Final Project Report

Pridhvi Myneni

November 2020

Contents

1 Introduction 1

2 Project Proposal 1

2.1 Group Members . . . 1

2.2 Language . . . 2

2.3 Description . . . 2

2.4 Rough Work Breakdown . . . 2

2.4.1 Nicholas Dodd . . . 2

2.4.2 Pridhvi Myneni . . . 2

2.4.3 R. Joey Rupert . . . 2

2.5 Approach . . . 2

3 Week 1 Report 3

3.1 Repository Information . . . 3

3.1.1 Repository Organization . . . 3

3.1.2 Repository Access . . . 3

3.2 Progress . . . 3

3.2.1 Nicholas Dodd . . . 3

3.2.2 Pridhvi Myneni . . . 3

3.2.3 R. Joey Rupert . . . 4

1 Introduction

This group consists of Nicholas Dodd, Pridhvi Myneni, and Joey Rupert. We are working on the RK45 and Predictor-Corrector methods of numerical estimation. Our GitHub repository contains both the documentation and code for our project.

2 Project Proposal

2.1 Group Members

Nicholas Dodd Pridhvi Myneni Joey Rupert

2.2 Language

C++

2.3 Description

Partial differential equation solvers for a gravity simulator. In order to compute where bodies are in a system, we compute the accelerations on those bodies by summing all forces due to all bodies. For an n-body system, this is $O(n^2)$. For a system like the solar system with one star and 9 planets, this is doable, but if you try to compute the locations of all the asteroids of 1km and above, computation increases dramatically. If the goal is to compute the motion of a galaxy with n=1011 stars, it is computationally intractable. Tree-based techniques have been used. Obviously a faraway body like Pluto does not have any significant effect on Earth. Computing the average mass and position of faraway objects can be used to reduce complexity. For example, if calculating the motion of our galaxy, you can use the center of mass of the nearby galaxies, you do not have to model 100 billion stars.

Also, stepping forward in time uses differential equation solvers. The simplest one (Euler) is just $v_{\text{new}} = v_{\text{old}} + a * dt$, $x_{\text{new}} = x_{\text{old}} + v_{\text{old}} * dt + 0.5 * a * dt^2$.

The problem is that a is not constant, so a small dt must be chosen. Higher order methods determine estimates for a so they can take bigger timesteps. Best algorithms include RK45 and Predictor-Correctors. Either half of this project is suitable for a group, and two groups can work together to achieve an entire program. I have a 3d front end that would make the result visually beautiful if you choose to use it.

2.4 Rough Work Breakdown

2.4.1 Nicholas Dodd

Predictor Corrector research and implementation

2.4.2 Pridhvi Myneni

RK 45 research and implementation

2.4.3 R. Joey Rupert

Predictor Corrector research and implementation

2.5 Approach

Pick a known 1st order diffeq with a known analytical solution. Solve using Euler, RKF45, Adams-Bashforth and verify your code works

Compare timesteps and see which is most efficient. Probably Adams-Bashforth is the most efficient but who knows because this one $f'(x)$ is cheap to compute

Repeat for a 2nd order system (ie gravity) Do for a 2-body system like earth-moon with a circular orbit Don't even move the earth. Earth = $(0, 0)$ You can check stability with Energy should be the same

Once you have this, you can try with solar system. First try, just do planets out to Jupiter, everyone gravity affects everyone is easiest. Or you can reverse sort, and say everyone with mass $> k$ affects everyone to reduce computation.

(Obviously, the earth affects the moon and vice versa because the moon is really close, so this is not a full solution)

If you want to use the assumption that every orbit is circular, average perihelion and aphelion. Pick a random starting angle around the circle. You will have to calculate velocity for circular orbit or elliptical orbits

Then try every planet, every major moon. Should be fun.

Reference:

Circular Orbits

Elliptical Orbits(vis-viva equation)

RK45

Predictor Corrector

More to be added as research continues

https://web.mit.edu/10.001/Web/Course_Notes/Differential_Equations$_{N}otes/node7.html$

3 Week 1 Report

3.1 Repository Information

3.1.1 Repository Organization

Our GitHub repository was setup this past week with our organizational structure, a root directory `docs` for our documentation, and `src` (not yet created) for our codebase.

3.1.2 Repository Access

Since our final project is not being attempted by another group, and our work on this project could aid us in our job/internship searches, we are making our repository public. Therefore, its contents, as well as its history, should be viewable by Professor Kruger and our TA, Fan Yang.

3.2 Progress

3.2.1 Nicholas Dodd

As per the GitHub repository commit history, Nick wrote an introduction to Predictor-Corrector methods, as well as an approach for time-stepping in a proposed preliminary implementation of the Euclidean method for numerical estimation.

3.2.2 Pridhvi Myneni

I reorganized the repository structure to match the current system.

Additionally, I started learning how the RK45 system works and wrote part of an introduction to the Runge Kutta class of solvers. I wrote and illustrated an example of the first order solution, the Euclidean estimation, and got stuck attempting to compare the efficiency of computing the exact solution to my proposed hypothetical situation, against the Euclidean estimation.

Specifically, how do we calculate the derivative of a function for a computer? Do we use the Taylor Series approximation? Can a computer do that? I understand that `CORDIC` might be the solution; however, I have yet to read up on it, and if this means that polynomials are not more efficient to estimate(?). My introduction can be viewed here.

Although Joey had already written a separate introduction to RK45, I did not understand the equations and had to review the material for myself, as this was the part of the project I was responsible for coding in terms of the breakdown of work in the project proposal.

3.2.3 R. Joey Rupert

As per the GitHub repository commit history, Joey worked on a separate introduction to RK45, which covered what the algorithm was used for and some equations.

[Authors note: I didn't understand the equations, so I am likely understating Joey's contributions.]