Mathematical Tripos Part IB: Lent 2010

Numerical Analysis – Lecture 9[1]

Formally, $\boldsymbol{y}(t_{n+1}) = \boldsymbol{y}(t_n) + \int_{t_n}^{t_{n+1}} \boldsymbol{f}(t, \boldsymbol{y}(t))\mathrm{d}t$, and this can be 'approximated' by

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + h \sum_{l=1}^{\nu} b_l \boldsymbol{f}(t_n + c_l h, \boldsymbol{y}(t_n + c_l h)). \tag{4.11}$$

except that, of course, the vectors $\boldsymbol{y}(t_n + c_l h)$ are unknown! *Runge–Kutta methods* are a means of implementing (4.11) by replacing unknown values of $\boldsymbol{y}$ by suitable linear combinations. The general form of a *ν-stage explicit Runge–Kutta method (RK)* is

$$\begin{aligned}
\boldsymbol{k}_1 &= \boldsymbol{f}(t_n, \boldsymbol{y}_n),\\
\boldsymbol{k}_2 &= \boldsymbol{f}(t_n + c_2 h, \boldsymbol{y}_n + h c_2 \boldsymbol{k}_1),\\
\boldsymbol{k}_3 &= \boldsymbol{f}(t_n + c_3 h, \boldsymbol{y}_n + h(a_{3,1}\boldsymbol{k}_1 + a_{3,2}\boldsymbol{k}_2)), \qquad a_{3,1} + a_{3,2} = c_3,\\
&\vdots\\
\boldsymbol{k}_\nu &= \boldsymbol{f}\left(t_n + c_\nu h, \boldsymbol{y}_n + h \sum_{j=1}^{\nu-1} a_{\nu,j}\boldsymbol{k}_j\right), \qquad \sum_{j=1}^{\nu-1} a_{\nu,j} = c_\nu,\\
\boldsymbol{y}_{n+1} &= \boldsymbol{y}_n + h \sum_{l=1}^{\nu} b_l \boldsymbol{k}_l.
\end{aligned}$$

The choice of the *RK coefficients* $a_{l,j}$ is motivated at the first instance by order considerations.

Example Set $\nu = 2$. We have $\boldsymbol{k}_1 = \boldsymbol{f}(t_n, \boldsymbol{y}_n)$ and, Taylor-expanding about $(t_n, \boldsymbol{y}_n)$,

$$\begin{aligned}
\boldsymbol{k}_2 &= \boldsymbol{f}(t_n + c_2 h, \boldsymbol{y}_n + c_2 h \boldsymbol{f}(t_n, \boldsymbol{y}_n))\\
&= \boldsymbol{f}(t_n, \boldsymbol{y}_n) + h c_2 \left[\frac{\partial \boldsymbol{f}(t_n, \boldsymbol{y}_n)}{\partial t} + \frac{\partial \boldsymbol{f}(t_n, \boldsymbol{y}_n)}{\partial \boldsymbol{y}} \boldsymbol{f}(t_n, \boldsymbol{y}_n)\right] + \mathcal{O}(h^2).
\end{aligned}$$

But

$$\boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y}) \qquad \Rightarrow \qquad \boldsymbol{y}'' = \frac{\partial \boldsymbol{f}(t, \boldsymbol{y})}{\partial t} + \frac{\partial \boldsymbol{f}(t, \boldsymbol{y})}{\partial \boldsymbol{y}} \boldsymbol{f}(t, \boldsymbol{y}).$$

Therefore, substituting the exact solution $\boldsymbol{y}_n = \boldsymbol{y}(t_n)$, we obtain $\boldsymbol{k}_1 = \boldsymbol{y}'(t_n)$ and $\boldsymbol{k}_2 = \boldsymbol{y}'(t_n) + h c_2 \boldsymbol{y}''(t_n) + \mathcal{O}(h^2)$. Consequently, the *local* error is

$$\begin{aligned}
\boldsymbol{y}(t_{n+1}) - \boldsymbol{y}_{n+1} = {} & [\boldsymbol{y}(t_n) + h\boldsymbol{y}'(t_n) + \tfrac{1}{2}h^2 \boldsymbol{y}''(t_n) + \mathcal{O}(h^3)]\\
& - [\boldsymbol{y}(t_n) + h(b_1 + b_2)\boldsymbol{y}'(t_n) + h^2 b_2 c_2 \boldsymbol{y}''(t_n) + \mathcal{O}(h^3)].
\end{aligned}$$

We deduce that the RK method is of order 2 if $b_1 + b_2 = 1$ and $b_2 c_2 = \frac{1}{2}$. It is easy to demonstrate that no such method may be of order ≥ 3 (e.g. by applying it to $y' = \lambda y$).

General RK methods A general ν-stage *Runge–Kutta method* is

$$\boldsymbol{k}_l = \boldsymbol{f}\left(t_n + c_l h, \boldsymbol{y}_n + h \sum_{j=1}^{\nu} a_{l,j}\boldsymbol{k}_j\right) \quad \text{where} \quad \sum_{j=1}^{\nu} a_{l,j} = c_l, \qquad l = 1, 2, \ldots, \nu,$$

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + h \sum_{l=1}^{\nu} b_l \boldsymbol{k}_l.$$

Obviously, $a_{l,j} = 0$ for all $l \leq j$ yields the standard *explicit* RK. Otherwise, an RK method is said to be *implicit*.

[1]Corrections and suggestions to these notes should be emailed to `A.Iserles@damtp.cam.ac.uk`. All handouts are available on the WWW at the URL `http://www.damtp.cam.ac.uk/user/na/PartIB/Handouts.html`.

Example Consider the 2-stage method

$$\boldsymbol{k}_1 = \boldsymbol{f}\left(t_n, \boldsymbol{y}_n + \tfrac{1}{4}h(\boldsymbol{k}_1 - \boldsymbol{k}_2)\right), \qquad \boldsymbol{k}_2 = \boldsymbol{f}\left(t_n + \tfrac{2}{3}h, \boldsymbol{y}_n + \tfrac{1}{12}h(3\boldsymbol{k}_1 + 5\boldsymbol{k}_2)\right),$$
$$\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + \tfrac{1}{4}h(\boldsymbol{k}_1 + 3\boldsymbol{k}_2).$$

In order to analyse the order of this method, we restrict our attention to scalar, autonomuous equations of the form $y' = f(y)$. (This procedure might lead to loss of generality for methods of order ≥ 5.) For brevity, we use the convention that all functions are evaluated at $y = y_n$, e.g. $f_y = \mathrm{d}f(y_n)/\mathrm{d}y$. Thus,

$$k_1 = f + \tfrac{1}{4}hf_y(k_1 - k_2) + \tfrac{1}{32}h^2 f_{yy}(k_1 - k_2)^2 + \mathcal{O}(h^3),$$
$$k_2 = f + \tfrac{1}{12}hf_y(3k_1 + 5k_2) + \tfrac{1}{288}h^2 f_{yy}(3k_1 + 5k_2)^2 + \mathcal{O}(h^3).$$

We have $k_1, k_2 = f + \mathcal{O}(h)$ and substitution in the above equations yields $k_1 = f + \mathcal{O}(h^2)$, $k_2 = f + \frac{2}{3}hf_y f + \mathcal{O}(h^2)$. Substituting again, we obtain

$$k_1 = f - \tfrac{1}{6}h^2 f_y^2 f + \mathcal{O}(h^3),$$
$$k_2 = f + \tfrac{2}{3}hf_y f + h^2\left(\tfrac{5}{18}f_y^2 f + \tfrac{2}{9}f_{yy}f^2\right) + \mathcal{O}(h^3)$$
$$\Rightarrow \quad y_{n+1} = y + hf + \tfrac{1}{2}h^2 f_y f + \tfrac{1}{6}h^3(f_y^2 f + f_{yy}f^2) + \mathcal{O}(h^4).$$

But $y' = f \Rightarrow y'' = f_y f \Rightarrow y''' = f_y^2 f + f_{yy}f^2$ and we deduce from Taylor's theorem that the method is at least of order 3. (It is easy to verify that it isn't of order 4, for example applying it to the equation $y' = \lambda y$.)

4.4 Stiff equations

Linear stability Consider the linear system

$$\boldsymbol{y}' = A\boldsymbol{y} \qquad \text{where} \qquad A = \begin{bmatrix} -100 & 1 \\ 0 & -\frac{1}{10} \end{bmatrix}.$$

The exact solution is a linear combination of $\mathrm{e}^{-t/10}$ and e^{-100t}: the first decays gently, whereas the second becomes practically zero almost at once. Suppose that we solve the ODE with the *forward Euler* method. As will be shown soon, the requirement that $\lim_{n\to\infty} \boldsymbol{y}_n = \boldsymbol{0}$ (for fixed $h > 0$) leads to an unacceptable restriction on the size of h.

With greater generality, let us solve $\boldsymbol{y}' = A\boldsymbol{y}$, for general $N \times N$ constant matrix A, with Euler's method. Then $\boldsymbol{y}_{n+1} = (I + hA)\boldsymbol{y}_n$, therefore $\boldsymbol{y}_n = (I + hA)^n \boldsymbol{y}_0$. Let the eigenvalues of A be $\lambda_1, \ldots, \lambda_N$, with corresponding linearly-independent eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_N$. Let $D = \mathrm{diag}\boldsymbol{\lambda}$ and $V = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_N]$, whence $A = VDV^{-1}$. We assume further that $\mathrm{Re}\,\lambda_l < 0$, $l = 1, \ldots, N$. In that case it is easy to prove that $\lim_{t\to\infty} \boldsymbol{y}(t) = \boldsymbol{0}$, e.g. by representing the exact solution of the ODE explicitly as $\boldsymbol{y}(t) = \mathrm{e}^{tA}\boldsymbol{y}_0$, where $\mathrm{e}^{tA} = \sum_{k=0}^{\infty} \frac{1}{k!}t^k A^k = V\mathrm{e}^{tD}V^{-1}$. However, $\boldsymbol{y}_n = V(I + hD)^n V^{-1}\boldsymbol{y}_0$, where $A = VDV^{-1}$ and the matrix D is diagonal, therefore $\lim_{n\to\infty} \boldsymbol{y}_n = \boldsymbol{0}$ for all initial values $\boldsymbol{y}_0$ iff $|1 + h\lambda_l| < 1$, $l = 1, \ldots, N$. In our example we thus require $|1 - \frac{1}{10}h|, |1 - 100h| < 1$, hence $h < \frac{1}{50}$.

This restriction, necessary to recovery of correct asymptotic behaviour, has *nothing* to do with local accuracy, since, for large n, the genuine 'unstable' component is exceedingly small. Its purpose is solely to prevent this component from leading to an unbounded growth in the numerical solution.

Stiffness We say that the ODE $\boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y})$ is *stiff* if (for some methods) we need to depress h to maintain *stability* well beyond requirements of accuracy. An important example of stiff systems occurs when an equation is linear, $\mathrm{Re}\,\lambda_l < 0$, $l = 1, 2, \ldots, N$, and the quotient $\max|\lambda_k| / \min|\lambda_k|$ is large: a ratio of 10^{20} is not unusual in real-life problems!

Stiff equations, mostly nonlinear, occur throughout applications, whenever we have two (or more) different timescales in the ODE. A typical example are equations of *chemical kinetics,* where each timescale is determined by the speed of reaction between two compounds: such speeds can differ by many orders of magnitude.