Ordinary Differential Equation Solvers

MATH-151: Mathematical Algorithms in Matlab

October 16, 2023

Embracing Change

- In physical systems one of the most common things we want to answer is how things change, and further, how that change creates other change
    - For example, me speeding up my legs accelerates me and changes the speed I'm walking. This changes how much where I am changes.
- These relationships are almost always described as **differential equations**, which are equations that show how some quantity relates to its derivatives. We will see them in the form

$$y^{(m)}(t) = f(t, y, y', \ldots, y^{(m-1)})$$

  where $y^{(k)}$ is the kth derivative of the function $y(t)$
- We'll see some specific examples of these in a bit, but often to solve these we need to have some other information which come in two "flavors"
    - Sometimes we know a lot about where we start and want to know what happens afterwards, this is an **initial value problem** or IVP
    - Other times we know what happens at two points and want to find what has happened in between, this is a **boundary value problem** or BVP

Starting on the Right Foot

- We'll start with IVPs, in these cases we are given where our value of interest begins and use the differential equation to find how it changes.
- For example, we can write the equations for me starting to walk from a standstill as follows

$$\begin{aligned} \frac{dv}{dt} &= a(t) \\ \frac{dp}{dt} &= v(t) \\ p(0) = 0&, \quad v(0) = 0 \end{aligned}$$

- This is a **system of differential equations**, to know where I am we need to solve both of these together!
  - Since we have two equations to solve, we have two pieces of information to start with
  - Where was I standing and how fast was I traveling to begin with
- The first equation tells me how my acceleration changes my speed, then once I know my speed I can see how that changes my position!

Forward Euler Method

- The most straightforward method for solving IVPs is to start at the beginning and "march forward" by numerically integrating our equation to get the value at the next step
- What is the simplest way to do this? Since $y' = f(t, y)$ we can calculate $f(t, y_n)$ so we will use left rectangular rule to estimate the change!
- Doing this leaves us with the **forward Euler method**

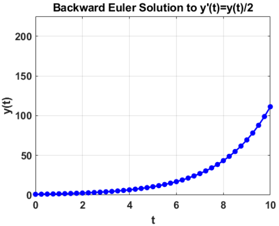
$$y_{n+1} = y_n + f(t, y_n)h$$

- Let's try this with the IVP $y' = \frac{1}{2}y$ where $y(0) = 1$, with step size $h = 0.1$

$$y_{n+1} = y_n + h\frac{y_n}{2}$$

We can find the true solution is

$$y(t) = e^{\frac{t}{2}}$$


Backward Euler Solution to y'(t)=y(t)/2

Forward Euler Method

- The most straightforward method for solving IVPs is to start at the beginning and "march forward" by numerically integrating our equation to get the value at the next step
- What is the simplest way to do this? Since $y' = f(t,y)$ we can calculate $f(t, y_n)$ so we will use left rectangular rule to estimate the change!
- Doing this leaves us with the **forward Euler method**

$$y_{n+1} = y_n + f(t, y_n)h$$

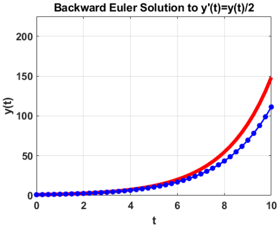
- Let's try this with the IVP $y' = \frac{1}{2}y$ where $y(0) = 1$, with step size $h = 0.1$

$$y_{n+1} = y_n + h\frac{y_n}{2}$$

We can find the true solution is

$$y(t) = e^{\frac{t}{2}}$$

Unfortunately, our approximation looks exponential, but it lags behind the true solution...

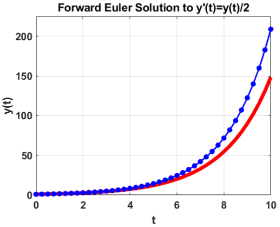


Backward Euler Method

- Forward Euler method is similar to our left rectangular rule, so maybe we might want to use right rectangular rule!

$$y_{n+1} = y_n + f(t, y_{n+1})h$$

- But wait! How can we calculate $f(t, y_{n+1})$ if we don't know y_{n+1}?
- We don't know it yet, but we can use our nonlinear solvers to solve $y_{n+1} - y_n - f(t, y_{n+1})h = 0$
- Solving these equations for each of our y_n values gives us the **backward Euler method**



- This still seems to be off, maybe we need to find a middle-ground between these two

Runge-Kutta Methods

- Runge-Kutta methods are a class of ODE solver algorithms that find weighted averages of solutions
- The most commonly seen is the 4th-order Runge-Kutta algorithm, often called just RK4. It is similar to applying Simpson's rule to integrating our differential equation. We perform it as below

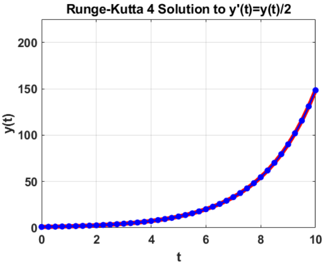
$$\begin{aligned}
y_{n+1} &= y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\
k_1 &= f(t_n, y_n) \\
k_2 &= f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1) \\
k_3 &= f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2) \\
k_4 &= f(t_n + h, y_n + hk_3)
\end{aligned}$$

- A quick way to think about it is that we keep trying to take half-steps forward and refining our guess of our slope
- It looks like a lot of work at first, but once you get comfortable it works very well!

Runge-Kutta Example Case

- The general form may look scary, but lets look at it for our $y' = \frac{1}{2}y$ example, so $f(t,y) = \frac{1}{2}y$

$$
\begin{aligned}
y_{n+1} &= y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\
k_1 &= \frac{1}{2}y_n \\
k_2 &= \frac{1}{2}(y_n + \frac{h}{2}k_1) \\
k_3 &= \frac{1}{2}(y_n + \frac{h}{2}k_2) \\
k_4 &= \frac{1}{2}(y_n + hk_3)
\end{aligned}
$$



- This is surprisingly straightforward to code up and it gives us VERY accurate solutions!
    - Our maximum error in this example was 0.0014

Setting Some Boundaries

- Another major class of ODEs are **Boundary Value Problems** or BVPs, which are cases where we know the value of our function at two endpoints and want to find what the function does in between.
- Lets return to the problem of me walking around the room. Suppose instead I know I started here and ended up

$$\frac{dv}{dt} = a(t)$$
$$\frac{dp}{dt} = v(t)$$
$$p(0) = 0, \quad p(60) = 30$$

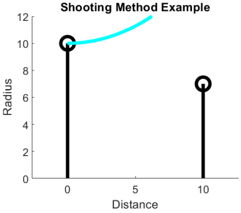
- Now this becomes less of a question of “where did I go?” and more of a question of “how did I get there?”
- These are more difficult to solve, we don’t even have enough information to start “marching forward” line we did for IVPs...

Shooting Method

- ... but what if we just guess at what we think the initial conditions would be? We can then solve it like an IVP and see if we get to the right place
- If its wrong we can "adjust our aim" and try again. We repeat this process using a nonlinear solver until we hit the target!
- Let's see it in practice. Suppose we have a soap bubble connected along two rings being pull apart. This bubble's shape will minimize the volume of a surface of revolution and satisfy the following BVP

$$\frac{du'}{dx}(x) = \frac{1 + u'(x)^2}{u(x)}$$
$$\frac{du}{dx}(x) = u'(x)$$
$$u(0) = 10, \; u(10) = 7$$

Lets aim flat



Shooting Method

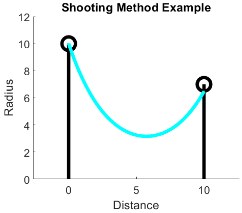
- ... but what if we just guess at what we think the initial conditions would be? We can then solve it like an IVP and see if we get to the right place
- If its wrong we can "adjust our aim" and try again. We repeat this process using a nonlinear solver until we hit the target!
- Let's see it in practice. Suppose we have a soap bubble connected along two rings being pull apart. This bubble's shape will minimize the volume of a surface of revolution and satisfy the following BVP

$$\frac{du'}{dx}(x) = \frac{1+u'(x)^2}{u(x)}$$

$$\frac{du}{dx}(x) = u'(x)$$

$$u(0) = 10,\ u(10) = 7$$

Too high, lets aim lower



Shooting Method

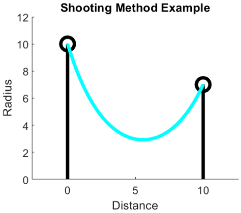
- ... but what if we just guess at what we think the initial conditions would be? We can then solve it like an IVP and see if we get to the right place
- If its wrong we can "adjust our aim" and try again. We repeat this process using a nonlinear solver until we hit the target!
- Let's see it in practice. Suppose we have a soap bubble connected along two rings being pull apart. This bubble's shape will minimize the volume of a surface of revolution and satisfy the following BVP

$$\frac{du'}{dx}(x) = \frac{1 + u'(x)^2}{u(x)}$$

$$\frac{du}{dx}(x) = u'(x)$$

$$u(0) = 10,\ u(10) = 7$$

We keep adjusting until


Shooting Method Example
Radius
Distance