CS 530: High-Performance Computing

Seminar 1: A Survey of Computational Physics

Nathan Chapman

Department of Computer Science
Central Washington University

April 22, 2024

Contents

List of Figures

Chapter 1

Introduction

Each of the five pillars of physics (Classical Mechanics, Electromagnetism, Relativity, Quantum Mechanics, and Thermodynamics) has well-posed problems for which the mathematical models are either unreasonable or impossible to solve by hand. As such, computational methods are needed to get an approximate solution.

There are many numerical methods to support the evolution of physical models for each domain. Whether it be using a fourth-order Runge-Kutta method to solve the first-order ordinary differential equation that models a ballistic object under the influence of gravity and air-resistance, or using an eigth-order symplectic Yoshida integrator to model a system of bodies orbiting each other while accounting for the produced gravitational waves, there are methods for every physical context. Though, because these methods are using approximations to their analytic counterparts in calculus, there is error that needs to be considered.

Numerical analysis is the practice of tracking and rigorously accounting for the discrepancies that arise when approximating analytic mathematical methods by their finite and counterparts. These errors can arise not only from approximating descriptions from calculus, but also from operations in linear algebra as they're applied to finding the eigenvalues of quantum operators.

Chapter 2

Classical Mechanics

The "everyday" world as we know it is described by a model of physics that has been studied for millenia. As such, we call this model of the behavior of nature "Classical Mechanics". Now I hear you ask "Why?". Well "mechanics" is the study of motion, and we include the "classical" preface to distinguish it from "modern" physics which encompasses all the physics discovered after around 1905 or 1925 (depending on who you ask); those physics will be covered later in sections 4 and 5. For now we take a look at the mathematical model that describes the motion of not-too-small, not-too-large objects moving at slow velocities.

The motion of "classical" objects, ranging in scale from biological cells, sand, ants, birds, cars, airplanes, planets, up to stars, can described as the solution to what's called the *equation of motion*. The equation of motion for an object with mass m under the influence of forces F_i is determined by Newton's 2nd law of motion

$$\sum_i \vec{F}_i = m\ddot{\vec{x}}, \tag{2.1}$$

where $\ddot{\vec{x}}$ represents the second time derivative of the position $\vec{x}$ known as *acceleration*. These forces completely determine the how objects move through space in time.

Mathematically, the resulting equation is a vector second-order ordinary differential equation, compactly representing the underlying system of differential equations. In many cases, this differential equation is homogenous to represent no time-dependent external forces supplying the physical system with energy. Likewise, sometimes these forces can be represented by nonlinear terms; making the equation of motion much more complex.

The following is a survey of topics in classical mechanics whose forces yield an equation of motion either too complex or too unreasonable to solve by hand.

2.1 N-Body Orbits

While so-called N-Body problems are prolific in physics and other sciences, a quintessential example is the gravitational interaction of astrophysical bodies i.e. massive bodies in space.

2.1.1 Mathematical Model: Systems of Ordinary Differential Equations

In the case of multi-body gravitaitonal interactions, the governing equation of motion is determined via Newton's law of gravitation between two objects with masses m_i and m_j

$$F_{ij} = \frac{Gm_im_j}{r_{ij}^2}\hat{r} \tag{2.2}$$

where $G = 6.67 \times 10^{-11} \frac{Nm^2}{kg^2}$ is the universal gravitaional constant, r_{ij} is the distance between the two objects, and $\hat{r}$ is the unit vector pointing from object i toward j. Not only are is there an equation of motion for each object, but each equation is coupled to every other equation. This results in the three-dimensional vector differential equation[2]

$$\sum_{j;i\neq j} \frac{Gm_im_j}{(x_i-x_j)^2+(y_i-y_j)^2+(z_i-z_j)^2} \frac{(x_j-x_i)\hat{x}+(y_j-y_i)\hat{y}+(z_j-z_i)\hat{z}}{\sqrt{(x_i-x_j)^2+(y_i-y_j)^2+(z_i-z_j)^2}} = m_i(\ddot{x}_i\hat{x}+\ddot{y}_i\hat{y}+\ddot{z}_i\hat{z}) \tag{2.3}$$

for object i, and a similar equation for every other object. For example, a system of 3 objects interacting only via gravitational forces has the equations of motion

$$\frac{Gm_2}{((x_1-x_2)^2+(y_1-y_2)^2+(z_1-z_2)^2)^{3/2}}\begin{bmatrix}x_2-x_1\\y_2-y_1\\z_2-z_1\end{bmatrix}+\frac{Gm_3}{((x_1-x_3)^2+(y_1-y_3)^2+(z_1-z_3)^2)^{3/2}}\begin{bmatrix}x_3-x_1\\y_3-y_1\\z_3-z_1\end{bmatrix}=\begin{bmatrix}\ddot{x}_1\\\ddot{y}_1\\\ddot{z}_1\end{bmatrix}$$
$$\tag{2.4}$$
$$\frac{Gm_1}{((x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2)^{3/2}}\begin{bmatrix}x_1-x_2\\y_1-y_2\\z_1-z_2\end{bmatrix}+\frac{Gm_3}{((x_2-x_3)^2+(y_2-y_3)^2+(z_2-z_3)^2)^{3/2}}\begin{bmatrix}x_3-x_2\\y_3-y_2\\z_3-z_2\end{bmatrix}=\begin{bmatrix}\ddot{x}_2\\\ddot{y}_2\\\ddot{z}_2\end{bmatrix}$$
$$\tag{2.5}$$
$$\frac{Gm_1}{((x_3-x_1)^2+(y_3-y_1)^2+(z_3-z_1)^2)^{3/2}}\begin{bmatrix}x_1-x_3\\y_1-y_3\\z_1-z_3\end{bmatrix}+\frac{Gm_2}{((x_3-x_2)^2+(y_3-y_2)^2+(z_3-z_2)^2)^{3/2}}\begin{bmatrix}x_2-x_3\\y_2-y_3\\z_2-z_3\end{bmatrix}=\begin{bmatrix}\ddot{x}_3\\\ddot{y}_3\\\ddot{z}_3\end{bmatrix}$$
$$\tag{2.6}$$

Because this problem is in three dimensions, there are actually 9 differential equations to solve in order to fully model the evolution of the position and velocity over time of each object.

2.1.2 Computational Model

Finite Differences

Because computers can only store numbers of finite precision, approximations must be used in order to implement these differntial equations in a form that can be understood by the computer. One such translation is by approximating the analytic derivatives and differential operators with *finite differences*. In its most basic form, the derivative can be approximated as a ratio of differences in the form

$$\frac{df}{dt} \approx \frac{f(t+\Delta t) - f(t)}{\Delta t}. \tag{2.7}$$

This equation is known as the first-order-accurate, forward-finite-difference representation of the first-order derivative [**NEEDS CITATION**]. There are similar representations for higher-order derivatives, more accurate approximations, and more accurate approximations of higher-order derivatives. Similarly, there are also *central* and *backward* finite-differences.

Each of these finite-difference schemes can be represented via coefficients. For example, the derivative of order m with accuracy n has $2p + 1 = 2\lfloor \frac{m+1}{2} \rfloor - 1 + n$ central coefficients $a_{-p}, a_{-p+1}, \ldots, a_{p-1}, a_p$ defined such that

$$\begin{bmatrix} 1 & 1 & \ldots & 1 & 1 \\ -p & -p+1 & \ldots & p-1 & p \\ (-p)^2 & (-p+1)^2 & \ldots & (p-1)^2 & p^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ (-p)^{2p} & (-p+1)^{2p} & \ldots & (p-1)^{2p} & p^{2p} \end{bmatrix} \begin{bmatrix} a_{-p} \\ a_{-p+1} \\ \vdots \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ m! \\ \vdots \\ 0 \end{bmatrix} \tag{2.8}$$

where the only non-zero value on the right side of the equation is at the $m + 1$-th index[**NEEDS CITATION**].

Once the derivative operators are discretized using finite-differences, there are different algorithms in which the differences are used to solve the differential equations.

Symplectic Integrators

While there are many methods of numerically solving differential equations (known as *integrators*), one of the most important classes when considering a context such as orbital mechanics and oscillatory situations (i.e. Hamiltonian systems) is that of *symplectic* integrators. This type of integrator is so important to computational physics because it conserves energy throughout time [**NEEDS CITATION**]. Without energy conservation, the system becomes unphysical, which can lead to unnatural behavior such as a mass on a spring extending to infinity. Two of the simplest symplectic integrators are the *Semi-implicit Euler Method*[**NEEDS CITATION**]

$$v_{n+1} = v_n + a_n \Delta t \tag{2.9a}$$

$$x_{n+1} = x_n + v_{n+1} \Delta t \tag{2.9b}$$

and the *Velocity Verlet Algorithm*[**NEEDS CITATION**]

$$v_{n+1} = v_n + \frac{a_n + a_{n+1}}{2}\Delta t \tag{2.10a}$$

$$x_{n+1} = x_n + v_n\Delta t + \frac{1}{2}a_n\Delta t^2 \tag{2.10b}$$

The velocity-Verlet algorithm in C[NEEDS CITATION TO ALGORITHM ARCHIVE]:

Parallel Computing

Because there are so many terms and equations, and each term is only needs the current step, this problem naturally lends itself to being parallelized[1].

2.1.3 Numerical Analysis

Discretization

Error associated with the size of the time-step is dictated by which specific finite-difference method is chosen.

Energy Drift

In the case of orbital mechanics, there is no loss of mechanical energy (ignoring the effects of general relativity; more on this later in section 5). Therefore, a constraint is imposed on the system that the total mechanical energy (kinetic energy + potential energy) of the system must remain constant at each point in time. When using non-symplectic methods, this constraint is not adhered to which leads to *energy drift*.

2.2 Fluid Dynamics

Fluid dynamics is

2.2.1 Mathematical Model

- Navier-Stokes

2.2.2 Computational Tools

lattice boltzmann method in parallel and on GPU

2.2.3 Numerical Analysis

NS is a PDE so CFL

2.3   Projectile Motion with Drag

Projectile motion is a fundamental concept in introductory physics as it allows students to learn how to work with problems involving forces and motion in multiple dimensions. These considerations always neglect air-resistance, also known as *drag*. This simplification is done to not only highlight the fundamentals of motion, but also because including even low-order drag terms greatly increases the complexity of the equation of motion.

2.3.1   Mathematical Model

Beginning again from Newton's second law as described with equation (2.1), we can fully determine the motion of an object under the influence of gravity and both linear and quadratic drag as[2]

$$m\vec{g} - b\left|\left|\dot{\vec{r}}\right|\right|\hat{r} - c\left|\left|\dot{\vec{r}}\right|\right|^2\hat{r} = m\ddot{\vec{r}} \tag{2.11}$$

where b, c are physical constants $\vec{g}$ is the acceleration of gravity near the surface of Earth (i.e. where the air is). Because this equation cannot be represented as the application of a differential operator on a function or vector of functions, this equation of motion is a *nonlinear* differential equation. In fact, because this is again a vector equation, there are actually three nonlinear differential equations that, due to the magnitude of the velocity, all rely on each other; thus this differential equation is said to be a *coupled system* of differential equations. Expanded out, the equation of motion becomes

$$-b\sqrt{\dot{r}_x^2 + \dot{r}_y^2}\hat{r} - c\sqrt{\dot{r}_x^2 + \dot{r}_y^2}\,\dot{r}_x = m\ddot{r}_x \tag{2.12a}$$

$$mg - c\sqrt{\dot{r}_x^2 + \dot{r}_y^2}\,\dot{r}_y = m\ddot{r}_y \tag{2.12b}$$

2.3.2   Computational Tools

2.3.3   Numerical Analysis

Chapter 3

Electromagnetism

3.1 Fringing Electric Fields of Non-Ideal Capacitors

3.1.1 Mathematical Model

- maxwell's equations

3.1.2 Computational Tools

- relaxation method

3.1.3 Numerical Analysis

- spatial mesh

3.2 Antenna Radiation Patterns

3.2.1 Mathematical Model

- maxwell's equations

3.2.2 Computational Tools

-

3.2.3 Numerical Analysis

-

Chapter 4

Quantum Mechanics

4.1 Quantum Chemistry

4.1.1 Mathematical Model

- The Schrödinger Equation

- The Many-Body Schrödinger Equation

4.1.2 Computational Tools

4.1.3 Numerical Analysis

4.2 Quadrupole-Quadrupole Interactions in a BEC

4.2.1 Mathematical Model

- The Schrödinger Equation

- The Many-Body Schrödinger Equation

4.2.2 Computational Tools

4.2.3 Numerical Analysis

Chapter 5

Relativity

5.1 Effects of Eccentricity on Mercury's Perihelion Shift

5.1.1 Mathematical Model

5.1.2 Computational Tools

5.1.3 Numerical Analysis

5.2 Mercury's Gravitational Wave Chirp

5.2.1 Mathematical Model

5.2.2 Computational Tools

5.2.3 Numerical Analysis

5.3 Strong-Field General Relativity

5.3.1 Mathematical Model

5.3.2 Computational Tools

5.3.3 Numerical Analysis

Chapter 6

Conclusion

Bibliography

[1] Dhananjay Saikumar. *Parallel N-body simulations of planetary systems: a direct approach.* 2022. arXiv: `2208.13562` `[astro-ph.EP]`.

[2] J.R. Taylor. *Classical Mechanics.* G - Reference,Information and Interdisciplinary Subjects Series. University Science Books, 2005. ISBN: 9781891389221. URL: `https://books.google.com/books?id=P1kCtNr-pJsC`.

Index