Mathematical Tripos Part IB: Lent 2020
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 $(t_n, \boldsymbol{y}_n)$,

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

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 $\boldsymbol{k}_2 = \boldsymbol{y}'(t_n) + h c_2 \boldsymbol{y}''(t_n) + \mathcal{O}\big(h^2\big)$. Consequently, the *local* error is

$$\begin{aligned}
\boldsymbol{y}(t_{n+1}) - (\boldsymbol{y}(t_n) + h b_1 \boldsymbol{k}_1 + h b_2 \boldsymbol{k}_2) &= [\boldsymbol{y}(t_n) + h\boldsymbol{y}'(t_n) + \tfrac{1}{2}h^2 \boldsymbol{y}''(t_n) + \mathcal{O}\big(h^3\big)]\\
&\quad - [\boldsymbol{y}(t_n) + h(b_1 + b_2)\boldsymbol{y}'(t_n) + h^2 b_2 c_2 \boldsymbol{y}''(t_n) + \mathcal{O}\big(h^3\big)].
\end{aligned}$$

We deduce that the RK method is of order 2 if $b_1 + b_2 = 1$ and $b_2 c_2 = \frac{1}{2}$. We can demonstrate that no such method may be of order ≥ 3. To show this consider the ODE $y' = y$ with $y(0) = 1$ whose solution is $y(t) = \mathrm{e}^t$. For this ODE we can write the local error explicitly: indeed we have $k_1 = f(t_n, y(t_n)) = \mathrm{e}^{t_n}$ and $k_2 = f(t_n + c_2 h, y(t_n) + c_2 h k_1) = y(t_n) + c_2 h k_1 = \mathrm{e}^{t_n}(1 + c_2 h)$. Then the local error is

$$\begin{aligned}
y(t_{n+1}) - (y(t_n) + h b_1 k_1 + h b_2 k_2) &= \mathrm{e}^{t_{n+1}} - \mathrm{e}^{t_n} - \mathrm{e}^{t_n}(h b_1 + h b_2 + h^2 b_2 c_2)\\
&= \mathrm{e}^{t_n}(\mathrm{e}^h - 1 - h(b_1 + b_2) - h^2(b_2 c_2))\\
&= \mathrm{e}^{t_n}\left(h(1 - b_1 - b_2) + h^2(1/2 - b_2 c_2) + \frac{h^3}{6} + \mathcal{O}\big(h^4\big)\right).
\end{aligned}$$

[1]Corrections and suggestions to these notes should be emailed to `h.fawzi@damtp.cam.ac.uk`.

We see that there is no choice of b_1, b_2, c_2, c_2 that will make the term h^3 vanish, and so the method cannot have order ≥ 3.

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.*

4.4 Stiff equations

Consider the linear scalar system

$$\begin{cases} y' = \lambda y \\ y(0) = 1 \end{cases}$$

where $\lambda < 0$. The solution is $y(t) = \mathrm{e}^{\lambda t}$ which decays to 0 as $t \to \infty$. If we solve our ODE using a numerical method, we would like our sequence (y_n) to also decay to zero. For example with Euler's method we get $y_{n+1} = y_n + h\lambda y_n = (1 + h\lambda)y_n$ whose solution is $y_n = (1 + h\lambda)^n$. Thus the sequence y_n converges to 0 as $n \to \infty$ provided that $|1 + h\lambda| < 1$, i.e., $h < 2/|\lambda|$. For large λ this can be a severe restriction on h: for example for $\lambda = -1000$ this implies $h < 2/1000 = 0.002$.

Consider now the implicit Euler method. Here we have $y_{n+1} = y_n + h\lambda y_{n+1}$ which gives $y_{n+1} = (1 - h\lambda)^{-1} y_n$ and so $y_n = (1 - h\lambda)^{-n}$ which converges to 0 for any choice of $h > 0$ (we assumed $\lambda < 0$)!

Definition Suppose that a numerical method, applied to $y' = \lambda y$, $y(0) = 1$, with constant h, produces the solution sequence $\{y_n\}_{n \in \mathbb{Z}^+}$. We call the set

$$\mathcal{D} = \{h\lambda \in \mathbb{C} : \lim_{n \to \infty} y_n = 0\}$$

the *linear stability domain* of the method. Noting that the set of $\lambda \in \mathbb{C}$ for which $y(t) \overset{t \to \infty}{\longrightarrow} 0$ is the left half-plane $\mathbb{C}^- = \{z \in \mathbb{C} : \operatorname{Re} z < 0\}$, we say that the method is *A-stable* if $\mathbb{C}^- \subseteq \mathcal{D}$.

Example We have already seen that for the explicit Euler's method $y_n \to 0$ iff $|1 + h\lambda| < 1$, therefore $\mathcal{D} = \{z \in \mathbb{C} : |1 + z| < 1\}$ and the explicit Euler method is not A-stable. Moreover, solving $y' = \lambda y$ with the implicit Euler method we have seen that $y_n \to 0$ iff $|1 - h\lambda|^{-1} < 1$, therefore the linear stability domain is $\mathcal{D} = \{z \in \mathbb{C} : |1 - z| > 1\}$, hence the implicit Euler method is A-stable.