9 Numerical Integration

We want to evaluate a numerical approximation to the integral

$$\int_{x_0}^{x_N} f(x)\,dx.$$

We are able to evaluate the function $f(x)$ at a set of equally spaced points $x_0, x_1, x_2, ..., x_N$, with $x_k = x_0 + kh$, where $h = (x_N - x_0)/N$ is the interval size.

A given scheme is said to *converge* if the error in the integral tends to zero as $N \to \infty$ and $h \to 0$.

9.1 Trapezoidal rule

Approximate the function $f(x)$ as a piecewise linear function, and hence approximate the integral as a sum of the areas of the trapezoids thus formed.

$$\int_{x_0}^{x_1} f(x)\,dx = h\left[\frac{1}{2}f_0 + \frac{1}{2}f_1\right] + O\left(h^3 f''\right). \tag{8}$$

$$\begin{aligned}&\int_{x_0}^{x_N} f(x)\,dx = \\ &h\left[\frac{1}{2}f_0 + f_1 + f_2 + ... + f_{N-1} + \frac{1}{2}f_N\right] + O\left(\frac{(x_N - x_0)^3 f''}{N^2}\right).\end{aligned} \tag{9}$$

9.2 Simpson's rule

Approximate the function $f(x)$ as a piecewise quadratic function, with each quadratic fitted through three successive data points. Then approximate the integral by the sum of the areas under the piecewise quadratics.

$$\int_{x_0}^{x_2} f(x)\,dx = h\left[\frac{1}{3}f_0 + \frac{4}{3}f_1 + \frac{1}{3}f_2\right] + O\left(h^5 f^{iv}\right). \tag{10}$$

For even N

$$\begin{aligned}&\int_{x_0}^{x_N} f(x)\,dx = \\ &h\left[\frac{1}{3}f_0 + \frac{4}{3}f_1 + \frac{2}{3}f_2 + ... + \frac{2}{3}f_{N-2} + \frac{4}{3}f_{N-1} + \frac{1}{3}f_N\right] + O\left(\frac{(x_N - x_0)^5 f^{iv}}{N^4}\right).\end{aligned} \tag{11}$$

Note the alternating 2/3, 4/3 pattern of the coefficients continues for all points except the first and last.

9.3 Romberg integration

The dominant error in the trapezoidal rule is $O(1/N^2)$, and the next contribution is $O(1/N^4)$. Suppose we appoximate an integral with the trapezoidal rule using N intervals,

$$\int_{x_0}^{x_N} f(x)\,dx = h\left[\frac{1}{2}f_0 + f_1 + f_2 + ... + f_{N-1} + \frac{1}{2}f_N\right] + \frac{C}{N^2} + O\left(\frac{1}{N^4}\right), \tag{12}$$

and also using $N/2$ intervals

$$\int_{x_0}^{x_N} f(x)\,dx = 2h\left[\frac{1}{2}f_0 + f_2 + ... + f_{N-2} + \frac{1}{2}f_N\right] + \frac{4C}{N^2} + O\left(\frac{1}{N^4}\right). \tag{13}$$

We can eliminate the dominant error by taking 4/3 times (12) minus 1/3 times (13) to obtain

$$\int_{x_0}^{x_N} f(x)\,dx = h\left[\frac{1}{3}f_0 + \frac{4}{3}f_1 + \frac{2}{3}f_2 + ... + \frac{2}{3}f_{N-2} + \frac{4}{3}f_{N-1} + \frac{1}{3}f_N\right] + O\left(\frac{1}{N^4}\right). \tag{14}$$

This is just Simpson's rule!

The generalization of this idea, using the trapezoidal rule with different numbers of intervals to eliminate the dominant error terms, is known as *Romberg integration*.

10 Numerical Differentiation

10.1 First order Euler finite difference scheme

By rearranging the Taylor series for $f(x+h)$ we obtain the Forward Euler formula

$$f'(x) = \frac{f(x+h) - f(x)}{h} + O(h). \tag{15}$$

Similarly, by rearranging the Taylor series for $f(x-h)$ we obtain the Backward Euler formula

$$f'(x) = \frac{f(x) - f(x-h)}{h} + O(h). \tag{16}$$

Both are first order accurate—i.e. the leading truncation error scales like h to the power 1 as $h \to 0$.

10.2 Second order centred difference scheme

By combining the Taylor series for $f(x+h)$ and $f(x-h)$ we obtain the centred difference formula

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2). \tag{17}$$

This scheme is second order accurate.

10.3 Fourth order centred difference scheme

By combining the Taylor series for $f(x+2h)$, $f(x+h)$, $f(x-h)$, and $f(x-2h)$, we obtain the fourth order centred difference formula

$$f'(x) = \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h} + O(h^4). \tag{18}$$

10.4 Centred difference formula for second derivative

By combining Taylor series for $f(x+h)$ and $f(x-h)$ we can obtain a formula for the second derivative

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + O(h^2). \tag{19}$$

11 Time stepping schemes

We can use some of the finite difference formulas of section 10 as the basis for time stepping schemes. Suppose we want to integrate in time the equation

$$\frac{dy}{dt} = F(y)$$

from some initial condition $y(0) = y_0$, and that we want to find the solution values at regularly spaced time steps $t^{(n)} = n\Delta t$, $n = 0,\ 1,\ 2,\ \ldots$ with Δt a constant.

11.1 Forward Euler scheme

If we use the forward Euler formula to approximate the time derivative we obtain

$$\frac{y^{(n+1)} - y^{(n)}}{\Delta t} = F\left(y^{(n)}\right)$$

where superscript (n) means at timestep n.

This scheme is first order accurate. This means that if we substitute the true solution into the finite difference equation the residual is order Δt to the power 1.

However, a given order of accuracy does not necessarily imply that the scheme will work well in practice; it must also be stable. There are many definitions of stability used in numerical analysis. Here we will adopt a simple and practical one: a scheme will be called stable if perturbations to the numerical solution do not grow when perturbations to the true solution do not grow.

Consider a simple example in which $F(y) = -\lambda y$ with λ real. Then the true solution is $y = y_0 \exp(-\lambda t)$, so perturbations to the true solution do not grow if and only if $\lambda \geq 0$. How does the numerical solution behave?

The numerical scheme becomes

$$\frac{y^{(n+1)} - y^{(n)}}{\Delta t} = -\lambda y^{(n)}$$

or

$$y^{(n+1)} = (1 - \lambda \Delta t)\, y^{(n)}.$$

Thus the numerical solution changes by a factor $A = (1 - \lambda \Delta t)$ at each step. A is called the *amplification factor*.

- If $\lambda \Delta t < 1$ then $0 < A < 1$; the numerical solution decays by a factor A at each step, so the scheme is stable.
- If $1 < \lambda \Delta t < 2$ then $-1 < A < 0$; the numerical solution still decays in amplitude but oscillates in sign. The scheme is still stable, but not very accurate.
- If $2 < \lambda \Delta t$ then $A < -1$; the numerical solution oscillates and grows. The scheme is now unstable.

For this problem the forward Euler scheme is *conditionally stable*; its stability depends on the size of Δt.

Equations with oscillating solutions are also of interest. E.g., $F(y) = i\omega y$ with ω real. The true solution now is $y = y_0 \exp(i\omega t)$, so perturbations to the true solution do not grow for any value of ω. However, the amplification factor for the forward Euler scheme is $A = (1 + i\omega \Delta t)$. This always has $|A| > 1$, so the forward Euler scheme is unconditionally unstable for this type of equation.

11.2 General method of stability analysis

A general method for analysing the stability of linear schemes with constant coefficients is to seek solutions $y^{(n)} \propto A^n$. Substituting a solution of this form into the scheme leads to an equation for the amplification factor A. In general A may be a complex number. If we find $|A| > 1$ when the true solution does not grow then the scheme is unstable.

If the coefficients in the problem are not constant then this method can still be used, but only gives guidance as to the stability of the scheme rather than a definitive answer.

For nonlinear problems we must linearize the function $F(y)$ in order to examine the stability of a numerical scheme. Suppose $y(t)$ is a solution of the original equation. Then a small perturbation δy will evolve according to

$$\frac{d\delta y}{dt} = F'(y)\delta y$$

The above argument then applies with $-F'(y)$ in place of λ. Again, it will only give guidance as to the stability of the scheme rather than a definitive answer.

11.3 Leapfrog scheme

Approximate the time derivative by a centred difference:

$$\frac{y^{(n+1)} - y^{(n-1)}}{2\Delta t} = F\left(y^{(n)}\right).$$

This scheme is second order accurate.

For $F(y) = -\lambda y$ or $F(y) = i\omega y$, stability analysis gives a quadratic equation with two roots, A_1 and A_2 say, for the amplification factor A. This means the general form of the numerical solution is

$$y^{(n)} = C_1 A_1^n + C_2 A_2^n$$

for some constants C_1 and C_2.

One of the roots approaches 1 as $\Delta t \to 0$; it corresponds to the true solution of the equation and is called the *physical mode*. The other root does not approach 1 as $\Delta t \to 0$; it is an artefact of the solution method and is called the *computational mode*. (Schemes that use three or more time levels, like the leapfrog scheme, tend to have computational modes because they lead to a quadratic, or higher order equation, for A).

For $F(y) = i\omega y$ we find $|A| = 1$ for both roots provided $|\omega \Delta t| \leq 1$, so the scheme is conditionally stable. However, for $F(y) = -\lambda y$ the computational mode is always unstable.

11.4 Implicit schemes

Improved stability can often be obtained by using implicit schemes like the *backward Euler scheme*

$$\frac{y^{(n+1)} - y^{(n)}}{\Delta t} = F\left(y^{(n+1)}\right)$$

or the *trapezoidal implicit scheme* (also called the *Crank-Nicolson scheme*)

$$\frac{y^{(n+1)} - y^{(n)}}{\Delta t} = \frac{1}{2}\left[F\left(y^{(n)}\right) + F\left(y^{(n+1)}\right)\right]$$

However, implicit schemes require a (possibly nonlinear) equation to be solved at each time step for $y^{(n+1)}$. Schemes similar to these are often used in modelling atmospheric chemistry, where the very fast reaction rates for some species would require unfeasibly short time steps with an explicit scheme.

11.5 Multistep methods

Integrating the original equation from $t^{(n)}$ to $t^{(n+1)}$ gives

$$y^{(n+1)} - y^{(n)} = \int_{t^{(n)}}^{t^{(n+1)}} F(y(\tau))\, d\tau.$$

Multistep schemes approximate the average value of $F(y)$ on the right hand side using $F(y^{(k)})$ from a number of time steps k. The trapezoidal implicit scheme is one example. The *second order Adams-Bashforth* scheme

$$\frac{y^{(n+1)} - y^{(n)}}{\Delta t} = \frac{1}{2}\left[3F\left(y^{(n)}\right) - F\left(y^{(n-1)}\right)\right]$$

is another. The second order Adams Bashforth scheme is a three-time-level scheme, so it has a computational mode, but the computational mode is strongly damped. However, the physical mode is weakly unstable for the problem $F(y) = i\omega y$. Higher order Adams-Bashforth schemes are possible using additional F values from even earlier times.

11.6 Multistage methods

Multistage methods attempt to obtain higher accuracy by estimating F at time levels intermediate between $t^{(n)}$ and $t^{(n+1)}$. For example, the *second order Runge-Kutta method* is

$$\begin{aligned} \frac{y^* - y^{(n)}}{\Delta t} &= \frac{1}{2} F\left(y^{(n)}\right) \\ \frac{y^{(n+1)} - y^{(n)}}{\Delta t} &= F\left(y^*\right). \end{aligned}$$

This particular Runge-Kutta scheme is weakly unstable for the problem $F(y) = i\omega y$. Higher order Runge-Kutta schemes are possible using more intermediate values of F. For example, the fourth order Runge-Kutta scheme is often used for Lagrangian air parcel trajectory calculations.