PyElastica: Run-time visualization of simulations

Hamzah Hashim - London, UK (UTC + 0)

Vast systems and their dynamics can be highly complex in their nature, and developing strong and robust simulation tools is vital to be able to better understand how these complex systems behave. A huge range of both natural and artificial systems, from contact lenses to spider silk, can be modelled using slender structures, which are thin structures such as wires and shells that we can compose large systems from.

Cosserat rod theory is a method of modelling 1D slender rods, accounting for all the modes of deformation these rods can undergo. Building up systems such as human muscles from these cosserat rods, we can simulate the complex deformations that these muscles undergo during contraction and relaxation, allowing for a greater insight into how some of these structures function. PyElastica is an open-source library that can be used to assemble cosserat rods into structures and simulate their behaviour.

A useful aspect of numerical simulations, particularly when simulating these dynamic physical systems, is being able to visualize them. Having the ability to visualize simulations allows you to gain a more intuitive understanding of the dynamics of the system than might be obvious from numerical data, and is useful for things such as proof-of-concept that you simulation is working as intended and validating initial predictions. Strong visualisation features are therefore invaluable for numerical simulation libraries, and I hope to be able to help improve PyElastica this Summer in creating visualisation tools for simulations.

Project Goals – Overview

I hope to be able to improve the visualisation tools of PyElastica, and implement a fast and lightweight run-time visualisation this Summer as part of the Google Summer of Code program. The simulation should be built upon the current workings of the PyElastica library, and be able to be used for both real-time visualisation and as a post-process call back as well. The visualisation tools should be able to allow interaction with the scene, such as rotation and panning and be feature rich. All code and features will be fully documented and tested.

Specifically, based on visualisation toolkits for other simulation libraries that work well and discussions with project mentors, the main points that should be focused on are:

- **Accuracy –** There is not much use for a visualisation that is only accurate at small scales or is only accurate part of the time. With simulations producing large amounts of numerical data from which it can be very difficult to distinguish the behaviour of the system without visualising, it is crucial that the visualisation tools developed represent the behaviour of the system to a high degree of accuracy.

- **Reliability** – PyElastica simulations can be large and complex in scale and size, consisting of many different and varying objects. The visualisation tools developed during this project should be able to handle the variety of systems that can be simulated with PyElastica with minimal user input, troubleshooting or errors.

- **Interactivity –** While obtaining simple videos and renders of simulations from PyElastica can be useful, being able to interact with the visualisations can provide better insight and information into the behaviour of these simulated systems. Features, even basic ones such as panning, zooming and moving around the system can provide far greater insight into the workings and behaviours of these different systems.

- **Usefulness –** Ultimately, I hope that the visualisation tools developed during this project can provide more than just cool-looking videos. Working my mentors and the PyElastica community, I hope to implement features into the visualisation tools that can provide real insight into the behaviour of these systems, and help the users of PyElastica to better understand, troubleshoot and improve their systems and simulations.

Estimated Project Timeline

~25 hours per week (but can be increased/decreased depending on requirements of project. I am also happy to extend to a Large project if that would work with the mentors :))

Logs and tests are kept and documented throughout program.

May 20th - June 12th : Familiarise myself with the PyElastica library, the community, documentation, contributing guidlines and gain a brief understanding of the mathematics of Cossat rods. Research into different visualisation toolkit libraries in python eg. VTK, Fury etc.

June 13th – June 26th: Start project. Gain familiarity with visualisation toolkit that will be used (depending on discussion with mentors perhaps try different libraries) and their APIs. Implement visualisation of a basic numerical simulation using numerical data from callback of simulation as a proof of concept.

June 27th – July 10th: Begin to integrate visualisation during run-time for very basic simulations with the aim to get a basic working version. Trial/consider different ways of adding the visualisation function (build upon existing function, separate function/class method)

July 11th – July 24th: Improve implementation and test and document existing code more thoroughly. Test with different types/sizes of simulations and identify and profile bugs, bottlenecks etc. Focus more heavily on profiling the visualisation tools in this stage, and identify optimisations that can be made to ensure visualisations tools are as efficient as possible.

July 25th – August 7th: Improve functionality of visualisation, adding different features/arguments based on project aims/discussions with mentors. Possible features that could be really beneficial includes features such as color mapping to represent information such as stress and strain through the various rods in the simulation,

August 8th – August 21st: Begin getting code ready for integration into production. Ensure thorough documentation and testing code coverage, and that features of the visualisation are functioning properly.

August 22nd – August 29th: Finalise project and code and commit and pull into production.

(2 weeks have been left unassigned for any extra time needed on specific parts of the project)

A bit about me

My name is Hamzah Hashim, and I am a penultimate year student studying for a masters in Theoretical Physics at Imperial College London expecting to graduate in 2023 (GPA 4.0). I am passionate about software engineering and open-source software, and would love to be able to contribute to PyElastica this summer. I have significant experience with Python, and experience with other languages as well such as C/C++ etc. I have quite a bit of past experience using numerical simulation libraries (particularly GEANT4 in C++ for particle physics simulations) as well as developing my own (raytracing simulation, PDE heat flow solvers), and have a good understanding of developing in Python.