the tables for me for the matrices (matrices that are holding the information of nodes and elements) that I provide. And by using the text! command I put them on the appropriate axes. The important thing here is that I have again used observables to form and write the tables, so that they are updated every time the matrices for nodes and elements are updated.

Evrytime the elements matrix is updated, besides updating the tables, function also draws the graph for this elements using the code:

```
empty!(axisOfMesh)
for i = 1:size(elem[], 1)
    lines!(axisOfMesh,[coords[][elem[][i],1]; coords[][elem[][i,2],1]],
    ↪  [coords[][elem[][i],2]; coords[][elem[][i,2],2]], color = :black)
```

```
4     scatter!(axisOfMesh,[coords[][elem[][i],1]; coords[][elem[][i,2],1]],
      ↪  [coords[][elem[][i],2]; coords[][elem[][i,2],2]], color = :black)
5 end
```

In this code instead of drawing all the elements with a single array, I preferred to draw them a line for each element. As one can see, the graphs are drawn on "axisOfMesh". Before drawing, to clean the previous draws by using "empty!". Also to make the nodes seeable, I also added a scatter graph as well which puts dots on each node. It actually puts more than one dots for each node, but it is not necessary since they are all drawn on the same graph. It is just like overwriting.

6 2D Truss Analysis

In this section we will analyze 2D truss systems. This will be the first analysis type Space Vortex provides. It is a good starting point since it is not extremely hard and will provide me the chance of trying my skills with the interface and etc.

7 Update Notes

7.1 Version 0.1

- Added main menu
- Added node and element addition screen for 2D Truss Analysis