Runge-Kutta Methods

$u'(t) = f(u,t)$

"Stages" $1 \le i \le s$

$$Y_i = U^n + k\sum_{j=1}^{s} a_{ij} f(Y_j, t_n + kc_j)$$

$$U^{n+1} = U^n + k\sum_{j=1}^{s} b_j f(Y_j, t_n + kc_j)$$

Butcher Tableau

$$\begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1s} \\ \vdots & \vdots & & \\ c_s & a_{s1} & \cdots & a_{ss} \\ \hline & b_1 & \cdots & b_s \end{array} \qquad \begin{array}{c|c} c & A \\ \hline & b^T \end{array}$$

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array} \qquad \tau = O(k^2)$$

$$Y_1 = U^n$$

$$Y_2 = U^n + kf(Y_1, t_n)$$

$$U^{n+1} = U^n + \frac{k}{2}\left(f(Y_1, t_n) + f(Y_2, t_n + k)\right)$$

RK4

$$\begin{array}{c|cccc} 0 & 0 \\ 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1/2 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$$

$$\tau = \mathcal{O}(k^4)$$

$$c_i = \sum_j a_{ij}$$

c_i : "abscissas"

b_j : "weights"

Gauss-Legendre $\gamma = \frac{\sqrt{3}}{6}$

$$\begin{array}{c|cc} \frac{1}{2}-\gamma & \frac{1}{4} & \frac{1}{4}-\gamma \\ \frac{1}{2}+\gamma & \frac{1}{4}+\gamma & \frac{1}{4} \\ \hline & 1/2 & 1/2 \end{array}$$

$$\tau = \mathcal{O}(k^4)$$

Classes of RK methods:

Fully implicit: A is a full matrix
Need to solve $s \times m$ equations at once

Diagonally implicit: A is lower triangular
Need to solve s systems of m equations

Explicit: A is strictly lower triangular
No need to solve eqns.

Apply a RK method to

$$u'(t) = \lambda u$$

$$Y_i = U^n + k \sum_{j=1}^{s} a_{ij} \lambda Y_j \Big|_{1 \le i \le s}$$

$$U^{n+1} = U^n + k \sum_{j=1}^{s} b_j \lambda Y_j$$

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_s \end{bmatrix} \quad \mathbb{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^s$$

$$z = k\lambda$$

$$Y = U^n \mathbb{1} + zAY$$

$$U^{n+1} = U^n + z b^T Y$$

$$(I - zA) Y = U^n \mathbb{1}$$

$$Y = U^n (I - zA)^{-1} \mathbb{1}$$

$$\Rightarrow U^{n+1} = U^n + z b^T (I - zA)^{-1} \mathbb{1} U^n$$

$$U^{n+1} = \underbrace{\left(1 + z b^T (I - zA)^{-1} \mathbb{1}\right)}_{R(z)} U^n$$

In general, $R(z)$ is a rational function.

$$\frac{1}{1-x} = 1 + x + x^2 + \cdots = \sum_{j=0}^{\infty} x^j$$

$$(I - zA)^{-1} = \sum_{j=0}^{\infty} z^j A^j$$

$$R(z) = 1 + \sum_{j=0}^{\infty} z^{j+1} b^T A^j \mathbb{1} = 1 + \sum_{j=1}^{\infty} z^j b^T A^{j-1} \mathbb{1}$$

If A is strictly lower triangular, then $A^s = 0$

So $R(z) = 1 + \sum_{j=0}^{s-1} z^{j+1} b^T A^j \mathbb{1}$ (explicit methods)

For instance, for RK4

$$R(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24}$$

Exact solution of $u'(t) = \lambda u$, $u(0) = 1$

$$u(t) = e^{\lambda t}$$

$$u(t_n + k) = e^{\lambda(t_n + k - t_n)} u(t_n)$$

$$u(t_n + k) = e^{z} u(t_n)$$

Compare with: $U^{n+1} = R(z) U^n$

We want

$$R(z) \approx e^z = \sum_{j=0}^{\infty} \frac{z^j}{j!}$$

This means $b^T A^{j-1} \mathbb{1} = \frac{1}{j!}$ $(*)$ $j = 1, 2, \ldots, p$ for a method of order p

Is $(*)$ sufficient to have $\tau = \mathcal{O}(k^p)$?

No.

For instance, for order 3 we also need

$$\sum_i b_i \Big(\sum_j a_{ij}\Big)^2 = \frac{1}{3}$$

For 4th order, there 3 additional conditions.

$$\underline{u(t_{n+1}) = u(t_n) + k u'(t_n) + \frac{k^2}{2} u''(t_n) + \cdots}$$

Consider $u : \mathbb{R} \to \mathbb{R}^m$

$$u_i'(t) = f_i(u)$$

$$u_i''(t) = \sum_j \frac{\partial f_i}{\partial u_j} \frac{du_j}{dt} = \sum_j \frac{\partial f_i}{\partial u_j} f_j(u)$$

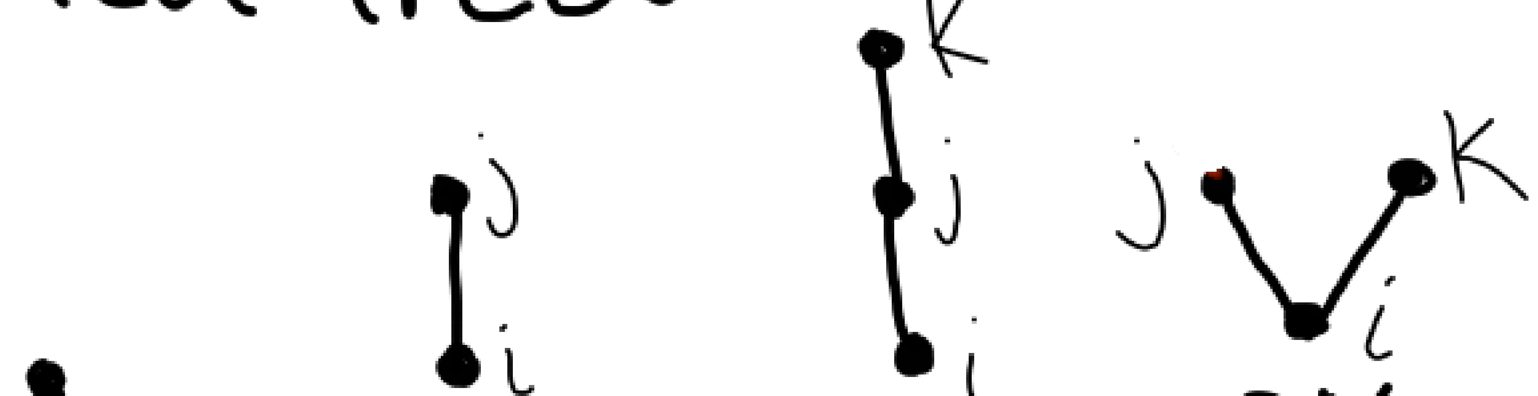
$$= f'f$$

$$u'''(t) = f'f'f + f''(f,f)$$

$$u^{(4)}(t) = f'''(f,f,f) + 3f''(f, f'f) + f'f''(f,f) + f'f'f'f$$

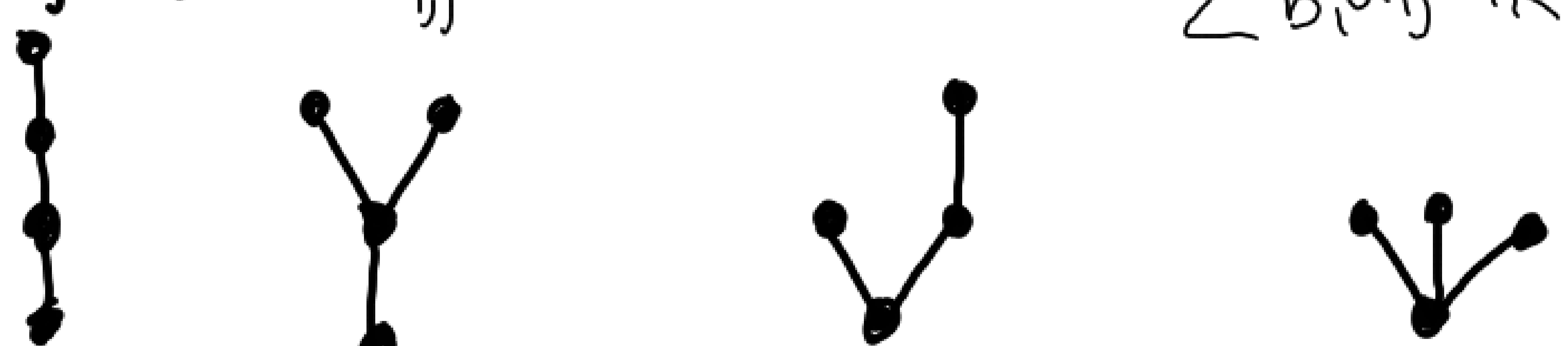
$\vdots$

There is a 1-1 map between these terms and rooted trees:



f $\quad$ $f'f$ $\quad$ $f'f'f$ $\quad$ $f''(f,f)$

$\sum_j b_j = 1$ $\quad$ $\sum_{i,j} b_i a_{ij} = \frac{1}{2}$ $\quad$ $\sum b_i a_{ij} a_{jk} = \frac{1}{6}$ $\quad$ $\sum b_i a_{ij} a_{ik} = \frac{1}{3}$



$f'f'f'f$ $\quad$ $f'f''(f,f)$ $\quad$ $f''(f,f'f)$ $\quad$ $f'''(f,f,f)$

The RK methods form a group under composition.

Error Estimation and Step Size Control

How do we choose a step size?

- Accuracy (we want a small local error)
- Stability (the errors shouldn't be amplified too much)

Focus on this for now

Ideally we would limit the global error, but this is very costly!

In practice we control the local truncation error.

We want to use the largest k such that

$$\tau < \varepsilon \leftarrow \text{error tolerance}$$

τ depends on derivatives, so we should adapt k.

Error estimation

① Richardson extrapolation

a) Take step with size $K \to U_K^{n+1}$

b) Take two steps with size $\frac{K}{2} \to U_{K/2}^{n+1}$

$|U_K^{n+1} - U_{K/2}^{n+1}|$ is an estimate of the local error:

$$U_K^{n+1} = u(t_{n+1}) + \tau_K^n \qquad \tau_K^n = CK^{p+1}$$

$$U_{K/2}^{n+1} = u(t_{n+1}) + \tau_{K/2}^n \qquad \tau_{K/2}^n = C\left(\frac{K}{2}\right)^{p+1}$$

$$U_K^{n+1} - U_{K/2}^{n+1} = C\left(K^{p+1} - \left(\frac{K}{2}\right)^{p+1}\right) = CK^{p+1}\left(1 - \frac{1}{2^{p+1}}\right)$$

Embedded Runge-Kutta Pairs

$$\mathcal{O}(k^p) - \mathcal{O}(k^{p-1}) = \mathcal{O}(k^{p-1})$$

$$Y_i = U^n + k \sum_{j=1}^{s} a_{ij} f(Y_j)$$

$$U^{n+1} = U^n + k \sum_{j=1}^{s} b_j f(Y_j) \qquad \mathcal{L} = \mathcal{O}(k^{p+1})$$

$$\hat{U}^{n+1} = U^n + k \sum_{j=1}^{s} \hat{b}_j f(Y_j) \qquad \mathcal{L} = \mathcal{O}(k^p)$$

$$|U^{n+1} - \hat{U}^{n+1}| = \mathcal{O}(k^p) \rightarrow \text{estimate of one-step error}$$

No extra evaluations of f

Step size adaptation

1. Take a step and estimate the local error by δ
2. if $\delta > \varepsilon$ (tolerance) retake step with smaller k.
3. if $\delta < \varepsilon$ accept the step and continue (possibly with larger k)