

7 Sept 23 - Notes: Numerical Integration in 1D

Principal Ideas for this unit will come from Newman Chapter 8, Section 1, which is the reading assigned below.

We have seen how we can quickly exhaust our analytical tools when we investigate systems with interesting and complex behavior. We can see much from the perspective of phase space in terms of the qualitatively different behaviors (e.g., oscillations vs rotations, both clockwise and counter-clockwise, in the large angle pendulum). But what if we want to find the specific trajectory of a given set of initial conditions.

That is we want to be able to take a more general differential equation:

$$\ddot{x} = f(x, \dot{x}, t)$$

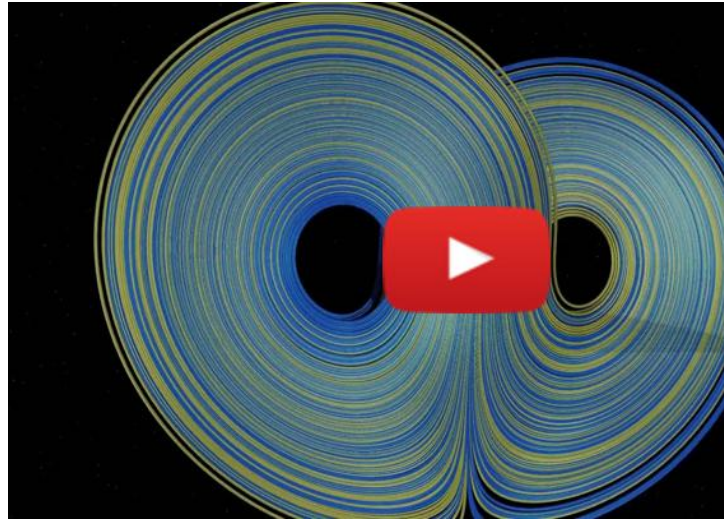
to a solution for all (or some specified amount) of time, $x(t)$...in (most) cases...because sometimes integrals don't converge. We can use **numerical integration** to find these trajectories.

Numerical Integration

Numerical Integration is a vast and wide topic with lots of different approaches, important nuances, and difficult problems. Some of the most high profile numerical integration was done by NASA's human computers – a now well-known story thanks to the film Hidden Figures. Black women formed a core group of these especially talented scientists (including Mary Jackson, Katherine Johnson, and Dorothy Vaughn), without whom, John Glenn would not have orbited the Earth in 1962. This is also a very interesting story about the importance of Historically Black Colleges and Universities to American science.

Below is a video describing 3 coupled ODEs, the Lorenz equations, that are quite famous. You will be able to model systems like this, but for now think of it as motivation for why we want to learn numerical integration.

> [Image not embedded: remote images are not included in PDF ex-



port. Check the original file for the image.]

- Non-Commercial Link: <https://inv.tux.pizza/watch?v=aAJkLh76QnM?> -
Commercial Link: <https://youtube.com/watch?v=aAJkLh76QnM?>

Nonlinear Science

Some of the most interesting ODEs to work with are those that lead to chaos, have interesting behaviors in different parts of phase space, are highly sensitive to initial conditions, or that have complex bound orbits (in real or phase space). This area of study in physics overlaps with a broad interdisciplinary field called Nonlinear Science, which studies all manner of systems where the results are not simply proportional to the input features. This work includes theory, applied science, and experimental work. As it is difficult to find truly linear systems in the real world, nonlinear science is quite interdisciplinary with physics research in the areas of magnetohydrodynamics, fluid mechanics, bio-inspired design, biophysics (cellular), biophysics (animal locomotion), atomic and molecular physics, and many more fields.

We will focus on integrating 1 dimensional ODEs (that is, ODEs of only one variable like the SHO). To that end the reading this week aims to inform you with the simplest of the integrators the Euler-Cromer integration technique. This approach is intuitive, straight-forward, and forms the basis for better and far more accurate methods like Runge-Kutta.

Through this week's activities, we will introduce several integration methods, but you will learn to use the built-in integrator (`odeint`) from `scipy`, which are more efficient than what we could write ourselves.

How Does Numerical Integration actually work?

A computer understands things like updating individual variables with a change. It turns out this process of updating things in steps is the basis for numerical

integration. We need a set of update equations. Making those update equations is effectively choosing our integrator.

Update equations

The critical part of numerical integration is approximating the change to variables you are investigating. Going back to our differential equations, we can rewrite them as approximate equation, which a computer understands because it involves discrete steps. How we choose to approximate this update indicates which integration routine we've chosen and sets the irreducible error we are stuck with (i.e., $O((\Delta t)^2)$, $O((\Delta t)^3)$, etc.).

My handwritten notes (below) provide derivations for the Euler-Cromer method. Other forms of integration are more complex, but the basic idea is the same: *we are trying to approximate the “area under the curve” by sampling the slope of the function at different points.*

We will illustrate three approximations to the slope of these functions:

- **Euler-Cromer (EC)** - definitely the most intuitive of the approaches, where we approximate the slope with two points separated by Δt in time. It is quick to write, slow to solve, and requires small steps for accurate results. Even so, it fails to integrate periodic motion well because it doesn't always conserve energy in periodic motion. Turns out it's the best tool to use when you have random noise added to the model though (e.g., $\eta_n(\sigma(t))$). For a first order eqn, $\dot{x} = f(x, t)$,

$$x(t + \Delta t) = x(t) + \text{change} = x(t) + \Delta t \left(f(x(t + \frac{1}{2}\Delta t), t + \frac{1}{2}\Delta t) \right)$$

- **Runge-Kutta 2nd order (RK2)** - just a step above Euler-Cromer; it uses three points to approximate the slope giving two measures of the slope (hence, 2nd order). It's not much more complex than Euler-Cromer, but gives an order of magnitude lower error. It's a good starting point for simple systems. For a first order eqn, $\dot{x} = f(x, t)$,

$$\begin{aligned} k_1 &= \Delta t (f(x, t)), \\ k_2 &= \Delta t \left(f\left(x + \frac{1}{2}k_1, t + \frac{1}{2}\Delta t\right) \right), \\ x(t + \Delta t) &= x(t) + \text{change} = x(t) + k_2 \end{aligned}$$

- **Runge Kutta 4th order (RK4)** - this is the gold standard. Most researchers start with RK4 on most problems. It uses 5 points to build 4 slope profiles and integrates the system in 4 steps. It is highly adaptable and supported – it can be modified to take smaller or longer steps depending on the specific nature of the problem at the time. I mean that it can change

step size in the middle of its work; including within the step it is taking presently. For a first order eqn, $\dot{x} = f(x, t)$,

$$\begin{aligned}k_1 &= \Delta t (f(x, t)), \\k_2 &= \Delta t \left(x + \frac{1}{2}k_1, t + \frac{1}{2}\Delta t \right), \\k_3 &= \Delta t \left(x + \frac{1}{2}k_2, t + \frac{1}{2}\Delta t \right), \\k_4 &= \Delta t (x + k_3, t + \Delta t), \\x(t + \Delta t) &= x(t) + \text{change} = x(t) + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4)\end{aligned}$$

We don't expect you memorize these approaches or to derive them, but to understand how they work and what their limitations are. We will implement them in class using known tools.

Additional Resources

Handwritten Notes

- Numerical Integration - 1D Spatial integrals
- Numerical Integration - Euler Cromer for ODEs

Book Readings

These two readings are useful preparation for what we are going to do with numerical integration.

Newman's Computational Physics - Chapter 8

The first one is from Chapter 8 of Mark Newman's book, Computational Physics, which is an excellent introduction to most of the computational physics approaches we use in physics. It uses Python and modern libraries, which is rare for a computational physics textbook for undergraduate students.

Read This Sections 8.1 and 8.1.1. We will discuss and work with ideas from 8.1.2 and 8.1.3, but I don't expect you to read those in detail. We will discuss how these work and then make use of the built-in integrators from scipy. The core one will be `odeint` (Documentation).

Paper Readings

Here's the original paper from Alan Cromer about the correction to the Euler step that makes it more accurate. It's a good read, but not required.

- Cromer, Alan H. "Stable solutions using the Euler approximation." American Journal of Physics 49.5 (1981): 455-459.