Classical mechanics simulations

Class	
Created	@November 18, 2021 10:38 PM
Materials	
Reviewed	☐
Type	

Be sure to join the LMS for this course available at https://learn.naxxatra.com/course/. It will be constantly updated with useful links, meeting notes, and example code snippets.\

Meeting Schedule

We'll be meeting once a week, on Saturday evening 7:30 PM.

Aside from that, we can meet anytime of the week, per your convivence, to debug, and clear any potential doubts with help from each other.

Syllabus Outline

Week 0

Basic Equations of motion. Displacement, Velocity, acceleration.

Try making motion in 1D, displacement, velocity, and finally add acceleration

Vectors: What are vectors, why do we need vectors, definition of vectors, and representing Vectors in our code.

Try making motion in 2D, displacement, velocity, and finally add acceleration.

Setting walls for the particles.

Euler's method of solving Differential Equations. [If people are interested and time permits]

Week 1

Play around with various acceleration vectors on the one body we made last week.

Vector Addition. When to add, and how to add vectors in the code.

Object Oriented Programming: Create classes of bodies with different mass.

See how different bodies will move about given the same set of forces.

Week 2

Collision mechanisms. Brownian Motion

Newton's gravitational law. Inverse Square forces.

Interaction of a static body and a free moving body.

Interaction of two free moving bodies

Interaction of multiple free moving bodies with each other. (Complex Dynamics)

Interaction of Multiple free moving bodies with one another, and a static gravitational attractor.

Project Work: Non-Elastic Collisions and try the above topics again.

Croll-Milankowitch Cycles [If people are interested and Time Permits. Otherwise we can toss this to homework]

Week 3

Spring Forces. Simple Harmonic Motion.

Constrained Motion. String Constraint.

1D Spring

Polar representation of Vectors. Manipulating Vectors by changing the angle, rather than the x and y components.

Spring Force + Another force.. Basic Pendulum.

Project Work: Try making a Double Pendulum, or a pendulum suspended by a string.

End of Classical Mechanics Simulations.

Other Details

Idk. Just addressing some FAQs that *might* pop up.

Applications?

Where is this stuff applied? I'm sure that's going to be a question if you're going to be spending the next four weeks learning this.

Well, for one, computation has become an important pillar of scientific research, be it materials science or Astronomy. And many of the concepts covered here are directly used in modelling the world in computer simulations.

For another, Game Development involves a lot of physics. If you have played any game, be it Far Cry 6 or even something as simple as Pong, you wouldn't have noticed, but Physics makes up an important part of the mechanics, with the way things interact with one another. While we're just working with two dimensions, it's extremely easy to extend these concepts into the third dimension, and use these concepts for a 3D game. And when building 3D models for things to use in the game, defining the object's behavior would generally require a good understanding of the physics it's going to be subjected to.

Programming Background

Ideally, you don't have to know too much programming. Basic Arithmetic operations, loops, and working with basic datatypes, like Lists or Dictionaries.

If you've done Python in 11th grade CBSE, that's more than enough to keep up with we're doing here.

For Programming in the club, we're going to be using one of two languages, and you're free to choose whichever of the two you're more comfortable with:

1. Python and Pygame: Pygame is one of the simplest ways to render 2D images and interact with them in Python. It's a library intended for Game Development, but you can do a lot with it, and in our case, with some basic logic, we can draw objects which obey the laws of physics we define.
2. p5.Js and Javascript: "p5.js is a JavaScript library for creative coding, with a focus on making coding accessible and inclusive for artists, designers, educators, beginners, and anyone else!", as their website says. The Tutorial series we're following, Nature of Code, is going to be using Javascript for the coding sections.

Recommended Resources

Regardless of which language you choose to use, I highly recommend keeping HC Verma's Concepts of Physics (Book 1) next to you as you code. I don't believe there's a better source that'll let you quickly reach an answer to any physics doubt, when it comes to 11th CBSE level 😂

Aside from the resources described here, be sure to be on the lookout for the meeting notes that I'll be taking, and sharing. If we come across something interesting in our discussion in the meeting, I'll include it there, and those may not be available elsewhere in the internet.

Javascript/p5.Js

https://p5js.org/ The official website for p5.Js. They have a web editor that you can use, if you're too lazy to setup Javascript on your device.

https://natureofcode.com/ Nature of Code, the book. You can either buy the book, or read it online through interactive web-pages.

https://www.youtube.com/channel/UCvjgXvBlbQiydffZU7m1_aw The Coding Train Youtube Channel. It's run by the same person who wrote the nature of Code, and has a lot of advanced projects which he didn't cover in Nature of Code as well.

https://www.kadenze.com/courses/the-nature-of-code/info The book, adapted as a course on Kadenze. The same is available for Free on The Coding Train website, with a different version of the videos.

Python

While the above references can be used if you're using python as well, here are a couple more if you want Python Specific.

http://www.pygame.org/docs/tut/PygameIntro.html The pygame docs have some pretty good explanation on how to get started with the library, and it's always a good idea to read the docs when you're looking for some advanced features of the library.

https://www.petercollingridge.co.uk/tutorials/pygame-physics-simulation/ This is a pretty good source as well. It has some pretty good explanations and generally covers what we'll be going through. But be careful copying the code. it uses Python 2.x, while you'll probably be using python 3.x. Python isn't Backwards compatible.

What next?

Physical Simulation techniques don't end here. There's a whole another branch which uses a different method called Cellular Automaton. That is where we'll be

categorizing space into little cells, and simulate the bulk properties of each of those cells. This is more useful in things like Fluid simulations, or Atmosphere simulations. If All things go well, we'll be starting another club on that side of simulations in mid/late December, after the end of this.

While going through this set of topics, you should be able to see how easy it is going to be to simulate and visualize complex systems. If you're a part of the Complex Systems tribe, you could try implementing and visualizing those topics, using the techniques described here. (And if time permits, we'll try to spend some dedicated time to work on complex systems)