Numerics

$x_n = x(nh)$, $h = \Delta t$.

- Ideally,

$$x_{n+1} = x_n + f(x_n)\cdot h + \frac{1}{2} f'(x_n) h^2 + \frac{1}{6} f''(x_n) h^3$$

$$+ \ldots + \frac{1}{p!} f^{(p)}(x_n) h^p + O(h^{p+1})$$

$\Rightarrow$ $(p+1)$-th order scheme

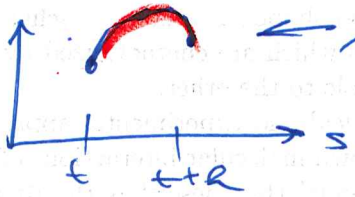
But:
We do not know $f^{(i)}$ in general.

- Get inspiration from numerical integration of definite integrals

$$\int_t^{t+h} f(\xi)\, d\xi = h\left[\frac{1}{6} f(t) + \frac{2}{3} f\left(t + \frac{h}{2}\right) + \frac{1}{6} f(t+h)\right]$$

$\equiv$ Simpson rule $+ O(h^5)$

$\equiv$ Keplersche Fassregel.



← integrate f with quadratic polyn.

Proof:
Plug in Taylor expansion and see how terms cancel.

The classical Runge-Kutta method implements a 'Simpson's rule' for ODEs.

① $k_1 = f(x_n)$

② $x_{n+1/2}^{\cdot} = x_n + \frac{h}{2} k_1$

$\Rightarrow k_2 = f(x_{n+1/2}^{\cdot})$

③ $x_{n+1/2}^{\cdot\cdot} = x_n + \frac{h}{2} k_2$

$\Rightarrow k_3 = f(x_{n+1/2}^{\cdot\cdot})$

④ $x_{n+1}^{\cdot} = x_n + h\, k_3.$

$\Rightarrow$

$$\left| x_{n+1} = x_n + h \cdot \left(\frac{1}{6} k_1 + \frac{1}{3} k_2 + \frac{1}{3} k_3 + \frac{1}{6} k_4 \right) + O(h^5) \right|$$

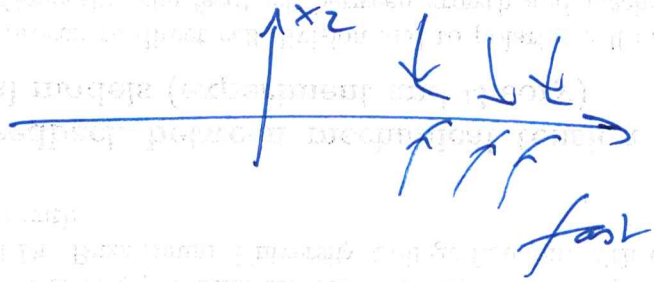
Bells & Whistles.

- go up to order 8th if you want
- Nest 4th and 5th order Runge-Kutta ⇒ error estimate
- use error estimate for adaptive time step control.
- Straightforward generalization to
  - non-holonomic case $f(t,x)$
  - dynamics on $\mathbb{R}^n$: $f(x)$.

Stiff ODEs

example: $\dot{x}_1 = 1$

$\dot{x}_2 = -\lambda x_2, \quad \lambda >> 1$



fast transients.

Pragmatic definition of stiffness

- Stability requirements, not accuracy dictate step size
- Some components of the solution decay much faster than others.

Math. Definition for linear systems

$\dot{\underline{x}} = \underline{\underline{J}}\,\underline{x}, \qquad eig(\underline{\underline{J}}) = \{\lambda_1, \dots, \lambda_n\}$.

Stiff $\Leftrightarrow$ $\dfrac{\max |\lambda_i|}{\min |\lambda_i|} >> 1$.

explicit (e.g. Euler)

$$\underline{x}_{n+1} = \underline{x}_n + f(\underline{x}_n)\, h$$

- mathematically correct for $h \to 0$.
- prone to numerical instability (blow up, oscillations)
- higher order methods: only partial remedy.

implicit schemes

$$\underline{x}_{n+1} = \underline{x}_n + f(\underline{x}_{n+1})\cdot h$$

- more costly: must solve fixed point eqn.
- usually stable.

Exercise: Google 'matlab ode45 ode15s'

A note on integrating PDEs.

Example (Diffusion equation)

$c(x,t) \equiv$ concentration field.

$$\frac{d}{dt} c(x,t) = D \frac{\partial^2}{\partial x^2} c(x,t).$$

Known solution:

$$c(x,0) = M_0 \, \delta(x) \quad \left[\frac{1}{m}\right] \quad \text{(mole)}$$

$$c(x,t) = \frac{1}{\sqrt{2\pi D t}} \exp - \frac{x^2}{2\pi D t}. \qquad \left[\frac{\text{mole}}{m}\right].$$

discretize space and time

$t_n = n \Delta t$.

$x_n = n \Delta x$ ← bins

$c_n = c(x_n, t) \Rightarrow c_n dx \equiv$ mass in nth bin.

$$c_n \leftarrow c_n + D \cdot \frac{c_{n+1} + c_{n-1} - 2c_n}{dx^2} dt.$$

Courant criterion:

Courant upwind scheme

$$\frac{d}{dt} c(x,t) = -\frac{d}{dx}\left[v(x)\, c(x,t)\right].$$

- right derivative: $\frac{d}{dx} f(x) \approx \frac{f(x_{n+1}) - f(x_n)}{dx}$
- left derivative: $\frac{d}{dx} f(x) \approx \frac{f(x_n) - f(x_{n-1})}{dx}$.

Q: which?

A: choose for each bin, depending on direction of v.

- $V(x_n) > 0$

$V_n = v(x_n)$
$c_n = c(x_n)$.

$$c_n \leftarrow c_n - \frac{c_n V_n}{dx} dt.$$

$$c_{n+1} \leftarrow c_{n+1} + \frac{c_n V_n}{dx} dt.$$

- $V(x_n) < 0$.

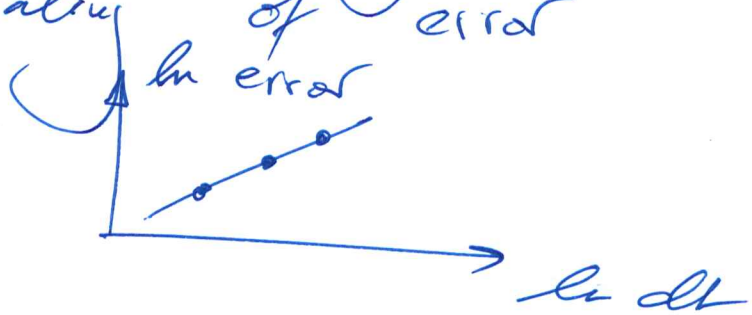
$$c_n \leftarrow c_n - \frac{c_n |V_n|}{dx} dt.$$

$$c_{n-1} \leftarrow c_{n-1} + \frac{c_n |V_n|}{dx} dt.$$

Comment:

- Speed up:
  parallelize code, e.g.
  use matrix operations,
  not loops.

- Check code.
  - mass conservation
  - symmetries.
  - special cases with known analytic solutions
  - scaling of error



- Modern developments:
  - particle methods
  - adaptive meshes
  - fast BEM.