3.11.2023

1) Difference between Approximation Error and Interpolation Error ?

Interpolation Error refers specifically to interpolation

Approximation Error is "general"

$\underline{Ex}$ EW $\lambda \approx \lambda_n$ $\qquad |\lambda - \lambda_n|$

Ex $f \in C^1[0,1]$

$f: [0,1] \to \mathbb{R}$, f cont. differentiable

$f_n \approx f$ $\qquad \|f - f_n\|_{\infty \atop 2}$

given function values

$y_0 = f(t_0)$, $y_1 = f(t_1)$, $y_2 = f(t_2)$, ..., $y_n = f(t_n)$

construct Approximation $f_n(t)$ such that

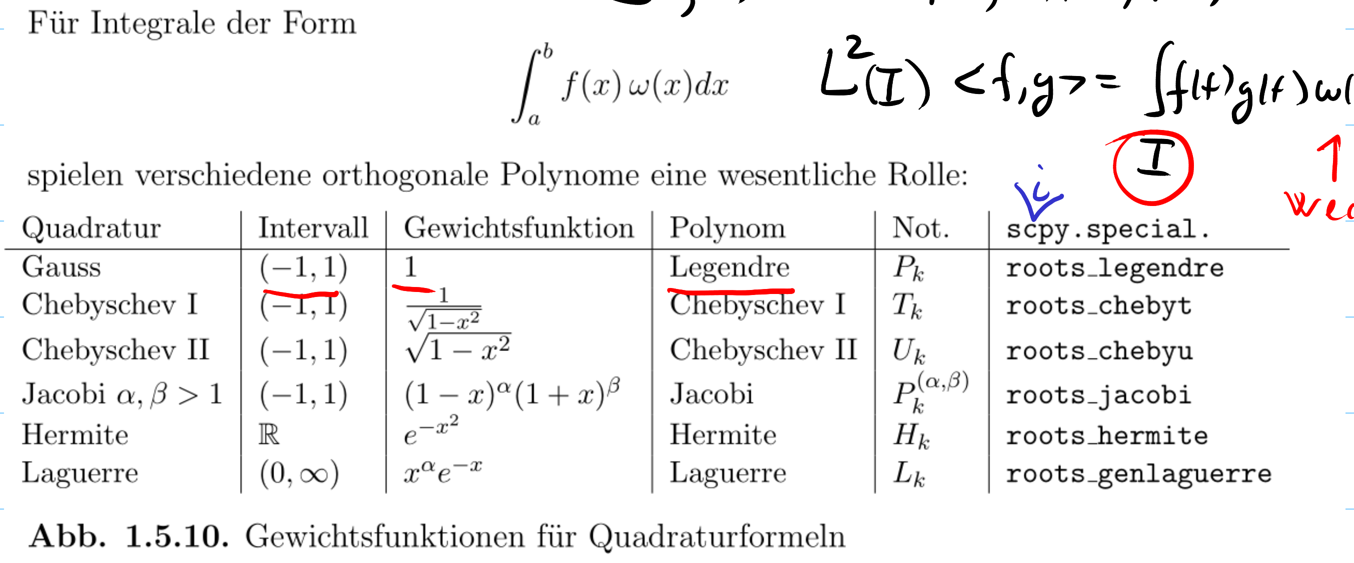
$f_n(t_j) = y_j$ for $j = 0, 1, \dots, n$

$\|f - f_n\|_{\infty \atop 2}$

2) Legendre Polynomials

Gram-Schmidt produces orthogonal elements of a lin. space with scalar product

$\langle \cdot, \cdot \rangle \rightsquigarrow p_0, p_1, \dots, p_n, \dots$



Für Integrale der Form

$$\int_a^b f(x)\,\omega(x)dx$$

$L^2(I)$ $\langle f, g \rangle = \int f(t)g(t)w(t)dt$

spielen verschiedene orthogonale Polynome eine wesentliche Rolle:

i $\quad$ I $\quad$ weight

Quadratur	Intervall	Gewichtsfunktion	Polynom	Not.	scpy.special.
Gauss	$(-1,1)$	1	Legendre	P_k	roots_legendre
Chebyschev I	$(-1,1)$	$\frac{1}{\sqrt{1-x^2}}$	Chebyschev I	T_k	roots_chebyt
Chebyschev II	$(-1,1)$	$\sqrt{1-x^2}$	Chebyschev II	U_k	roots_chebyu
Jacobi $\alpha, \beta > 1$	$(-1,1)$	$(1-x)^\alpha(1+x)^\beta$	Jacobi	$P_k^{(\alpha,\beta)}$	roots_jacobi
Hermite	$\mathbb{R}$	e^{-x^2}	Hermite	H_k	roots_hermite
Laguerre	$(0,\infty)$	$x^\alpha e^{-x}$	Laguerre	L_k	roots_genlaguerre

Abb. 1.5.10. Gewichtsfunktionen für Quadraturformeln

3) Chebyshev (I-kind)

$T_0(x) = 1, T_1(x) = x, \quad T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$

$T_n(x) = \cos(n \arccos x) \quad , x \in [-1,1]$

Polynomials of degree n

zeros: Chebyshev-nodes for $[a,b]$ of $T_{n+1}(x)$ are:

$$x_k = a + \frac{1}{2}(b-a)\left(\cos\left(\frac{2k+1}{2(n+1)}\pi\right) + 1\right) \quad k = 0,1,\dots,n$$

optimal points for Interpolation

$\| \cdot \|_\infty$

extrema of Chebyshev-Polynomials of I-kind T_n:

$(\pm 1)^n$ achieved in the Chebyshev-alternates abscisa

$$x_k = a + \frac{1}{2}(b-a)\left(\cos\left(\frac{k}{n}\pi\right) + 1\right) \quad k = 0,1,\dots,n$$

if we do not want $a, b \Rightarrow x_1, \dots, x_{n-1}$

4) change of variable $\Rightarrow$

Chebyshev interpolation / approximatio | quadratura

$\updownarrow$

Fourier interpolation / approximatio | quadratura.

$\rightarrow$ exploits fast convergence which of the weight

One can use Chebyshev-Nodes

"math. correct", a bit cumbersome.

Chebyshev-Alternates

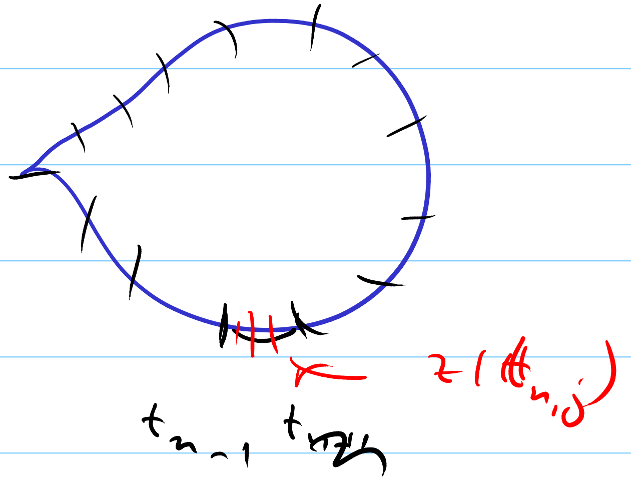
math. correct (asympt.), better in practice.

5)

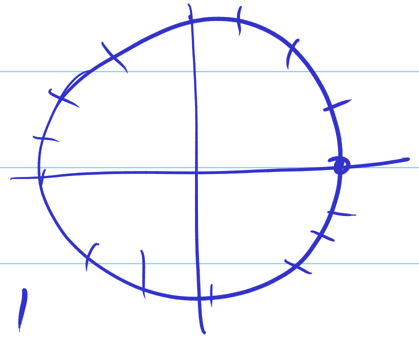
$$\int_\Gamma f(z)dz = \int_\partial f(z(t))\,\dot z(t)\,dt \approx$$

$$= \sum_{n=1}^{N} \int_{t_{n-1}}^{t_n} f(z(t))\,\dot z(t)\,dt$$

$$\approx \sum_{n=1}^{N} \sum_{j=1}^{d} f(z(t_{n,j}))\,\dot z(t_{n,j}) \cdot w_{n,j}$$



Circle. $z(t) = e^{it}$



$\Rightarrow$ Fourier type!

6) $\varepsilon_n = c n^{-p}$

$$\frac{\varepsilon_n}{\varepsilon_m} = \frac{n^{-p}}{m^{-p}} = \left(\frac{n}{m}\right)^{-p} \le \frac{1}{2}$$

$-p(\log n - \log m) \le -\log 2$

$\varepsilon_m \le \frac{1}{2}\varepsilon_n$ $\qquad \frac{\varepsilon_n}{\varepsilon_n} = \frac{1}{2}$

$\log n - \log m \geq \frac{\log 2}{p}$

$\log m \le \log n + \frac{\log 2}{p}$

$\log m \le \log(n 2^{p})$

$m \le n \cdot 2^{+p}$

$p=1$: $\quad m \le \frac{n}{2} \qquad 2n$

$p=2$ $\quad m \le \frac{n}{4} \qquad 4n$

10.11.2023

① Case $\varepsilon_i = c \cdot n_i^{-p}$

$\varepsilon_{i+1} = c \cdot n_{i+1}^{p}$

$\Rightarrow \frac{n_i^{-p}}{n_{i+1}^{-p}} = \frac{\varepsilon_i}{\varepsilon_{i+1}} = 2$

$\varepsilon_{i+1} = \frac{1}{2}\varepsilon_i \Leftrightarrow \frac{\varepsilon_i}{\varepsilon_{i+1}} = 2$

$\frac{n_i}{n_{i+1}} = 2^{-\frac{1}{p}} \Leftrightarrow \boxed{n_{i+1} = 2^{\frac{1}{p}} n_i}$

linear convergence $p=1$ $n_{i+1} = 2 n_i$

quadratic converg $p=2$ $n_{i+1} = 2^{\frac{1}{2}} n_i = \sqrt{2}\, n_i$

cubic converg. $p=3$ $n_{i+1} = 2^{\frac{1}{3}} n_i = \sqrt[3]{2}\, n_i$

Case

$$\left.\begin{array}{l} \varepsilon_i = c \cdot e^{-\beta n_i} \\ \varepsilon_{i+1} = c\, e^{-\beta n_{i+1}} \end{array}\right\} \Rightarrow 2 = \frac{\varepsilon_i}{\varepsilon_{i+1}} = \frac{e^{-\beta n_i}}{e^{-\beta n_{i+1}}}$$

$$2 = e^{\beta(n_{i+1} - n_i)}$$

$$\ln 2 = \beta(n_{i+1} - n_i) \Rightarrow \frac{1}{\beta}\ln 2 = n_{i+1} - n_i$$

$$n_{i+1} = n_i + \underline{\frac{1}{\beta}\ln 2}$$

② $\varepsilon_{k+1} \simeq C\, \varepsilon_k^{p}$ ← Order $p>1$ convergence

$\shortparallel$ ← Error in step k.

Error in step $k+1$

③ optical illusion 🙂

④ We expect perfect results (i.e. error at machine precision) in case of the global Gauss quadrature and P_{12}

$\Rightarrow$ rule C for Plot #3

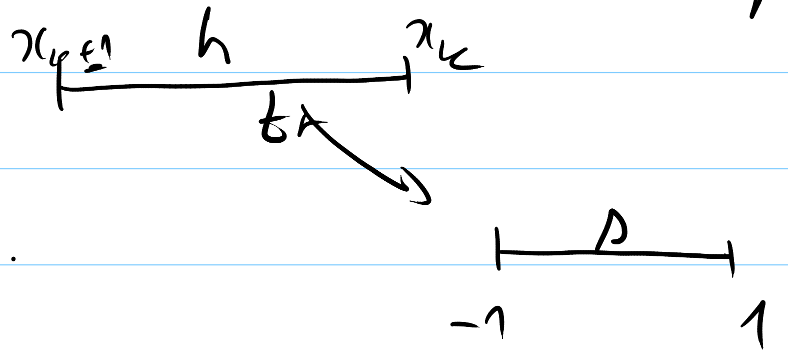
red = P_{12} f_C

blue = $C^\infty(I)$ fuction f_A

black = $C^0(I) \setminus C^1(I)$ f_B

Plot 1: convergen of order 4 $\Leftarrow$ comp. 2-point Gauss

Plot 2: 2 $\Leftarrow$ comp trapezoidal rule



5

$$s(t) = \frac{t - x_k}{x_k - x_{k-1}} + \frac{t - x_{k-1}}{x_k - x_{k-1}} =$$

$$\int_{x_{k-1}}^{x_k} f(t)\,dt =$$

$$s(t) = \frac{t - x_{k-1}}{h} + \frac{t - x_k}{h}$$

$$hs = 2t - x_{k-1} - x_k$$

$$= \frac{h}{2}\int_{-1}^{1} f\left(\frac{h}{2}s + \frac{x_k + x_{k-1}}{2}\right)ds \qquad t = \frac{hs}{2} + \frac{x_k + x_{k-1}}{2}$$

$$dt = \frac{h}{2}\,ds$$

$$\approx \frac{h}{2}\left[f\left(-\frac{h}{2}\frac{1}{\sqrt{3}} + \underbrace{\frac{x_{k-1}+x_k}{2}}_{m_k}\right) + f\left(\frac{h}{2}\frac{1}{\sqrt{3}} + \frac{x_{k-1}+x_k}{2}\right)\right]$$

$$EST_k = \frac{h}{2}\left[f\left(m_k + \frac{h}{2\sqrt{3}}\right) + f\left(m_k - \frac{h}{2\sqrt{3}}\right)\right] - h\,f(m_k)$$

Cost of EST_k: 3 function evaluations ☹

Note: as the Gauss points are not nested when dividing the interval we cannot reuse the information, i.e. function values at that points!

Φ : $x_{k+1} = \Phi(x_k)$ $x_k \to x$ Fixpoint of Φ

$$s_k = \frac{1}{k+1}\sum_{j=0}^{k} x_j \qquad \underline{s_{k+1} = \Psi(k, s_k)}$$

?

$$s_{k+1} = \frac{1}{k+2}\sum_{j=0}^{k+1} x_j = \frac{k+1}{k+2}\,\frac{1}{k+1}\left(\sum_{j=0}^{k} x_j + x_{k+1}\right) =$$

$$= \frac{k+1}{k+2}\left(s_k + \frac{1}{k+1} x_{k+1}\right) = \frac{k+1}{k+2} s_k + \frac{1}{k+2} x_{k+1}$$

$$\Psi(k, s_k) = \frac{k+1}{k+2} \underset{\to s}{s_k} + \underbrace{\frac{1}{k+2}}_{0} \underset{\substack{\downarrow \\ x}}{\Phi(x_k)} \to s \Rightarrow$$ a fixed point iteration!

Suppose $s_k \to s$ for $k \to \infty$

17.11.2023

Solving Algebraic Nonlinear Equations

1) Break down in Newton / bisection

may happen if ⊕ dir. points in the wrong direction

⊕ correction is too large

↑ ⊕ $\underline{D}f(x_k)$ singular

too far away from 0

2) 8.4.1.4. Bisection

$f(1)<0$, $f(2)>0$, f continuous, rel. error $<10^{-6}$

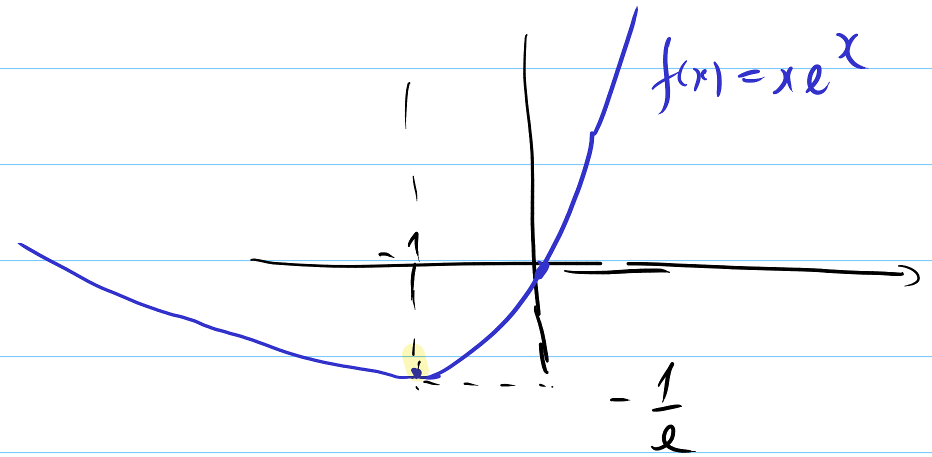
according to 8.4.1.1. bisection is linear convergent

$$\text{Nr. steps} \geq \log_2 \frac{|b-a|}{tol} = \log_2 \frac{1}{10^{-6}} = 6 \log_2 10 \approx 19.93$$

$\Rightarrow$ need 20 steps!

3) 8.4.2.16B Lambert-W-function.

$W(x)\, e^{W(x)} = x \Leftrightarrow W$ inverse of $f(x) = x e^{x}$



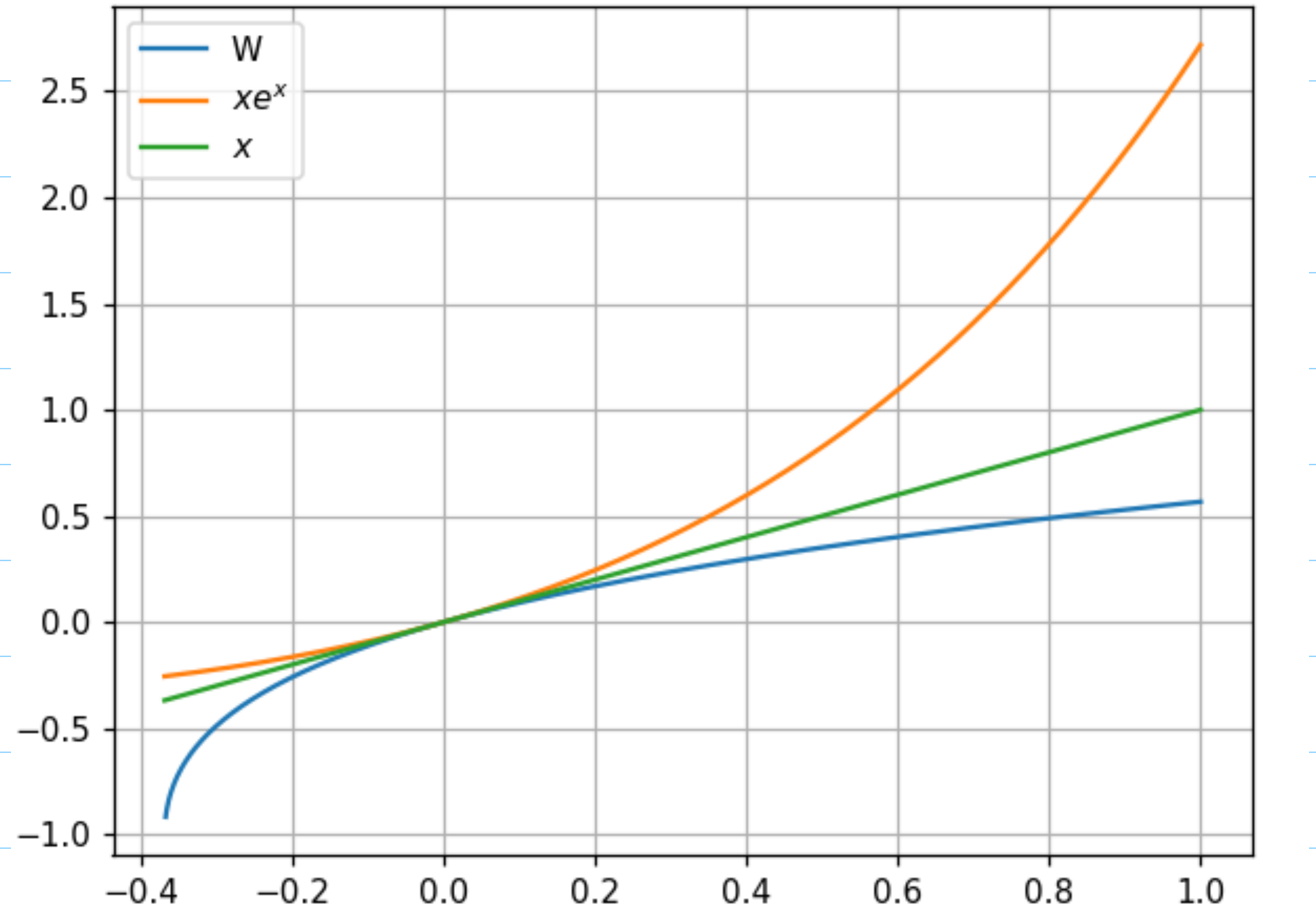
Inverse: $x \geq -\frac{1}{e}$ $W: [-\frac{1}{e}, \infty] \to \mathbb{R}$

For given x:

solve $F(w)=0$, where $F(w) = we^{w} - x$

$DF(w) = (1+w)e^{w}$

apply Newton!



4) 8.4.2.33

$$\ldots \qquad e_{k+1} = c \cdot e_k \cdot e_{k-1}$$

assume $e_k = M e_{k-1}^{p}$; $e_{k+1} = M e_k^{p} = M\left(M e_{k-1}^{p}\right)^{p} = M^{p+1} e_{k-1}^{p^2}$

$$e_{k+1} = c M e_{k-1}^{p} e_{k-1} = cM e_{k-1}^{p+1}$$

$\Big\} \Rightarrow$

$$\Rightarrow \quad cM e_{k-1}^{p+1} = e_{k+1} = M^{p+1} e_{k-1}^{p^2}$$

$$\Rightarrow \quad \underbrace{cM^{-p}}_{\text{does not depend on}} = \underbrace{e_{k-1}}_{\text{depends on } k}^{p^2-p-1} \Rightarrow p^2-p-1=0$$

$$\left[\text{or} \quad \underbrace{\log cM^{-p}}_{\text{constant}} = \underbrace{\overset{\overset{0}{\shortparallel}}{(p^2-p-1)} \; \log \overset{}{\underset{\downarrow\, 0}{e_{k-1}}}}_{\downarrow\, -\infty}\right]$$

5) 8.4.2.39 inverse iteration?

$$\begin{tikzcd} x^* \arrow[r, bend left, "F"] & 0 \arrow[l, bend left, "F^{-1}"] \end{tikzcd}$$

find x^* s.t. $F(x^*)=0 \iff$ compute $x^* = F^{-1}(0)$

$\approx$

$P(0)$

modell function that approximates $F^{-1}(x)$

e.g. take $p(x)$ = Polynomial of degree $\leq m-1$

decide p e.g. by interpolation on some points

take $x_0, x_1, \ldots, x_{m-1}$

request $p(x_j) = F^{-1}(x_j) \iff P(F(x_j)) = x_j$

for $j = 0, 1, \ldots, m-1$

$\underline{ex}$ $m=2 \Rightarrow$ secant method

$m=3 \Rightarrow$ quadratic inverse interpolation

6) empiric convergence of Newton?

table experiment 8.5.2.1 ☺

7) $\underline{F}(\underline{x})=0$; $\underline{\underline{A}}$ arbitrary : $F(x)=0 \iff \underline{\underline{A}}\,\underline{F}(\underline{x})=0$
invertible

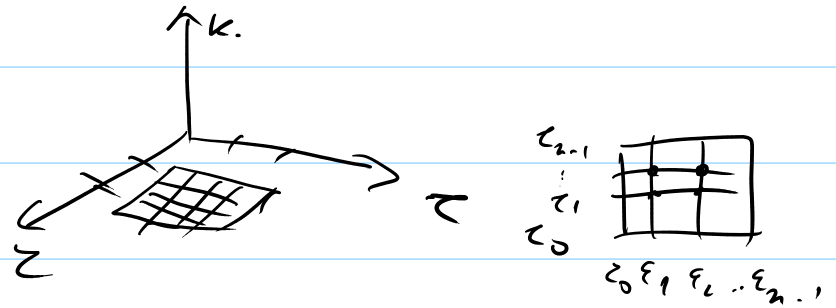
A method is affine invariant if the iterates $(\tilde{x}_k)$
for $\underline{\underline{A}}\,\underline{F}(\underline{x})$ are the same as (x_k)
for $\underline{F}(\underline{x})$

8) 8-2 "–" cosmetics ☺

9) 8-5 meshgrid = a set of $(\varepsilon, \tau) \in [0, c\ldots] \times [10^{-5}, 10^{-1}]$

for each $\varepsilon, \tau)$ compute k

Plot


k
τ
ε
τ_{n-1}
τ_1
τ_0
ε_0 ε_1 ε_2 .. ε_{n-1}

24.11.2023

1) 8-9 d) Newton 1-Step.

$$\underline{x}_{k+1} = \underline{x}_k - \underbrace{\underline{\underline{DF}}(\underline{x}_k)^{-1}}_{\text{never compute this!}} \underline{F}(\underline{x}_k) = \underline{x}_k - \underline{s}$$

NEVER COMUTE THE INVERSE! $\underline{\underline{A}} = \underline{\underline{DF}}(x_k)$

$\underline{s}$ = solution of $\underline{\underline{A}}\,\underline{s} = \underbrace{\underline{F}(\underline{x}_k)}_{\underline{b}}$

$$\underline{\underline{A}}\,\underline{s} = \underline{b}$$

① compute the LU-Decomposition

$$\underline{\underline{P}}\,\underline{\underline{A}} = \underline{\underline{L}}\,\underline{\underline{U}} \leftarrow \text{expensive } O(n^3)$$

$$\underline{\underline{P}}\,\underline{\underline{A}}\,\underline{s} = \underline{\underline{P}}\,\underline{b} \iff \underline{\underline{L}}\,\underbrace{\underline{\underline{U}}\,\underline{s}}_{\underline{y}} = \underline{\underline{P}}\,\underline{b}$$

② $\underline{\underline{L}}\,\underline{y} = \underline{\underline{P}}\,\underline{b} \Rightarrow \underline{y}$ (fast)

③ then. $\underline{\underline{U}}\,\underline{s} = \underline{y} \Rightarrow \underline{s}$ (fast)

Note: simplified Newton: reuse $\underline{\underline{A}} = \underline{\underline{DF}}(x_k)$ for several k

reuse the factors $\underline{\underline{L}}, \underline{\underline{U}}$.

2) 8-10 a) $\underline{\underline{A}}(x)\,\underline{x} = \underline{b}$

at step $k+1$: use $\underline{\underline{A}}(x_k) \Rightarrow$

$$\underline{\underline{A}}(x_k)\,\underline{x}_{k+1} = \underline{b} \iff \underline{x}_{k+1} = \underline{\underline{A}}(x_k)^{-1}\underline{b}$$

ITERATION!

if (x_k) convergent to $\underline{x}^* \Rightarrow \underline{\underline{A}}(x^*)\,\underline{x}^* = \underline{b}$

8-10 d) Newton for $\underline{F}(\underline{x}) = \underline{\underline{A}}(\underline{x})\,\underline{x} - \underline{b}$

$$\underline{\underline{DF}}(x) = \underline{\underline{A}}(\underline{x}) + \underline{x}\,\underline{\underline{D}}\,\underline{\underline{A}}(\underline{x})$$

$$\underline{\underline{A}}(x) = \underline{\underline{B}} + \gamma(x)\,\underline{\underline{I}} \quad \text{with } \underline{\underline{B}} = \begin{bmatrix} 0 & 1 & & 0 \\ 1 & 0 & 1 & \\ & 1 & \ddots & 1 \\ 0 & & 1 & 0 \end{bmatrix}$$

$$\gamma(x) = 3 + \|\underline{x}\|_2$$

$$\underline{\underline{D}}\,\underline{\underline{A}}(\underline{x}) = \underline{\underline{D}}\,(\underline{\underline{B}} + \gamma(x)\,\underline{\underline{I}}) = D\gamma(x)\,\underline{\underline{I}}$$

↓ compute by hand $\frac{\partial}{\partial x_1}\|x\|_2 \Rightarrow$

$$D\gamma(x) = D\|\underline{x}\|_2 = \frac{x^T}{\|x\|_2}$$

$$\underline{\underline{DF}}(x) = \underline{\underline{A}}(x) + \frac{x\,x^T}{\|x\|_2}$$

3) autonomous ODE

$$\dot{\underline{y}} = f(\underline{y})$$

↳ no explicit dependence of f on t

ex. $\dot{y} = \cos(y(t))$ is autonomous.

$\dot{y} = y^2$

$\dot{y} = t^2 + y^2$ is not autonomous

Note: every ODE can be made autonomous

$\dot{\underline{y}} = \underline{f}(t, \underline{y})$ $\qquad f : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$

Denote $\underline{z} = \begin{bmatrix} \underline{y} \\ t \end{bmatrix} \in \mathbb{R}^{d+1}$

Here: we add t as unknown.

$\underline{z} : \mathbb{R} \to \mathbb{R}^{d+1}$

$\underline{z}(t) = \begin{bmatrix} \underline{y}(t) \\ t \end{bmatrix}$

$\dot{\underline{y}} = \underline{f}(t, \underline{y}) \Rightarrow$

$$\dot{\underline{z}} = \begin{bmatrix} \dot{\underline{y}}(t) \\ 1 \end{bmatrix} = \begin{bmatrix} \underline{f}(t, \underline{y}) \\ 1 \end{bmatrix}$$

Denote $g : \mathbb{R}^{d+1} \to \mathbb{R}^{d+1}$

$$\underline{g}(\underline{z}) = \begin{bmatrix} f(z_{d+1}, [z_1, z_2, \dots, z_d]^T) \\ 1 \end{bmatrix}$$

$\Rightarrow$ ODE becomes $\dot{\underline{z}} = g(\underline{z})$

Note autonomous eq. are invariant to translations in time!

$$\begin{cases} \dot{\underline{y}} = f(\underline{y}) \\ \underline{y}(t_0) = y_0 \end{cases} \Longleftrightarrow \begin{cases} \dot{y} = f(\underline{y}) \\ \underline{y}(0) = \underline{y}_0 \end{cases}$$

4) Q11.2.3.4.c

$$\dot{y} = y^2$$

implicit Euler ⇒ $y_{k+1} = y_k + h f(t_{k+1}, y_{k+1})$ is

$$y_{k+1} = y_k + h y_{k+1}^2 \quad \text{implicit}$$

unknown $x = y_{k+1}$

eq. $x = y_k + h x^2 \Leftrightarrow h x^2 - x + y_k = 0$

$$\Delta = 1 - 4hy_k$$

$$x_{1,2} = \frac{1 \pm \sqrt{1-4hy_k}}{2h}$$

$$\boxed{0 < h < \frac{1}{4y_k}}$$

$0 < 1 - 4hy_k \Leftrightarrow 4hy_k < 1 \Leftrightarrow h < \frac{1}{4y_k}$

⇒ implicit methods are not easy ☺

5) Q11.3.1.17B

(i) Relationship between single step methods for

$$\int_a^b \begin{cases} \dot{y} = f(t,y) \\ y(t_0) = y_0 \end{cases} \quad \text{and} \quad \int_a^b f(\tau)d\tau, \quad f:[a,b] \to \mathbb{R}$$

$$\int_a^b \dot{y}(t)\,dt = \int_a^b f(\tau)d\tau \Rightarrow$$

$$\int_a^b f(\tau)dt = y(b) - y(a) = y(b) \approx \underset{\uparrow}{y_M}$$

From the numerical method for ODE

with $\begin{cases} \dot{y}(t) = f(t) \\ y(a) = 0 \end{cases}$

(ii) Method of order p for ODE ⇒

if uniform time step $\frac{b-a}{M}$ ⇒

$$\underset{\overset{\shortparallel}{|y(b) - y_M|}}{\left|\int_a^b f(\tau)dt - y_M\right|} = \left(\frac{b-a}{M}\right)^p$$

(iii) use iMP for ODE

$$y_{k+1} = y_k + h f\left(\tfrac{1}{2}(t_k+t_{k+1}) , \tfrac{1}{2}(y_k+y_{k-1})\right)$$

$h = \frac{b-a}{M}$; $y_0 = y(a) = 0$

$$\int_a^b f(\tau)\,d\tau = y(b) \approx y_M = 0 + h \sum_{k=1}^{M-1} \underbrace{f\left(\tfrac{1}{2}(t_k+t_{k+1}), \tfrac{1}{2}(y_k+y_{k+1})\right)}_{f(t_k+\frac{h}{2})}$$

↗ MP for quadrature!

b) Q 11.3.2.34c

$$\dot{\underline{y}} = \underline{f}(\underline{y}) \qquad f : D \subset \mathbb{R}^d \to D$$

$$\Psi^h \underline{y} = y + h\,\underline{f}(\underline{y}) + \frac{h^2}{2} D\underline{f}(\underline{y})\,\underline{y}$$

order of convergence p:

$$\|\Psi^h \underline{y}(t) - \Phi^h \underline{y}(t)\| \le C \cdot h^{p+1} \quad , h \text{ small} \ , t \in]0,T[$$

(local)

$$\max_{k=1,\dots,M} \|\underline{y}_k - \underline{y}(t_k)\| \le \tilde{C} h^p \qquad \left(h = \frac{T}{M}\right)$$

Taylor for the exact solution is:

$$\Phi^h \underline{y}_0 = \underline{y}(h) = \underline{y}(0) + h \underset{f(\underline{y}(0))}{\underset{\|}{\dot{\underline{y}}(0)}} + \frac{h^2}{2} \underset{D\underline{f}(\underline{y}(0))\underline{y}(0)}{\underset{\|}{\ddot{\underline{y}}(0)}} + O(h^3)$$

$$\Rightarrow \quad \Phi^h y_0 = y(h) = \underbrace{y(0) + h f(y(0)) + \frac{h^2}{2} Df(y(0))\, y(0)}_{\Psi^h y_0} + \underset{c \cdot h^3}{O(h^3)}$$

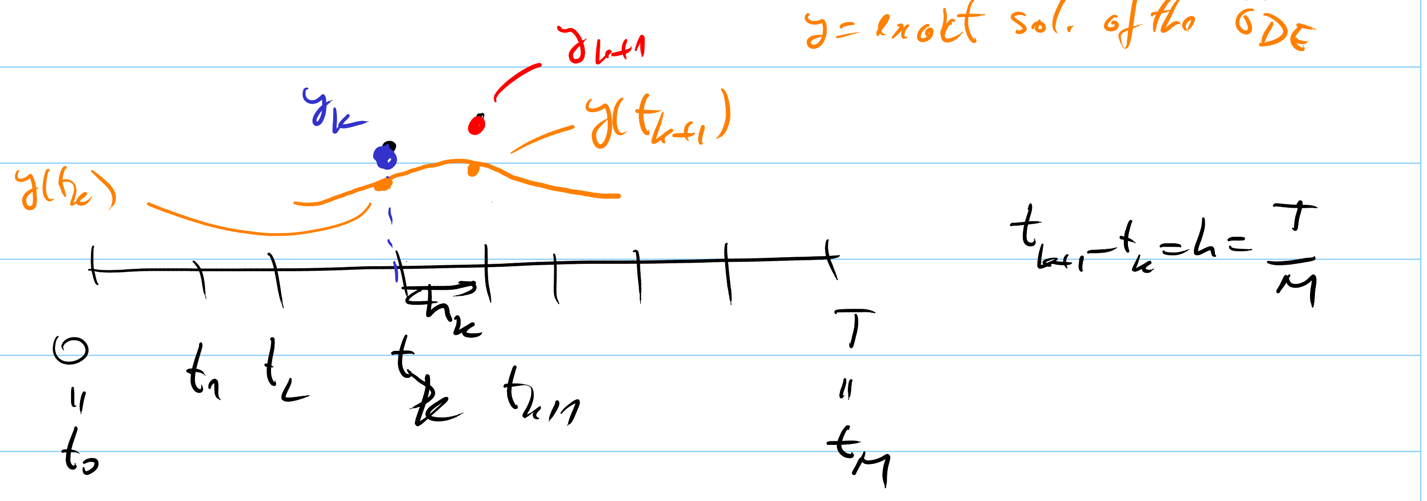
$\Rightarrow$ error $\underline{\text{in one step}}$ $|\Phi^h y_0 - \Psi^h y_0| \le c\, h^3$

↑

Local error

$\Rightarrow$ global error:

$$\max_{k=1,\dots,M} \| y_k - y(t_k) \| \le \tilde{C}\, h^2$$



$t_{k+1} - t_k = h = \frac{T}{M}$

Single step method:

proposes an y_{k+1} approximation for $y(t_{k+1})$

using only information from (the previous step) y_k

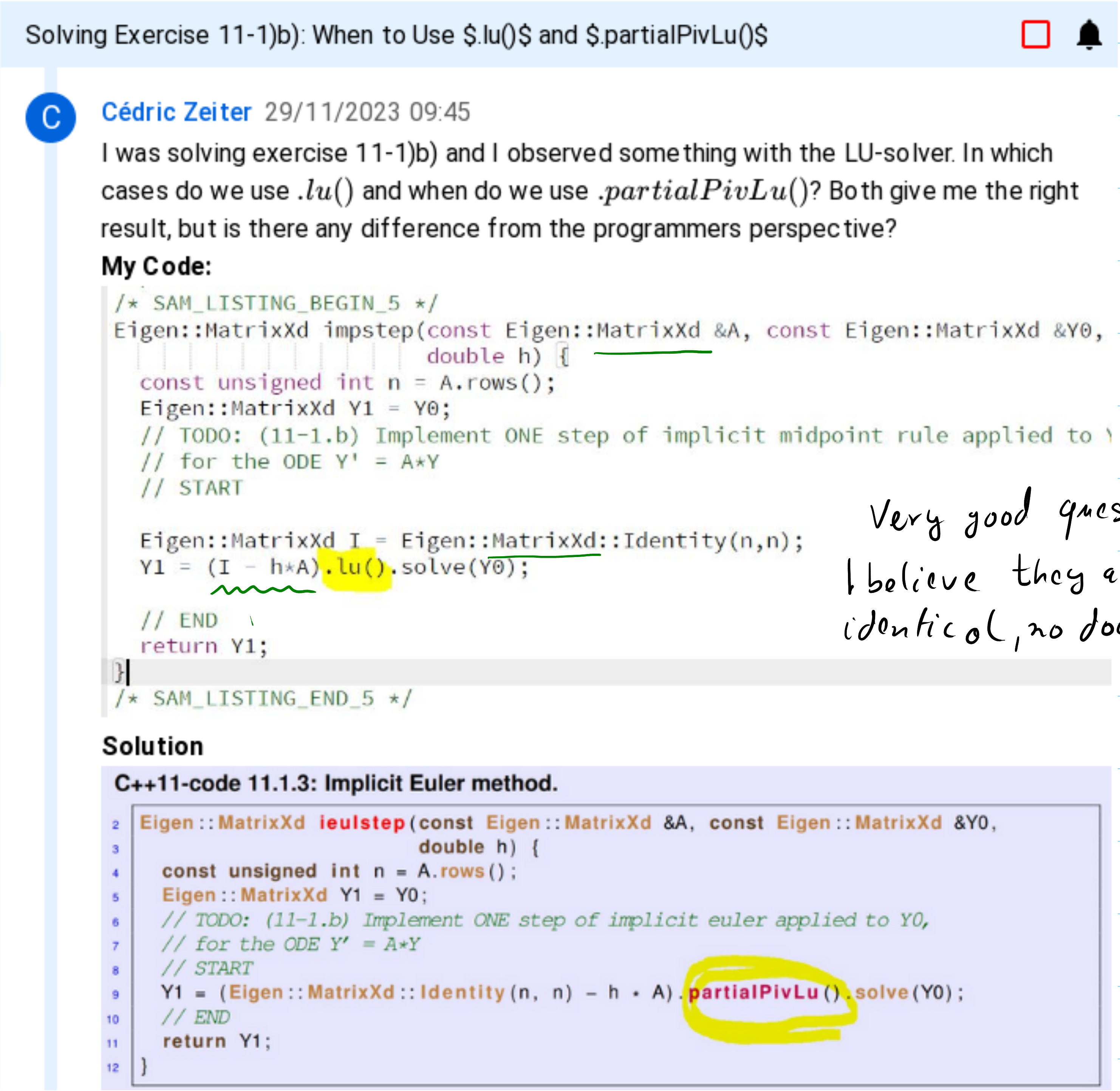
Formal way of writing:

$$y_{k+1} = \Psi(h_k, y_k)$$

↑ depend only on the interval length because ODE autonomous.

Note: a 2-step method: $y_{k+1} = \Psi(h_k, h_{k-1}, y_k, y_{k-1})$

01.12.2023



Solving Exercise 11-1)b): When to Use $.lu()$ and $.partialPivLu()$

Cédric Zeiter 29/11/2023 09:45

I was solving exercise 11-1)b) and I observed something with the LU-solver. In which cases do we use $.lu()$ and when do we use $.partialPivLu()$? Both give me the right result, but is there any difference from the programmers perspective?

My Code:

```
/* SAM_LISTING_BEGIN_5 */
Eigen::MatrixXd impstep(const Eigen::MatrixXd &A, const Eigen::MatrixXd &Y0,
                        double h) {
  const unsigned int n = A.rows();
  Eigen::MatrixXd Y1 = Y0;
  // TODO: (11-1.b) Implement ONE step of implicit midpoint rule applied to Y
  // for the ODE Y' = A*Y
  // START

  Eigen::MatrixXd I = Eigen::MatrixXd::Identity(n,n);
  Y1 = (I - h*A).lu().solve(Y0);

  // END
  return Y1;
}
/* SAM_LISTING_END_5 */
```

Solution

C++11-code 11.1.3: Implicit Euler method.

```
Eigen::MatrixXd ieulstep(const Eigen::MatrixXd &A, const Eigen::MatrixXd &Y0,
                         double h) {
  const unsigned int n = A.rows();
  Eigen::MatrixXd Y1 = Y0;
  // TODO: (11-1.b) Implement ONE step of implicit euler applied to Y0,
  // for the ODE Y' = A*Y
  // START
  Y1 = (Eigen::MatrixXd::Identity(n, n) - h * A).partialPivLu().solve(Y0);
  // END
  return Y1;
}
```

Very good question :)
I believe they are identical, no documentation found!

Read the code!

(*) Q 12.2.0.17B

Damped pendulum.

$$\ddot{w} = -\sin w - \lambda \dot{w}$$

For which λ is this ODE stiff near $w=0, \dot{w}=0$?

$$u = \dot{w} \Rightarrow \dot{u} = -\sin w - \lambda u$$

$$\underline{y} = \begin{bmatrix} w \\ u \end{bmatrix} \Rightarrow \begin{cases} \dot{y}_2 = -\sin y_1 - \lambda y_2 \\ \dot{y}_1 = y_2 \end{cases}$$

$$\dot{\underline{y}} = \underline{f}(\underline{y}) \text{ wit } \underline{f}(\underline{y}) = \begin{bmatrix} y_2 \\ -\sin y_1 - \lambda y_2 \end{bmatrix}$$

Linearisation arond $\underline{y}^*$:

Taylor for $\underline{f}$ around $\underline{y}^*$:

$$f(\underline{y}) = f(\underline{y}^*) + \underline{D}f(\underline{y}^*)(\underline{y}-\underline{y}^*) + O(\|\underline{y}-\underline{y}^*\|^2)$$

Here $\underline{y}^* = \begin{bmatrix}0\\0\end{bmatrix} \Rightarrow \underline{f}(\underline{y}) \approx \underline{0} + D\underline{f}\left(\begin{bmatrix}0\\0\end{bmatrix}\right)(\underline{y}-\underline{0}) =$

$\underline{f}\left(\begin{bmatrix}0\\0\end{bmatrix}\right) = \begin{bmatrix}0\\0\end{bmatrix} = \underline{0}$ $\qquad \underline{\underline{Df}}\left(\begin{bmatrix}0\\0\end{bmatrix}\right)\underline{y}$

Test problem is $\dot{\underline{y}} = \underline{A}\,\underline{y}$ with $\underline{A} = \underline{\underline{Df}}\left(\begin{bmatrix}0\\0\end{bmatrix}\right)$

$$\underline{\underline{Df}}(\underline{y}) = \begin{bmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} \\ \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial y_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\cos y_1 & -\lambda \end{bmatrix}$$

$$\underline{A} = \underline{\underline{Df}}\left(\begin{bmatrix}0\\0\end{bmatrix}\right) = \begin{bmatrix} 0 & 1 \\ -1 & -\lambda \end{bmatrix}$$

$$\underline{A} - \mu \underline{\underline{I}} = \begin{bmatrix} -\mu & 1 \\ -1 & -\mu-\lambda \end{bmatrix}$$

$$\det(\underline{A} - \mu I) = \mu(\mu+\lambda) + 1 = \mu^2 + \lambda\mu + 1$$

$$\mu_{1,2} = \frac{-\lambda \pm \sqrt{\lambda^2-4}}{2}$$

$\lambda > 2$: $\mu_1 = \dfrac{-\lambda - \sqrt{\lambda^2-4}}{2}$, $\mu_2 = \dfrac{-\lambda + \sqrt{\lambda^2-4}}{2}$

$\boxed{\lambda \gg 2}$ $\quad \approx -2\lambda \qquad$ near <0 $\Rightarrow$ stiff!

$|\lambda| < 2 \Rightarrow \mu_{1,2} = \dfrac{-\lambda \pm i\sqrt{4-\lambda^2}}{2}$ → oscillations

→ bounded. decay if $0 < \lambda < 2$

easy!

$\lambda < -2$: this makes no sense, because exact sol. explodes

$\Rightarrow$ no physical meaning!

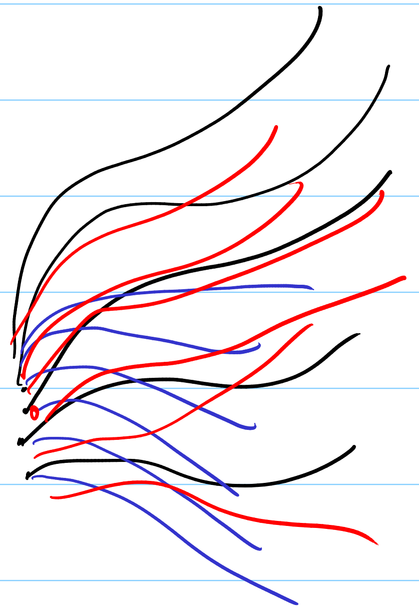
$\lambda = 2$: $\Rightarrow \mu_1 = \mu_2 = -\frac{2}{2} = -1$

$\underline{A}$ not diagonalisebar.

8.12.2024

1) Splitting.

IMPORTANT: autonomous ODEs!



$$\dot{\underline{y}} = \underline{f}(\underline{y}) \qquad (1)$$

$$\dot{\underline{y}} = \underline{g}(\underline{y}) \qquad (2)$$

$$\dot{\underline{y}} = \underline{f}(\underline{y}) + \underline{g}(\underline{y}) \qquad (3)$$

simplest case:

$\underline{f}(\underline{y}) = \underline{\underline{A}}\,\underline{y}$ and $\underline{g}(\underline{y}) = \underline{\underline{B}}\,\underline{y}$

$$f(y) + g(y) = (\underline{\underline{A}} + \underline{\underline{B}})\,\underline{y}$$

(1) $\dot{\underline{z}} = \underline{\underline{A}} \cdot \underline{z} \Rightarrow \underline{z}(t) = e^{\underline{\underline{A}}t}\,\underline{z}(0) = \Phi_{\underline{\underline{A}}}^t\,\underline{z}(0)$

(2) $\dot{\underline{v}} = \underline{\underline{B}}\,\underline{v} \Rightarrow \underline{v}(t) = e^{\underline{\underline{B}}t}\,\underline{v}(0) = \Phi_{\underline{\underline{B}}}^t\,\underline{v}(0)$

(3) $\dot{\underline{y}} = (\underline{\underline{A}} + \underline{\underline{B}})\,\underline{y} \Rightarrow \underline{y}(t) = e^{(\underline{\underline{A}} + \underline{\underline{B}})t}\,\underline{y}(0) = \Phi_{A+B}^t\,\underline{y}(0)$

$$e^{(\underline{\underline{A}}+\underline{\underline{B}})t} = (\underline{\underline{A}}+\underline{\underline{B}})^0 t^0 + (\underline{\underline{A}}+\underline{\underline{B}})^1 t^1 + \frac{1}{2}(\underline{\underline{A}}+\underline{\underline{B}})^2 t^2 + \dots$$

$$= \underline{\underline{I}} + \underline{\underline{A}}t + \underline{\underline{B}}t + \underline{\frac{1}{2}\left(\underline{\underline{A}}^2 + \underline{\underline{B}}^2 + \underline{\underline{A}}\,\underline{\underline{B}} + \underline{\underline{B}}\,\underline{\underline{A}}\right)t^2} + \dots$$

$$e^{\underline{\underline{A}}t} = \underline{\underline{I}} + \underline{\underline{A}}t + \boxed{\frac{1}{2}\underline{\underline{A}}^2 t^2} + \dots$$

$$e^{\underline{\underline{B}}t} = \underline{\underline{I}} + \underline{\underline{B}}t + \boxed{\frac{1}{2}\underline{\underline{B}}^2 t^2} + \dots$$

$$e^{(\underline{\underline{A}}+B)h} \approx \left.\begin{array}{l} e^{\underline{\underline{A}}t}\,e^{\underline{\underline{B}}t} \\ \text{or } e^{\underline{\underline{B}}h}\,e^{\underline{\underline{A}}t} \end{array}\right\} + O(h^2)$$

lokally => $O(h)$ globally

Lie-Trotter splitting

$$e^{(\underline{\underline{A}}+B)h} \approx e^{\underline{\underline{A}}\frac{h}{2}}\,e^{\underline{\underline{B}}h}\,e^{\underline{\underline{A}}\frac{h}{2}} + O(h^3)$$

$$\text{or } e^{\underline{\underline{B}}\frac{h}{2}}\,e^{\underline{\underline{A}}h}\,e^{\underline{\underline{B}}\frac{h}{2}} + O(h^3)$$

Strong-splitting.

Idea: do the same for nonlinear autonomous ODE.

Lie Trotter: $\Phi^h_{f+g} = \Phi^h_f\,\Phi^h_g$ $\qquad O(h)$ globally

$$\Phi^h_{f+g} = \Phi^h_g\,\Phi^h_f$$

Strong: $\Phi^h_{f+g} = \Phi_f^{h/2}\,\Phi_g^h\,\Phi_f^{h/2}$ $\qquad O(h^2)$

$$\Phi^h_{f+g} = \Phi_g^{h/2}\,\Phi_f^h\,\Phi_g^{h/2}$$

Many very usefull higher order schemes

```
for k = 1,2,..,n:        Propagation from 0 to nh
    $y_1 = \Phi_f^{h/2}\Phi_g^h\Phi_f^{h/2} y_0$
    $y_0 = y_1$
```

works only if ODEs are autonomous

we used here tacitly the time invariance!

Propagations with $\overset{f}{g}$ from 0 to h.

Strong splitting ∈ Processing splittings.

$$\Phi_f^{h/2}\,\Phi_g^h\,\overbrace{\Phi_f^{h/2}\,\Phi_f^{h/2}}^{\Phi_f^h}\,\Phi_g^h\,\overbrace{\Phi_f^{h/2}\,\Phi_f^{h/2}}^{\Phi_f^h}\,\Phi_g^h\,\Phi_f^{h/2}\,y_0$$

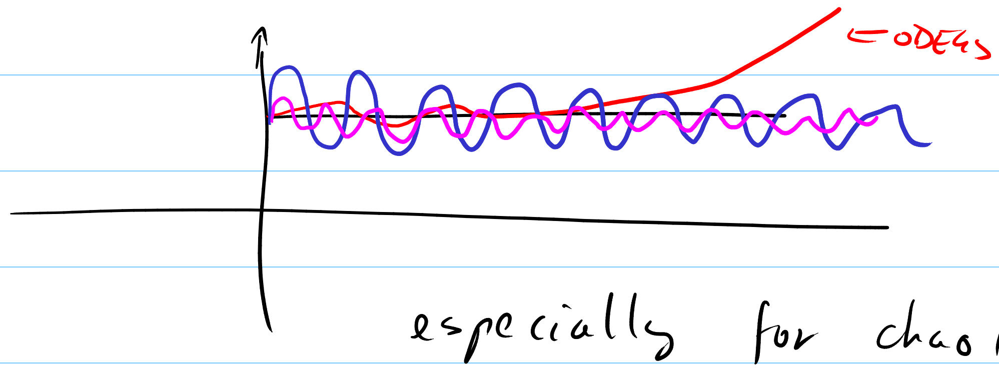
$\approx y(3h)$ $\qquad \approx y(2h)$ $\qquad \approx y(h)$

Lie-Trotter with the pre- / post- processor

If we care only of $y(nh)$ then we can compute faster: LT inside = n-1 steps of LT with only once
- pre processor $\Phi_f^{h/2}$
- post processor $\Phi_f^{h/2}$

⇒ cost of LT with convergence of Strong splitting!

Use: for conservation problems



especially for chaotic systems.

2) $\dot{y} = -Ay + (\sin \pi y_j)_{j=1}^{N}$ ← how?

split

$\dot{\underline{z}} = -A\underline{z}$ ← $\underline{z}(t) = e^{-\underline{A}h} \underline{z}(0)$

$$\begin{cases} \dot{v}_1 = \sin(\pi v_1) \\ \dot{v}_2 = \sin(\pi v_2) \\ \vdots \\ \dot{v}_N = \sin(\pi v_N) \end{cases}$$ ← exactly

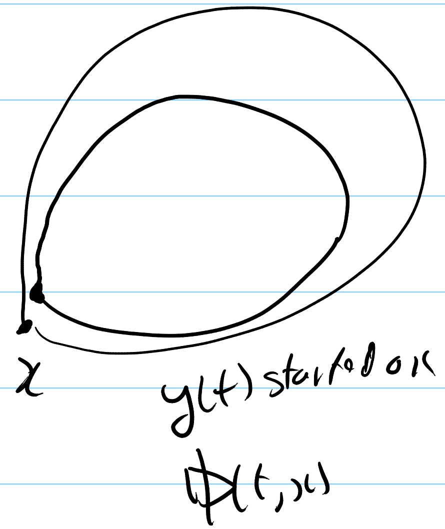
3) (11.9 d) $h = \sqrt{EPS}$ because

we compute numerically the derivative epsilon-machine.
with the divided differences
which is numerically instable for $h \approx EPS$

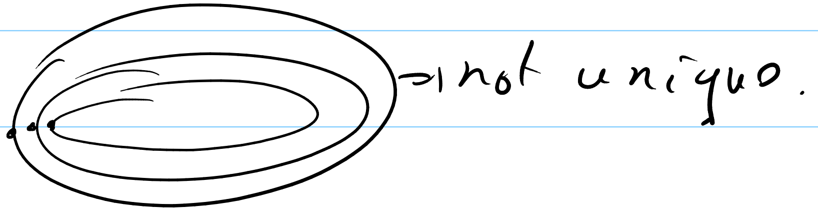
(11.7) $\begin{cases} \underline{\dot{y}} = f(\underline{y}) \\ \underline{y}(0) = \underline{x} \end{cases}$ $\longrightarrow$ $\Phi(t, \underline{x}) = y(t)$ Solution of

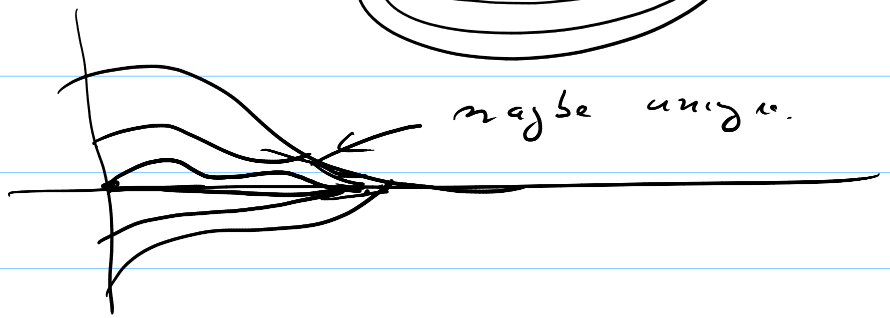
$\Phi: \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$

$D_{\underline{x}} \Phi \in \mathbb{R}^{d \times d}$



$F(x) = \Phi(T, x) - x$




maybe unique.