$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array}$$

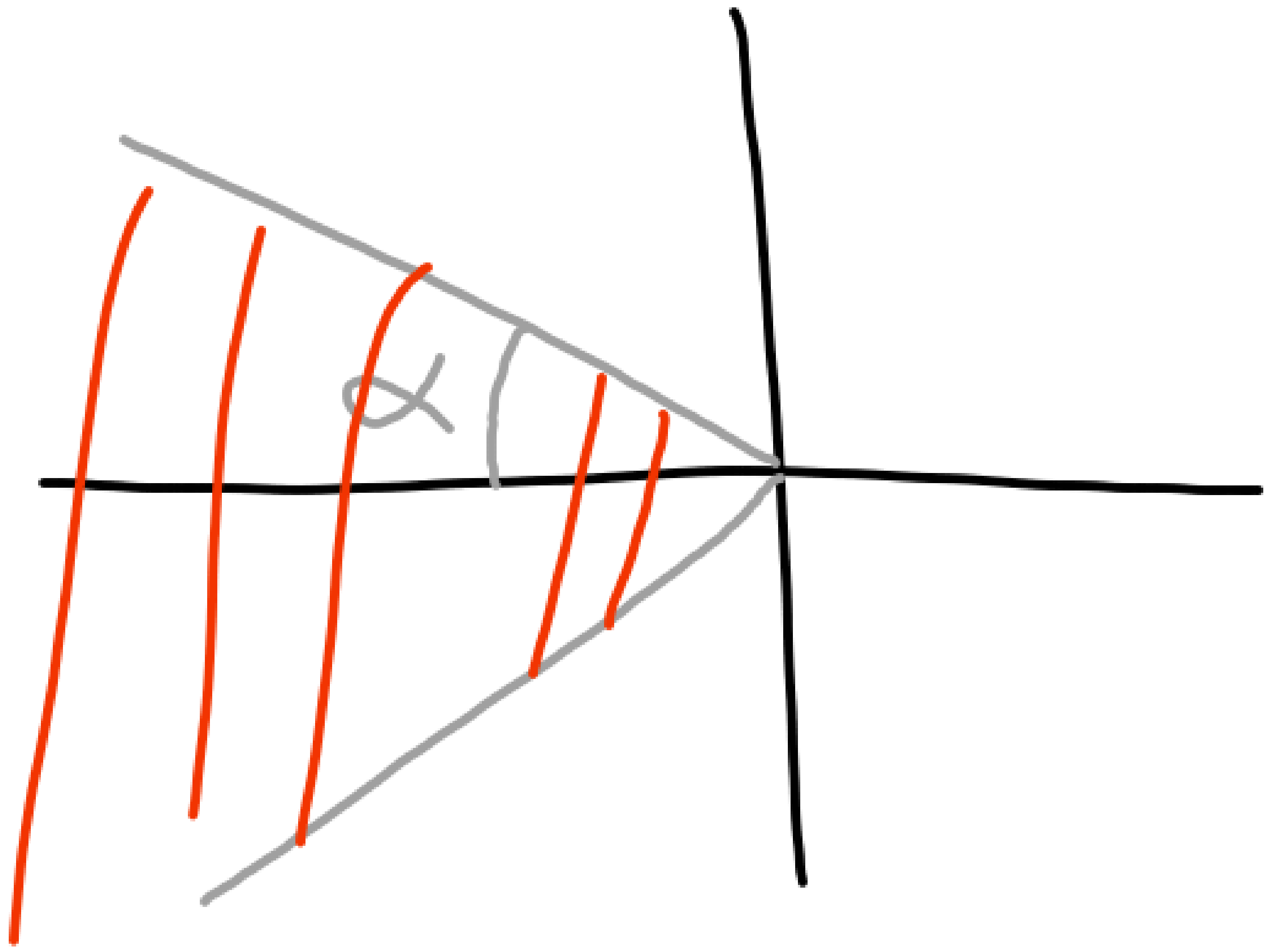
$$\sum_{j=0}^{r} \alpha_j U^{n+j} = K\sum_{j=0}^{r} \beta_j f(U^{n+j})$$

Adams-Bashforth: $U^{n+r} = U^{n+r-1} + K\sum_{j=0}^{r-1} \beta_j f(U^{n+j})$

Adams-Moulton: $U^{n+1} = U^{n+r-1} + K\sum_{j=0}^{r} \beta_j f(U^{n+j})$

Backward difference formula (BDF): $\sum_{j=0}^{r} \alpha_j U^{n+j} = K\beta_r f(U^{n+r})$

$A(\alpha)$-stability



For $\quad U'(t) = \lambda U(t) \quad U(0) = \eta$

any RK method gives

$$U^{n+1} = R(z) U^n \Rightarrow \frac{U^{n+1}}{U^n} = R(z)$$

The exact solution: $\quad u(t) = e^{\lambda t}\eta$

$$\frac{u(t_{n+1})}{u(t_n)} = \frac{e^{\lambda(t_n+k)}\eta}{e^{\lambda t_n}\eta} = e^{\lambda k} = e^{z}$$

We want $R(z) \approx e^{z}$

$$e^z = \sum_{j=0}^{\infty} \frac{z^j}{j!}$$

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

So we need

$$\left.\begin{aligned} \sum_{i=1}^{r} b_i &= b^T \mathbb{1} = 1 \\ b^T A \mathbb{1} &= \frac{1}{2} \\ b^T A^2 \mathbb{1} &= \frac{1}{6} \end{aligned}\right\}$$

Gives 3rd order accuracy for linear ODEs.

$$\sum_{i=1}^{r} b_i c_i^2 = \frac{1}{3}$$

← Required for nonlinear ODEs

Error estimation

We'd like to bound

$$E^n = \|U^n - u(t_n)\| < \varepsilon$$

Instead we typically try to bound the local error:

$$\mathcal{L}^n = k\tau^n$$

We want $\|\mathcal{L}^n\| < \varepsilon$

Richardson estimation

① Compute $U_k^{n+1} \approx u(t_n+k)$ using 1 step of size k

② Compute $U_{\frac{k}{2}}^{n+1} \approx u(t_n+k)$ using 2 steps of size $\frac{k}{2}$

Since $\mathcal{L} = \mathcal{O}(k^{p+1}) \approx Ck^{p+1}$

we have $\mathcal{L}_k = U_k^{n+1} - \underbrace{u(t_n+k)}_{\substack{\text{exact solution}\\ \text{starting from}\\ U^n,\, t_n}} = Ck^{p+1}$

and $\mathcal{L}_{k/2} = U_{\frac{k}{2}}^{n+1} - u(t_n+k) = C\left(\frac{k}{2}\right)^{p+1}$

$$= Ck^{p+1}\frac{1}{2^{p+1}} = \mathcal{L}_k \cdot \frac{1}{2^{p+1}}$$

So $U_k^{n+1} - U_{k/2}^{n+1} =$

$$U^{n+1} - u(t_n+k) - \left(U_{k/2}^{n+1} - u(t_n+k)\right)$$

$$= \mathcal{L}_k - \mathcal{L}_{k/2} = \left(1 - \frac{1}{2^{p+1}}\right)\mathcal{L}_k$$

estimates the one-step error.

Adapting the step size

Given an error estimate

$$\mathcal{L}^n$$

and tolerance ε,

if $\|\mathcal{L}^n\| > \varepsilon$

we go back and redo the step with a smaller K.

If $\|\mathcal{L}^n\| < \varepsilon$, we continue but may choose a smaller K for the next step.

Control theory is used for these algorithms.

Embedded Runge-Kutta Pairs

$$Y_1 = U^n$$

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

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

2nd-order accurate

Let $\hat{U}^{n+1} = Y_2$ be the solution from Euler's method.

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

$$\approx \frac{k^2}{2}\left(\frac{u'(t_n+k) - u'(t_n)}{k}\right)$$

$$\approx \frac{k^2}{2}u''(t_n)$$

One-step error for Euler's method

We got an error estimate "for free".

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \\ & 1 & 0 \end{array}$$

In general we can use 2 RK methods with identical A, c but different weights b:

$$\begin{array}{c|c} c & A \\ \hline & b^T \\ & \hat{b}^T \end{array}$$

The first method has order p
The second has order $p-1$.

Then $U^{n+1} - \hat{U}^{n+1} = \left(U^{n+1} - u(t_n+k)\right) - \left(\hat{U}^{n+1} - \right.$

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

$$\approx O(k^p)$$

$-u(t_n+k))$