Today

methods IVPs

BVP

Predictor-Corrector Methods

Idea: Obtain intermediate result, improve it (with same or different method).

For example:

1. *Predict* with forward Euler: $\tilde{y}_{k+1} = y_k + hf(y_k)$
2. *Correct* with the trapezoidal rule: $y_{k+1} = y_k + \frac{h}{2}(f(y_k) + f(\tilde{y}_{k+1}))$.

This is called Heun's method.

Runge-Kutta/'Single-step'/'Multi-Stage' Methods

Idea: Compute intermediate 'stage values':

$0 \le c_i \le 1$

y_k $\quad$ y_{k+1}

t_k $\quad$ $t_{k+1} = t_k + h$

$y' = f(t, y)$

stage values $\rightarrow$

$$\begin{aligned}
r_1 &= f(t_k + c_1 h, y_k + (a_{11} \cdot r_1 + \cdots + a_{1s} \cdot r_s) h) \\
\vdots & \qquad \vdots \\
r_s &= f(t_k + c_s h, y_k + (a_{s1} \cdot r_1 + \cdots + a_{ss} \cdot r_s) h)
\end{aligned}$$

Then compute the new state from those:

$$y_{k+1} = y_k + (b_1 \cdot r_1 + \cdots + b_s \cdot r_s) h$$

$y' = \lambda y$

$y' = Ay$

$\hookrightarrow w' = [\ddots] w$

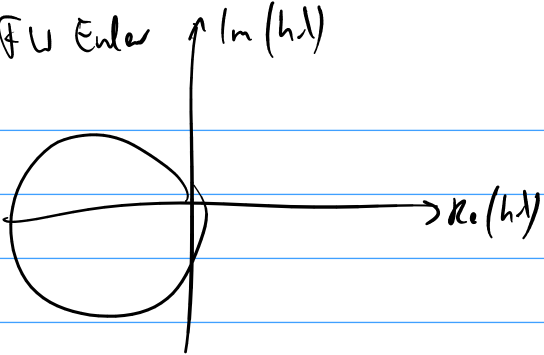
Can summarize in a *Butcher tableau*:

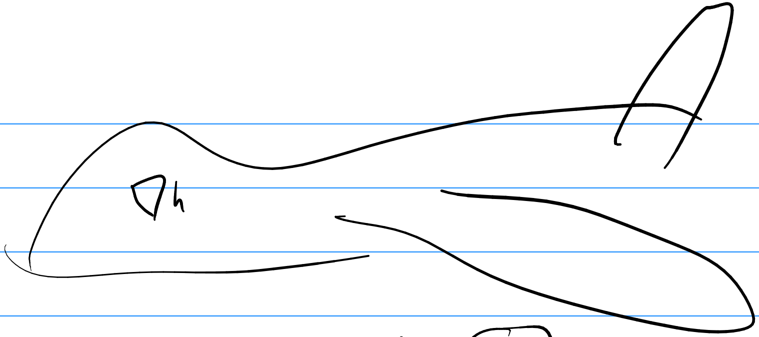
$$\begin{array}{c|ccc}
c_1 & a_{11} & \cdots & a_{1s} \\
\vdots & \vdots & & \vdots \\
c_s & a_{s1} & \cdots & a_{ss} \\
\hline
 & b_1 & \cdots & b_s
\end{array}$$

$$y_{k+1} = y_k + \int_{t_k}^{t_{k+1}} f(y(\tau))\, d\tau$$

$\hookrightarrow$ order conditions

FW Euler





$\mathrm{Error}(h) = O(h) = 0.1$

~~100x bigger~~ $\rightarrow \mathrm{Error}\left(\frac{h}{10}\right) = 0.01$

$\mathrm{Error}(h) = O(h^5) = 0.1$

Runge-Kutta: Properties

$$\begin{array}{c|ccc} c & a_{11} & & a_{1n} \\ \vdots & & & \\ & a_{n1} & \cdots & a_{nn} \\ \hline & b_1 & \cdots & b_n \end{array}$$

When is an RK method explicit?

$\Longleftrightarrow$

$$\begin{array}{c|ccc} c & a_{11} & & a_{1n} \\ \vdots & & & \\ & a_{n1} & \cdots & a_{nn} \\ \hline & b_1 & \cdots & b_n \end{array}$$

only those coeff. nonzero

When is it implicit?

otherwise

When is it *diagonally implicit*? (And what does that mean?)

can solve one at a time

$$\begin{array}{c|ccc} c & a_{11} & & a_{1n} \\ \vdots & & & \\ & a_{n1} & \cdots & a_{nn} \\ \hline & b_1 & \cdots & b_n \end{array}$$

nonzero

ESDIRK

Heun and Butcher

Stuff Heun's method into a Butcher tableau:

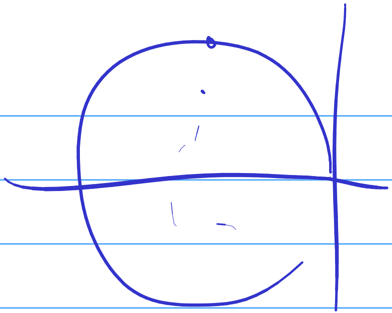
1. $\tilde{y}_{k+1} = y_k + hf(y_k)$ v_0
2. $y_{k+1} = y_k + \frac{h}{2}(f(y_k) + f(\tilde{y}_{k+1}))$. v_1

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

What is RK4?

Demo: Dissipation in Runge-Kutta Methods

$$y' = \begin{pmatrix} & 1 \\ -1 & \end{pmatrix} y$$



Multi-step/Single-stage/Adams Methods/Backward Differencing Formulas (BDFs)

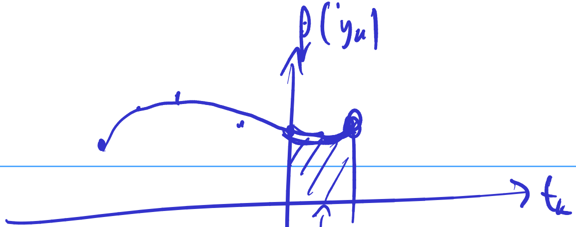
Idea: Instead of computing stage values, use *history* (of either values of f or y–or both):

$$y_{k+1} = \sum_{i=1}^{M} \alpha_i y_{k+1-i} + h \sum_{i=1}^{N} \beta_i f(y_{k+1-i})$$

Extensions to implicit possible.

Method relies on existence of history. What if there isn't any? (Such as at the start of time integration?)

use RK(n) until you have history.



$$\int_{t_k}^{t_{k+1}} f(y)\, d\tau = y_{k+1} - y_k$$

Stability Regions

Why does the idea of stability regions still apply to more complex time integrators (e.g. RK?)

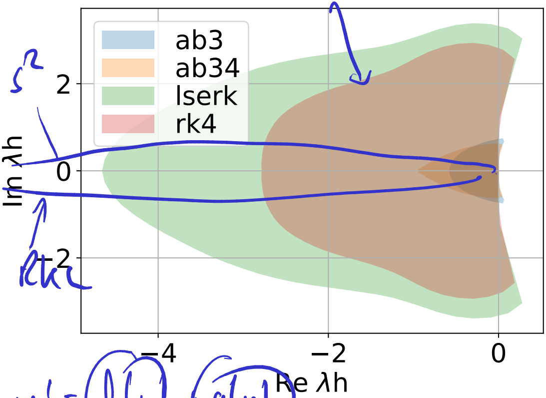
because diagonalizing the ODE also diagonalizes the time stepper

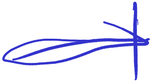
Demo: Stability regions

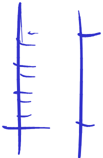
More Advanced Methods

Discuss:

- What is a good cost metric for time integrators?
- AB3 vs RK4
- Runge-Kutta-Chebyshev
- LSERK and AB34
- IMEX and multi-rate
- Parallel-in-time ("Parareal")







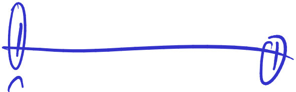
$$y' = \underbrace{f(y)}_{ex} + \underbrace{g(y)}_{im}$$

In-Class Activity: Initial Value Problems

In-class activity: Initial Value Problems

Outline

BVP Problem Setup: Second Order



Example: Second-order linear ODE

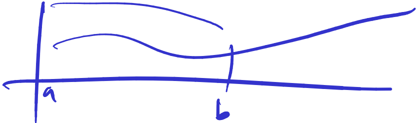
$$u''(x) + p(x)u'(x) + q(x)u(x) = r(x)$$

with *boundary conditions ('BCs')* at a:

- *Dirichlet* $u(a) = u_a$
- or *Neumann* $u'(a) = v_a$
- or *Robin* $\alpha u(a) + \beta u'(a) = w_a$

and the same choices for the BC at b.

Note: BVPs in time are rare in applications, hence x (not t) is typically used for the independent variable.



$$y'' = f(y, y')$$

$$y(a) = \cdot$$

$$y'(a) = \cdot$$

$$\cancel{y(b)} = \cdot$$

BVP Problem Setup: General Case

$y' = f(y)$
$y(a) = 15$
$y(b) = 17$

ODE:

$$\mathbf{y}'(x) = \mathbf{f}(\mathbf{y}(x)) \quad \mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$$

BCs:

$$\mathbf{g}(\mathbf{y}(a), \mathbf{y}(b)) = \mathbf{0} \quad \mathbf{g} : \mathbb{R}^{2n} \to \mathbb{R}^n$$

(Recall the rewriting procedure to first-order for any-order ODEs.)

Does a first-order, scalar BVP make sense?

no – that's an IVP
(already solvable w/o other cond.)

Example: Linear BCs

$$g(y(a), y(b)) = B_a\mathbf{y}(a) + B_b\mathbf{y}(b) - \mathbf{c} = 0$$

Is this Dirichlet/Neumann/...?

? can't know
derivatives become other variables

Does a solution even exist? How sensitive are they?

General case is harder than root finding, and we couldn't say much there.
$\rightarrow$ Only consider linear BVP.

$$(*) \begin{cases} \mathbf{y}'(x) = A(x)\mathbf{y}(x) + \mathbf{b}(x) \\ B_a\mathbf{y}(a) + B_b\mathbf{y}(b) = \mathbf{c} \end{cases}$$

To solve that, consider *homogeneous IVP*

$$\mathbf{y}_i'(x) = A(x)\mathbf{y}_i(x)$$

with initial condition

$$\mathbf{y}_i(a) = \mathbf{e}_i.$$

Note: $\mathbf{y} \neq \mathbf{y_i}$. $\mathbf{e}_i$ is the ith unit vector. With that, build the fundamental solution matrix

$$Y(x) = \begin{bmatrix} | & & | \\ \mathbf{y}_1 & \cdots & \mathbf{y}_n \\ | & & | \end{bmatrix}$$