Fall 2020 ME751 Final Project Report

University of Wisconsin-Madison

SimEngine3D Development

Logan Rapp

December 12, 2020

**Abstract**

This report describes the progress on the development of a python code - SimEngine 3D – that is used to simulate multibody dynamic problems.

<https://github.com/lrapp/simEngine3D>

Contents

[1. General information 4](#_Toc58159897)

[2. Problem statement 4](#_Toc58159898)

[3. Solution description 4](#_Toc58159899)

[3.1 System object & body/constraint information 4](#_Toc58159900)

[4. Overview of results. Demonstration of your project 7](#_Toc58159901)

[5. Deliverables: 8](#_Toc58159902)

[6. Conclusions and Future Work 8](#_Toc58159903)

[References 8](#_Toc58159904)

# General information

* Home Department: Mechanical Engineering – Solar Energy Lab
* Current Status: PhD student
* Individuals working on the Final Project: Logan Rapp
* I am not interested in releasing my code as open source code.

# Problem statement

I will continue development of my simEngine3D code to include support for all joints and driving constraints discussed in class and an improved method of providing model definition. My research is not closely aligned with topics covered in this course, so I was not able tie this final project with my research.

Explain in clear terms what you wanted to accomplish. If you go w/ a default project, simply indicate so. Please use here the material that you provided in your Final Project proposal.

As part of this section, touch on the motivation/rationale for your project selection. Explain why you chose to work on this project. For instance, if it’s work related, explain in rough terms what the big process is, and what part you’re trying to take care of.

# Solution description

### 3.1 System object & body/constraint information

Indicate how you went about implementing your solution. Explain data structures, algorithms used, code structure, function you implemented, etc. Provide a panoramic snapshot of your Final Project effort.

I used a custom python class definition, I call it “SYS”, to hold all the components of the problem and the results. This includes a list of the bodies in the simulation, a list of the constraints and other important items. A screen shot of the “sys” definition is shown in Figure 1.

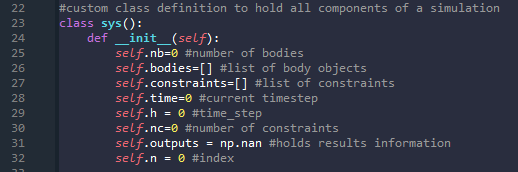


Figure 1 - Definition of custom class to hold information related to the simulation

I’ve created a .txt file input system in order to facilitate inputting the body and constraint information. The file parser function I use is call “data\_file” and is located in the Python script “simEngine3D\_dataload.py”. I first define a custom body class, as shown in Figure 2.

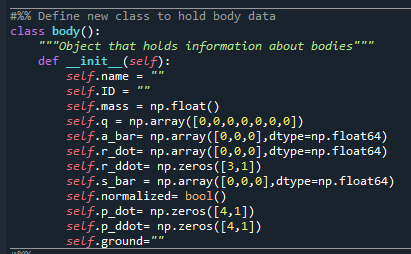


Figure 2 - Definition of custom class to hold body information

The “data\_file” function requires a filepath to be passed to it, and the .txt file must follow the format shown in Figure 3.

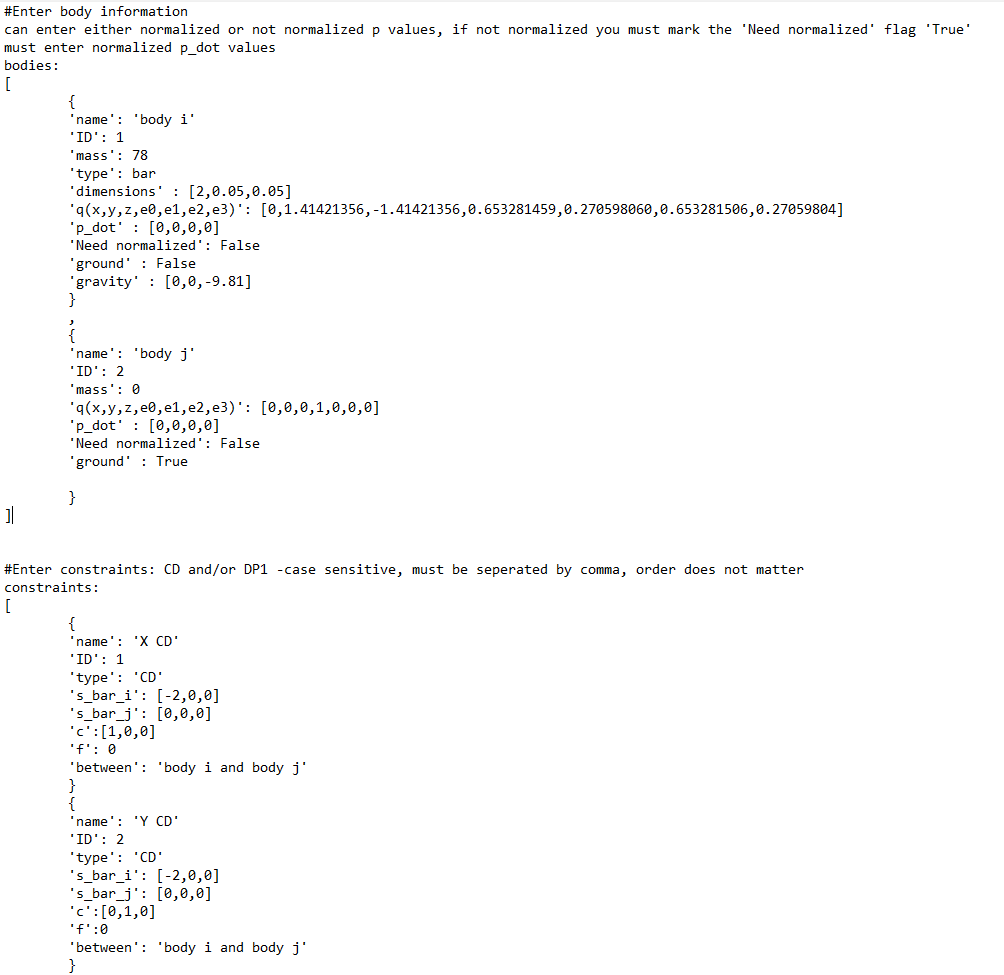


Figure 3 - Example of input file

The function reads in all the information and then finds the “bodies” and the “constraints” headings. For the bodies, it then counts the number of “{“ to determine how many bodies will be read in. I then create a list of the body object as described in Figure 2. Looping through the bodies data, I use the “{“ and “}” to separate the information for the different bodies. For each body, I create a list of dictionaries that hold all of the body information listed in the input file such as “name”, “ID”, “mass”, etc. Then looping through the list of dictionaries I assign the correct information to the variable in the body object. The function returns a list of the body objects with length equal to the number of bodies in the system.

The constraints are read into Python in a similar manner as described for the bodies above. The function that reads in the constraint information is titled “constraints\_in” and is also part of the “simEngine3D\_dataload.py” Python script. The “constraints\_in” file is passed a path to the input file and returns a list of constraint objects. The constraint object contains all information regarding the constraint, and the class definition is shown in Figure 4.

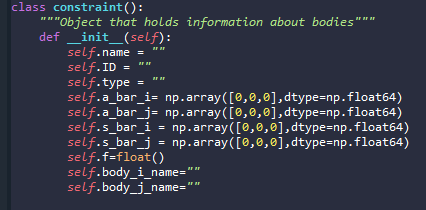


Figure 4 - Definition of the constraint custom class

### 3.2 Using the simEngine3D

To begin a simulation, I first create the “SYS” object. I then pass that object, along the start time, end time, time step, and path to the input file to one of the two analysis tools implemented thus far. Those two options are inverse dynamics analysis and dynamic analysis. For example, Figure 5 shows the first few lines of the inverse dynamics function.

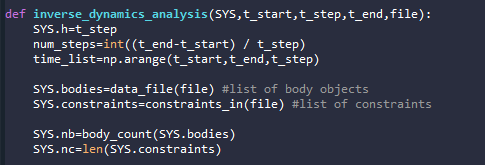


Figure 5 - First few lines of inverse dynamics function

You can see that first I assign SYS.h to equal the inputted t\_step, then I calculate the number of steps and create an array of time for this solution. Next I use the functions “data\_file” and “constraints” to read in the bodies and constraints as described in section 3.1. I then proceed with the inverse dynamics analysis steps outlined during lecture. This includes solving for the accelerations using only the set of constraints, computing the LaGrange multipliers using the Newton-Euler form of the equations of motion, and finally computing the reaction forces/torques of the prescribed motion. All of this can be found in the “inverse\_dynamics\_analysis” function included in the “simEngine3D.py” Python script.

I made use of a Pandas DataFrame structure to hold portions to facilitate easier plotting and interpreting of results.

# Overview of results. Demonstration of your project

Explain here what you obtained, explained why the results are good/bad. This is the place where you talk about the outcomes of your Final Project effort. It is not the end of the world if your code doesn’t work as anticipated. Explain here how far you have made it.

Most often, you have a comparison against sequential code, perhaps via a scaling analysis. Make sure you include plots and/or tables to show your results.

# Deliverables:

Discuss what is delivered for this Final Project. Important points:

* This report should be in Canvas.
  + On multi-student teams, each team member should submit a final report even if the reports end up identical. However, the code should be in one repo
* Tell us what is in your git repo and explain how we can run your code
  + If we cannot run your code, explain why that is the case

# Conclusions and Future Work

# References

[1] Make sure to give credit where it’s due.