Why is timestepping so hard?

Jemma Shipton

January 2023

Solving time dependent problems

Ordinary differential equation

We wish to approximate the solution of

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0 \tag{1}$$

where f is a ‘nice’ function that we know.

We know, or can evaluate,

1. the value of y at t_0,
2. the *slope*, given by $\frac{dy}{dt}$ for any value of y.

Integrating equation 1

$$y(t) = y(t_0) + \int_{t_0}^{t} f(\tau, y(\tau)) d\tau \tag{2}$$

Euler's method

Approximate $f(t, y) \approx f(t_0, y_0)$ for $t_0 < t < t_0 + \Delta t$:

$$\begin{aligned} y(t) &= y(t_0) + \int_{t_0}^{t} f(\tau, y(\tau)) d\tau, \\ &\approx y_0 + (t - t_0) f(t_0, y_0). \end{aligned}$$

Given a sequence $t_0, t_0 + \Delta t, t_0 + 2\Delta t, \ldots$ we can generate a sequence of approximate solutions

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

where y_n is our approximate solution for y at time $t_n = t_0 + n\Delta t$.

Euler's method

Approximate $f(t, y) \approx f(t_0, y_0)$ for $t_0 < t < t_0 + \Delta t$:

$$y(t) = y(t_0) + \int_{t_0}^{t} f(\tau, y(\tau)) d\tau,$$
$$\approx y_0 + (t - t_0) f(t_0, y_0).$$

Given a sequence $t_0, t_0 + \Delta t, t_0 + 2\Delta t, \ldots$ we can generate a sequence of approximate solutions

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

where y_n is our approximate solution for y at time $t_n = t_0 + n\Delta t$.

But how do we know if it is any good?

Other methods

Trapezoidal rule

Instead of approximating the derivative by a constant value at t_0, use the average of the endpoint values:

$$\begin{aligned} y(t) &= y(t_0) + \int_{t_0}^{t} f(\tau, y(\tau)) d\tau, \\ &\approx y_0 + \frac{1}{2}(t - t_0)(f(t_0, y_0) + f(t, y)). \end{aligned}$$

which gives the trapezoidal rule:

$$y_{n+1} = y_n + \frac{1}{2}\Delta t(f(t_n, y_n) + f(t_{n+1}, y_{n+1})).$$

Theta method

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

Other methods

Implicit midpoint rule

Instead of approximating the derivative by the average of the endpoint values, take the value at the midpoint:

$$y'(t) \approx f\left(t_n + \frac{1}{2}\Delta t, \frac{1}{2}(y_n + y_{n+1})\right).$$

which gives the implicit midpoint rule:

$$y_{n+1} = y_n + \Delta t f\left(t_n + \frac{1}{2}\Delta t, \frac{1}{2}(y_n + y_{n+1})\right).$$

But how do we know if it is any good?

Local truncation error

This is the error that the method makes in one step:

$$\tau = ||y(t_0 + \Delta t) - y_1||$$

Global truncation error

This is the error that the method makes over many steps:

$$E = ||y(T) - y_N||, \quad T = N\Delta t$$

For numerical methods for solving ordinary differential equations, the global truncation error is one order less than the local truncation error.

But how do we know if it is any good?

For the forward Euler method we have:

Local truncation error

$$\begin{aligned}\tau &= ||y(t_0+\Delta t) - y_1||,\\ &= ||y_0 + \Delta t y'(t_0) + \mathcal{O}(\Delta t^2) - (y_0 + \Delta t f(t_0, y_0))||,\\ &= \mathcal{O}(\Delta t^2).\end{aligned}$$

where we have used the Taylor series expansion for $y(\Delta t)$, substituted for y_1 from the forward Euler formula, and used the original ODE to cancel terms.

Global truncation error

This is one order less than the local truncation error, meaning that forward Euler has global truncation error of $\mathcal{O}(\Delta t)$.

Consistency

A consistent method is one for which the local truncation error approaches zero as $\Delta t \to 0$.

Convergence

A convergent method produces a numerical solution that tends to the true solution, i.e. the global trunction error goes to zero, as the timestep is decreased.

Order

If the local truncation error is of order $\mathcal{O}(\Delta t^{p+1})$ then the method is said to have *order of accuracy* p.

Accuracy

- Formal definitions of accuracy rely on Taylor series and therefore assume smoothness of solutions that may not be realistic.
- The accuracy of the solution is considered as $\Delta t \to 0$, which is not what happens in practice.
- Durran: ‘If the function is sufficiently rough and t is sufficiently coarse, neither formula is likely to produce a good approximation, and the superiority of one over the other will be largely a matter of chance.’

Stability

Consistency of order p and *stability* imply convergence of order p. Define the *amplification factor* A as the ratio of the numerical solution at consecutive timesteps:

$$A = \frac{y_{n+1}}{y_n}.$$

Dahlquist equation

Simple, but important, test problem:

$$y_t = \lambda\phi, \quad y(0) = y_0, \quad \lambda \in \mathbb{C}.$$

This has solution

$$y(t) = y_0 e^{\lambda t}.$$

Solutions grow if $\Re(\lambda) > 0$ and decay if $\Re(\lambda) < 0$.

Timestepping notebook

Exercises to implement different timestepping schemes and investigate the stability.

Partial differential equations

Dahlquist's equation is very simple but still relevant for thinking about time discretisation of partial differential equations - consider the solution as a superposition of Fourier modes.

Diffusion equation

This partial differential equation governs diffusion of a substance ϕ:

$$\frac{\partial \phi}{\partial t} = \nu \frac{\partial^2 \phi}{\partial x^2},$$

where ν is the diffusivity.

This equation is also known as the heat equation as it can be used to model the conduction of heat.

Transport equation

This partial differential equation governs the transport, or advection, of a substance ϕ:

$$\frac{\partial\phi}{\partial t} + c\frac{\partial\phi}{\partial x} = 0,$$

where c is the transport speed.

The solution of this equation is of the form

$$\phi(x,t) = F(x - ct).$$

[You can see this by substituting this expression into the transport equation and using the chain rule.]

More complexity...

We can build up the complexity of the problems we consider by:

Combining advection and diffusion:

$$\frac{\partial \phi}{\partial t} + c\frac{\partial \phi}{\partial x} = \nu \frac{\partial^2 \phi}{\partial x^2}.$$

Introducing nonlinearity:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}.$$

Considering higher dimensions:

$$\frac{\partial \phi}{\partial t} + \mathbf{c} \cdot \nabla \phi = \nu \nabla^2 \phi.$$

Introducing some forcing:

$$\frac{\partial \phi}{\partial t} + c\frac{\partial \phi}{\partial x} = \nu \frac{\partial^2 \phi}{\partial x^2} + f(t, x).$$