ODEs: Backward Euler

October 11, 2021

Forward Euler

- Problem: Solve the ordinary differential equation $y'(t) = f(y,t)$ with initial data $y(0) = y_0$ on the domain $[0, L]$

Forward Euler

- Problem: Solve the ordinary differential equation $y'(t) = f(y,t)$ with initial data $y(0) = y_0$ on the domain $[0, L]$

- We can derive Forward Euler's Method by using Taylor series:

$$\begin{aligned} y_{n+1} = y(t_n + \Delta t) &= y(t_n) + \Delta t y'(t_n) + \frac{1}{2}(\Delta t)^2 y''(t_n) + \cdots \\ &= y(t_n) + \Delta t f(t_n, y_n) + \frac{1}{2}(\Delta t)^2 y''(t_n) + \cdots \end{aligned}$$

  so that the scheme become

$$y_{n+1} = y_n + \Delta t f(y_n, t_n)$$

Backward Euler

- Forward Euler:

$$y_{n+1} = y_n + \Delta t f(y_n, t_n)$$

- For Backward Euler, we use Taylor series, but expand differently:

$$\begin{aligned} y(t_n) = y(t_{n+1} - \Delta t) &= y(t_{n+1}) - \Delta t y'(t_{n+1}) + \frac{1}{2}(\Delta t)^2 y''(t_{n+1}) + \cdots \\ &= y(t_{n+1}) - \Delta t f(t_{n+1}, y_{n+1}) + \frac{1}{2}(\Delta t)^2 y''(t_{n+1}) + \cdots \end{aligned}$$

- This leads to the scheme

$$y_{n+1} = y_n + \Delta t f(t_{n+1}, y_{n+1})$$

Backward Euler

- Backward Euler:

$$y_{n+1} = y_n + \Delta t f(t_{n+1}, y_{n+1})$$

- This is an *implicit* method because it involves solving an equation for y_{n+1}.

Backward Euler

- Backward Euler:

$$y_{n+1} = y_n + \Delta t f(t_{n+1}, y_{n+1})$$

- This is an *implicit* method because it involves solving an equation for y_{n+1}.
- For example, the ODE $y' = y + yt + 1$ has $f(t, y) = y + yt + 1$.

Backward Euler

- Backward Euler:

$$y_{n+1} = y_n + \Delta t f(t_{n+1}, y_{n+1})$$

- This is an *implicit* method because it involves solving an equation for y_{n+1}.
- For example, the ODE $y' = y + yt + 1$ has $f(t, y) = y + yt + 1$.
- Therefore, the method becomes

$$y_{n+1} = y_n + \Delta t f(t_{n+1}, y_{n+1}) = y_n + \Delta t \left(y_{n+1} + y_{n+1} t_{n+1} + 1\right)$$

Backward Euler

- Backward Euler:
$$y_{n+1} = y_n + \Delta t f(t_{n+1}, y_{n+1})$$
- This is an *implicit* method because it involves solving an equation for y_{n+1}.
- For example, the ODE $y' = y + yt + 1$ has $f(t, y) = y + yt + 1$.
- Therefore, the method becomes

$$y_{n+1} = y_n + \Delta t f(t_{n+1}, y_{n+1}) = y_n + \Delta t \left(y_{n+1} + y_{n+1} t_{n+1} + 1\right)$$

or that

$$y_{n+1} \left(1 - \Delta t - (\Delta t) t_{n+1}\right) = y_n + \Delta t$$

or

$$y_{n+1} = \frac{y_n + \Delta t}{1 - \Delta t - (\Delta t) t_{n+1}}$$

Backward Euler

- Backward Euler:

$$y_{n+1} = y_n + \Delta t f(t_{n+1}, y_{n+1})$$

- Another example: consider the ODE $y' = y \cos y$ which has $f(t, y) = y \cos y$.

Backward Euler

- Backward Euler:

$$y_{n+1} = y_n + \Delta t f(t_{n+1}, y_{n+1})$$

- Another example: consider the ODE $y' = y \cos y$ which has $f(t, y) = y \cos y$.
- In this case, the scheme becomes

$$y_{n+1} - \Delta t y_{n+1} \cos y_{n+1} = y_n$$

which has no closed form solution.

Backward Euler

- Backward Euler:

$$y_{n+1} = y_n + \Delta t f(t_{n+1}, y_{n+1})$$

- Another example: consider the ODE $y' = y \cos y$ which has $f(t,y) = y \cos y$.
- In this case, the scheme becomes

$$y_{n+1} - \Delta t y_{n+1} \cos y_{n+1} = y_n$$

  which has no closed form solution.
- Here, we need to use some root finder, like Newton's Method or the Bisection Method:

$$y_{n+1} - \Delta t y_{n+1} \cos y_{n+1} - y_n = 0$$

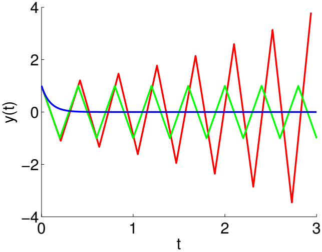
Backward Euler

- Why use Backward Euler, where the complexity of writing the program is much higher?

Backward Euler

- Why use Backward Euler, where the complexity of writing the program is much higher?

- Recall the problem $y' = -10y$ with $y(0) = 1$. The exact solution is $y(x) = e^{-10t}$.
- There is a stability requirement: our step size needs to be sufficiently small.



Stability

- Why do we need this stepsize requirement? Consider Forward Euler's method for this problem

$$y_{n+1} = y_n + \Delta t(-10y_n) = y_{n-1}(1 - 10\Delta t)^2 = y_0(1 - 10\Delta t)^{n+1}$$

- So in order for our solution to decay (like we expect), we need

$$|1 - 10\Delta t| < 1$$

or that

$$\Delta t < 0.2$$

Stability

- What stepsize requirement do we need for Backward Euler?

$$y_{n+1} = y_n + \Delta t(-10y_{n+1})$$

Stability

- What stepsize requirement do we need for Backward Euler?

$$y_{n+1} = y_n + \Delta t(-10 y_{n+1})$$

- This reduces to

$$(1 + 10\Delta t)\, y_{n+1} = y_n$$

or

$$y_{n+1} = \left(\frac{1}{1+10\Delta t}\right) y_n = \left(\frac{1}{1+10\Delta t}\right)^{n+1} y_0$$

- So in order for our solution to decay, we need

$$\left|\frac{1}{1+10\Delta t}\right| < 1$$

or that

$$1 < |1 + 10\Delta t|$$

- This is satisfied for any value of Δt! There is no time-step requirement!