§1 Quadratur

§1.1. Motivation

Bsp $\begin{cases} \dot{y} = f(t,y) \\ y(t_0) = y_0 \end{cases} \Big| \int_{t_0}^{T} dt \Rightarrow y(T) = y(t_0) + \int_{t_0}^{T} f(t,y(t))dt$

Bsp Was ist die Periode eines Pendulums?

$$T = 4\sqrt{\frac{\ell}{g}}\, K\left(\sin\frac{\alpha_0}{2}\right) \quad \text{wobei}$$

$$K(a) = \int_0^{\pi/2} \frac{1}{\sqrt{1 - a^2 \sin^2 s}}\, ds$$

Bsp Plank' Gesetz für die Strahlung eines schwarzen Körpers

T = Temperatur in k

λ = in μm

$$E(\lambda, T) = \frac{c_1}{\lambda^5 \left(e^{\frac{c_2}{\lambda T}} - 1\right)}$$

Energie verlust per Flächeneinheit:

$$Q = \int_0^\infty E(\lambda, T)\, d\lambda = \sigma T^4$$

σ = Stefen-Boltzmann-Konstante

Ziel: berechne σ!

$$J = \int_a^b f(x)dx \approx Q(f,a,b) = \sum_{j=1}^{s} \omega_j f(x_j)$$

ω_j: Gewichte, x_j: Knoten

Ziel: Welche Knoten und Gewichte sollen wir wählen um den Fehler $|J - Q|$ klein

mit möglichst wenige Funktionsauswertungen zu haben.

IDEE: $f \approx$ einfache Funktion, dessen Integral leicht/analytisch berechenbar ist

z.B: $f \approx$ Polynom $\alpha_0 + \alpha_1 x + \dots + \alpha_n x^n$

oder $f \approx$ trigonometrischen Polynom.

$$f \approx \alpha_0 + \alpha_1 e^{i1x} + \alpha_2 e^{i2x} + \dots + \alpha_{-1} e^{-i1x} + \alpha_{-2} e^{-i2x} + \dots$$

Gegeben knoten $x_0, x_1, \dots, x_n$ kann man $\alpha_0, \alpha_1, \dots, \alpha_n$ berechnen, so dass

$p_n(x) = \alpha_0 + \alpha_1 x + \dots + \alpha_n x^n$ erfüllt

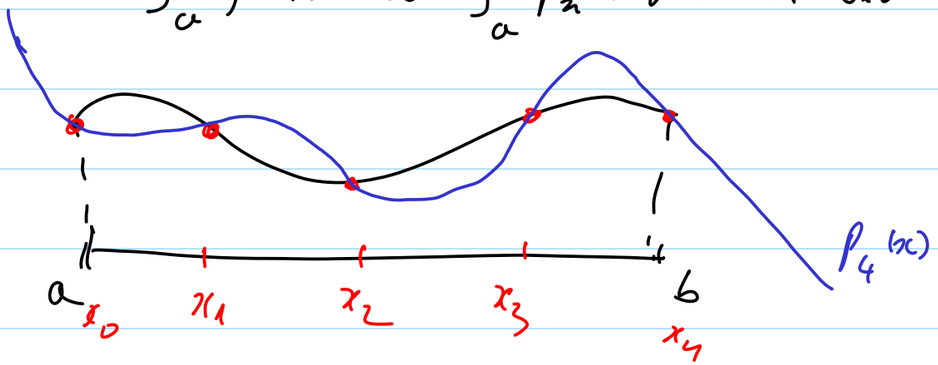
$p_n(x_j) = f(x_j)$ für $j = 0, 1, 2, \dots, n$

Im Prinzip, das ist das LGS

$$\begin{bmatrix} 1 & x_0 & \dots & x_0^n \\ 1 & x_1 & & x_1^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \dots & x_n^n \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} f(x_0) \\ \vdots \\ f(x_n) \end{bmatrix}$$

Somit $f(x) \approx p_n(x)$ und dann

$$\int_a^b f(x)\,dx \approx \int_a^b p_n(x)\,dx = \dots \text{ exakt}$$

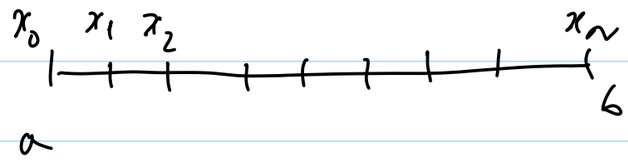


Man kann beweisen:

$$f \in C^n[a,b] \Rightarrow \left| \int - Q \right| \le \frac{1}{n!}(b-a)^{n+1} \max_{z \in [a,b]} \left| f^{(n)}(z) \right|$$

Glattheit ist wichtig — Länge des Intervals ist wichtig!

IDEE: Zerlege $\int_a^b f(x)dx = \sum_{k=0}^{N-1} \int_{x_k}^{x_{k+1}} f(x)\,dx$



$x_k = x_0 + kh$ mit $h = \frac{b-a}{N}$ klein

und wende die Quadraturformel auf jedes Interval der Länge h (klein)

$\Rightarrow$ zusammengesetzte Quadraturformel.

Fehler: $\left| \int_a^b f(x)dx - \sum_{k=0}^{N-1} Q(f, x_k, x_{k+1}) \right| \leq$

$\leq \sum_{k=0}^{N-1} \left| \int_{x_k}^{x_{k+1}} f(x)dx - Q(f, x_k, x_{k+1}) \right| \leq$

$\leq \sum_{k=0}^{N-1} \frac{1}{n!} \underbrace{(x_{k+1} - x_k)}_{h}^{n+1} \max_{z \in [x_k, x_{k+1}]} |f^{(n)}(z)| =$

$\sum_{k=0}^{N-1} \frac{h^{n+1}}{n!} \max_{z \in [x_k, x_{k+1}]} |f^{(n)}(z)| \leq C \cdot \frac{1}{n!} h^{n+1} \cdot N$

$h = \frac{b-a}{N} \Rightarrow N = \frac{b-a}{h}$

Somit: Fehler $\leq \frac{C}{n!} h^{n+1} \cdot \frac{(b-a)}{h} = \frac{(b-a)}{n!} C h^n$

!

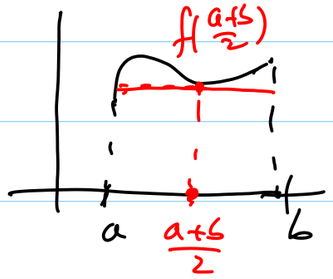
Def Quadraturformel hat Ordnung $n+1$ wenn sie Polynome von Grad maximal n exakt integriert.
(das erste falsche Ergebniss: x^{n+1})

$\int_a^b p(x)dx = Q(p, a, b)$ für alle Polynome vom Grad $\leq n$

und es gibt ein Polynom $\tilde{p}$ vom Grad $n+1$:
$\int_a^b \tilde{p}(x)\,dx \neq Q(\tilde{p}, a, b)$.

Bsp 1) Mittelpunktsregel

$$Q^M(f,a,b) = (b-a)\, f\left(\frac{a+b}{2}\right)$$



Bemerkung: 1) Polynome von Grad 0 und 1 werden exakt integriert (mittels MPR)

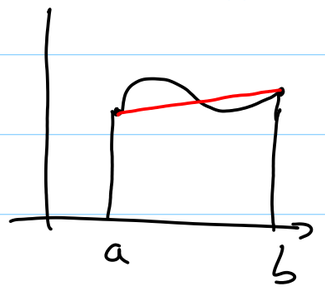
$\Rightarrow$ MPR hat Ordnung 2

Fehler: $\frac{(b-a)^3}{24} f''(z)$ mit $z \in [a,b]$

Bsp 2) Trapezregel:

$$Q^T(f,a,b) = \frac{b-a}{2} f(a) + \frac{b-a}{2} f(b)$$

Gewichte, Knoten



$$= \sum_{j=1}^{2} w_j \cdot f(x_j) \qquad x_1 = a,\ x_2 = b,\ w_1 = w_2 = \frac{b-a}{2}$$

Bem 1)

Ordnung 2, Fehler: $\frac{1}{12}(b-a)^3 f^{(2)}(z)$ mit $z \in [a,b]$

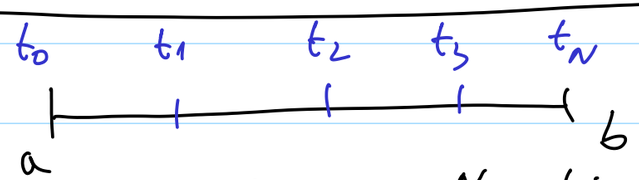
Bem 2) geschlossene Quadraturformel:

Enden von $[a,b]$ sind Knoten.

Bem 2) offene Quadraturformel:

Enden von $[a,b)$ sind keine Knoten

Implementieren:



N Intervale

Argument: f, a, b, N;

$$\int_a^b f(x)dx = \sum_{j=1}^{N} \int_{t_{j-1}}^{t_j} f(x)\,dx$$

MPR: $\int_a^b f(x)dx \approx \sum_{j=1}^{N} \underbrace{(t_i - t_{i-1})}_{h = \frac{b-a}{N}} f\left(\frac{t_{i-1}+t_i}{2}\right) =$

$$= \frac{b-a}{N} \sum_{j=1}^{N} f(x_j) \quad \text{wobei } x_j = t_{i-1} + \frac{h}{2}$$

Bsp 3) Simpson Regel

$$Q^S(f,a,b) = \underbrace{\frac{b-a}{6}}_{\omega_1} \underbrace{f(a)}_{x_1} + \underbrace{\frac{b-a}{6}}_{\omega_2} \cdot 4 \underbrace{f\left(\frac{a+b}{2}\right)}_{x_2} + \underbrace{\frac{b-a}{6}}_{\omega_3} \underbrace{f(b)}_{x_3}$$

$$= \sum_{j=1}^{3} \omega_j f(x_j)$$

Bemerkung Ordnung 4; Fehler: $\frac{1}{90}\left(\frac{b-a}{2}\right)^5 f^{(4)}(z)$

mit $z \in [a,b]$.

Input: f, a, b, N

$h = \frac{b-a}{N}$

$\underline{t} = [t_0, t_1, \ldots, t_N]$ mit $t_0 = a$, $t_N = b$

(linspace(a, b, N+1))

$\underline{x} = \underline{t}[:-1] + \frac{h}{2}$

Output: $h\,\mathrm{sum}(f(\underline{x}))$

TR:
$$\int_a^b f(x)\,dx = \sum_{j=1}^{N} \int_{t_{j-1}}^{t_j} f(x)\,dx \approx \sum_{j=1}^{N} \left(\frac{h}{2} f(t_{j-1}) + \frac{h}{2} f(t_j) \right)$$

$$= h \sum_{j=1}^{N} \left(\frac{1}{2} f(t_{j-1}) + \frac{1}{2} f(t_j) \right) =$$

$j=1$ $\quad$ $j=2$ $\quad$ $j=N$

$$= \frac{h}{2} f(t_0) + \frac{h}{2} f(t_1) + \frac{h}{2} f(t_1) + \frac{h}{2} f(t_2) + \ldots + \frac{h}{2} f(t_{N-1}) + \frac{h}{2} f(t_N)$$

$$= \frac{h}{2} f(t_0) + h f(t_1) + h f(t_2) + \ldots + h f(t_{N-1}) + \frac{h}{2} f(t_N)$$

$$= \frac{h}{2} f(t_0) + h \sum_{j=1}^{N-1} f(t_j) + \frac{h}{2} f(t_N)$$

$\underline{x} = \underline{t}[1:-1]$

Output: $\mathrm{sum}(h f(\underline{x})) + \frac{h}{2} f(a) + \frac{h}{2} f(b)$

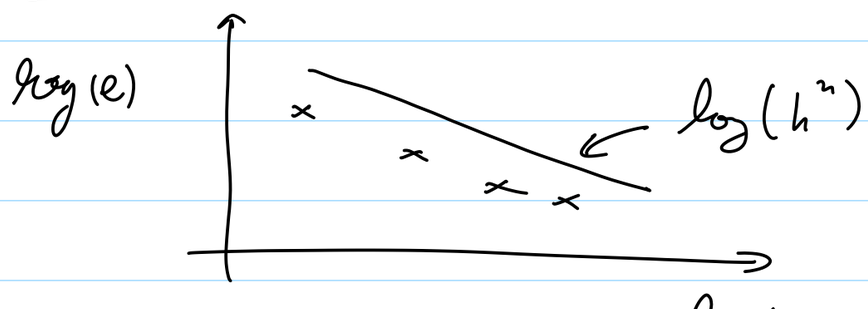
Simpson: ähnlich

$$\sum_{j=1}^{N} \left(\frac{h}{6} f(t_{j-1}) + \frac{h}{6} 4 f\left(\frac{t_{j-1} + t_j}{2}\right) + \frac{h}{6} f(t_j) \right)$$

```
from scipy.integrate import quad
```

Fehler: $C \cdot h^{(n)}$

$e(h,n) = |\square - Q(h)|$ welcher n?



$\log(c \cdot h^n) = n(\log h) + \log c$

Implementiere 3 Funktionen

mpr, tr, simp wie besprochen

Anwende für $\sin x$; $\frac{1}{1+(5x)^2}$; $\sqrt{x}$

Plank-konstante usw.

Fehlerplots.

Simpson:

$$
\begin{aligned}
I(f;a,b) &\approx \sum_{i=0}^{N-1} \frac{h}{6}\left(f(x_i) + 4f\left(\frac{x_i + x_{i+1}}{2}\right) + f(x_{i+1})\right) \\
&= \frac{h}{6}\left(f(a) + 2\sum_{i=1}^{N-1} f(x_i) + 4\sum_{i=1}^{N} f\left(\frac{x_{i-1} + x_i}{2}\right) + f(b)\right) \\
&= \frac{\tilde{h}}{3}\left(f(\tilde{x}_0) + 2\sum_{i=1}^{N-1} f(\tilde{x}_{2i}) + 4\sum_{i=1}^{N} f(\tilde{x}_{2i-1}) + f(\tilde{x}_{2N})\right),
\end{aligned}
$$

wobei $\tilde{h} = \frac{h}{2}$ und $\tilde{x}_{2i} = x_i$ und $\tilde{x}_{2i-1} = \frac{x_{i-1}+x_i}{2}$.

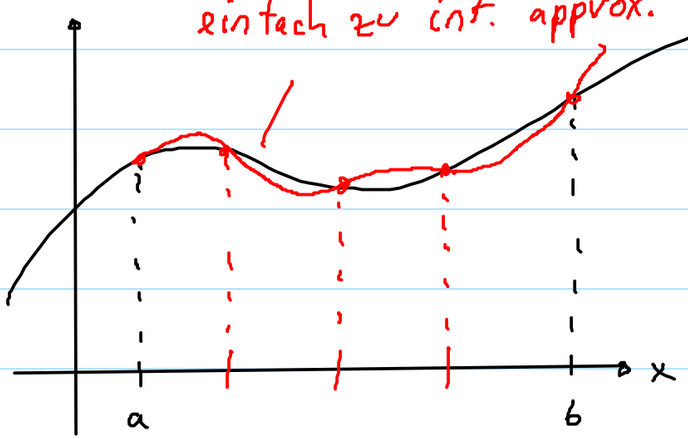
Kurz Zusammenfassung

Ziel: Approx. von

$$Q(f) \approx \int_a^b f(x)\,dx = I(f)$$

Wozu: Oft leider nicht exakt berechenbar

Idee:



Einfache Fkt.: Polynome!

Polynomiale Interpolation

Geg. $n+1$ paarweise verschiedene Stützstellen/Knoten $x_0, x_1, \dots, x_n$ und dazugehörige Stützwerte $y_0, y_1, \dots, y_n$

finde das Polynom n-ten Grades, Interp.-Polynom (IP)

$$p_n(x) = a_0 + a_1 \cdot x + \dots + a_n x^n \in \mathbb{P}_n$$

welches die Interpolationsbedingungen (IB) erfüllt

$$p_n(x_j) = y_j \qquad (j = 0, \dots, n)$$

Die $n+1$ Koeff. $a_0, \dots, a_n$ ergeben sich durch Lösen eines linearen Gleichungssystem (LGS)

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & & x_1^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Das IP lässt sich auch direkt mit der sog. Lagrange'schen Interpolationsformel (LF)

$$p_n(x) = \sum_{j=0}^{n} y_j \cdot L_j^n(x)$$

wobei

$$L_j^n(x) = \prod_{\substack{i=0 \\ i \neq j}}^{n} \frac{x - x_i}{x_j - x_i} \qquad (j = 0, 1, \ldots, n)$$

(LP)

die Lagrange-Polynome sind

Die LP haben folgende Eigenschaften

(LP1) $L_j^n(x)$ sind Polynome von Grad n

(LP2) $L_j^n(x_k) = \delta_{jk} = \begin{cases} 1 & j = k \\ 0 & \text{sonst} \end{cases}$

Wegen (LP2) funktioniert

$$p_n(x_i) = \sum_{j=0}^{n} y_j \cdot L_j^n(x_i)$$

$$= 0 + \ldots + 0 + y_i \cdot \underbrace{L_i^n(x_i)}_{1} + 0 \ldots$$

$$= y_i$$

IB erfüllt ✓.

Soweit waren die Stützwerte beliebig.

Sei nun $f: I=[a,b] \to \mathbb{R}$ und wir wollen mit einen IP approx.

$$p_n[f|x_0,\dots,x_n](x) \in \mathbb{P}_n$$

Dieses erfüllt die IB

$$p_n[f|x_0,\dots,x_n](x_j) = f(x_j) \quad (j=0,1,\dots,n)$$

Numerische Integration = Quadratur

Ziel: Approx. von $I(f) = \int_a^b f(x)\,dx$

Idee: Verwende polynomiale Interpolation um $f(x)$ zu approx. und integriere

$p[f|x_0,\dots,x_n]$

(... Poly. sind so einfach!)

Def.: Eine endliche Rechenvorschrift der Form

$$Q(f) = \sum_{j=0}^{n} w_j \cdot f(x_j)$$

zur Approx. von $I(f)$ nennt man Quadraturregel (QR) oder Quad.-formel

Die $x_j \in [a,b]$ nennt man (Quadratur-) Knoten oder Integrationsstützstelle

und die w_j (Quadratur-) Gewichte

QR können nun ganz einfach hergeleitet werden.

Seien $x_0, x_1, \dots, x_n \in I$ uns gegebe Knoten.

Dann ist das IP

$$p[f|x_0,\dots,x_n] = \sum_{j=0}^{n} f(x_j) \cdot L_j^n(x)$$

Da $p[f|x_0,\dots,x_n] \approx f(x)$

können wir versuchen

$$\int_a^b f(x)\,dx \approx \int_a^b p[f|x_0,\dots,x_n]\,dx$$

$$= \int_a^b \sum_{j=0}^{n} f(x_j)\cdot \mathcal{L}_j^n(x)\,dx$$

$$= \sum_{j=0}^{n} \int_a^b f(x_j)\cdot \mathcal{L}_j^n(x)\,dx$$

$$= \sum_{j=0}^{n} f(x_j)\cdot \underbrace{\int_a^b \mathcal{L}_j^n(x)\,dx}$$

Gewichte!

$$= \sum_{j=0}^{n} f(x_j)\cdot w_j$$

Also die Gewichte sind gegeben durch

$$w_j = \int_a^b \mathcal{L}_j^n(x)\,dx$$

w_j hängen nicht von f ab!

Geg. Knoten: berechne w_j ein für alle Mal und tabelliere

Wichtige Bsp.

Bsp.: Mittelpunktsregel (MPR)

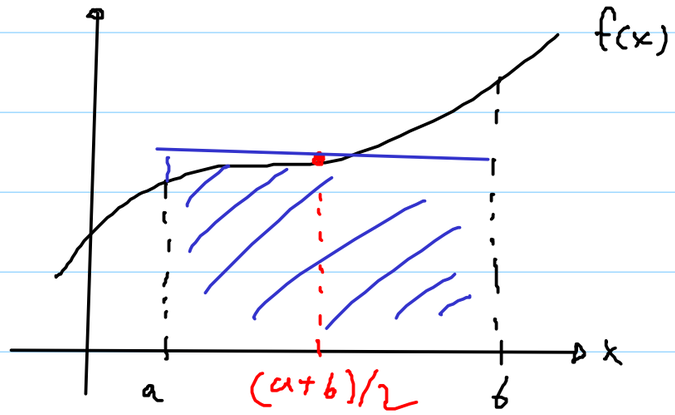
Knoten: $x_0 = \frac{a+b}{2}$

LP: $\mathcal{L}_0^0(x) = 1$

Gewichte: $w_0 = \int_a^b \mathcal{L}_0^0(x)\,dx = b-a$

Damit

$$Q_0(f) = (b-a)\cdot f\left(\frac{a+b}{2}\right)$$



Bsp.: Trapezregel (TR)

Knoten: $x_0 = a$, $x_1 = b$

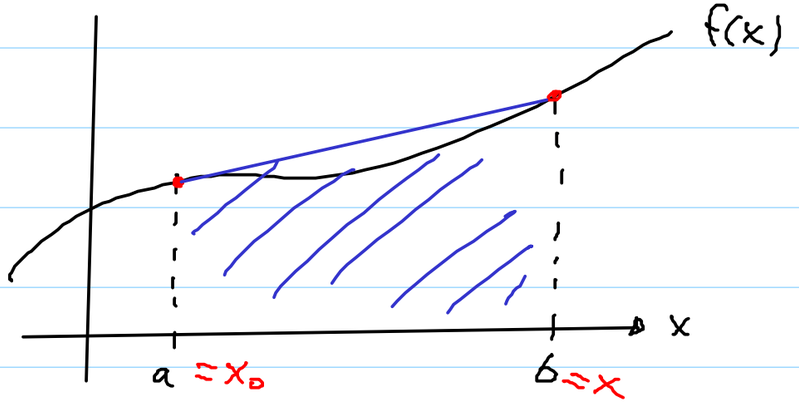
LP : ...

Gewichte: $w_0 = \frac{b-a}{2}$

$w_1 = \frac{b-a}{2}$

Damit:

$$Q_1(f) = \frac{b-a}{2}\left(f(a) + f(b)\right)$$



Bsp.: Simpson-Regel (SR)

Knoten: $x_0 = a$, $x_1 = \frac{a+b}{2}$, $x_2 = b$

LP : ...

Gewichte: $w_0 = \frac{b-a}{6}$

$w_1 = \frac{4(b-a)}{6}$

$w_2 = \frac{b-a}{6}$

Damit

$$Q_2(f) = \frac{b-a}{6}\left(f(a) + 4 \cdot f\left(\frac{a+b}{2}\right) + f(b)\right)$$

Die MPR, TR und SR sind Teil einer grossen Familie: Newton-Cotes QR_n (NC)

Bei NC wählt man die Knoten äquidistant verteilt im Integrations-Intervall

Bem.: (i) NPR, TR und SR sind sehr populär

(ii) NC QRn mit $n > 6$ werden numerisch unbrauchbar (weil negative Gewicht auftauchen)

Quadraturfehler

Def.: $E(f) = |Q(f) - I(f)|$ Quadraturfehler (QF)

Def.: Eine QR hat Genauigkeitsgrad (GG) $q \in \mathbb{N}$ falls sie alle Polynome bis zu und mit Grad q exakt integriert und q der Größte Wert mit dieser Eigenschaft
(Manchmal Exaktheitsgrad)

Def.: Die Ordnung einer QR ist $s = q+1$

Es genügt den GG für die Monome zu berechnen:

$$Q(x^k) = I(x^k) \quad k = 0, 1, \dots, q$$

$$Q(x^{q+m}) \neq I(x^{q+1})$$

[Wegen der Linearität von I und Q, und die Monome bilden eine Basis von $\mathcal{P}$

Weiter ist es bequem auf einem Referenzintervall (RI) $[-1, 1]$ zu arbeiten

Jedes Intervall $[a, b]$ lässt sich durch eine Substitution auf das RI transformieren

$$x = \frac{b-a}{2}\,t + \frac{a+b}{2} \quad \text{mit} \quad t \in [-1,1]$$

Transformiere

$$I(f) = \int_a^b f(x)\,dx$$

$$= \int_{-1}^{1} f\left(\frac{b-a}{2}\,t + \frac{a+b}{2}\right) \frac{b-a}{2}\,dt$$

$$= \frac{b-a}{2} \int_{-1}^{1} \hat{f}(t)\,dt$$

$$\approx \frac{b-a}{2} \sum_{j=0}^{n} \hat{\omega}_j\, \hat{f}(t_j)$$

Gewichte / Knoten auf $\hat{I}$

Übung: bestimme q (und s) für TPR, TR und SR

Für den QF lässt sich zeigen

$$E(f) \le \frac{\|f^{(q+1)}\|_\infty}{(q+1)!}\,(b-a)^{q+2} \qquad (QF1)$$

s — Glattheit von f

$q+1$ — Ordnung

$q+2$ — GG

Intervall-Breite

Zusammengesetzte QR$_n$

Idee: Zerteile $I=[a,b]$ in Teil-Intervalle und wende QR auf jedes an

Mit gleich grossen Teil-Intervallen

TI

$$I_j = [x_{j-1}, x_j] \quad (j = 1, \dots, N)$$

$\#TI$

und

$$x_j = a + h \cdot j \quad (j = 0, \dots, N)$$

$$h = \frac{b-a}{N}$$

Fehler?

$$\epsilon^N(f) = |I(f) - Q_n^N(f)|$$

$$= \left| \sum_{j=1}^{N} I(f \circ I_j) - Q_n(f \circ I_j) \right|$$

Δ-Ungl.

$$\leq \sum_{j=1}^{N} \underbrace{|I(f \circ I_j) - Q_n(f \circ I_j)|}_{\epsilon(f \circ I_j)}$$

(QF1)

$$\leq \sum_{j=1}^{N} \frac{\max_{x \in I_j} |f^{(q+1)}(x)|}{(q+1)!} \underbrace{|x_j - x_{j-1}|}_{h}^{q+2}$$

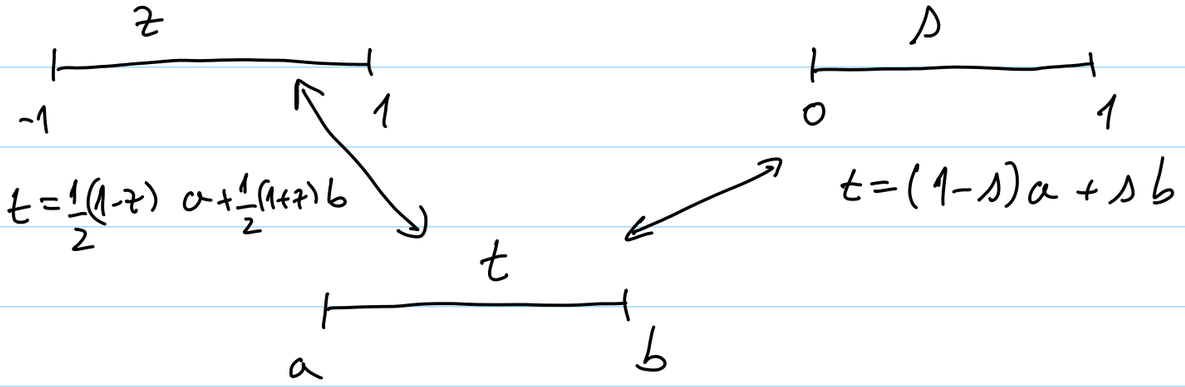
$$\leq N \cdot \frac{\|f^{(q+1)}\|_\infty}{(q+1)!} h^{q+2}$$

$N = \frac{b-a}{h}$

$$= \frac{\|f^{(q+1)}\|_\infty}{(q+1)!} (b-a) \cdot h^{q+1}$$

q q

§1.2. Referenzintervale und symmetrische QF

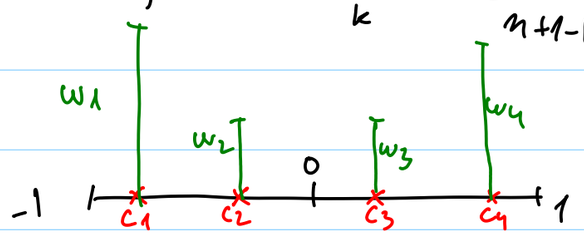


$$\int_a^b f(t)\,dt = \frac{b-a}{2}\int_{-1}^{1} \hat{f}(z)\,dz \approx \frac{b-a}{2}\sum_{j=1}^{n} \hat{\omega}_j \hat{f}(\hat{c}_j)$$

$\uparrow$ $[-1,1]$

mit $\hat{f}(z) = f\left(\frac{1}{2}(1-z)a + \frac{1}{2}(1+z)b\right)$

Definition QF auf $[-1,1]$ heisst symmetrisch falls $c_k = -c_{n+1-k}$, $w_k = w_{n+1-k}$



Theorem Die Quadraturordnung einer symmetrischen QF ist gerade.

Beweis

Annahme, QF exakt für Polynome vom Grad (max) $2n-2$. Nehme $f(x) = ax^{2n-1} + g(x)$

$g(x)$ = Polynom vom Grad $2n-2$.

$$\int_{-1}^{1} f(x)\,dx = a\int_{-1}^{1} x^{2n-1}\,dx + \int_{-1}^{1} g(x)\,dx = \sum_{k=1}^{n} \hat{\omega}_k g(\hat{c}_k)$$

$\int_{-1}^{1} = 0$

$$Q[f,-1,1] = a\sum_{k=1}^{n} \hat{c}_k^{2n-1}\cdot\hat{\omega}_k + \sum_{k=1}^{n} \hat{\omega}_k g(\hat{c}_k)$$

Es reicht zu zeigen $\sum_{k=1}^{n} \hat{\omega}_k \hat{c}_k^{2n-1} = 0$.

$$\sum_{k=1}^{n} \hat{\omega}_k \hat{c}_k^{2n-1} = \sum_{k=1}^{n} \hat{\omega}_{n+1-k}\left(-\hat{c}_{n+1-k}\right)^{2n-1} = -\sum_{k=1}^{n} \hat{\omega}_{n+1-k}\hat{c}_{n+1-k}^{2n-1} =$$

$j = n+1-k$

$$= -\sum_{j=1}^{n} \omega_j \hat{c}_j^{2n-1} \Rightarrow \sum_{k=1}^{n} \omega_k \hat{c}_k^{2n-1} = 0.$$

§1.3 Fehler für Quadratur auf [0,1]

Fehler $E(g) = \int_0^1 g(t)dt - \sum_{j=1}^{n} b_j \, g(c_j)$

c_j → Knoten in $[0,1]$

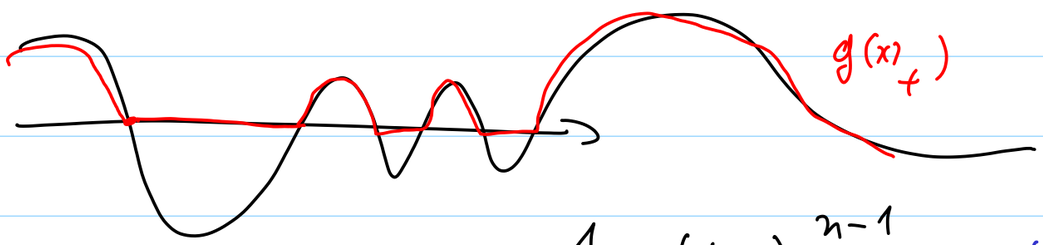
b_j ↓ Gewichte

linear in g.

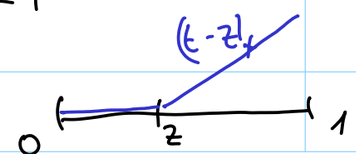
$E(Ag+Bf) = A\,E(g) + B\,E(f)$

$A, B \in \mathbb{R}$, g, f Funktionen.

das positive Teil $g(x)_+ = \begin{cases} g(x) \text{ für } x \text{ so dass } g(x)>0 \\ 0, \text{ sonst} \end{cases}$



Peano-Kern: $\alpha(z,t) = \frac{1}{(n-1)!}(t-z)_+^{n-1}$

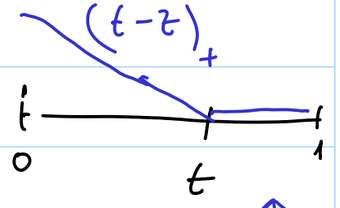


$$K_n(z) = E(\alpha(z,\cdot)) = \int_0^1 \frac{1}{(n-1)!}(t-z)_+^{n-1} dt - \sum_{j=1}^{n} b_j \frac{(c_j-z)_+^{n-1}}{(n-1)!} = \frac{(1-z)^n}{n!} - \sum_{j=1}^{n} b_j \frac{(c_j-z)_+^{n-1}}{(n-1)!}$$

Theorem Sei Q eine QF mit Q-Ordnung n und sei g n-mal stetig diff-bar.

Dann

$$E(g) = \int_0^1 K_n(z)\, g^{(n)}(z)dz.$$



Beweis Taylor:

$$g(t) = g(0) + \dots + \frac{t^{n-1}}{(n-1)!} g^{(n-1)}(0) + \int_0^t \frac{(t-z)^{n-1}}{(n-1)!} g^{(n)}(z)dz$$

$q(t)$ Grad $n-1$

$$= \int_0^1 \frac{(t-z)_+^{n-1}}{(n-1)!} g^{(n)}(z)\,dz$$

E linear $\Rightarrow$ $E(g) = \underbrace{E(q)}_{=0} + \int_0^1 \underbrace{E(\alpha(\tau,\cdot))}_{K_n(\tau)} g^{(n)}(\tau)\,d\tau$

qed.

Wenden wir den Satz auf

$$g(t) = f(x_0 + th)$$

$$\int_{x_0}^{x_0+h} f(x)\,dx - h\sum_{j=1}^{s} b_j f(x_0 + c_j h) =$$

$$= h\int_0^1 g(t)\,dt - h\sum_{j=1}^{s} b_j\, g(c_j) = h\,E(g) =$$

$$= h\,h^n \int_0^1 K_n(\tau) f^{(n)}(x_0 + h\tau)\,d\tau = h^{n+1}\int_0^1 K_n(\tau) f^{(n)}(x_0+h\tau)\,d\tau$$

Variablenwechsel und n-te Ableitung

$g(t) = f(x_0 + ht) \Rightarrow g'(t) = h f'(x_0+ht) \Rightarrow \dots$

$$g^{(n)}(t) = h^n f^{(n)}(x_0 + ht)$$

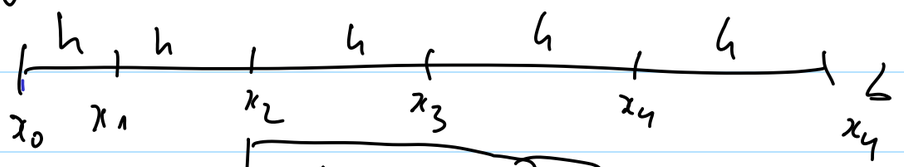
Zusammengesetzte Quadraturformel

$$|E(f)| = \left|\int_a^b f(x)\,dx - \sum_{k=1}^{N}\sum_{j=1}^{s} b_j f(x_{k-1} + c_j h)\right| \le$$

$$\le c\cdot h^n \max_{x\in[a,b]} |f^{(n)}(x)|$$

Glattheit von f wichtig

Ordnung der lokalen QF.

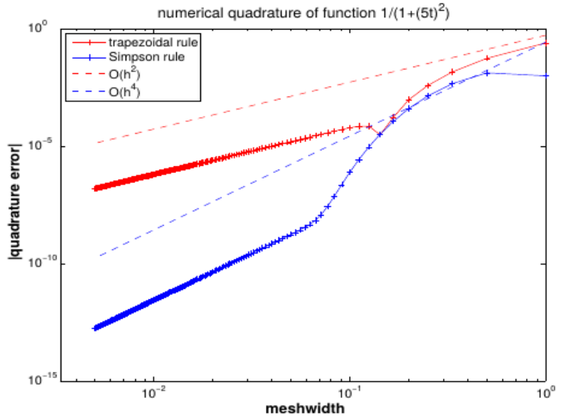


$$|\quad| \le h^{n+1}\int_0^1 |K_n(\tau)|\,d\tau \cdot \max_{x\in[x_0,x_0+h]} |f^{(n)}(x)|$$

Konstante

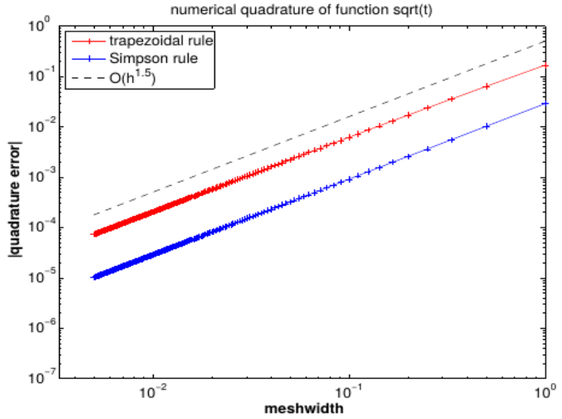
MPR, TR $= \frac{1}{12}$

Simpson: $\frac{1}{2880}$



Quadratur-Fehler für $f_1(t) := \frac{1}{1+(5t)^2}$

Ordnung 2 für die Trapez-Regel und Ordnung 4 für Simpson-Regel



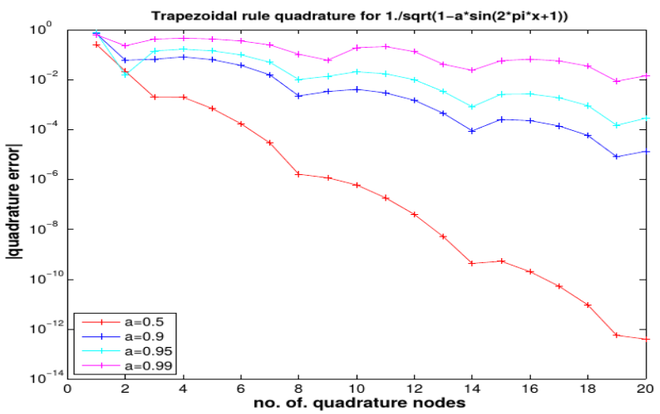
Quadratur-Fehler für $f_2(t) := \sqrt{t}$

Unglattheit von f beschränkt die Konvergenzordnung

Die Trapez-Regel auf äquidistante Punkte hat Ordnung 2, aber angewendet auf den 1-periodischen glatten (analytischen) Integrand

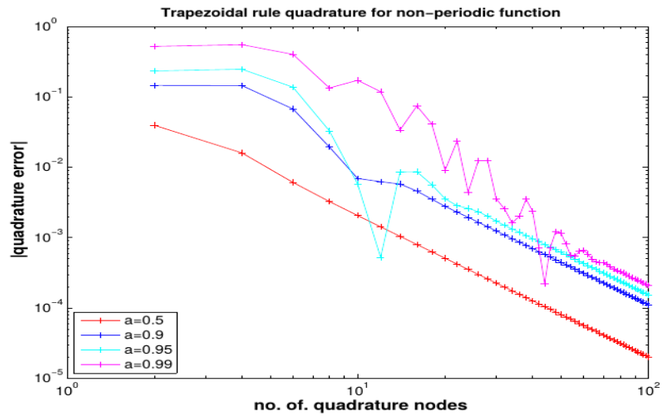
$$f(t) = \frac{1}{\sqrt{1 - a\sin(2\pi t - 1)}} \quad , \quad 0 < a < 1$$

erhalten wir folgende erstaunliche Ergebnisse (als “exakten Wert des Integrals” verwenden wir T_{500}):



Quadratur-Fehler für $T_n(f)$ auf $[0, 1]$

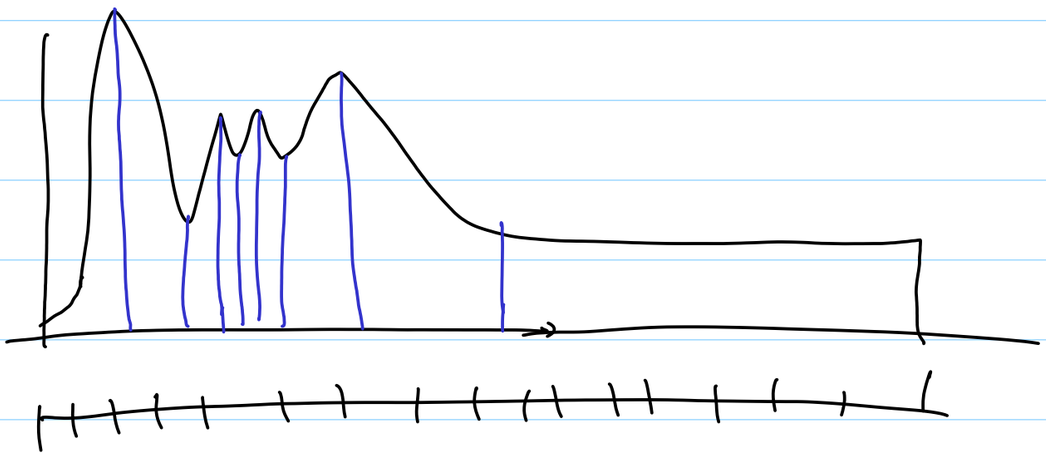
Exponentielle Konvergenz!



Quadratur-Fehler für $T_n(f)$ auf $[0, \frac{1}{2}]$

nur algebraische Konvergenz. . .

§1.4. Adaptive Quadratur



Optimiere: Anzahl Funktionsauswertungen.

Intervallänge, Gewichte, Knoten $[0,1]$

$$\left| \int_a^b f(x)\,dx - \sum_{k=1}^{N} h_k \sum_{j=1}^{s} b_j f(x_{k-1} + c_j h_k) \right| \le$$

$(b-a) \max_{k=1,2,\dots,N} \varepsilon_k$ wobei

$$\varepsilon_k = \left| \int_{x_{k-1}}^{x_k} f(x)\,dx - \sum_{j=1}^{s} b_j f(x_{k-1} + c_j h_k) \right|$$

lokale Fehler auf $[x_{k-1}, x_k]$

Lokaler Fehler ~ Glattheit der Funktion.

$$h^n \max_{x \in [x_{k-1}, x_k]} \left| f^{(n)}(x) \right|$$

es lohnt h kleiner, dann wenn $f^{(n)}$ gross,
sonst nicht.

IDEE: wähle h_k klein nur dort wo $|f^{(n)}(x)|$ gross

Wo?

Wie schätze ich während der Rechnung der lokale Fehler, ohne weiter Informationen über f?

$$\int_{x_{k-1}}^{x_k} f(x)\,dx = Q^T(f, x_{k-1}, x_k) + c\,h^3 \max_{z\in[x_{k-1},x_k]} |f''(z)|$$

$$\int_{x_{k-1}}^{x_k} f(x)\,dx = Q^S(f, x_{k-1}, x_k) + c\,h^5 \max_{z\in[x_{k-1},x_k]} |f^{IV}(z)|$$

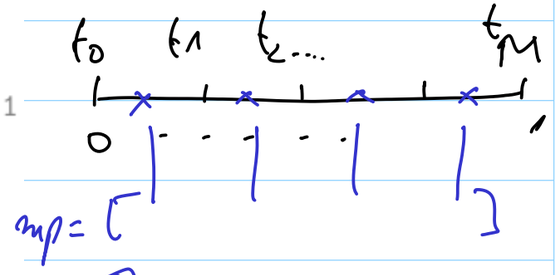
$$\left| \int_{x_{k-1}}^{x_k} f(x)\,dx - Q^T(f, x_{k-1}, x_k) \right| \approx$$

$$z^2\, Q^S(f, x_{k-1}, x_k)$$

$|Q^S - Q^T|$ = Schätzung des lokalen Fehlers

IDEE: Verfeinere dort wo diese Schätzung gross ist!

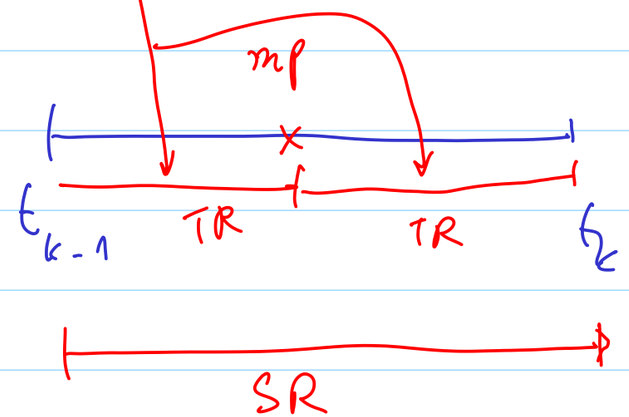
```
if __name__ == '__main__':
    f = lambda x: exp(6*sin(2*pi*x))
    #f = lambda x:  1.0/(1e-4+x*x)
    M = arange(11.)/10 # 0, 0.1, ...  0.9, 1
    rtol = 1e-6; abstol = 1e-10
    I = adaptquad(f,M,rtol,abstol)
    exact,e = integrate.quad(f,M[0],M[-1])
    print 'adaptquad:',I, '"exact":',exact
    print 'error:',abs(I-exact)
```



$[h_1, h_2, \ldots, h_M]$

```
    h = diff(M)                          # compute lengths of mesh intervals
    mp = 0.5*( M[:-1]+M[1:] )            # compute midpoint positions
    fx = f(M); fm = f(mp)                # evaluate function at positions and
midpoints
    trp_loc = h*( fx[:-1]+2*fm+fx[1:] )/4   # local trapezoid rule
    simp_loc= h*( fx[:-1]+4*fm+fx[1:] )/6   # local simpson rule
```

lokal!



```
    I = sum(simp_loc)                       # use simpson rule value as
intermediate approximation for integral value
    est_loc = abs(simp_loc - trp_loc)       # difference of values obtained from
local composite trapezoidal rule and local simpson rule is used as an estimate
for the local quadrature error.
    err_tot  = sum(est_loc)                 # estimate for global error (sum
moduli of local error contributions)
    # if estimated total error not below relative or absolute threshold, refine
mesh
    if err_tot > rtol*abs(I) and err_tot > abstol:
        refcells = nonzero( est_loc > 0.9*sum(est_loc)/size(est_loc) )[0]
        I = adaptquad(f,sort(append(M,mp[refcells])),rtol,abstol)   # add
midpoints of intervalls with large error contributions, recurse.
    return I
```

$loc > \frac{9}{10} \cdot tot \cdot \frac{1}{\#Intervale} \Rightarrow$

↑ array. ↑ Zahl

TT FFTF FTT
1100 1 00 11
(true, false, ...)

nonzero ⇒ indizes, wo das array keine 0 hat

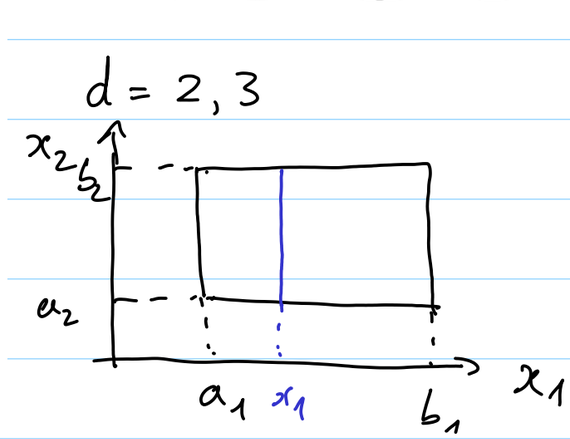
↳ Intervale, die zu verfeinern sind.

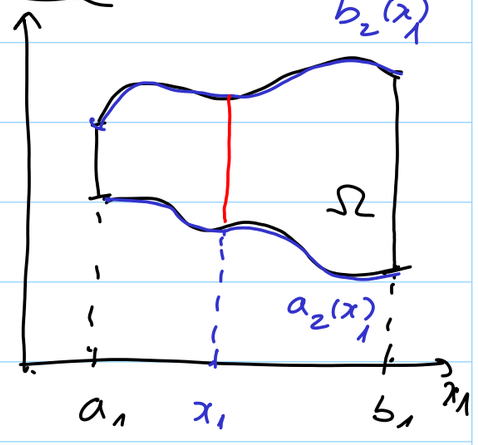
nehme die Mittelpunkte nur davon

append

, sortiere ⇒ die neue Liste M.

§1.5 Quadratur in $\mathbb{R}^d$





$$I = \int_\Omega f(x_2, x_1)\, dx_2\, dx_1 = \int_{a_1}^{b_1} \int_{a_2(x_1)}^{b_2(x_1)} f(x_2, x_1)\, dx_2\, dx_1$$

$$= \int_{a_1}^{b_1} F(x_1)\, dx_1$$

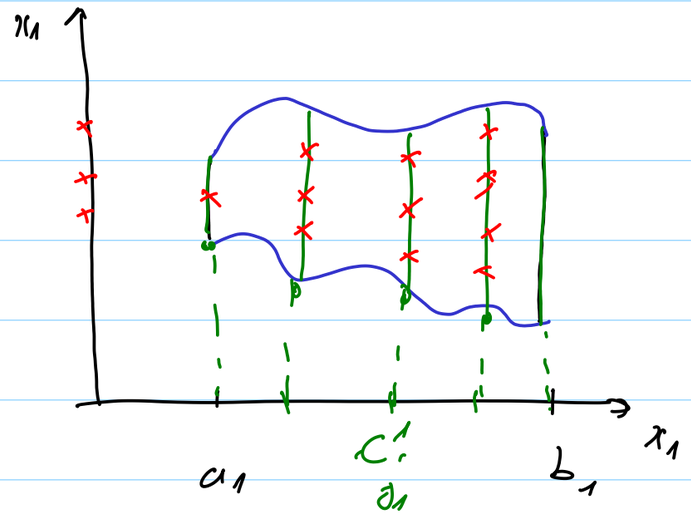
$$\approx \sum_{j_1=1}^{N_1} F(c_{j_1}^1)\, w_{j_1}^1$$

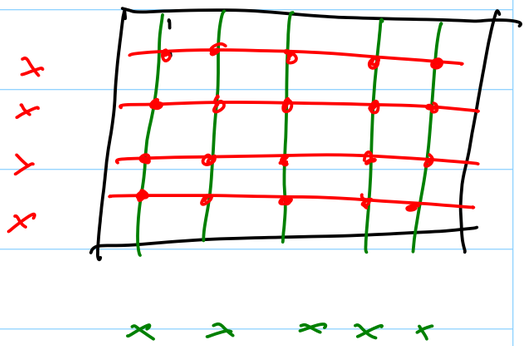
$c_{j_1}^1$: Knoten in Ox_1-Richtung $\in [a_1, b_1]$

$$= \sum_{j_1=1}^{N_1} \int_{a_2(c^1_{j_1})}^{b_2(c^1_{j_1})} f(x_2, c^1_{j_1}) \, dx_2 \cdot w^1_{j_1} \approx$$

$$= \sum_{j_1=1}^{N_1} \sum_{j_2=1}^{N_2} f(c^2_{j_2}, c^1_{j_1}) \, w^2_{j_2} \, w^1_{j_1}$$

$N_1 \cdot N_2$ Auswertungen von f





* Dünne Gitter / sparse grids
* Monte Carlo
* hochoszilatorische Funktionen.

$$\Omega = [0,1]^d \subset \mathbb{R}^d$$

$$\int_{[0,1]^d} f(\underline{x}) \, d\underline{x} \approx \sum_{j_1=1}^{N_1} \sum_{j_2=1}^{N_2} \cdots \sum_{j_d=1}^{N_d} f(c^d_{j_d}, c^{d-1}_{j_{d-1}}, \ldots, c^1_{j_1}) \cdot w^d_{j_d} \, w^{d-1}_{j_{d-1}} \cdots w^1_{j_1}$$

$N_1 \cdot N_2 \cdots N_d$ Auswertungen von f

§1.6. Quadratur mit erhöhter Ordnung

- s Knoten auf Referenzintervall.
- s Gewichte so bestimmt, dass Polynome höchstes Grades exakt mit der QF integriert werden.

Frage: Wenn wir frei sind sowohl Gewichte b_j als auch Knoten c_j, wie hoch kann die Quadraturordnung werden?

c_j äquidistant $\Rightarrow$ s

$2s$ Unbekannte, $2s$ Gleichungen.

$$\begin{cases} \int_0^1 1 \, dt = Q_s(1, 0,1) \\ \int_0^1 t \, dt = Q_s(t, 0,1) \\ \dots \\ \int_0^1 t^{2s-1} dt = Q_s(t^{2s-1}, 0,1) \end{cases}$$

☹

Möchte Ordnung $p = s+m$

jedes Polynom von Grad $\leq s+m-1 = p-1$ soll exakt integriert werden (mit QF)

IDEE: dividire f durch $M(x) = (x-c_1)(x-c_2)\dots(x-c_s)$

$\text{Grad } M = s$

$$f(x) = M(x)\, g(x) + r(x) \quad \text{mit} \quad \text{Grad}(r) \leq s-1$$

$$\int_0^1 f(t)dt = \int_0^1 M(t) g(t) dt + \int_0^1 r(t) dt$$

$$\sum_{j=1}^{s} b_j f(c_j) = \sum_{j=1}^{s} b_j M(c_j) g(c_j) + \sum_{j=1}^{s} b_j r(c_j)$$

$$\Rightarrow \int_0^1 M(t) g(t) dt = 0 \Rightarrow \langle M, g \rangle = 0 \text{ für alle}$$

Polynome g mit $\text{Grad}(g) \le m-1$.

$$\langle M, g\rangle = \int_0^1 M(t)g(t)dt$$ Skalarprodukt auf dem Raum der Polynome

Theorem Ordnung der QF ist $\ge m \iff$ $\langle M, g\rangle = 0$ für alle Polynome g vom Grad $\le m-1$.

$P_m = \text{span}\{1, t, \dots, t^{m-1}\}$: $M \perp P_m$

Theorem Ordnung einer QF ist höchstens $2n$.

Beweis Annahme: $p \ge 2n+1 \Rightarrow$

$\int_0^1 M(t)\, g(t)dt = 0$ für alle Polynome vom Grad $\le n+1$ $\Big\} \Rightarrow$

nehme $g = M$

$$\int_0^1 M(t)\,M(t)dt = 0 \iff \int_0^1 M(t)^2 dt = 0 \Rightarrow$$

$\Rightarrow M(t) \equiv 0$ ↯

$\Rightarrow$ wahr ist das Gegenteil der Annahme:

$p \le 2n$.

Orthogonale Polynome

$w:]a,b[\to \mathbb{R}$ Gewichtfunktion

stetig $w(x) > 0$ für alle $x \in]a,b[$

$$\int_a^b |x|^k w(x)dx < \infty \quad \text{für } k = 0, 1, 2, \dots$$

Betrachte den linearen Raum:

$$V = \left\{ f:]a,b[\to \mathbb{R},\ f \text{ stetig},\ \int_a^b |f(x)|^2 w(x)dx < \infty \right\}.$$

Bem Alle Polynome liegen in V auf V: Skalarprodukt

$$\langle f,g\rangle = \int_a^b f(x)\,g(x)\,w(x)\,dx.$$

T [Gram-Schmidt]:

Es existiert eine eindeutige Folge von Polynomen $p_0, p_1, \dots$ mit $p_k(x) = x^k +$ Polynom vom Grad $\le k-1$ so dass $p_k \perp \mathcal{P}_{k-1}$ bzgl. $\langle\cdot,\cdot\rangle$.

3-Term Rekurenz:

$$p_{k+1}(x) = (x-\beta_{k+1})\,p_k(x) - \gamma_{k+1}^2\,p_{k-1}(x)$$

mit $p_0(x)=1$, $p_{-1}(x)=0$ und

$$\beta_{k+1} = \frac{\langle x p_k, p_k\rangle}{\langle p_k, p_k\rangle}, \quad \gamma_{k+1}^2 = \frac{\langle p_k, p_k\rangle}{\langle p_{k-1}, p_{k-1}\rangle}.$$

Bem $c_1, c_2, \dots, c_s$ sind die Nullstellen von p_s.

Bsp $w(x)\equiv 1 \Rightarrow$ orthogonale Polynome bzgl.

$$\int_0^1 f(x)\,g(x)\,dx \qquad a=0,\ b=1$$

Legendre Polynome

QF: Gauss-Quadratur

$]a,b[=]-\infty,\infty[$ $w(x) = e^{-x^2} \Rightarrow$ Hermite Polynome

QF $\Rightarrow$ Hermite-Quadratur.

Bsp Gauss-Quadratur:

1) $n=1$ auf $[0,1]$, $p_1(x) = x - \frac{1}{2}$; $c_1 = \frac{1}{2}$, $b_1 = 1$

auf $[-1,1]$ $c_j = \frac{1}{2}(1+\gamma_j)$

$c_1 = \frac{1}{2}(1+\gamma_1) \Rightarrow$

$\frac{1}{2} = \frac{1}{2}(1+\gamma_1) \Rightarrow$

$0 = \frac{1}{2}\gamma_1 \Rightarrow \gamma_1 = 0$

Ordnung $2n = 2\cdot 1 = 2$

2) $n=2$ auf $[-1,1]$, $p_2(x) = \frac{2}{2}x^2 - \frac{1}{2} = \frac{3}{2}\left(x^2 - \frac{1}{3}\right)$

$$x_{1,2} = \pm\frac{1}{\sqrt{3}} \; ; \quad c_1 = \frac{1}{2} - \frac{\sqrt{3}}{6}, \quad c_2 = \frac{1}{2} + \frac{\sqrt{3}}{6}$$

Ordnung $2n = 2\cdot 2 = 4$.

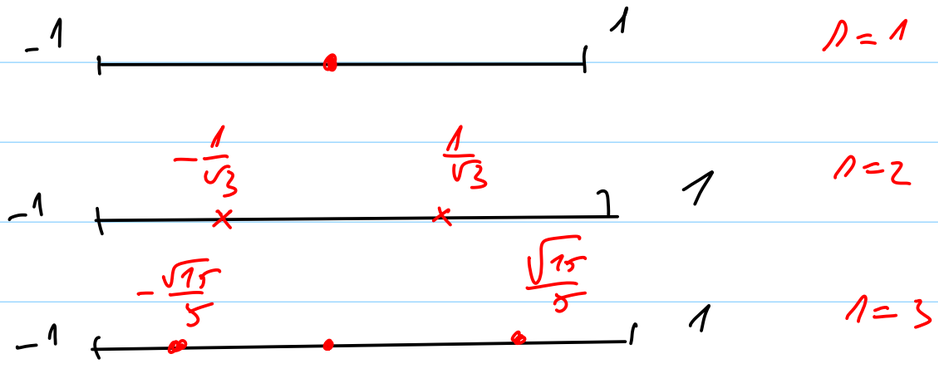
3) $n=3$ auf $[-1,1]$, $p_3(x) = \frac{5}{2}x^3 - \frac{3}{2}x$

$$x_2 = 0, \quad x_{1,3} = \pm\frac{\sqrt{15}}{5} \; | \; b_2 = \frac{8}{18}, \quad b_1 = b_3 = \frac{5}{18}$$

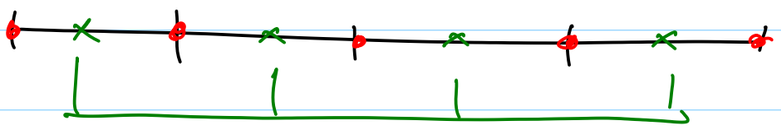
$$c_2 = \frac{1}{2}, \quad c_{1,3} = \frac{1}{2} \pm \frac{\sqrt{15}}{10}$$

Ordnung $2n = 6$

Bem Gewichte der Gauss-QF sind positiv.



Bem Gauss-Knoten sind nicht verschachtelt $\neq$ Trapez / Simpson Nachteil.



Bem Knoten & Gewichte der Gauss-QF berechnet man via Eigenwertberechnung.

3-Term-Rekurenz

$$\begin{cases} p_k(x) = (a_k x + b_k)\, p_{k-1}(x) - c_k\, p_{k-2}(x) \\ p_{-1}(x) = 0, \quad p_0(x) = 1 \end{cases} \Rightarrow$$

$$\begin{cases} x\, p_{k-1}(x) = \boxed{\frac{c_k}{a_k}}\, p_{k-2}(x) \boxed{- \frac{b_k}{a_k}}\, p_{k-1}(x) + \boxed{\frac{1}{a_k}}\, p_k(x) \\ \text{für} \quad k = 1, 2, 3, \ldots, n \end{cases}$$

$$x \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_{k-1} \\ \vdots \\ p_{n-2} \\ p_{n-1} \end{bmatrix} = \begin{bmatrix} -\frac{b_1}{a_1} & \frac{1}{a_1} & & & & \\ \frac{c_2}{a_2} & -\frac{b_2}{a_2} & \frac{1}{a_2} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \frac{c_k}{a_k} & -\frac{b_k}{a_k} & \frac{1}{a_k} & \\ & & & \ddots & \ddots & \frac{1}{a_{n-1}} \\ & & & & \frac{c_n}{a_n} & -\frac{b_n}{a_n} \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_{n-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \frac{1}{a_n} \end{bmatrix} p_n$$

$\underline{p}(x)$ $\qquad$ $\underline{\underline{A}}$ $\qquad$ $\underline{e}_n = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$

3-Term-Rekurenz: $x\,\underline{p}(x) = \underline{\underline{A}}\,\underline{p}(x) + \frac{1}{a_n} p_n(x)\,\underline{e}_n$

Knoten $\equiv$ Nullstellen von $p_n(x)$ $\Rightarrow$ $p_n(t_k) = 0$
$\hookrightarrow$ $t_1, t_2, \dots, t_n$ für $k = 1, 2, \dots, n$.

$\Rightarrow$ $t_k \underline{p}(t_k) = \underline{\underline{A}}\,\underline{p}(t_k) + 0$ $\Rightarrow$

t_k EW der Matrix $\underline{\underline{A}}$
Eigenwerte.
$\underline{p}(t_k)$ ist EV der Matrix $\underline{\underline{A}}$
Eigenvektor.

Bem Es gibt ziemlich gute numerische Verfahren für die Berechnung von Approximationen an EW, EV

$\Rightarrow$

$$\text{eig}(\underline{\underline{A}}) \Rightarrow \text{knoten}.$$

Gauss: Gewichte $2\,\underline{p}(t_0)$.

Bem 1) Gauss nicht verschachtelt, nicht immer anwendbar
2) a, b sind keine Knoten $\Rightarrow$ Gauss-QF ist offen.
3) Manchmal braucht man a oder b oder beide als knoten, möchte trotzdem QF höchster Ordnung $\Rightarrow$ $2n-1$, $2n-2$
$\Rightarrow$ Selbe Prozedur aber mit 2 feste Knoten $\Rightarrow$.

$[-1,1]$, $\omega(x) = (1-x)(1+x)$, $x_r = -1$ oder $x_l = 1$

$\Rightarrow$ Radau- Quadratur.

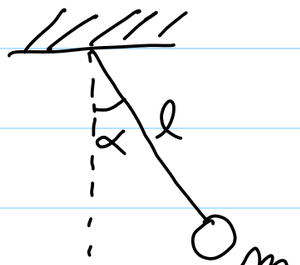
Beide $-1, 1$ Knoten $\Rightarrow$ Lobatto- Quadratur.

§2. Einfache Verfahren für ODEs 1. Ordnung

ODE = ordinary differential equation
gewöhnliche Differentialgleichungen

§2.1 Linearisierung: global und lokal

$$m l \ddot{\alpha}(t) = -mg \sin\alpha(t)$$



$$(0)\quad \begin{cases} \ddot{\alpha}(t) = -\frac{g}{l}\sin\alpha(t) \\ \alpha(0) = \alpha_0 \\ \dot{\alpha}(0) = \dot{\alpha}_0 \end{cases}$$

ODE 2. Ordnung da $\ddot{\alpha}$

autonom: t erscheint nicht explizit

$$\ddot{y}(t) = f(y(t))$$

$$\dot{\alpha}(t) = \frac{d}{dt}\alpha(t), \qquad \ddot{\alpha}(t) = \frac{d^2}{dt^2}\alpha(t)$$

Die meiste Software ist für ODE 1. Ordnung

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

"global": linearisiere die rechte Seite:

$$\sin\alpha = \alpha - \frac{1}{3!}\alpha^3 + \dots = \alpha + O(\alpha^3)$$

für kleines α!

Neues Model.

$$(1)\quad \begin{cases} \ddot{\beta}(t) = -\frac{g}{l}\beta(t) \\ \beta(0) = \alpha_0 \\ \dot{\beta}(0) = \dot{\alpha}_0 \end{cases}$$

$$\omega^2 = \frac{g}{l}$$

exakte Lösung

$$\beta(t) = \frac{\dot{\alpha}_0}{\omega}\sin(\omega t) + \alpha_0\cos(\omega t).$$

Bsp Trick: Reduktion der Ordnung der ODE:

Notiere $p(t) = \dot{\alpha}(t)$

$$\dot{p}(t) = \ddot{\alpha}(t) = -\frac{g}{\ell}\sin\alpha(t)$$

(0) $\Leftrightarrow$ System ODEs 1. Ordnung:

$$\begin{cases} \dot{\alpha} = p \\ \dot{p} = -\frac{g}{\ell}\sin\alpha \\ \alpha(0) = \alpha_0 \\ p(0) = \dot{\alpha}_0 \end{cases} \qquad \underline{y} = \begin{bmatrix} \alpha \\ p \end{bmatrix}; \ \underline{y}(t) = \begin{bmatrix} \alpha(t) \\ p(t) \end{bmatrix} \in \mathbb{R}^2$$

$$\underline{f}(\underline{y}) = \begin{bmatrix} y_2 \\ -\frac{g}{\ell}\sin y_1 \end{bmatrix}$$

(0) $\Leftrightarrow$ (2) $\begin{cases} \dot{\underline{y}} = \underline{f}(\underline{y}) \\ \underline{y}(0) = \begin{bmatrix} \alpha_0 \\ \dot{\alpha}_0 \end{bmatrix} \end{cases}$

$\underline{y} : [0,T] \to \mathbb{R}^2$

Unbekannte Funktion.

ODEs 1. Ordnung

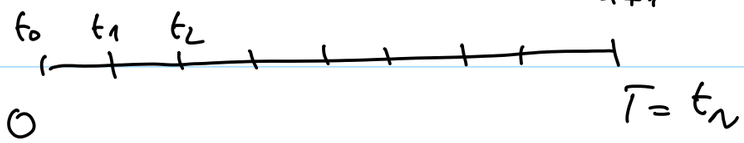
autonom

Ziel: Möchte Approximation von

$\underline{y}(T)$ = exakte Lösung zur Endzeit T

Zeitgitter

$$0 = t_0 < t_1 < t_2 < \dots < t_n < t_{n+1} < \dots < t_N = T$$



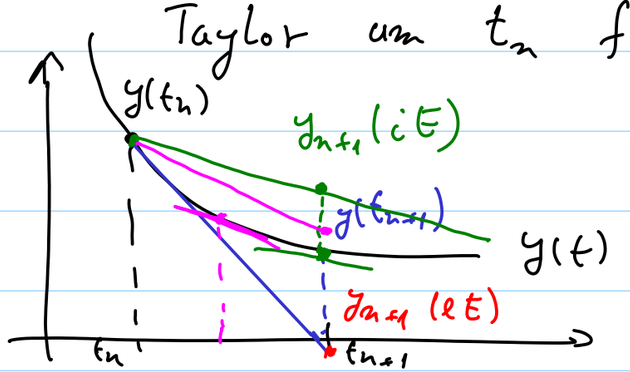
Zeitschritt $h_n = t_{n+1} - t_n$

Wir werden Approximationen bauen

$\underline{y}_n \approx \underline{y}(t_n)$

IDEE: Lokal: Linearisiere die Lösung lokal:

Taylor um t_n für $\underline{y}(t)$



$$\underset{\text{"}}{y}(t_n+h) = \underset{\text{"}}{y}(t_n) + h\,\underset{\text{"}}{\dot{y}}(t_n) + O(h^2)$$

$\underset{\text{"}}{y}_{n+1}$ $\quad$ $\underset{\text{"}}{y}_n$ $\quad$ $\underset{\text{"}}{f}(t_n, \underset{\text{"}}{y}_n)$ $\quad$ (z)

Methode: $\underset{\text{"}}{y}_{n+1} = \underset{\text{"}}{y}_n + h\,\underset{\text{"}}{f}(t_n, \underset{\text{"}}{y}_n)$ $\quad$ $(e\in)$

expliziter Euler

mit lokalen Fehler $O(h^2)$

Andere Herleitung

$$\underset{\text{"}}{f}(t_n, \underset{\text{"}}{y}(t_n)) = \underset{\text{"}}{\dot{y}}(t_n) = \lim_{h\to 0} \frac{\underset{\text{"}}{y}(t_n+h) - \underset{\text{"}}{y}(t_n)}{h} \approx$$

$$\approx \frac{\underset{\text{"}}{y}_{n+1} - \underset{\text{"}}{y}_n}{h}$$

$$\Rightarrow \underset{\text{"}}{f}(t_n, \underset{\text{"}}{y}(t_n)) = \frac{\underset{\text{"}}{y}_{n+1} - \underset{\text{"}}{y}_n}{h} \Rightarrow$$

$$y_{n+1} = y_n + h\,\underset{\text{"}}{f}(t_n, \underset{\text{"}}{y}(t_n)) \qquad (e\in)$$

$$(e\in): \begin{cases} \underset{\text{"}}{y}_0 = \underset{\text{"}}{y}(0) \\ \underset{\text{"}}{y}_{n+1} = \underset{\text{"}}{y}_n + h\,\underset{\text{"}}{f}(t_n, \underset{\text{"}}{y}_n) \quad \text{für } n=0,1,2,\dots,N-1 \end{cases}$$

Taylor um t_{n+1} für $\underset{\text{"}}{y}(t)$:

$$\underset{\text{"}}{y}(t_{n+1}-h) = \underset{\text{"}}{y}(t_{n+1}) - h\,\underset{\text{"}}{\dot{y}}(t_{n+1}) + O(h^2)$$

$\underset{\text{"}}{y}_n$ $\quad$ $\underset{\text{"}}{y}_{n+1}$ $\quad$ $f(t_{n+1}, \underset{\text{"}}{y}_{n+1})$

$$\Rightarrow \boxed{\underset{\text{"}}{y}_n = \underset{\text{"}}{y}_{n+1} - h f(t_{n+1}, \underset{\text{"}}{y}_{n+1})} \qquad (i\in)$$

Lokaler Fehler $O(h^2)$ $\quad$ impliziter Euler

Bem $f(t,y) = -\frac{g}{l}\sin y \Rightarrow y_n = y_{n+1} - h\left(-\frac{g}{l}\right)\sin y_{n+1}$

↳ algebraisches Problem (Nullstelle!)

Neue Idee: linearisiere in der Mitte des Intervals:

Taylor um $t^* = \frac{1}{2}(t_n + t_{n+1}) = t_n + \frac{1}{2}h$ für $\underline{y}$:

$$\underline{y}(t_{n+1}) = \underline{y}(t^*) + \frac{h}{2}\,\dot{\underline{y}}(t^*) + \frac{1}{2}\left(\frac{h}{2}\right)^2 \ddot{\underline{y}}(t^*) + O\left(\left(\frac{h}{2}\right)^3\right)$$

$$\underline{y}(t_n) = \underline{y}(t^*) - \frac{h}{2}\,\dot{\underline{y}}(t^*) + \frac{1}{2}\left(\frac{h}{2}\right)^2 \ddot{\underline{y}}(t^*) - O\left(\left(\frac{h}{2}\right)^3\right)$$

$\ominus$

$$\Rightarrow \underline{y}(t_{n+1}) - \underline{y}(t_n) = h\,\dot{\underline{y}}(t^*) + O(h^3)$$

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

$$\Rightarrow \underline{y}(t_{n+1}) = \underline{y}(t_n) + h\underline{f}(t^*, \underline{y}(t^*)) + O(h^3)$$

$\underline{y}_{n+1} \quad \underline{y}_n \quad ?$

Bem $\underline{y}(t^*)$ stört, brauche noch ein Trick!

1)
$$\underline{y}(t^*) \approx \frac{1}{2}\left(\underline{y}(t_n) + \underline{y}(t_{n+1})\right) \Rightarrow$$

$$\underline{y}_{n+1} = \underline{y}_n + h\,\underline{f}\left(t_n + \frac{h}{2},\ \frac{1}{2}(\underline{y}_n + \underline{y}_{n+1})\right) \quad \text{(iMP)}$$

implizite Mittelpunktsregel.
lokaler Fehler $O(h^3)$

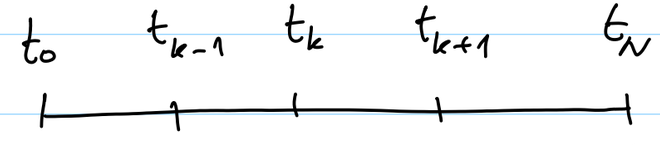
2)
$$\underline{y}_{n+1} = \underline{y}_n + h\,\frac{1}{2}\left(\underline{f}(t_n, \underline{y}_n) + \underline{f}(t_{n+1}, \underline{y}_{n+1})\right) \quad \text{(iTR)}$$

implizite Trapezregel, lokaler Fehler $O(h^3)$.

Bem Es geht so nur wenn die exakte Lösung $y(t)$ genügend glatt ist, d.h. $\ddot{\underline{y}}, \dddot{\underline{y}}, \ldots$ existieren!

§ 2.2. Störmer-Verlet Verfahren

$$\begin{cases} \ddot{y} = f(t, y) \\ y(t_0) = y_0 \\ \dot{y}(t_0) = v_0 \end{cases}$$



Bemerkung Addieren wir die 2 Gleichungen $\Rightarrow$

$$y(t_{n+1}) + y(t_n) = 2y(t^*) + \left(\frac{h}{2}\right)^2 \ddot{y}(t^*) + O(h^4) \Leftrightarrow$$

$$\Rightarrow \ddot{y}(t^*) = \frac{y(t_{n+1}) - 2y(t_n + \frac{h}{2}) + y(t_n)}{\left(\frac{h}{2}\right)^2} + O(h^2)$$

$$\Rightarrow \ddot{y}(t_n + \frac{h}{2}) \approx \frac{y_{n+1} - 2y_{n+\frac{1}{2}} + y_n}{\left(\frac{h}{2}\right)^2} \quad \text{mit}$$

lokaler Fehler $O(h^2)$

Bemerkung Selbe Rechnung um t_n $\Rightarrow$

$$\ddot{y}(t_n) \approx \frac{y_{n+1} - 2y_n + y_{n-1}}{h^2} \quad \text{mit lokalem Fehler } O(h^2)$$

$$f(t_k, y(t_k)) = \ddot{y}(t_k) \approx \frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} \Rightarrow$$

$$\boxed{y_{k+1} = -y_{k-1} + 2y_k + h^2 f(t_k, y_k)} \quad (St.V)$$

expliziter, 2-Schritt-Verfahren!

$\Rightarrow$ brauche Startwerte: $y(t_0) = y_0$

$y_1 \approx y(t_1)$

eine Möglichkeit: $\underline{y}_1 := \underline{y}_0 + h\,\underline{f}(t_0, \underline{y}_0)$ (e∈) ∴

Taylor: $\underline{y}_1 \approx \underline{y}(t_1) = \underline{y}(t_0) + h\,\dot{\underline{y}}(t_0) + \frac{h^2}{2}\,\ddot{\underline{y}}(t_0) + O(h^3)$

$\underline{y}(t_0) = \underline{y}_0$, $\dot{\underline{y}}(t_0) = \underline{v}_0$, $\ddot{\underline{y}}(t_0) = \underline{f}(t_0, \underline{y}_0)$

$$\underline{y}_1 := \underline{y}_0 + h\,\underline{v}_0 + \frac{h^2}{2}\,\underline{f}(t_0, \underline{y}_0)$$

für $k = 1, 2, 3, \ldots$

$$\underline{y}_{k+1} = -\underline{y}_{k-1} + 2\underline{y}_k + h^2 \underline{f}(t_k, \underline{y}_k)$$

lokal $O(h^3)$, global $O(h^2)$.

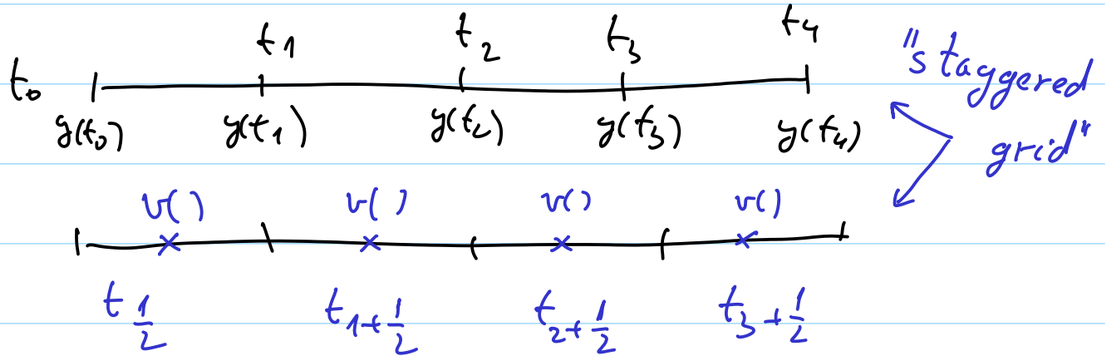
Notiere $\underline{v}_{k+\frac{1}{2}} = \dfrac{\underline{y}_{k+1} - \underline{y}_k}{h}$ entspricht $v(t_k + \frac{1}{2}h)$

(gute Approximation an $\dot{y}(t_k + \frac{1}{2}h)$)

in (St.V)

$$\begin{cases} \underline{v}_{k+\frac{1}{2}} = \underline{v}_{k-\frac{1}{2}} + h\,\underline{f}(t_k, \underline{y}_k) \\ \underline{y}_{k+1} = \underline{y}_k + h\,\underline{v}_{k+\frac{1}{2}} \end{cases}$$

"Leap-frog"



Besser noch: "velocity-Verlet"-Verfahren:

$$\begin{cases} \underline{y}_{k+1} = \underline{y}_k + h\,\underline{v}_k + \frac{h^2}{2}\,\underline{f}(t_k, \underline{y}_k) \\ \underline{v}_{k+1} = \underline{v}_k + h\,\frac{1}{2}\left(\underline{f}(t_k, \underline{y}_k) + \underline{f}(t_{k+1}, \underline{y}_{k+1})\right) \end{cases}$$

Begründung: Notiere $v_k = \frac{y_{k+1} - y_{k-1}}{2h}$ (StV)

$$v_k = \frac{-2y_{k-1} + 2y_k + h^2 f(t_k, y_k)}{2h} = \frac{y_k - y_{k-1}}{h} + \frac{h}{2} f(t_k, y_k)$$

Somit

$$v_{k+1} + v_k = \frac{y_{k+1} - y_k}{h} + \frac{h}{2} f(t_{k+1}, y_{k+1}) +$$

$$+ \frac{y_k - y_{k-1}}{h} + \frac{h}{2} f(t_k, y_k) =$$

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

$$\Rightarrow \quad v_{k+1} = v_k + \frac{h}{2}\left(f(t_{k+1}, y_{k+1}) + f(t_k, y_k)\right)$$

Vorteile: explizit, genauer als (eE), (iE), erhalten die Energie!

§2.3. Fehlerschätzung und Konvergenz

Theorem Taylor mit Rest als Integral:

$$f(x) = f(a) + \frac{x-a}{1!} f'(a) + \dots \frac{(x-a)^{n-1}}{(n-1)!} f^{(n-1)}(a) +$$

$$+ \int_a^x \frac{(x-t)^{n-1}}{(n-1)!} f^{(n)}(t)\, dt$$

Sei y exakte Lösung von ODE

$$\dot{y} = f(t, y) \quad \text{mit} \quad y(t_0) = y_0$$

y_n Approximation an $y(t_n)$, $t_n = t_0 + nh$

durch (eE) definiert

$f: [t_0, T] \times \mathbb{R}^d \to \mathbb{R}^d$ stetig differentierbar und Lipschitz:

$$\| \underline{f}(t, \underline{y}) - \underline{f}(t, \underline{z}) \| \le L \, \| \underline{y} - \underline{z} \|$$

L ↳ Konstante $\in \mathbb{R}$.

für alle $\underline{y}, \underline{z}, t$.

Theorem $\| \underline{y}_n - \underline{y}(t_n) \| \le M \cdot h$ für alle n, wobei

$$M = \frac{1}{L} \left(e^{L(T-t_0)} - 1 \right) \cdot \frac{1}{2} \max_{t \in [t_0, T]} \| \ddot{\underline{y}}(t) \|$$

Beweis 3 Schritte:

1) Lokaler Fehler: ein Schritt (eE) mit Start $\underline{y}(t_n)$

$$\underline{y}(t_{n+1}) - \underline{y}_{n+1} = \underline{y}(t_{n+1}) - \left(\underline{y}(t_n) + h \underline{f}(t_n, \underline{y}(t_n)) \right) =$$

$\underline{f}(t_n, \underline{y}(t_n)) = \dot{\underline{y}}(t_n)$ ← ODE

$$= \underline{y}(t_{n+1}) - \underline{y}(t_n) - h \dot{\underline{y}}(t_n)$$

Satz von Taylor mit Rest als Integral

$(a = t_n, \; x = t_n + h = t_{n+1}, \; f = \underline{y}) \Rightarrow$

$$\underline{y}(t_{n+1}) = \underline{y}(t_n) + h \dot{\underline{y}}(t_n) + \int_{t_n}^{t_{n+1}} (t_n + h - t)^1 \ddot{\underline{y}}(t) \, dt$$

$t = t_n + h\theta \qquad dt = h \, d\theta$

$$\Rightarrow \underline{y}(t_{n+1}) - \underline{y}(t_n) - h \dot{\underline{y}}(t_n) = h^2 \int_0^1 (1-\theta) \ddot{\underline{y}}(t_n + h\theta) \, d\theta$$

$\le \max_{[t_0, T]} \| \ddot{\underline{y}}(t) \|$

Somit:

$$\| \underline{y}(t_{n+1}) - \underline{y}_{n+1} \| \le \frac{1}{2} h^2 \cdot \max_{t \in [t_0, T]} \| \ddot{\underline{y}}(t) \| = c \cdot h^2$$

2) Fehlerfortpflanzung:

(eE) mit Startwert z_n: $z_{n+1} = z_n + h f(t_n, z_n)$

(eE) w_n: $w_{n+1} = w_n + h f(t_n, w_n)$

$$\Rightarrow \|z_{n+1} - w_{n+1}\| \le \|z_n - w_n\| + h \|f(t_n, z_n) - f(t_n, w_n)\| \le$$

$$\le \|z_n - w_n\| + hL \|z_n - w_n\| =$$

$$(1+hL) \|z_n - w_n\|$$

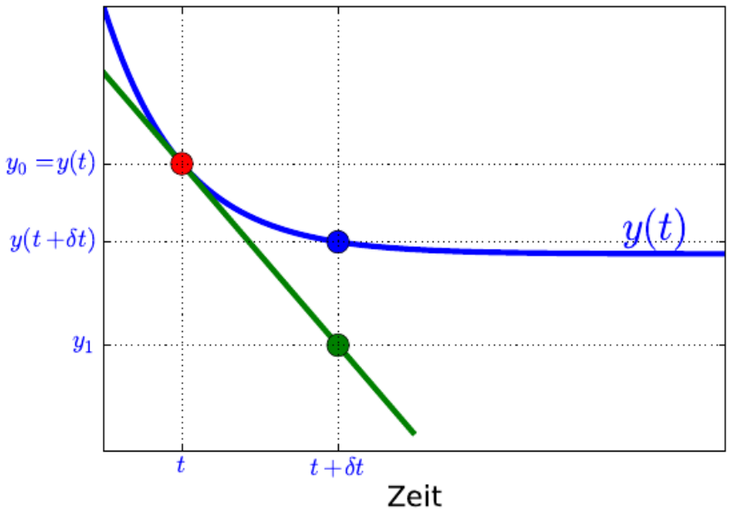
3) Fehlerakumulierung

$$\|y_n - y(t_n)\| \le ch^2 + ch^2(1+hL) + ch^2(1+hL)^2 + \dots +$$

$$+ ch^2(1+hL)^{n-1} =$$

$$= ch^2 \frac{(1+hL)^n - 1}{1+hL-1} = ch \frac{(1+hL)^n - 1}{L} \le$$

$$\le c\frac{h}{L}\left(e^{nhL} - 1\right) = ch \frac{e^{(t_n - t_0)L} - 1}{L}$$ q.ed.



$$\mathbf{y}_{n+1} = \mathbf{y}_n + h_n \mathbf{f}(t_n, \mathbf{y}_n), \quad n = 0, \ldots, N-1.$$

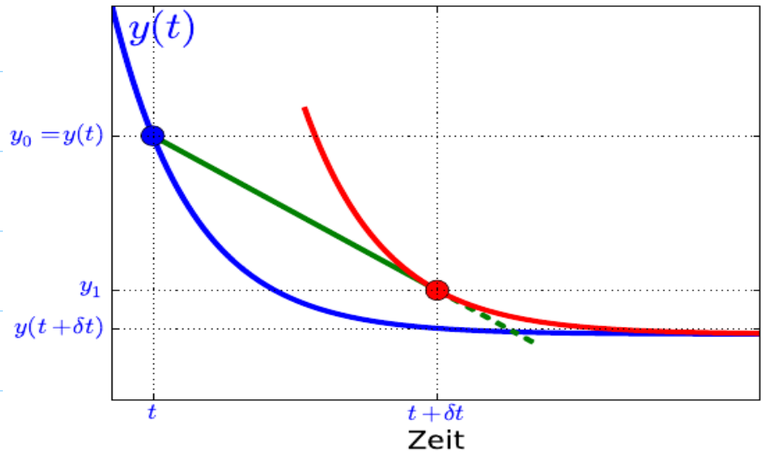
$$\mathbf{f}(t_n, \mathbf{y}(t_n)) = \dot{\mathbf{y}}(t_n) \approx \frac{\mathbf{y}(t_n + h_n) - \mathbf{y}(t_n)}{h_n}$$

$O(h^2)$ lokal

$\Downarrow$

$O(h)$

(eE)



$$\mathbf{y}_{n+1} = \mathbf{y}_n + h_n \mathbf{f}(t_{n+1}, \mathbf{y}_{n+1}), \quad n = 0, \ldots, N-1$$

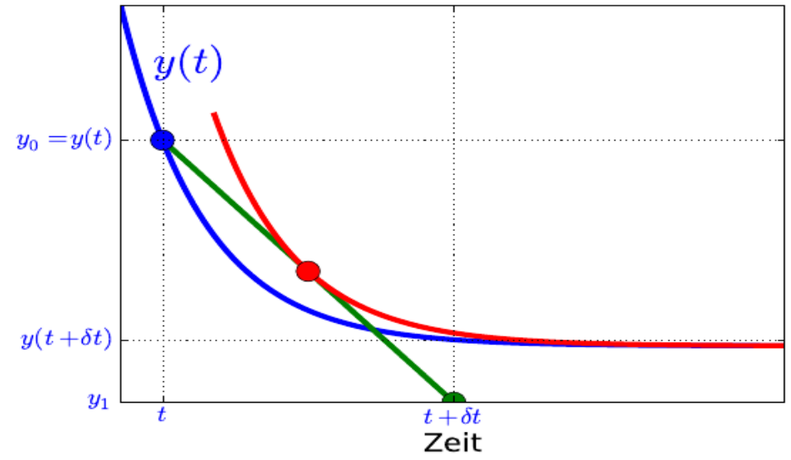
$$\mathbf{f}(t_{n+1}, \mathbf{y}(t_{n+1})) = \dot{\mathbf{y}}(t_{n+1}) \approx \frac{\mathbf{y}(t_{n+1} - h_n) - \mathbf{y}(t_{n+1})}{-h_n}$$

$O(h^2)$ lokal

$\Downarrow$

$O(h)$

(iE)



$$\mathbf{y}_{n+1} = \mathbf{y}_n + h_n \mathbf{f}(\frac{1}{2}(t_n + t_{n+1}), \frac{1}{2}(\mathbf{y}_n + \mathbf{y}_{n+1})) .$$

$$\mathbf{f}(\frac{1}{2}(t_n + t_{n+1}), \frac{1}{2}(\mathbf{y}_n + \mathbf{y}_{n+1})) = \dot{\mathbf{y}}(\frac{1}{2}(t_n + t_{n+1})) \approx \frac{\mathbf{y}(t_n + h_n) - \mathbf{y}(t_n)}{h_n} .$$

$O(h^3)$ lokal (IMP)

$\Downarrow$

$O(h^2)$ global.

(iMP) lokaler Fehler: $\bar{y} = \frac{1}{2}(y_n + y_{n+1})$, $t^* = \frac{1}{2}(t_n + t_{n+1})$

$$y(t_{n+1}) - y(t_n) - h f(t^*, \bar{y}) \overset{\text{Taylor}}{=}$$

$$= h\dot{y}(t^*) + O(h^3) - h f(t^*, \bar{y}) \overset{\text{ODE}}{=}$$

$$= h f(t^*, y^*) - h f(t^*, \bar{y}) + O(h^3) =$$

$$= h\left(f(t^*, y^*) - f(t^*, \bar{y})\right) + O(h^3)$$

$$\| f(t^*, y^*) - f(t^*, \bar{y}) \| \le L \| y^* - \bar{y} \|$$

$$y^* - \bar{y} = \frac{1}{2} y(t^*) + \frac{1}{2} y(t^*) - \frac{1}{2} y(t_n) - \frac{1}{2} y(t_{n+1}) =$$

$$= \frac{1}{2}\left(y(t^*) - y(t_n)\right) + \frac{1}{2}\left(y(t^*) - y(t_{n+1})\right)$$

$$\overset{\text{Taylor}}{=} \frac{h}{2}\dot{y}(t^*) + O(h^2) - \frac{h}{2}\dot{y}(t^*) + O(h^2) = O(h^2)$$

Somit:

$$\| f(t^*, y^*) - f(t^*, \bar{y}) \| \le L \cdot c \cdot h^2 \implies$$

$$\| y(t_{k+1}) - y_{k+1} \| \le h \cdot L \cdot c \cdot h^2 + O(h^3) = O(h^3).$$

(St.V.) Lokaler Fehler:

$$y(t_{n+1}) - \left(-y(t_{n-1}) + 2y(t_n) + h^2 f(t_n, y(t_n))\right) =$$

$$= y(t_{n+1}) + y(t_{n-1}) - 2y(t_n) - h^2 f(t_n, y(t_n)) =$$

$$\overset{\text{Taylor}}{=} \underbrace{h^2 \ddot{y}(t_n) - h^2 f(t_n, y(t_n))}_{\| \; \leftarrow \text{ODE}} + O(h^4) \implies$$

$$0$$

$\Rightarrow$ lokal $O(h^4)$ $\not\Rightarrow$ global $O(h^3)$

Startwert ist nur $O(h^3)$

ausserdem Fehler fortpflanzung $\Rightarrow$ $O(h^2)$.

§2.4. Vorgehensweise bei Implementierung

Pendelgleichung:

$$\begin{cases} \ddot{\alpha} = -\frac{g}{l}\sin\alpha(t) \\ \alpha(0) = \alpha_0 \\ \dot{\alpha}(0) = \dot{\alpha}_0 \end{cases}$$

Ziel: Approximation der Lösung & Energien mittels (mA), (eE), (iE), (CNP), (StV)

1. Schritt: Umschreiben in ODE 1. Ordnung:

$$p = \dot{\alpha}, \quad \underline{y} = \begin{bmatrix} \alpha \\ p \end{bmatrix}, \quad \underline{f}(\underline{y}) = \begin{bmatrix} p \\ -\frac{g}{l}\sin\alpha \end{bmatrix},$$

2. Schritt: Idee der Lösung.

$$\dot{\underline{y}} = \underline{f}(\underline{y}) \qquad y_n \approx y(t_n) \text{ für } t_n = 0 + nh$$

$\omega^2 = g/l$

(mA): $y_n = \frac{\dot{\alpha}}{\omega}\sin(\omega t_n) + \alpha_0 \cos(\omega t_n)$

(eE) $\underline{y}_{n+1} = \underline{y}_n + h\underline{f}(\underline{y}_n)$

(iE) $\underline{y}_{n+1} = \underline{y}_n + h\underline{f}(\underline{y}_{n+1})$

$y_0 = [\alpha_0, \dot{\alpha}_0]$, T, g, l gegeben, $\omega^2 = \frac{g}{l}$

↳ array

Rechte Seite: (input variable)

```
def f(y):
    result = zeros(2)
    result[0] = y[1]
    result[1] = -g/l * sin y[0]
    return result
```

(eE) N Zeitschritte.

$h = T/N$

$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \vdots & \end{bmatrix}$

$y =$?

$$\mathbf{f}(\frac{1}{2}(t_n + t_{n+1}), \frac{1}{2}(\mathbf{y}_n + \mathbf{y}_{n+1})) = \dot{\mathbf{y}}(\frac{1}{2}(t_n + t_{n+1})) \approx \frac{\mathbf{y}(t_n + h_n) - \mathbf{y}(t_n)}{h_n}.$$

$y[0]$ ..

für $n = 0, 1, \ldots, N-1$

$y[n+1] = y[n] + h\, f(y[n])$

⋮ ⋮ ⋮

plot(y[0,:], y[1,:])

$g(t)$

(nA) $\frac{d}{dt}\left(\frac{\dot{\alpha}}{\omega} \sin(\omega t) + \alpha_0 \cos(\omega t) \right) =$

$= \dot{\alpha} \cos(\omega t) - \alpha_0 \omega \sin(\omega t) = dg(t)$

$y[0] = y_0$

für $n = 0, 1, \ldots, N-1$

$y[n+1, 0] = g(y[n])$

$y[n+1, 1] = dg(y[n])$

def potE(y): $-g\cos(\alpha)$ $m = 1$, $l = 1$

return $(-g \cos y[0])$

def linE(y)

return $\frac{1}{2} y[1]^2$

def totE(y)

return potE(E) + linE(E)

(iE) für $n = 0, 1, 2, \ldots, N-1$

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

erste Möglichkeit: allgemein:

$\underline{z} = \underline{a} + h \underline{f}(\underline{z}) \iff$

$\underline{z} - \underline{a} + h \underline{f}(\underline{z}) = 0 \iff \underline{F}(\underline{z}) = 0$

mit $\underline{F}: \mathbb{R}^d \to \mathbb{R}^d$, $\underline{F}(\underline{z}) = \underline{z} - \underline{a} - h\,\underline{f}(\underline{z})$

Finde Nullselle(n) von $\underline{F}$

scipy.optimise.fsolve:

fsolve($\underline{F}$, $\underline{z}_0$)

↳ Startwert; Hinweis wo fsolve suchen soll ($\underline{z}_0$ sollte nah an der Nullstelle sein)

Startwert?

$\underline{z}_0 = \underline{y}_n$

$\underline{z}_0 :=$ was (eE) in diesem Schritt vorschlägt

$\underline{z}_0 = \underline{y}_n + h\underline{f}(\underline{y}_n)$

für $n = 0, 1, \dots, a-1$:

$\underline{z}_0 = \underline{y}_n + h\underline{f}(\underline{y}_n)$

$\underline{y}_{n+1} =$ fsolve($\underline{F}$, $\underline{z}_0$)

zweite Möglichkeit: nutze dass für den Pendel f einfach ist $\Rightarrow$

schreibe auf Papier das alg. Problem komponentenweise:

$$\begin{cases} z_1 - a_1 - h z_2 = 0 \\ z_2 - a_2 + h\frac{g}{\ell}\sin z_1 = 0 \Rightarrow z_2 = a_2 - h\frac{g}{\ell}\sin z_1 \end{cases}$$

$\Rightarrow \quad z_1 - a_1 - h a_2 + h^2 \frac{g}{\ell}\sin z_1 = 0$

$G(z) = z - a_1 - h a_2 + h^2 \frac{g}{\ell}\sin z$

↳ einfacher billiger!

y[n+1, 1] = fsolve(G, z_0)

§3 Strukturerhaltung

§3.1. Invariante und Hamilton'Systeme

Bsp

autonome Lotka-Volterra : $d=2$

$$\begin{cases} \dot u = (\alpha - \beta v)u \\ \dot v = (\delta u - \gamma)v \end{cases}$$

Konstanten $\alpha, \beta, \gamma, \delta > 0$

$u, v : \mathbb{R} \to \mathbb{R}$ Unbekannt

$$\dot u = (\alpha - \beta v)u = \left(\frac{\alpha}{v} - \beta\right) uv \quad \Big| \left(\delta - \frac{\gamma}{u}\right)$$
$$\dot v = (\delta u - \gamma)v = \left(\delta - \frac{\gamma}{u}\right) uv \quad \Big| \left(\frac{\alpha}{v} - \beta\right)$$
$$\Rightarrow$$

$$\Rightarrow \left(\delta - \frac{\gamma}{u}\right)\dot u = \left(\frac{\alpha}{v} - \beta\right)\dot v \Rightarrow$$

$$\frac{d}{dt}\underbrace{\left(\delta u - \gamma \log u - \alpha \log v + \beta v\right)}_{I(u(t), v(t))} = 0 \quad \text{für alle } t \Rightarrow$$

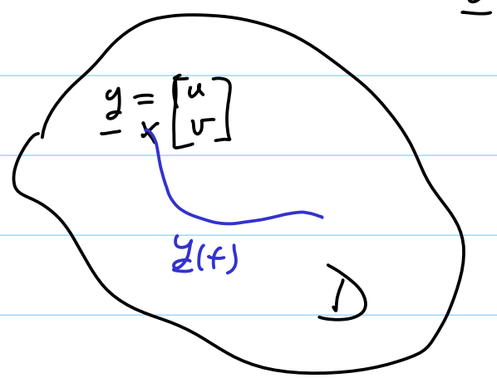
$$\frac{d}{dt}\Big(I(u(t), v(t))\Big) = 0 \quad \text{für alle } t \Rightarrow$$

$$\Rightarrow \quad I(u(t), v(t)) = \text{konstant}$$

Def I heisst erstes Integral / Invariante der ODE $\dot{\underline y} = \underline f(t, \underline y)$ wenn

$I(\underline y(t)) = \text{konstant}$ für jede Lösung $\underline y = \underline y(t)$ der ODE.

$$\underline I : \underline D \subset \mathbb{R}^d \to \mathbb{R}$$
$$\uparrow \underline y(t)$$



$\underline y : [0, T] \to \mathbb{R}^2$

$\underline y(t) \in D \subset \mathbb{R}^2$

Erinnerung $f: \mathbb{R}^d \to \mathbb{R}$, $f(\underline{x}) \in \mathbb{R}$.

partielle Ableitungen $\frac{\partial}{\partial x_1} f(\underline{x})$ = Ableitung von f nach der Variable x_1

$\frac{\partial}{\partial x_2}, \dots, \frac{\partial}{\partial x_d}$

$$\operatorname{grad} f(\underline{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(\underline{x}) \\ \vdots \\ \frac{\partial f}{\partial x_d}(\underline{x}) \end{bmatrix}$$

Bsp $f: \mathbb{R}^d \to \mathbb{R}$, $f(\underline{x}) = \|\underline{x}\|_2 = \left(\sum_{j=1}^{d} x_j^2 \right)^{1/2}$

Was ist $\operatorname{grad} f(\underline{x}) = ?$

$$\frac{\partial f}{\partial x_1}(\underline{x}) = \frac{1}{2} \left(\sum_{j=1}^{d} x_j^2 \right)^{-\frac{1}{2}} \cdot 2x_1 = \frac{x_1}{\|\underline{x}\|_2} \Rightarrow$$

$$\Rightarrow \operatorname{grad} f(\underline{x}) = \frac{1}{\|\underline{x}\|_2} \underline{x}$$

Theorem I ist Invariante für $\dot{\underline{y}} = \underline{f}(t, \underline{y})$

$\Longleftrightarrow$

$\operatorname{grad} I(\underline{y}) \cdot \underline{f}(t, \underline{y}) = 0$ für alle $(t, \underline{y}) \in [0,T] \times D$

(wo es eine (glatte) Lösung gibt)

Beweis

$\Leftarrow$: Annahme, $\underline{y} := \underline{y}(t)$ für jedes t, Lösung der ODE

$$0 = \operatorname{grad} I(\underline{y}(t)) \cdot \underline{f}(t, \underline{y}(t)) =$$

$$= \operatorname{grad} I(\underline{y}(t)) \cdot \dot{\underline{y}}(t) = \frac{d}{dt} I(\underline{y}(t)) \Rightarrow$$

(Kettenregel!)

$\Rightarrow$ $I(\underline{y}(t)) = \text{konstant} \Rightarrow I$ ist Invariante der ODE

$\Rightarrow$: Nehme $z_0 \in D$, $t_0 \in [0,T]$ beliebig.

verwende die ODE $\begin{cases} \dot{y} = f(t,y) \\ y(t_0) = z_0 \end{cases} \Rightarrow y(t) \Rightarrow$

I Invariante

$\Rightarrow$ grad $I(y(t)) \cdot f(t, y(t)) = 0$ für alle t

nehme $t = t_0 \Rightarrow$ grad $I(z_0) \cdot f(t_0, z_0) = 0$.

Da z_0, t beliebig $\Rightarrow$ grad $I(z) \cdot f(t,z) = 0$

für alle $(t,z) \in [t_0, T] \times D$.

q.e.d.

Notation: $H: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, $H(p,q)$

$$\left[\frac{\partial}{\partial p_1} H(p,q), \dots, \frac{\partial}{\partial p_d} H(p,q) \right]^T = \frac{\partial H}{\partial p} = \operatorname{grad}_p H(p,q)$$

Def Hamiltonische Differentialgleichung

$$H: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R} \qquad H(p,q)$$

$$\begin{cases} \dot{p}_j = -\dfrac{\partial H}{\partial q_j}(p(t), q(t)) \\ \dot{q}_j = \dfrac{\partial H}{\partial p_j}(p(t), q(t)) \\ \text{für } j = 1,2,\dots,d \end{cases} \qquad p, q : [0,T] \to \mathbb{R}^d$$

autonomes Hamilton-System mit der Hamilton-Funktion H.

Bsp 1. Pendel: $p, q : \mathbb{R} \to \mathbb{R}$ $(d=1)$ $(m=1)$

$$H(p,q) = \frac{1}{2}p^2 - \frac{g}{\ell}\cos(q) = E_{tot} \cdot \frac{1}{m\ell^2}$$

(Pendel. $q = \alpha$ = Winkel, $p = \dot{q}$)

2. Konservatives Feld, d.h. $f(\underline{x}) = -\operatorname{grad} U(\underline{x})$

$(\underline{x} \in \mathbb{R}^d)$

z.B. $U(\underline{x}) = G(\|\underline{x}\|_2)$

$$H(\underline{p},\underline{q}) = \frac{1}{2m}\|\underline{p}\|^2 + G(\|\underline{q}\|_2)$$

$p = m\dot{r}$ $\Rightarrow$ Bewegungsgleichungen
von Massenpunkt m
im konservativen Zentralfeld.

$$\begin{cases} \dot{q} = \frac{1}{m}p \\ \dot{p} = -G'(\|\underline{q}\|_2) \cdot \frac{\underline{q}}{\|\underline{q}\|_2} \end{cases}$$

Vergleiche mit Newton' Gleichung:

$$m\ddot{r}(t) = f(r(t))$$

$$f(\underline{q}) = -\operatorname{grad} G(\|\underline{q}\|) = -G'(\|\underline{q}\|_2) \cdot \operatorname{grad}\|\underline{q}\|_2 .$$

Bem $H(\underline{p},\underline{q})$ Invariante für das Ham. System.

Beweis $\underline{y} = \begin{bmatrix} \underline{p} \\ \underline{q} \end{bmatrix}$

$$\underline{\underline{J}} = \begin{bmatrix} \underline{\underline{0}} & \underline{\underline{I}}_d \\ -\underline{\underline{I}}_d & \underline{\underline{0}} \end{bmatrix} \overset{d=2}{=} \left[\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{array}\right]$$

Hamilton'sches System:

$$\dot{\underline{y}} = \underline{\underline{J}}^{-1} \operatorname{grad} H(\underline{y}) =: \underline{f}(\underline{y}) .$$

$I(\underline{y})$ ist Invariante für $\dot{\underline{y}} = \underline{f}(\underline{y}) \Leftrightarrow$

$$\operatorname{grad} I(\underline{y}) \cdot \underline{f}(\underline{y}) = 0 \quad \text{für alle } \underline{y}$$

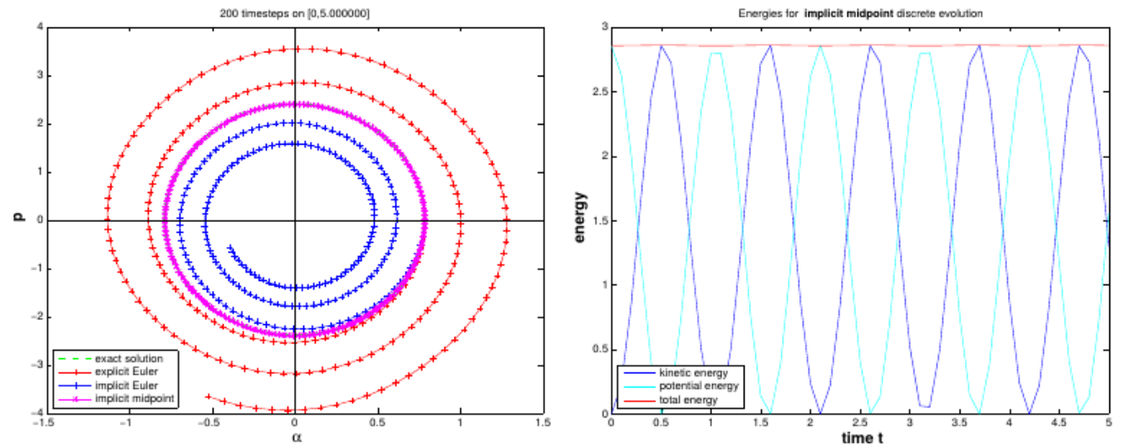
Überprüfen wir das für $I := H$:

$$\operatorname{grad} H(\underline{y}) \cdot \underline{f}(\underline{y}) = \operatorname{grad} H(\underline{y})^T \underline{\underline{J}}^{-1} \operatorname{grad} H(\underline{y}) =$$

$$\operatorname{grad} H(\underline{y}) = \begin{bmatrix} \underline{a} \\ \underline{b} \end{bmatrix}$$

$$= [\underline{a} \;\; \underline{b}] \begin{bmatrix} \underline{\underline{0}} & -\underline{\underline{I}} \\ \underline{\underline{I}} & \underline{\underline{0}} \end{bmatrix} \begin{bmatrix} \underline{a} \\ \underline{b} \end{bmatrix} = [\underline{a} \;\; \underline{b}] \begin{bmatrix} -\underline{b} \\ \underline{a} \end{bmatrix} = \underline{a}\underline{b} - \underline{b}\underline{a} = 0 .$$

qed.



a) Vergleich e.E., i.E., i.M.

b) implizite Mittelpunktsregel: konstante Energie

Abb. 2.2.7. Strukturerhaltungseigenschaften der impliziten Mittelpunktsregel

Bem iMP, St-V erhalten die Energie; $O(h^2)$.
Andere Methoden, die strukturerhaltend sind? höhere Ordnung?

Solche weniger präzise Methoden sind für solch Dgl. (wo Strukturerhaltung wichtig ist) zu bevorzugen!

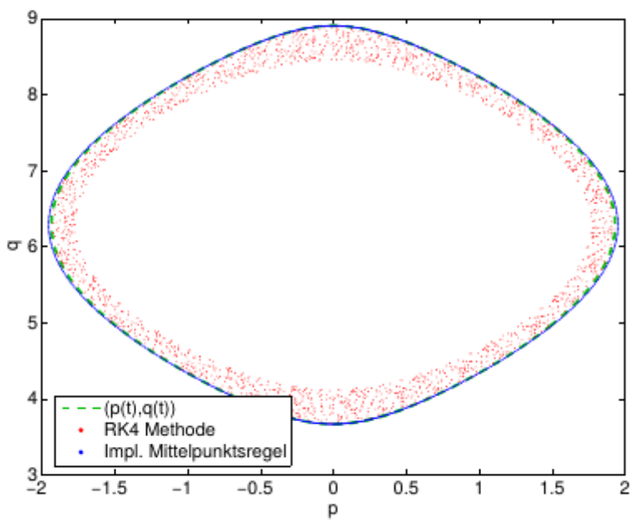
Theorem $I: \mathbb{R}^d \to \mathbb{R}$, $I(y) = \frac{1}{2} y^T B y$ mit

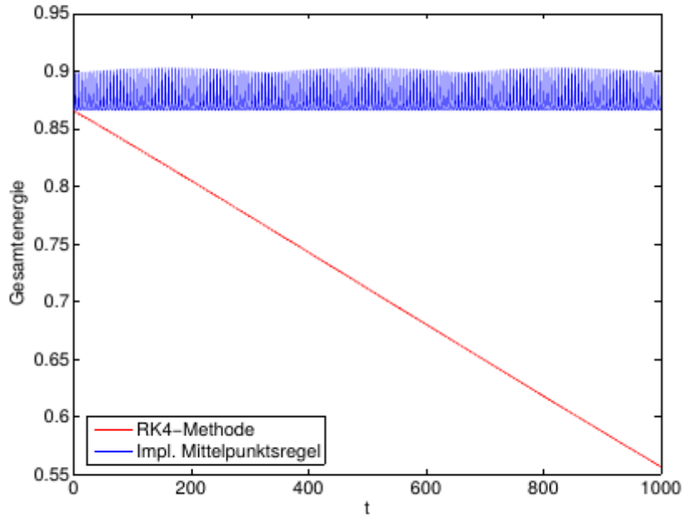
$B \in \mathbb{R}^{d \times d}$, I Invariante für $\dot{y} = f(y)$

f diffbar. Sei (y_k) aus iMP.

Dann $I(y_k) = I(y_0)$ für alle k.

Wir vergleichen das klassische Runge-Kutta-Verfahren (2.5.34) der Ordnung 4 mit der impliziten Mittelpunktsregel bei konstanter Zeitschrittweite $h = \frac{1}{2}$:





§3.2. Splitting Verfahren

autonome ODE:

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

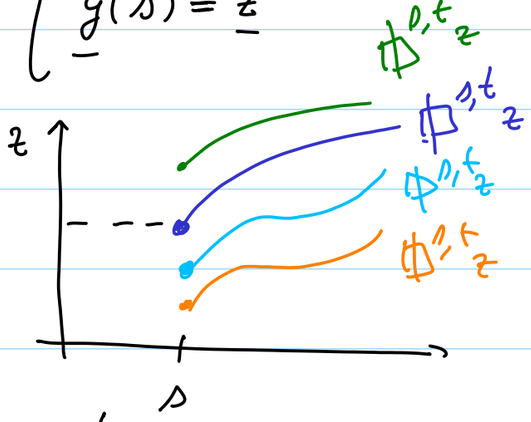
Bem Falls $\dot{y} = f(t,y)$ nicht autonom ist, dann kann man sie autonomisieren, und zwar mit dem Trick:

$$z = \begin{bmatrix} t \\ y \end{bmatrix} \Rightarrow \quad \dot{z} = \begin{bmatrix} 1 \\ f(z) \end{bmatrix} \text{ ist autonom.}$$

$$\dot{t} = 1$$

$$z(0) = \begin{bmatrix} t_0 \\ y_0 \end{bmatrix}.$$

ODE: $\begin{cases} \dot{y} = f(t,y) \\ y(s) = z \end{cases}$ Lösung: $y(t)$ für $t > s$



$\Phi^{s,t} : D \to D$ zweiparametrige Familie von Abbildungen "Fluss der ODE"

$\Phi^{t,t} = Id$; $\Phi^{s,t}z = \Phi^{r,t} \circ \Phi^{s,r} z$

wenn ODE autonom ist: $\Phi^{s,t} = \Phi^{0,t-s} = \Phi^{t-s}$

d.h. die Lösung der autonomen ODE

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

ist translationsinvariant!

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

$u(t) := y(t+\tau) \Rightarrow \frac{d}{dt} u(t) = \dot{y}(t+\tau) \cdot 1 =$

$= f(y(t+\tau)) = f(u)$

$\Rightarrow \dot{u} = f(u)$

Dem jede ODE kann man autonomisieren!

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

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

(ODE): $\dot{\underline{y}} = \underline{f}(\underline{y}) = \underline{f}_a(\underline{y}) + \underline{f}_b(\underline{y})$

Idee: wähle $\underline{f}_a$, $\underline{f}_b$ so dass

(a) $\dot{\underline{y}} = \underline{f}_a(\underline{y})$ und

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

einfach/exakt lösbar sind.

Bem Für lineares Problem:

$$\dot{\underline{y}} = \underline{\underline{M}}\,\underline{y} = (\underline{\underline{A}} + \underline{\underline{B}})\,\underline{y} = \underline{\underline{A}}\,\underline{y} + \underline{\underline{B}}\,\underline{y}$$

mit $\underline{\underline{A}}, \underline{\underline{B}}$ mit speziellen Eigenschaften....

exakte Lösung $\underline{y}(t) = e^{\underline{\underline{M}}t}\,\underline{y}_0 = e^{(\underline{\underline{A}}+\underline{\underline{B}})t}\,\underline{y}_0$

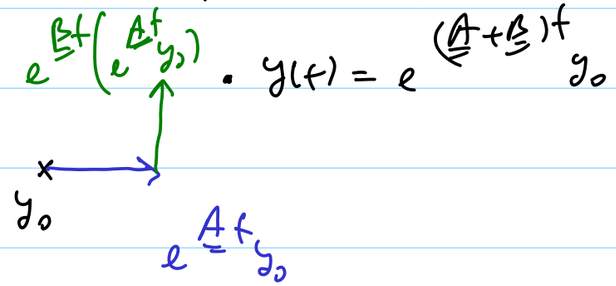
$$e^{\underline{\underline{M}}t} = \sum_{n=0}^{\infty} \frac{1}{n!}(\underline{\underline{M}}t)^n \quad \text{(nicht so rechnen)}$$

$$e^{(\underline{\underline{A}}+\underline{\underline{B}})t} \neq e^{\underline{\underline{A}}t}\,e^{\underline{\underline{B}}t} \quad \text{z.B. wenn } \underline{\underline{A}}\,\underline{\underline{B}} \neq \underline{\underline{B}}\,\underline{\underline{A}}$$

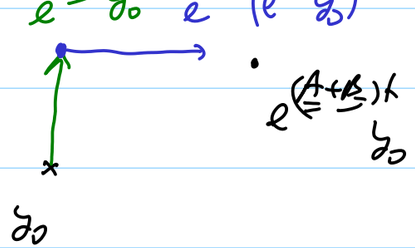
Lösung von $\begin{cases} \dot{\underline{y}} = \underline{\underline{A}}\,\underline{y} \\ \underline{y}(0) = \underline{y}_0 \end{cases}$ ist $e^{\underline{\underline{A}}t}$

$\begin{cases} \dot{\underline{y}} = \underline{\underline{B}}\,\underline{y} \\ \underline{y}(0) = \underline{y}_0 \end{cases}$ ist $e^{\underline{\underline{B}}t}$

$e^{\underline{\underline{B}}t}(e^{\underline{\underline{A}}t}\underline{y}_0)$ · $\underline{y}(t) = e^{(\underline{\underline{A}}+\underline{\underline{B}})t}\,\underline{y}_0$



$e^{\underline{\underline{B}}t}y_0$ $e^{\underline{\underline{A}}t}(e^{\underline{\underline{B}}t}y_0)$ · $e^{(\underline{\underline{A}}+\underline{\underline{B}})t}y_0$



$$e^{(A+B)t} = I + (A+B)t + \frac{1}{2}(A^2 + AB + BA + B^2)t^2 + \dots$$

$$(e^{At})(e^{Bt}) = \left(I + At + \frac{1}{2}A^2t^2 + \dots\right)\left(I + Bt + \frac{1}{2}B^2t^2 + \dots\right) =$$

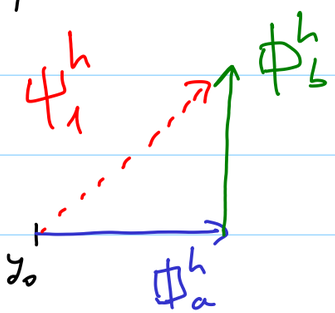
$$= I + (B+A)t + \left(ABt^2 + \frac{1}{2}A^2t^2 + \frac{1}{2}B^2t^2\right) + \dots$$

Der t^2-Term ist in $e^{(A+B)t}$: $\frac{1}{2}A^2 + \frac{1}{2}(AB+BA) + \frac{1}{2}B^2$

in $e^{At}e^{Bt}$: $\frac{1}{2}A^2 + AB + \frac{1}{2}B^2$

ungleich wenn $AB \neq BA$

Man kann beweisen, dass dies eine Approximation liefert!



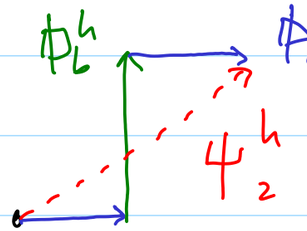
$\Psi_1^h := \Phi_b^h \circ \Phi_a^h$

Lie-Trotter splitting

$y_1 := \Phi_b^h \Phi_a^h y_0$ $\quad O(h)$ zur Endzeit T

$\Phi_a^h \Phi_b^h y_0$ $\quad O(h)$ zur Endzeit T

Symmetrie schenkt uns ein Ordnung mehr:



$\Psi_2^h := \Phi_a^{h/2} \Phi_b^h \Phi_a^{h/2}$

y_0 $\Phi_a^{h/2}$ Strang splitting

Allgemein: $\Psi_s^h = \prod_{i=1}^{s} \Phi_b^{b_i h} \Phi_a^{a_i h}$

mit $\sum_{i=1}^{s} a_i = 1, \quad \sum_{i=1}^{s} b_i = 1$

Bsp $s=1,\ a_1=1,\ b_1=1 \Rightarrow$ Lie-Trotter

$s=2,\ a_1=a_2=\frac{1}{2},\ b_1=1,\ b_2=0 \Rightarrow$ Strang

Welche a, b wählen damit wir höhe Ordnung?
f_a, f_b Erhaltungseigenschaften.

Beispiel 2.4.1. (Konvergenz einfacher Splitting-Verfahren)
Sei

$$\dot{y} = \underbrace{\lambda y(1-y)}_{=:f_a(y)} + \underbrace{\sqrt{1-y^2}}_{=:f_b(y)} \quad , \quad y(0) = 0 \, .$$

Die Evolutionsoperatoren der zwei Teile sind analytisch bekannt:

$$\mathbf{\Phi}_a^t y = \frac{1}{1 + (y^{-1} - 1)e^{-\lambda t}}, \text{ für } t > 0,\ y \in]0, 1] \text{ und}$$

$$\mathbf{\Phi}_b^t y = \begin{cases} \sin(t + \arcsin(y)), & \text{wenn } t + \arcsin(y) < \frac{\pi}{2} \, , \\ 1, & \text{sonst} \end{cases} \quad \text{für } t > 0,\ y \in [0, 1] \, .$$

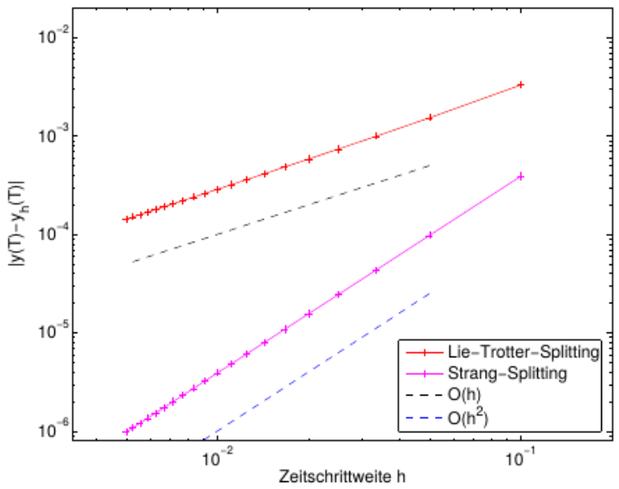
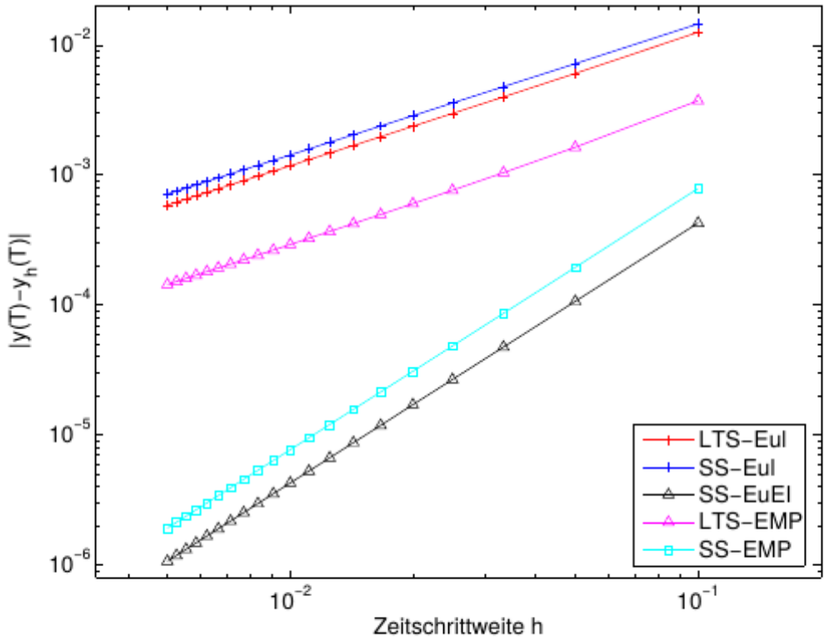


Abb. 2.4.2. Fehler zur Endzeit $T = 1$.

Numerisches Experiment:

$T = 1$, $\lambda = 1$, Vergleich einer sehr genauen numerisch Lösung.

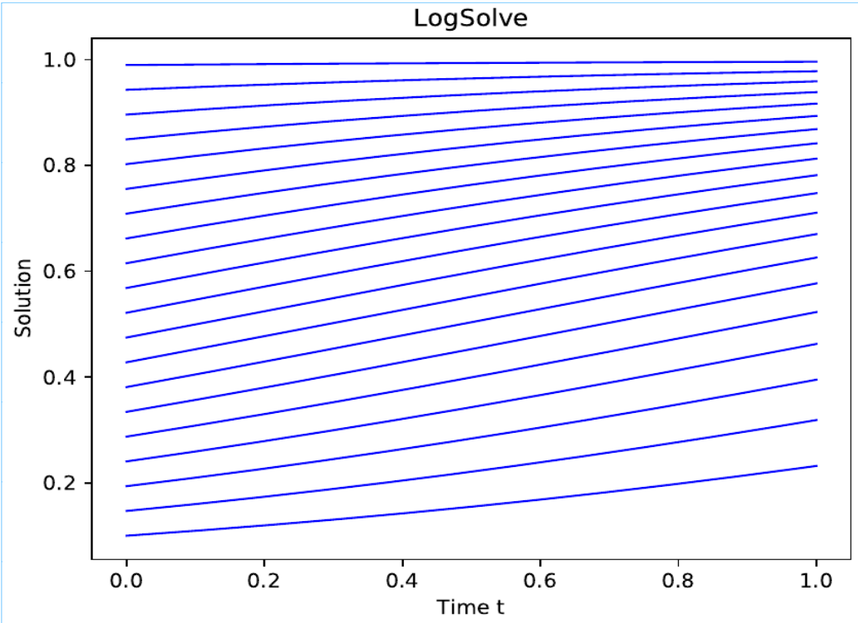


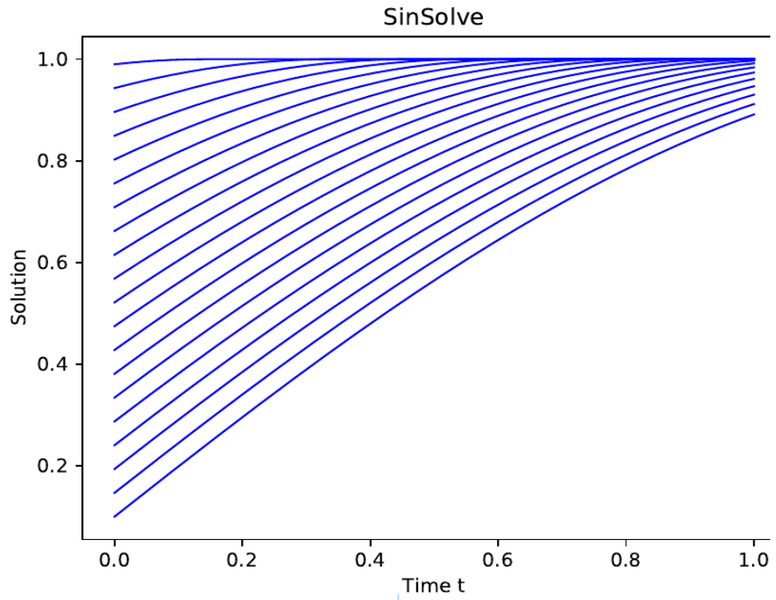
LTS-Eul	explizites Euler als $\Psi^h_{a,b}$, $\Psi^h_{a,b}$ und Lie-Trotter-Splitting
SS-Eul	explizites Euler als $\Psi^h_{a,b}$, $\Psi^h_{a,b}$ und Strang-Splitting
SS-EuEI	Strang-Splitting: explizites Euler als $\Psi^{h/2}_a$, exaktes Φ^h_b und implizites Euler als $\Psi^{h/2}_a$
LTS-EMP	explizite Mittelpunkt-Regel als $\Psi^h_{a,b}$, $\Psi^h_{a,b}$ und Lie-Trotter-Splitting
SS-EMP	explizite Mittelpunkt-Regel als $\Psi^h_{h,g}$, $\Psi^h_{h,f}$ und Strang-Splitting

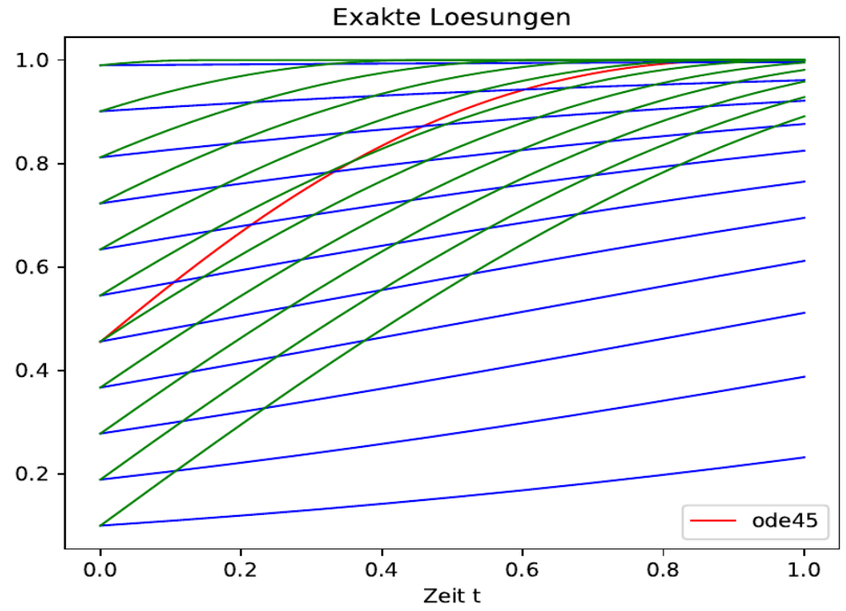
Abb. 2.4.5. Einfache Splitting-Verfahren

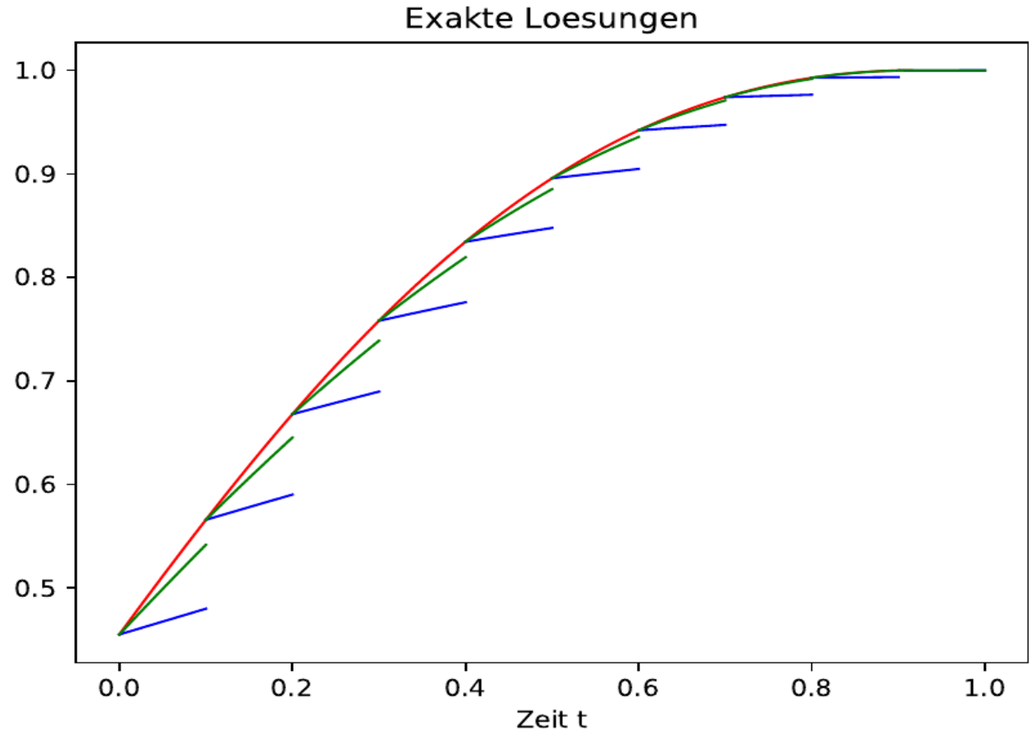
$y_1 = y_0 + h f\left(t_0 + \frac{h}{2}, y_0 + \frac{h}{2} y_0\right)$ $\quad O(h^2)$

ein halber Zeitschritt EE um $y(\frac{h}{2})$ zu approximieren.









Bsp (Splitting-Verfahren für Newtonsche Gleichung)

$$\ddot{r} = a(r) \iff \dot{y} = \begin{bmatrix} \dot{r} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ a(r) \end{bmatrix} =: F(y)$$

$$F(y) = \underbrace{\begin{bmatrix} 0 \\ a(r) \end{bmatrix}}_{f} + \underbrace{\begin{bmatrix} v \\ 0 \end{bmatrix}}_{g} \qquad \text{Startwert } y_0 = \begin{bmatrix} r_0 \\ v_0 \end{bmatrix}$$

Die exakten Evolutionsoperatoren sind:

$$\begin{cases} \dot{r} = 0 \\ \dot{v} = a(r) \end{cases}$$

von 0 zu h: $r(h) = r(0) = r_0$

$\dot{v} = a(r_0) \Rightarrow v(h) = v_0 + h\,a(r_0)$

$$\Phi_f^h \begin{bmatrix} r_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} r_0 \\ v_0 + h\,a(r_0) \end{bmatrix}$$

$$\begin{cases} \dot{r} = v \\ \dot{v} = 0 \end{cases} \Rightarrow v(h) = v_0$$

$\dot{r} = v_0 \Rightarrow r(h) = r_0 + h v_0$

$$\Phi_g^h \begin{bmatrix} r_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} r_0 + h v_0 \\ v_0 \end{bmatrix}$$

Kombinieren wir diese exakte Lösungen

(1) Lie-Trotter-Splitting:

$$\Psi^h \begin{bmatrix} r \\ v \end{bmatrix} = \left(\Phi_g^h \circ \Phi_f^h\right) \begin{bmatrix} r \\ v \end{bmatrix} = \begin{bmatrix} r + h(v + h\,a(r)) \\ v + h\,a(r) \end{bmatrix}$$

symplektische Euler-Verfahren

(v) Strang-Splitting:

$$\Psi^h\begin{bmatrix} r \\ v \end{bmatrix} = \Phi_g^{h/2} \circ \Phi_f^{h} \circ \Phi_g^{h/2}\begin{bmatrix} r \\ v \end{bmatrix} =$$

$$= \begin{bmatrix} r + \frac{h}{2}v + \frac{1}{2}h^2 a\left(r + \frac{1}{2}hv\right) \\ v + h\,a\left(r + \frac{1}{2}hv\right) \end{bmatrix}$$

$$\Phi_g^{h/2}\begin{bmatrix} r_0 + \frac{h}{2}v_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} r_0 + \frac{h}{2}v_0 \\ v_0 + h\,a(r_0) \end{bmatrix}$$

$$\Phi_g^{h/2} = \Phi_g^{\frac{h}{2}}\begin{bmatrix} r_0 + \frac{h}{2}v_0 \\ v_0 + h\,a\left(r_0 + \frac{h}{2}v_0\right) \end{bmatrix} =$$

$$= \begin{bmatrix} r_0 + \frac{h}{2}\left(v_0 + h\,a\left(r_0 + \frac{h}{2}v_0\right)\right) \\ v_0 + h\,a\left(r_0 + \frac{h}{2}v_0\right) \end{bmatrix}$$

$\Leftrightarrow$

Notation

$$\begin{cases} r_{k+\frac{1}{2}} = r_k + \frac{1}{2}h\,v_k \\ v_{k+1} = v_k + h\,a\left(r_{k+\frac{1}{2}}\right) \\ r_{k+1} = r_k + \frac{1}{2}h\,v_{k+1} \end{cases}$$

$\equiv$ ein-Schritt-Formulierung des (StV)!

Bem Trick auch für separablen Hamilton-Systeme:

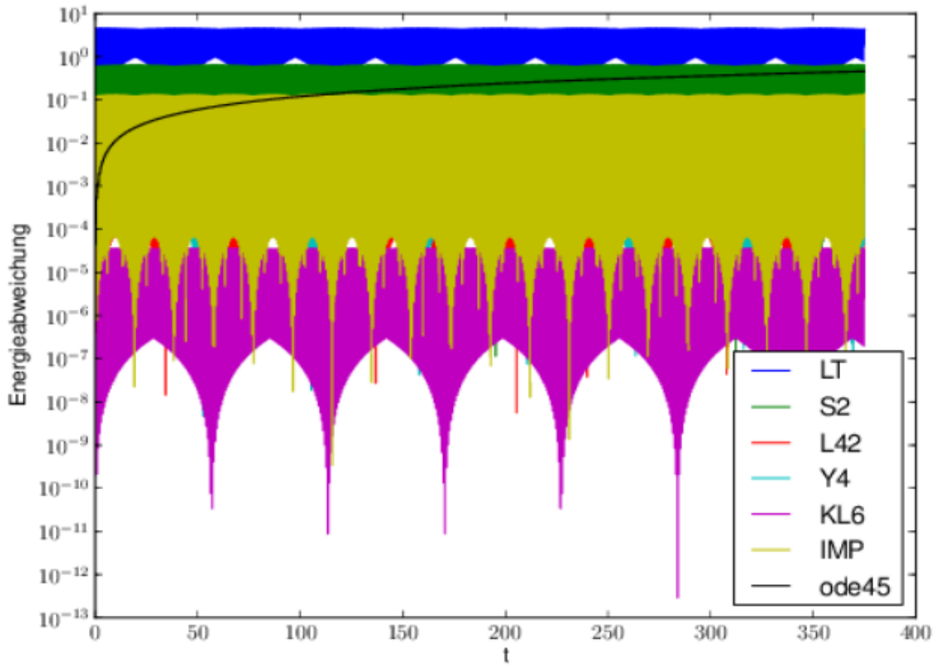
$$H(\underline{p},\underline{q}) = T(\underline{p}) + V(\underline{q})$$

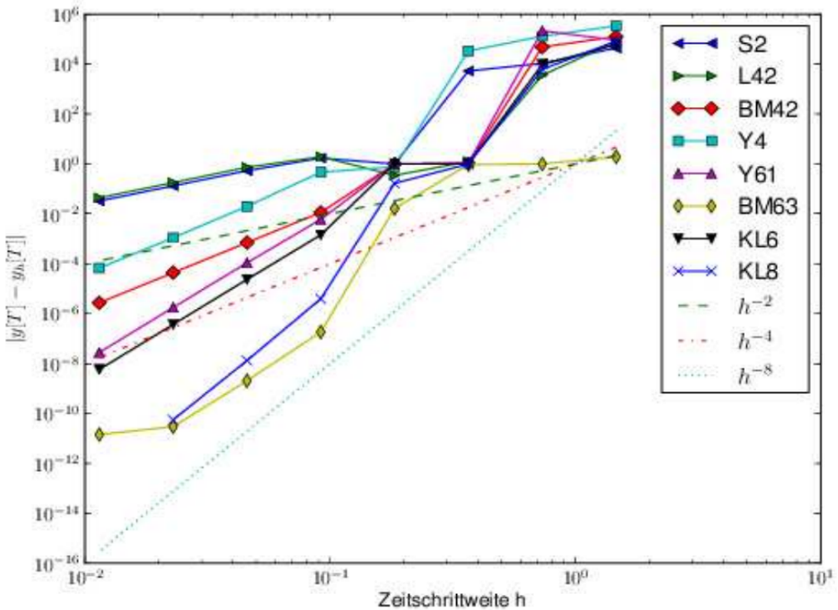
Lie-Trotter:

$$\begin{cases} \underline{p}_1 = \underline{p}_0 - h\,\mathrm{grad}\,V(\underline{q}_0) \\ \underline{q}_1 = \underline{q}_0 + h\,\mathrm{grad}\,T(\underline{p}_1) \end{cases}$$

Symplektische Euler-Verfahren

Strang-Splitting $\Rightarrow$ (St-V) !





Bsp $H(p,q) = \frac{1}{2}p^2 - \frac{g}{\ell}\cos q + (-q)\,A\cos(\omega t)$

$$\begin{cases} \dot p = -\frac{\partial H}{\partial q} = -\frac{g}{\ell}\sin q + A\cos(\omega t) \\ \dot q = \frac{\partial H}{\partial p} = p \end{cases}$$

Nicht autonom! ⇒ autonomisieren

Unbekannte $\dot t = 1$

(a):
$$\begin{cases} \dot q = 0 \Rightarrow q(h) = q(0) = q_0 \\ \dot t = 0 \Rightarrow t(h) = t(0) = t_0 \\ \dot p = -\frac{g}{\ell}\sin q + A\cos(\omega t) \Rightarrow \end{cases}$$

$$\dot p = -\frac{g}{\ell}\sin q_0 + A\cos(\omega t_0) \Rightarrow$$

$$p(h) = p_0 - \left(\frac{g}{\ell}\sin q\right)h + A\cos(\omega t_0)\,h$$

$$\underline{u} = \begin{bmatrix} q \\ t \\ p \end{bmatrix} \Rightarrow \qquad \dot{\underline{u}} = \underline{f}(\underline{u}), \quad \underline{f}(\underline{u}) = \begin{bmatrix} p \\ 1 \\ -\frac{g}{\ell}\sin q + A\cos(\omega t) \end{bmatrix}$$

$$\underline{f}(\underline{u}) = \underbrace{\begin{bmatrix} p \\ 1 \\ 0 \end{bmatrix}}_{(b)} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ -\frac{g}{\ell}\sin q + A\cos(\omega t) \end{bmatrix}}_{(a)} = \underline{f}_a(\underline{u}) + \underline{f}_b(\underline{u})$$

(b) $\begin{cases} \dot q = p & \Rightarrow q(h) = q_0 + p_0 h \\ \dot t = 1 & \Rightarrow t(h) = t_0 + h \\ \dot p = 0 & \Rightarrow p(h) = p_0 \end{cases}$

$\tau = a_i \cdot h$, $\tau = b_i \cdot h$ $\Rightarrow$ sein Lieblingssplitting!

Processing

$$\hat\Psi^h = \Pi^h \circ \Psi^h \circ (\Pi^h)^{-1}$$

post-processor ↑ pre-processor ↖

$$(\hat\Psi^h)^n = \Pi^h \circ \Psi^h \circ (\Pi^h)^{-1} \circ \Pi^h \circ \Psi^h \circ (\Pi^h)^{-1} \circ \dots \circ \Pi^h \circ \Psi^h \circ (\Pi^h)^{-1}$$

$$= \Pi^h \circ (\Psi^h)^n \circ (\Pi^h)^{-1}$$

Processing bringt Vorteile falls:

+ $\hat\Psi^h$ genauer als Ψ^h ist

+ Π^h, $(\Pi^h)^{-1}$ günstig.

+ keine / wenige Ausgaben vor Endzeit

Bsp Strang-Splitting:

$$\Psi_2^h = \Phi_a^{h/2} \circ \Phi_b^h \circ \Phi_a^{h/2} \Rightarrow$$

$$\uparrow \quad \Phi_a^{\frac{h}{2}} \circ \left(\Phi_a^{\frac{h}{2}}\right)^{-1}$$

$$\Rightarrow \Psi_2^h = \Phi_a^{\frac{h}{2}} \circ \Phi_b^h \circ \Phi_a^h \circ \left(\Phi_a^{\frac{h}{2}}\right)^{-1}$$

Lie-Trotter

$\Rightarrow$ Strang-Splitting = Processing von Lie-Trotter!

Leicht gestörte Probleme

$$\dot{y} = f_a(y) + \varepsilon f_b(y) \qquad \text{mit } \varepsilon \text{ klein}$$

optimierte Splitting-Verfahren:

$$O\left(\varepsilon h^{r_1} + \varepsilon^2 h^{r_2} + \varepsilon^3 h^{r_3} + \dots \right)$$

$r_1 \geq r_2 + 1, \dots$

$\varepsilon h^4 + \varepsilon^2 h^2$

§4 Runge-Kutta-Verfahren

§4.1. Grundidee

$$\begin{cases} \dot{\underline{y}} = \underline{f}(t, \underline{y}) \\ \underline{y}(t_0) = \underline{y}_0 \end{cases} \qquad \int_{t_0}^{t_1} \Rightarrow \underline{y}(t_1) = \underline{y}(t_0) + \int_{t_0}^{t_1} \underline{f}(t, \underline{y}(t))\,dt$$

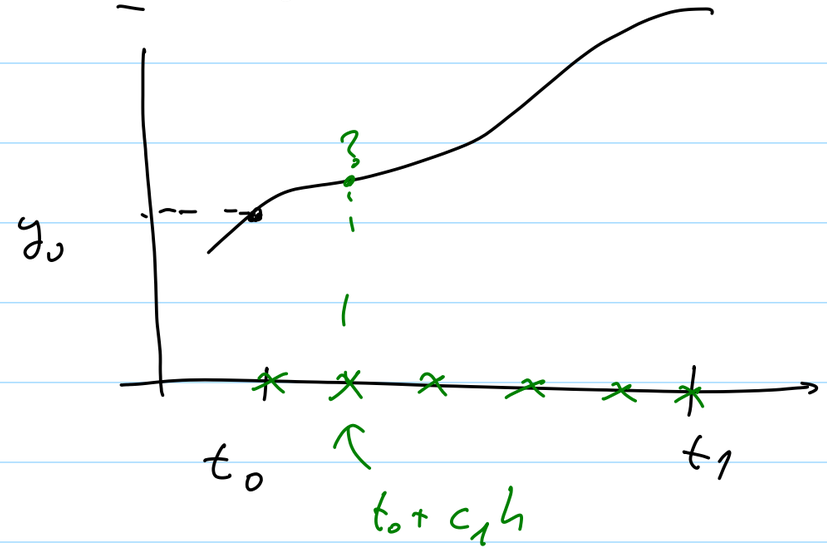
$h = t_1 - t_0$, Referenzinterval $[0,1]$ $\Rightarrow$

$$\underline{y}(t_1) = \underline{y}(t_0) + h \int_0^1 \underline{f}(t_0 + hz, \underline{y}(t_0 + hz))\,dz$$

QF mit Gewichten b_i, Knoten c_i $\Rightarrow$

$$\underline{y}(t_1) \approx \underline{y}(t_0) + h \sum_{i=1}^{s} b_i \underbrace{\underline{f}(t_0 + h c_i, \underline{y}(t_0 + h c_i))}_{k_i}$$

h erlaubt uns einen Fehler $O(h^{p+1})$ für $\underline{y}(t_1)$ zu bekommen, auch wenn für $\underline{y}(t_0 + h c_i)$ Approx. $O(h^p)$ verwende!



Bsp 1) QF = Trapezregel auf $[0,1]$:

$$s = 2, \quad c_1 = 0, \quad c_2 = 1, \quad b_1 = b_2 = \frac{1}{2}$$

$$\underline{y}_1 = \underline{y}_0 + h\left(\frac{1}{2} \underline{f}(\underbrace{t_0 + h\cdot 0}_{t_0}, \underbrace{\underline{y}(t_0 + h\cdot 0)}_{\underline{y}(t_0) = \underline{y}_0}) + \frac{1}{2} \underline{f}(\underbrace{t_0 + h1}_{t_1}, \underbrace{\underline{y}(t_0 + h1)}_{\underline{y}(t_1)}) \right)$$

Idee: verwende etwas bessseres für den inneren Term (unbekannt)

$$y(t_0+h) \approx y_0 + h f(t_0, y_0) \qquad (e \in) \qquad k_1$$

$$\begin{cases} k_1 := f(t_0, y_0) \\ k_2 := f(t_0+h, y_0+hk_1) \\ y_1 = y_0 + h\cdot\frac{1}{2}k_1 + h\frac{1}{2}k_2 \end{cases}$$

explizite Trapezregel.

$$\begin{cases} k_1 := f(t_0, y_0) \\ k_2 := f(t_0+\frac{h}{2}, y_0+\frac{h}{2}k_1) \\ y_1 = y_0 + hk_2 \end{cases}$$

explizite Mittelpunktsregel.

Def Runge-Kutta-Verfahren mit s Stufen.

Gegeben Butcher-Schema

c_1	
c_2	$A \in \mathbb{R}^{s\times s}$
$\vdots$	
c_s	
	$b_1\ b_2 \dots b_s$

so dass $b_1 + b_2 + \dots + b_s = 1$

$$\sum_{j=1}^{s} a_{ij} = c_i \qquad \text{für } i = 1, 2, \dots, s$$

Bsp 2) QF = Mittelpunktsregel

$$y_1 = y_0 + h f\left(t_0 + h\tfrac{1}{2},\ y(t_0+h\tfrac{1}{2})\right)$$

$\approx$ (e∈)

$$y_0 + h\tfrac{1}{2} f(t_0, y_0) \qquad k_1$$

$$\begin{cases} \underline{k}_i = \underline{f}\left(t_0 + c_i h,\ \underline{y}_0 + h \sum_{j=1}^{s} a_{ij}\, \underline{k}_j\right) \\ \text{für } i = 1, 2, \dots, s \end{cases}$$

bis $j \le i-1$ falls es explizit sein sollte!

Stufen

↖ ein $s \times s$ nichtlineares algebraisches Gleichungssystem

$$\underline{y}_1 = \underline{y}_0 + h \sum_{i=1}^{s} b_i\, \underline{k}_i$$

$$\underline{\underline{A}} = \begin{bmatrix} 0 & & 0 \\ & \ddots & \\ * & & 0 \end{bmatrix}$$

$a_{ij} = 0$ für alle $i \le j \Rightarrow$ ⇒ explizites Verfahren.

$$\underline{\underline{A}} = \begin{bmatrix} \diagdown & & 0 \\ & \diagdown & \\ * & & \diagdown \end{bmatrix}$$

diagonal implizites Verfahren.

Def Konsistenzordnung q:

lokale Fehler $\|y(t_0+h) - y_1\| \le c \cdot h^{q+1}$

Theorem

RK hat Konsistenzordnung $q \Rightarrow$
QF hat Ordnung q

(ist exakt für Polynome von Grad max $q-1$)

Beweis Nehme $\begin{cases} \dot{y} = t^n \\ y(0) = 0 \end{cases} \Rightarrow y(t) = \frac{1}{n+1} t^{n+1}$

Fehler: $|y(h) - y_1| = \left| \frac{1}{n+1} h^{n+1} - h \sum_{j=1}^{s} b_j (c_j h)^n \right| \le c \cdot h^{q+1}$

$$\Leftrightarrow \quad \left| \frac{1}{n+1} h^{n+1} - h^{n+1} \sum_{j=1}^{s} b_j c_j^n \right| \le c \cdot h^{q+1} \quad \Big| : h^{n+1} \Rightarrow$$

$$\left| \frac{1}{n+1} - \sum_{j=1}^{s} b_j c_j^n \right| \le c \cdot h^{q-n}$$

$h \to 0$ solange $q > n$

Für: $n = 0, 1, 2, \dots, q-1$:

$$\frac{1}{n+1} = \sum_{j=1}^{s} b_j c_j^n$$

$\Longleftrightarrow$ QF exakt für $p_n(t) = t^n$ q.e.d

Konsequenz RK s Stufen $\Rightarrow$ Konsistenzordnung $\le 2s$

Bem 1) $\sum_{j=1}^{s} b_j = 1 \Rightarrow$ mindestens Konsistenzordnung $q = 1$

2) RK hat mindestens Konsistenzordnung $q = 2$

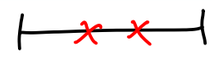
wenn $\sum_{j=1}^{s} b_j c_j = \frac{1}{2}$

3) $q = 3$ brauchen wir

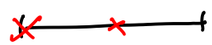
$$\sum_{j=1}^{s} b_j c_j^2 = \frac{1}{3}, \quad \sum_{j=1}^{s} b_j \sum_{n=1}^{s} a_{jn} c_n = \frac{1}{6}$$

Theorem RK explizit $\Rightarrow$ $q \le s$

Gauss-Quadratur $\Rightarrow$ $q = 2s$



Radau-Quadratur $\Rightarrow$



Radau-Verfahren für ODE

Lobatto-Quadratur

↳ Lobatto-Verfahren für ODE

Theorem RK hat Konsistenzordnung $q \Rightarrow$ (globale) Konvergenzordnung q

$$\| y(t_i) - y_i \| \le c \cdot h^q \quad \text{für } i = 1, 2, \dots, N.$$

§4.2. Kollokation

Def $c_1, c_2, \dots, c_s \in [0,1]$ verschieden.

Kollokationspolynom $u(t)$ von Grad s:

$$\begin{cases} u(t_0) = y_0 \\ \dot{u}(t_0 + c_i h) = f(t_0 + c_i h, u(t_0 + c_i h)) \end{cases} \quad \text{für } i = 1, 2, \dots, s$$

Bsp 1) $s = 1 \Rightarrow$ Polynom vom Grad 1:

$u(t) = y_0 + (t - t_0) k$

mit k bestimmt so, dass

$\dot{u}(t_0 + c_1 h) = f(t_0 + c_1 h, u(t_0 + c_1 h))$

$c_1 = 0 \Rightarrow$ (EE)

$c_1 = 1 \Rightarrow$ (IE)

$c_1 = \frac{1}{2} \Rightarrow$ (IMP)

2) $s = 2$; $c_1 = 0$, $c_2 = 1 \Rightarrow$ implizite Trapezregel

$c_{1,2} = \frac{1}{2} \pm \frac{\sqrt{3}}{6} \Rightarrow$ Gauss-Verfahren $O(h^4)$

Theorem Die Kollokation mit knoten $c_1, \dots, c_s$

$\Longleftrightarrow$

s-Stufiges RKV mit $a_{ij} = \int_0^{c_i} l_j(z)\,dz$,

$$b_i = \int_0^1 l_i(z)\,dz$$

wobei

$$l_i(z) = \prod_{\substack{j=1 \\ j \neq i}}^{s} \frac{z - c_j}{c_i - c_j}$$ Lagrange Polynom.

$$l_i(c_j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

Beweis Notiere $k_i := \dot{u}(t_0 + c_i h)$.

Eigenschaft von $l_i \Rightarrow \quad \dot{u}(t_0 + \tau h) = \sum_{j=1}^{s} k_j \cdot l_j(\tau)$ $\int_0^{c_i}$

Polynom vom Grad $s-1$

$$\Rightarrow u(t_0 + c_i h) = y_0 + h \sum_{j=1}^{s} k_j \underbrace{\int_0^{c_i} l_j(\tau)\, d\tau}_{a_{ij}}$$

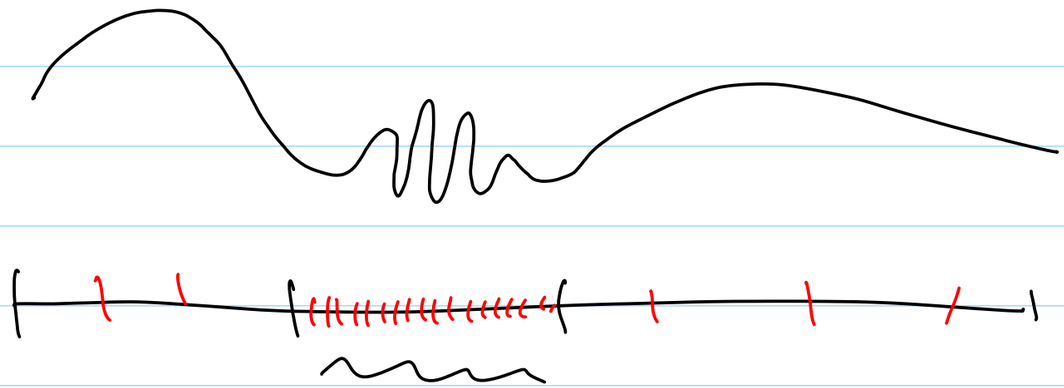
zwischen stellen.

$\int_0^1 \Rightarrow \quad y_1 = y_0 + h \sum_{j=1}^{s} k_j \underbrace{\int_0^1 l_j(\tau)\, d\tau}_{b_j}$

Konsequenz

Kollokationsmethode hat dieselbe Ordnung wie die entsprechende QF.

§4.3. Adaptivität



lokaler Fehler.

$\Phi - \Psi_h \Leftrightarrow$ Schätzung. $\tilde{\Psi}_h - \Psi_h$

$\Phi \approx \tilde{\Psi}_h$

genauere Methode.

$O(h^{p+2})$ $\qquad$ $O(h^{p+1})$

$$\left| \tilde{\Psi}^{t, t+h} y(t_k) - \Psi^{t, t+h} y(t_k) \right| = est_k$$

$est_k = c h^{p+1} \approx tol \Rightarrow \quad h^* := h \sqrt[p+1]{\frac{tol}{est_k}}$

SKRIPT !

§4.4. Partitionierte RK-Verfahren

System ODE, so partioniert:

$$\begin{cases} \dot{\underline{y}} = f(\underline{y}, \underline{z}) \\ \dot{\underline{z}} = g(\underline{y}, \underline{z}) \end{cases}$$

Idee: verwende 2 verschiedene RKV für $\underline{y}$ und $\underline{z}$:

$$\begin{cases} \underline{k}_i = \underline{f}\left(\underline{y}_0 + h\sum_{j=1}^{s} a_{ij}\underline{k}_j,\ \underline{z}_0 + h\sum_{j=1}^{s} \hat{a}_{ij}\underline{\ell}_j\right) \\ \underline{\ell}_i = g\left(\underline{y}_0 + h\sum_{j=1}^{s} a_{ij}\underline{k}_j,\ \underline{z}_0 + h\sum_{j=1}^{s} \hat{a}_{ij}\underline{\ell}_j\right) \end{cases}$$

für $\underline{y}$:

c	A
1	b

für $\underline{z}$:

$\hat{c}$	$\hat{A}$
1	$\hat{b}$

$$\underline{y}_1 = \underline{y}_0 + h\sum_{j=1}^{s} b_j \underline{k}_j$$

$$\underline{z}_1 = \underline{z}_0 + h\sum_{j=1}^{s} \hat{b}_j \underline{\ell}_j$$

Bsp 1)

(ℓE) mit $b_1 = 1$, $a_{11} = 1$ } ⇒ symplektische Euler-Verfahren

(ℰℓ) $b_1 = 1$, $\hat{a}_{11} = 0$ } (für Newton'Gleichg.)

Bsp 2)

0	0	0
1	$\frac{1}{2}$	$\frac{1}{2}$
	$\frac{1}{2}$	$\frac{1}{2}$

,

1/2	$\frac{1}{2}$	0
1/2	$\frac{1}{2}$	0
	$\frac{1}{2}$	$\frac{1}{2}$

PRK ⟹ Störmer-Verlet!

Eine Verallgemeinerung vom Störmer-Verlet kommt hier natürlich heraus, aus 3-stufigos Lobatto-Paar: (Ordnung h^4)

0	0	0	0
$\frac{1}{2}$	$\frac{5}{24}$	$\frac{1}{3}$	$-\frac{1}{24}$
1	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$
	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

0	$\frac{1}{6}$	$-\frac{1}{6}$	0
$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{3}$	0
1	$\frac{1}{6}$	$\frac{5}{6}$	0
	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

Bem Newton'Gleichung:

$$\ddot{\underline{y}} = g(t, \underline{y}, \dot{\underline{y}})$$

Umschreiben in System 1. Ordnung:

$$\begin{cases} \dot{\underline{y}} = \underline{z} \\ \dot{\underline{z}} = \underline{g}(t, \underline{y}, \underline{z}) \end{cases}$$

PRK für Rk-Nyström-Verfahren (RKN)

$$\begin{cases} \underline{k}_i = g\left(t, \underline{y}_0 + c_i h \dot{\underline{y}}_0 + h^2 \sum_{j=1}^{s} \bar{a}_{ij} \underline{k}_j, \dot{\underline{y}}_0 + h \sum_{j=1}^{s} \hat{a}_{ij} \underline{k}_j\right) \\ \underline{y}_1 = \underline{y}_0 + h \dot{\underline{y}}_0 + h^2 \sum_{i=1}^{s} \bar{b}_i \underline{k}_i \\ \dot{\underline{y}}_1 = \dot{\underline{y}}_0 + h \sum_{i=1}^{s} \hat{b}_i \underline{k}_i \end{cases}$$

mit

$$\bar{a}_{ij} = \sum_{k=1}^{\Lambda} a_{ik}\,\hat{a}_{kj}, \qquad \bar{b}_i = \sum_{k=1}^{\Lambda} b_k\,\hat{a}_{ki}$$

Wenn g nicht noch von j abhängt, dann dann braucht man $\hat{a}_{kj}$ gar nicht.

<u>Bem</u> PRK $\equiv$ Splitting mit

$$\underline{u} = \begin{bmatrix} \underline{x} \\ \underline{z} \end{bmatrix}, \quad \underline{f}_a = \begin{bmatrix} f(\underline{u}) \\ 0 \end{bmatrix}, \quad \underline{f}_b = \begin{bmatrix} 0 \\ g(\underline{u}) \end{bmatrix}$$

$\Rightarrow$ einfachere Anwendung!

BM42	$\rightarrow$	$O(h^4)$	symplektische
BM63	$\rightarrow$	$O(h^6)$	Verfahren.

§5 Steife Differentialgleichungen

§5.1. Einführung

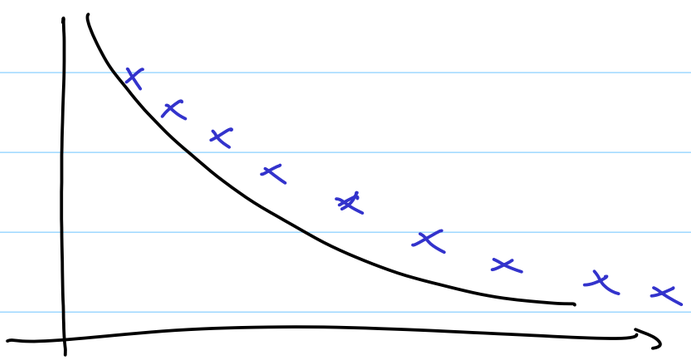
Modelproblem: $\dot{y} = -\lambda y$ mit $\lambda > 0$

$$y(t) = e^{-\lambda t} y_0 \to 0 \quad \text{für } t \to \infty$$

Der Abfall der Lösung ist schneller für grösseres λ.

Numerische Lösung: festes Zeitschritt h

$$\left(\psi_\lambda^{kh}\right)_{k=1,2,3,\dots} \to 0 \quad \text{für } k \to \infty ?$$



(EE) $y_1 = y_0 + h f(y_0) = y_0 - h\lambda y_0 = (1-h\lambda) y_0$

$y_2 = y_1 - h\lambda y_1 = (1-h\lambda) y_1 = (1-h\lambda)^2 y_0$

....

$y_k = (1-h\lambda)^k y_0$

für $k \to \infty$:

$|1-h\lambda| = 1 \Rightarrow y_k = \dots = y_0$ ☹

$|1-h\lambda| < 1 \Rightarrow y_k \to 0$ ☺

$|1-h\lambda| > 1 \Rightarrow y_k \to \infty$ ☹

$\Rightarrow$ Bedingung an h: $|1-h\lambda| < 1 \Rightarrow 0 < h < \frac{2}{\lambda}$ ($\lambda > 0$)

Def ODE heisst steif, falls explizite Methoden einen Zeitschritt h sehr klein brauchen, kleiner als die Genauigkeit verlangt!

Bsp

$$\frac{d}{dt}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} -50 & 49 \\ 49 & -50 \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

$\dot{\underline{y}} = \underline{\underline{B}}\,\underline{y}$ $\quad \underline{\underline{B}}$ symmetrisch: es gibt $\underline{\underline{S}}$ so dass $\quad (\underline{\underline{S}}\,\underline{\underline{S}}^T = \underline{\underline{I}})$

$$\underline{\underline{B}} = \underline{\underline{S}}\,\underline{\underline{D}}\,\underline{\underline{S}}^T$$

mit $\underline{\underline{D}}$ = Diagonalmatrix $\begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_d \end{bmatrix}$

$$\dot{\underline{y}} = \underline{\underline{S}}\,\underline{\underline{D}}\,\underline{\underline{S}}^T\underline{y} \Rightarrow \frac{d}{dt}(\underline{\underline{S}}^T\underline{y}) = \underline{\underline{D}}\,(\underline{\underline{S}}^T\underline{y})$$

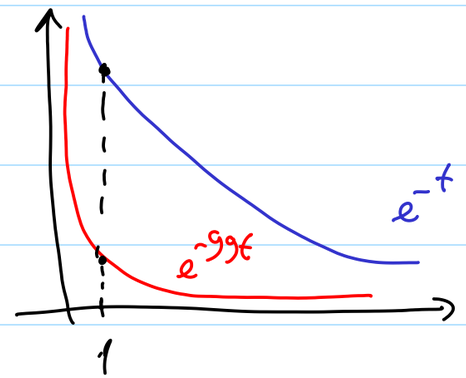
$\underline{\underline{S}}^T\,|$ $\qquad \underline{z} = \underline{\underline{S}}^T\underline{y}$ $\qquad \underline{z}$ $\Big\} \Rightarrow$

$$\dot{\underline{z}} = \underline{\underline{D}}\,\underline{z} \Leftrightarrow \begin{cases} \dot{z}_1 = \lambda_1 z_1 \\ \dot{z}_2 = \lambda_2 z_2 \\ \vdots \\ \dot{z}_d = \lambda_d z_d \end{cases}$$

$$\underline{\underline{D}} = \begin{bmatrix} -1 & 0 \\ 0 & -99 \end{bmatrix}$$

$$\begin{cases} \dot{z}_1 = -z_1 \Rightarrow h < 2 \\ \dot{z}_2 = -99\,z_2 \Rightarrow h < \frac{2}{99} \end{cases}$$

$$\begin{cases} y_1(t) = e^{-t} + e^{-99t} \\ y_2(t) = e^{-t} - e^{-99t} \end{cases}$$



zur Zeit $t = 1$

$y_1(1) = e^{-1} + e^{-99}$

sehr klein ⇒ für die exakte Lösung irrelevant!

aber ein explizites Verfahren will ein sehr kleines h, diktiert genau von e^{-99t} !

(i E) $y_1 = y_0 + h f(y_1) = y_0 - \lambda h y_1 \Rightarrow$

$\Rightarrow (1+\lambda h) y_1 = y_0 \Rightarrow y_1 = \frac{y_0}{1+\lambda h}$

$(h>0, \lambda>0)$

$y_2 = \frac{1}{1+\lambda h} y_1 = \frac{1}{(1+\lambda h)^2} y_0$

$\Rightarrow \dots$

$y_k = \left(\frac{1}{1+\lambda h}\right)^k y_0 \longrightarrow 0$ für $k \to \infty$

$\lambda>0, h>0 \Rightarrow \alpha < \frac{1}{1+\lambda h} < 1$ egal was für h!

Bem Implizite Verfahren stellen keine Bedingung an h.

Bsp explizite Trapezregel:

$\dot{y} = -\lambda y$

$$\begin{cases} k_1 = -\lambda y_0 \\ k_2 = -\lambda(y_0 + h k_1) \end{cases}$$

$y_1 = \underbrace{\left[1 - \lambda h + \frac{1}{2}(\lambda h)^2\right]}_{S(\lambda h)} y_0 = S(\lambda h) y_0$

$\Rightarrow \dots \quad y_k = S(\lambda h)^k y_0 \to 0$ nur wenn

$$|S(\lambda h)| < 1$$

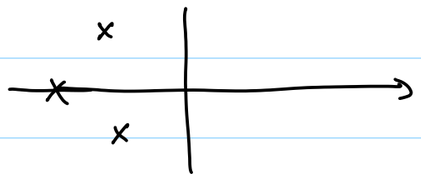
explizite RK (mit s Stufen): $S(z) =$ Polynom vom Grad s

implizite RK: $S(z) = \frac{p(z)}{q(z)}$ mit p, q Polynome vom Grad $\leq s$.

§ 5.2. Stabilität des RK-Verfahrens

Testproblem $\dot{y} = \lambda y$, $\lambda \in \mathbb{C}$, $\mathrm{Re}\,\lambda < 0$

$$y(t) = e^{\lambda t} y_0$$



Bsp

Falls $\lambda = 5i \in \mathbb{C} \Rightarrow y(t) = e^{5it} y_0 = y_0(\cos 5t + i \sin 5t)$

Falls $\lambda = -1 + 5i \in \mathbb{C} \Rightarrow y(t) = e^{-t} y_0(\cos 5t + i \sin 5t)$

$e^{-t} \to 0$ $t \to \infty$

$|y(t)| \to 0$ für $t \to \infty$

Numerische Verfahren: $y_n = S(\lambda h)^n y_0$

Frage: Wann $|y_n| \to 0$ für $n \to \infty$?

$S(z)$ = Stabilitätsfunktion $(z = \lambda h)$

(eE): $S(z) = 1 - z$

(iE): $S(z) = \dfrac{1}{1+z}$

(eTR): $S(z) = 1 - z + \dfrac{1}{2} z^2$

RK-Verfahren mit s Stufen:

$$y_1 = y_0 + h \sum_{i=1}^{s} b_i k_i$$

$$\begin{cases} k_i = f\left(t_0 + c_i h,\ y_0 + h \sum_{j=1}^{s} a_{ij} k_j\right) \\ i = 1, 2, \dots, s \end{cases}$$

c	A
	b

Für das Testproblem $\dot{y} = \lambda y$

$$\begin{cases} k_i = \lambda y_0 + \underbrace{\lambda h}_{z} \sum_{j=1}^{s} a_{ij} k_j \\ i = 1, 2, \dots, s \end{cases}$$

$$\underline{k} = \begin{bmatrix} k_1 \\ \vdots \\ k_s \end{bmatrix}$$

LGS: $\underline{k} = \lambda y_0 \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + z \underline{\underline{A}}\, \underline{k} \quad \Leftrightarrow$

$$\left(\underline{\underline{I}} - z \underline{\underline{A}} \right) \underline{k} = \lambda y_0 \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

falls $\underline{\underline{I}} - z\underline{\underline{A}}$ nicht invertierbar $\rightsquigarrow$

$\underline{\underline{I}} - z\underline{\underline{A}}$ invertierbar $\Rightarrow$

$$\underline{k} = \lambda y_0 \left(\underline{\underline{I}} - z \underline{\underline{A}} \right)^{-1} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \quad \Rightarrow$$

$$y_1 = y_0 + h\lambda y_0 \sum_{i=1}^{s} b_i \left(\left(\underline{\underline{I}} - z \underline{\underline{A}} \right)^{-1} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \right)_i \quad \Rightarrow$$

$$y_1 = y_0 \left(1 + z\, \underline{b}^T \left(\underline{\underline{I}} - z \underline{\underline{A}} \right)^{-1} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \right) \quad \Rightarrow$$

$$S(z) = 1 + z\, \underline{b}^T \left(\underline{\underline{I}} - z \underline{\underline{A}} \right)^{-1} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

Stabilitätsfunktion des s-Stufiges RkV

Theorem ~~Falls $\underline{\underline{I}} - z\underline{\underline{A}}$ invertierbar ist~~, dann ist S eine (komplexwertiges rationale Funktion

$$S(z) = \frac{P(z)}{Q(z)} \quad \text{mit } P, Q \text{ Polynome}$$

von Grad $\leq s$ und $Q(z) = 0$ für $z = \frac{1}{\mu}$ mit μ Eigenwert von $\underline{\underline{A}}$.

Falls RK - explizit, dann $Q(z) \equiv 1$

Beweis

RK-Verfahren explizit: $\underline{\underline{A}} = \begin{bmatrix} 0 & & 0 \\ 0 & \ddots & \\ * & & 0 \end{bmatrix}$ $p \times p$

$$\underline{\underline{A}}^2 = \underline{\underline{A}} \cdot \underline{\underline{A}} \cdots \underline{\underline{A}} = \begin{bmatrix} 0 & \cdots & 0 \\ & & \\ 0 & \cdots & 0 \end{bmatrix}$$

$$\left(\underline{\underline{I}} - z\,\underline{\underline{A}}\right)^{-1} = \underline{\underline{I}} + z\underline{\underline{A}} + (z\underline{\underline{A}})^2 + \ldots + (z\underline{\underline{A}})^{p-1} +$$

$$+ (z\underline{\underline{A}})^p + \ldots$$

($(z\underline{\underline{A}})^p$ und $\ldots$ jeweils $= 0$)

$$= \underline{\underline{I}} + z\underline{\underline{A}} + z^2\underline{\underline{A}}^2 + \ldots + z^{p-1}\underline{\underline{A}}^{p-1}$$

$$= \text{Polynom vom Grad } p-1 \text{ (in } z)$$

$$\underline{v} = \left(\underline{\underline{I}} - z\underline{\underline{A}}\right)^{-1} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \xrightarrow{\text{Cramer'sche Regeln}} v_i(z) = \frac{p_i(z)}{\det\left(\underline{\underline{I}} - z\underline{\underline{A}}\right)}$$

$$Q(z) = 0 \iff \det\left(\underline{\underline{I}} - z\underline{\underline{A}}\right) = 0 \iff z = \frac{1}{\mu}$$

mit $\mu \in EW$ von $\underline{\underline{A}}$.

Konsequenz

1) Falls $\frac{1}{\lambda h}$ nicht $\in EW$ von $\underline{\underline{A}}$, dann ist RK:

$$y_n = S(h\lambda)^n y_0 \quad \text{mit } n = 0, 1, 2, \ldots$$

mit wohldefinierte Stabilitätsfunktion $S(z)$

2) $y_0 = 1 \Rightarrow$ exakte Lösung $y(t) = e^{\lambda t}$

num. Lösung $y_n = S(h\lambda)^n$

RK mit Konvergenzordnung q:

Fehler $\left| e^{n\lambda h} - S(h\lambda)^n \right| \le O\left(|\lambda h|^{q+1}\right)$

$|e^z - S(z)| \leq O(|z|^{q+1}) \Rightarrow$

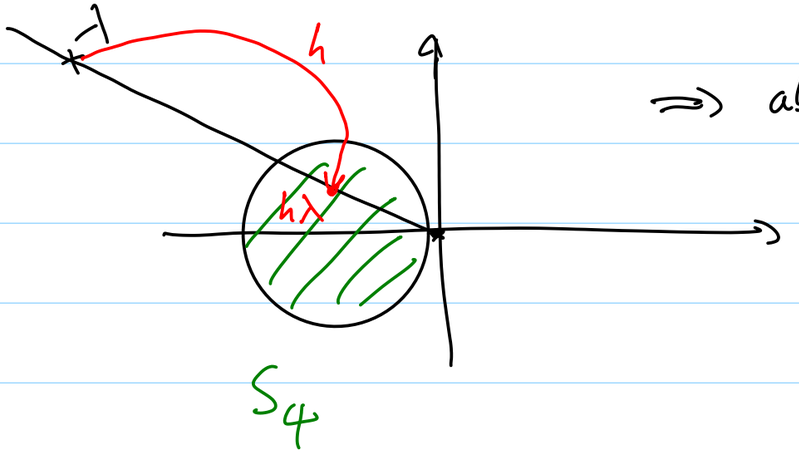
Stabilitätsfunktion $\approx e^z$ und.

Taylorpolynome von e^z und $S(z)$ um $z=0$ identisch bis zum Grad q

Definition Stabilitätsgebiet des Verfahrens Ψ

$$S_\Psi = \{ z \in \mathbb{C} \text{ so dass } |S(z)| < 1 \}$$

$y_n = S(z)^n y_0 \to 0$ nur wenn $|S(z)| < 1$

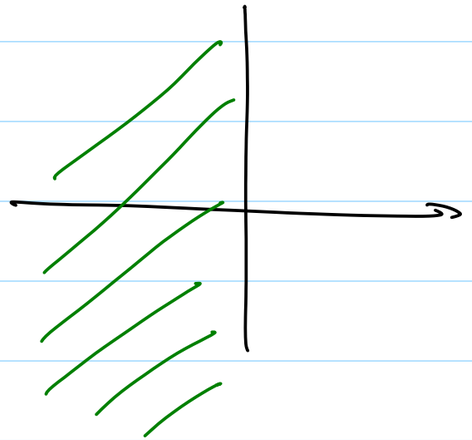


$\Rightarrow$ ablesen, wie klein muss h gewählt werden.

Bem RK explizit $\Rightarrow$ S_Ψ beschränkt $\Rightarrow$ immer eine Schranke an h.

Bem

ode45 / dopri5 verwenden explizite RK $\Rightarrow$ S_Ψ beschränkt $\Rightarrow$ brauchen h klein!

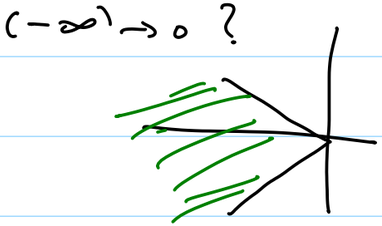


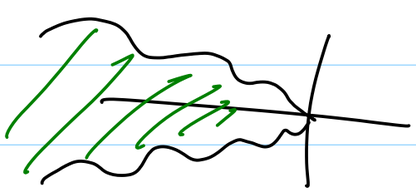
Def Ein Verfahren heisst A-stabil falls

$\{ z \in \mathbb{C} : \mathrm{Re}\, z < 0 \} \subset S_\Psi$.

Def $S(z) \approx e^z$; $z \to -\infty$, $S(-\infty) \to 0$?

Verfahren heisst L-stabil falls $\lim_{z \to -\infty} S(z) = 0$





$\dot{y} = f(y)$ f nicht linear

$y^* \in \mathbb{R}^d$ so dass

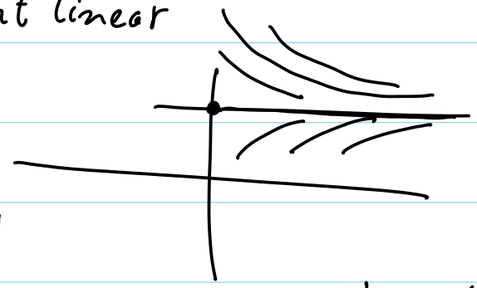
$f(y^*) = 0$ → Stationär!

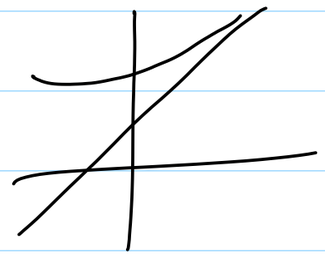
linearisiere

(Taylor um y^* von f)

$\dot{z} = Df(y^*)\, z.$

↳ Testproblem!





Bem RK L-stabil falls $b^T = a_{s\cdot}$ =

= letzte Zeile von A

c	A
1	b^T
	b^T

$c_s = 1$ ⇕

rechte Ecke des Referenzintervals ist ein Quadraturknoten

Solche Verfahren mit maximaler Konvergenzordnung $2s-1$:

Radau-Quadratur / Radau-Verfahren.

Radau-Verfahren von Ordnung 3, 5 → Skript

Bem in jedem Zeitschritt:

noch zu lösen: ein nicht-lineares algebraisches System mit $d \cdot s$ Gleichungen

(s stufen: $k_1, \dots, k_s \in \mathbb{R}^d$)

↳ sehr teuer.

$F(x) = 0$, fsolve mit Startwert

Idee: nicht-lineare alg. System linearisieren!

Ersetze:

$$k_i = f\Big(f_0 + c_i h,\ y_0 + h \sum_{j=1}^{s} a_{ij} k_j\Big)$$

falls autonom

durch:

$$\begin{cases} k_i = f(y_0) + h\, Df(y_0)\Big(\sum_{j=1}^{s} a_{ij} k_j\Big) \\ \uparrow \\ \mathbb{R}^d \end{cases}$$

$f(y_0)$: $\uparrow \mathbb{R}^d$; $Df(y_0)$: $\uparrow \mathbb{R}^{d\times d}$; $\sum_{j=1}^{s} a_{ij} k_j \in \mathbb{R}^d$

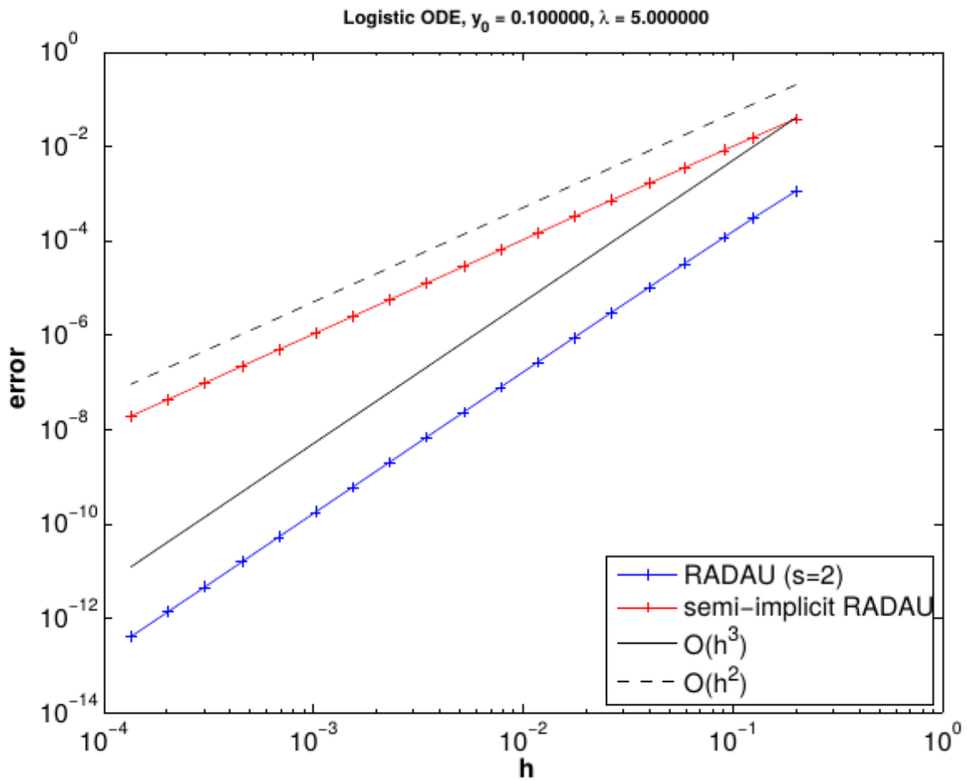
ein grosses lineares Gleichungssystem $\mathbb{R}^{dxs}$

$$Df(x) = \left[\frac{\partial f_k}{\partial x_\ell}(x)\right]_{k,\ell=1,2,\dots,d}$$

Bsp

$$\begin{cases} \dot{y} = \lambda y(1-y) & \lambda = 5 \\ y(0) = 0.1 \end{cases}$$

Skript: man verliert eine Konvergenzordnung!



Idee: Verwende ein Newton-Schritt für das nicht-lineare System $\underline{k} = \underline{f}(y_0 + h\underline{k})$

Bsp (iE) $\underline{y}_1 = \underline{y}_0 + h\underline{f}(\underline{y}_1)$

$$\underline{F}(\underline{z}) = \underline{z} - \underline{y}_0 - h\underline{f}(\underline{z})$$

Löse $\underline{F}(\underline{z}) = \underline{0}$ via Newton-Iteration (teuer ⇓ nur 1 Schritt) mit Startwert $\underline{z}_0 = \underline{y}_0$

$$\underline{\underline{DF}}(y_0) \Rightarrow \quad z_1 = z_0 - \underline{\underline{DF}}(y_0)^{-1}\,\underline{F}(z_0) \Rightarrow$$

$$\boxed{\underline{y}_1 = \underline{y}_0 + \left[\underline{\underline{I}} - h\underline{\underline{Df}}(y_0)\right]^{-1} h\underline{f}(y_0)}$$

Brauche zusätzlich ein sehr guten Startwert für Newton $\Rightarrow$ Ordnung retten!
Das geht bei diagonal-impliziten RK!

$$\underline{\underline{A}} = \begin{bmatrix} & & 0 \\ & \diagup & \\ & & \end{bmatrix} \Rightarrow \text{gestaffelten System}$$

$$\begin{cases} \underline{k}_i = \underline{f}\left(\underline{y}_0 + h\sum_{j=1}^{i} a_{ij}\,\underline{k}_j\right) \end{cases}$$

$$\underline{F}(\underline{k}) = \underline{k} - \underline{f}\left(\underline{y}_0 + \underline{z} + h\,a_{ii}\,\underline{k}\right)$$

wobei $\quad z = h\sum_{j=1}^{i-1} a_{ij}\,\underline{k}_j$

Newton-Schritt:

$$\underline{\underline{DF}}(\underline{k}) = \underline{\underline{I}} - \underline{\underline{Df}}\left(\underline{y}_0 + \underline{z} + h\,a_{ii}\,\underline{k}\right) h\,a_{ii}$$

Startwert $\underline{k}^{(0)}$ in Newton:

$$\underline{k}_i^{(0)} = \sum_{j=1}^{i-1} \frac{d_{ij}}{a_{ij}}\,\underline{k}_j$$

Spezielle Wahl von a_{ij}, d_{ij} rettet die Konvergenzordnung.

linear-implizite RK Rosenbok-Wanner-Methoden (ROW-Methoden)

$$\left(\underline{\underline{I}} - h a_{ii} \underline{\underline{J}}\right) \underline{k}_i = f\left(y_0 + h \sum_{j=0}^{i-1} (a_{ij} + d_{ij}) \underline{k}_j\right) - h \underline{\underline{J}} \sum_{j=1}^{i-1} d_{ij} \underline{k}_j$$

mit

$$\underline{\underline{J}} = \underline{\underline{D}} f\left(y_0 + \boxed{h \sum_{j=1}^{i-1} (a_{ij} + d_{ij}) \underline{k}_j}\right)$$

wie bei vereinfachten Newton lassen wir das weg!

$\Rightarrow$ ROW2, ROW3 $\Rightarrow$ ode23s nicht in standard python.

§6 Nichtlineare algebraische Gleichungen

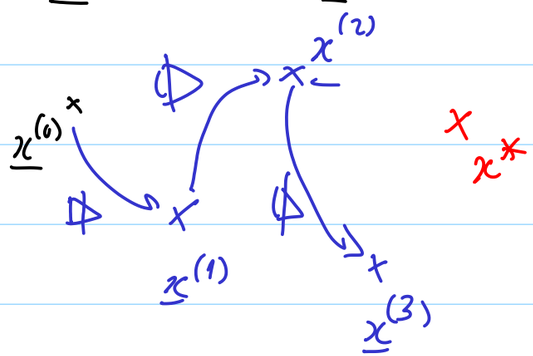
§6.1. Einführung.

Finde $\underline{x}^* \in \mathbb{R}^d$ so dass $\underline{F}(\underline{x}^*) = \underline{0}$

Bsp $d=1$, $F(x) = xe^x - 1$ $\left(: \quad x^* e^{x^*} - 1 = 0\right)$

Iterativ: $\underline{x}^{(0)}, \underline{x}^{(1)}, \underline{x}^{(2)}, \ldots, \underline{x}^{(k)} \to \underline{x}^*$

$\underline{x}^{(0)}$ gegeben $\underline{x}^{(k+1)} = \Phi(\underline{x}^{(k)})$



Def $\underline{x}^{(k+1)} = \Phi(\underline{x}^{(k)})$ heisst

linear konvergent nach $\underline{x}^*$

falls

es gibt $L < 1$ so dass

$$\|\underline{x}^{(k+1)} - \underline{x}^*\| \le L \|\underline{x}^{(k)} - \underline{x}^*\| \quad \text{für alle } k \in \mathbb{N}.$$

Bem

$$\|\underline{x}^{(k+1)} - \underline{x}^*\| \le L\|\underline{x}^{(k)} - \underline{x}^*\| \le L^2 \|\underline{x}^{(k-1)} - \underline{x}^*\| \le \ldots$$

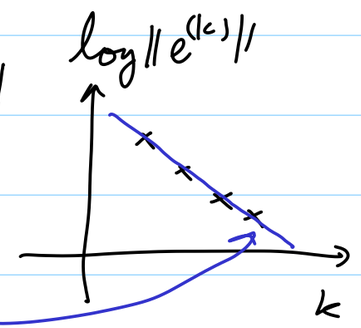
$$\le L^k \|\underline{x}^{(1)} - \underline{x}^*\| \le L^{k+1} \|\underline{x}^{(0)} - \underline{x}^*\|$$

$0 < L < 1$ ⟶ ↳ Konvergenz.

Fehler $\underline{e}_k = \underline{x}^{(k)} - \underline{x}^*$

$$\log \|\underline{e}^{(k)}\| \le k \log L + \log \|e^{(0)}\|$$

"linear" ↳ Steigung !

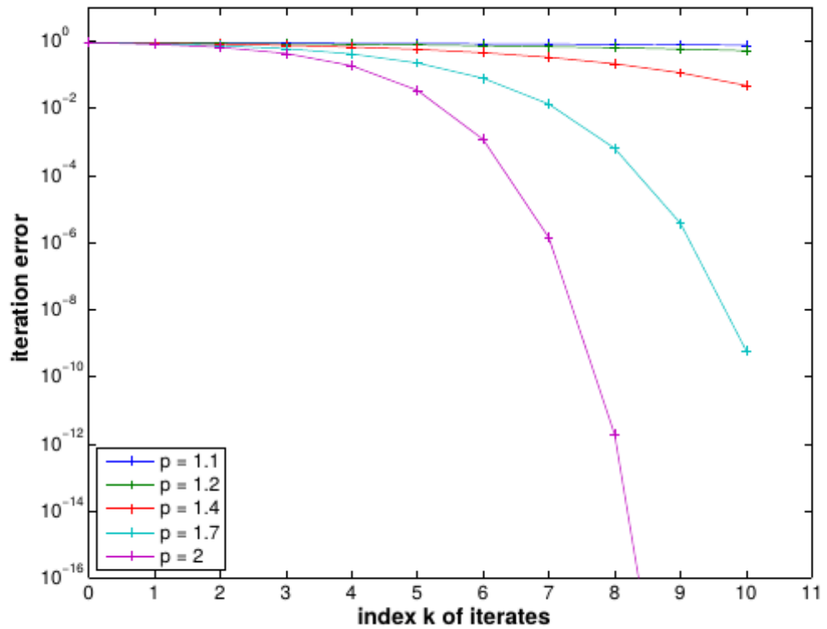


Def Konvergenz der Ordnung p des iterativen Verfahrens:

es gibt eine $C>0$ so dass:

$$\| x^{(k+1)} - x^* \| \leq C \| x^{(k)} - x^* \|^p$$

für alle $k \in \mathbb{N}$, mit $C<1$ für $p=1$.



§6.2. Fixpunktiteration

$F(x^*) = 0 \iff x^* = \Phi(x^*)$ Fixpunkt von Φ

$x^{(k+1)} = \Phi(x^k)$

falls $x^{(k)}$ konvergiert $\downarrow$ $\quad \downarrow \; k \to \infty$

$x^* \qquad \Phi(x^*)$

Bsp 1) $xe^x - 1 = 0 \iff xe^x = 1 \iff x = e^{-x}$

$\Phi_1(x) = e^{-x}$

starte mit $x^{(0)}$, $x^{(k+1)} = \Phi_1(x^{(k)})$

2) $xe^x - 1 = 0 \iff xe^x - 1 + x = x \iff x(e^x + 1) = x + 1 \iff$

$\iff x = \dfrac{x+1}{e^x+1}$

$\Phi_2(x) = \dfrac{x+1}{e^x+1}$

3) $x e^x - 1 = 0 \Rightarrow x e^x - 1 - x = -x \Leftrightarrow x = x + 1 - x e^x$

$\Phi(x) = x + 1 - x e^x$.

k	$x^{(k+1)} := \phi_1(x^{(k)})$	$x^{(k+1)} := \phi_2(x^{(k)})$	$x^{(k+1)} := \phi_3(x^{(k)})$
0	0.500000000000000	0.500000000000000	0.500000000000000
1	0.606530659712633	0.566311003197218	0.675639364649936
2	0.545239211892605	0.567143165034862	0.347812678511202
3	0.579703094878068	0.567143290409781	0.855321409174107
4	0.560064627938902	0.567143290409784	-0.156505955383169
5	0.571172148977215	0.567143290409784	0.977326422747719
6	0.564862946980323	0.567143290409784	-0.619764251895580
7	0.568438047570066	0.567143290409784	0.713713087416146
8	0.566409452746921	0.567143290409784	0.256626649129847
9	0.567559634262242	0.567143290409784	0.924920676910549
10	0.566907212935471	0.567143290409784	-0.407422405542253

?

↓ Lineare Konvergenz

↓ $p=2$ (quadratische) Konvergenz

↓ keine Konvergenz.

Bem Hinreichende Bedingung für lokale lineare Konvergenz:

U konvex, $\Phi: U \subset \mathbb{R}^d \to \mathbb{R}^d$ stetig differentier-bar

$$L = \sup_{x \in U} \| D\Phi(x) \| < 1$$

Wenn $\Phi(x^*) = x^*$ für $x^* \in U$, dann

konvergiert $x^{(k+1)} = \Phi(x^{(k)})$ gegen x^*

lokal mindestens linear.

↳ wir müssen schon nah an x^* starten!

Bem $\Phi: U \subset \mathbb{R} \to \mathbb{R}$ Φ $(m+1)$-mal differentierbar

$\Phi(x^*) = x^* \in U$.

Taylor: $\Phi(y) - \Phi(x) = \sum_{k=1}^{m} \frac{1}{k!} \Phi^{(k)}(x)(y-x)^k +$

$+ O(|y-x|^{m+1})$

Theorem

Voraussetzung: $\Phi^{(l)}(x^*) = 0$ für $l = 1, 2, \ldots, m \geq 1$

Dann konvergiert die Fixpunktiteration $x^{(k+1)} = \Phi(x^{(k)})$ gegen x^* lokal

mit der Ordnung $p \geq m+1$.

Beweis Taylor für $x = x^*$, $y = x^{(k)}$ $\Rightarrow$

$$x^{(k+1)} - x^* = \Phi(x^{(k)}) - \Phi(x^*) =$$

$$= \sum_{k=1}^{m} \frac{1}{k!} \Phi^{(k)}(x^*)(x^{(k)} - x^*)^k + O(|x^{(k)} - x^*|^{m+1})$$

$\Phi^{(k)}(x^*) = 0$

$$\Rightarrow |x^{(k+1)} - x^*| \leq C \cdot |x^{(k)} - x^*|^{m+1}.$$

Bsp $\Phi_2(x) = \frac{x+1}{e^x+1} \Rightarrow \Phi_2'(x) = \frac{e^x+1-(x+1)e^x}{(e^x+1)^2} = \frac{1-xe^x}{(e^x+1)^2}$

x^* Nullstelle von $1-xe^x$ gesucht $\Leftrightarrow$ $1 - x^* e^{x^*} = 0$ $\Rightarrow$

x^*

$\Phi_2'(x^*) = 0 \Rightarrow \Phi_2$ mindestens konvergenzordnung ?

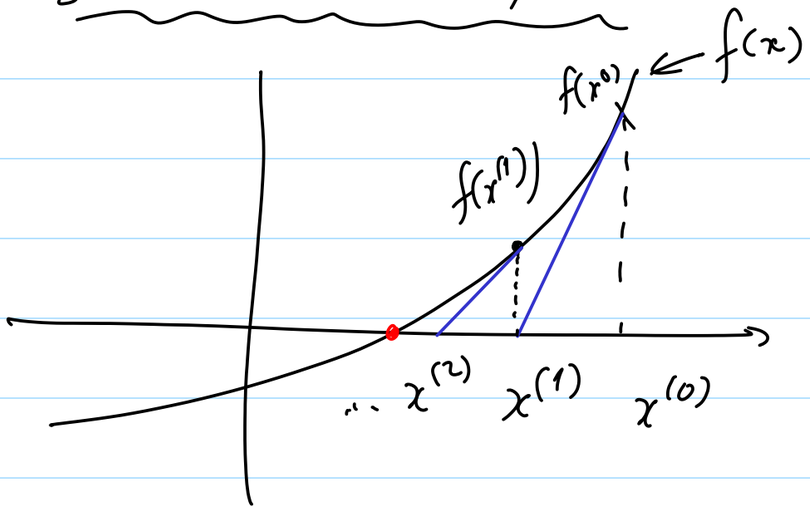
$$\Phi_2'(x^*) = \frac{1 - x^* e^{x^*}}{(e^{x^*}+1)^2} = \frac{0}{(e^{x^*}+1)^2}$$

Für $\Phi_1'(x^*) = -e^{-x^*} \in [-1, 0] \Rightarrow$ nur lineare Konv.

$\Phi_3'(x^*) = -\frac{1}{x^*} < -1$.

§ 6.3. Newton Verfahren

Idee



approximiere f lokal durch eine lineare Funktion

$x^{(0)}$ gegeben

ersetze $f(x) = 0$ durch $\tilde{f}(x) = 0$

Newton: $\tilde{f}(x) = f(x^0) + Df(x^{(0)})(x - x^0)$

Löse $\tilde{f}(x) = 0 \Leftrightarrow f(x^0) + Df(x^{(0)})(x^{(1)} - x^{(0)}) = 0$

(LGS) $Df(x^{(0)})\, x^{(1)} = -f(x^0) + Df(x^{(0)})\, x^{(0)}$

$\Rightarrow$ Newton Iteration:

gegeben $x^{(0)}$

für $k = 0, 1, 2, \ldots$

$$x^{(k+1)} = x^{(k)} - Df(x^{(k)})^{-1} f(x^{(k)})$$

Bem Berechne niemals die Inverse einer Matrix

sondern löse nur LGS

Gauss-Elimination; LU-Zerlegung.

Cholesky-Zerlegung; QR-Zerlegung.

Theorem Newton-Verfahren konvergiert mit Ordnung $p=2$ (lokal).

Beweis

$d=1$

$$x^{(k+1)} = x^{(k)} - f'(x^{(k)})^{-1} f(x^{(k)})$$

Fixpunkt iteration

$$x^* = \Phi(x^*); \quad \Phi(x) = x - f'(x)^{-1} f(x)$$

Nullstellenproblem: $f(x^*) = 0$

$$\Phi'(x) = 1 - \frac{f'(x)f'(x) - f(x)f''(x)}{(f'(x))^2} =$$

$$= \underbrace{1 - 1}_{0} + \frac{f(x)f''(x)}{(f'(x))^2} \Rightarrow$$

$$\Phi'(x^*) = \frac{f(x^*)f''(x^*)}{(f'(x^*))^2} = 0 \quad (\text{nur wenn } f'(x^*) \neq 0).$$

Bem Was tun, falls $f'(x)$ nicht bekannt?

1D: Sekanten verfahren: statt Tangente verwendo die Sekante

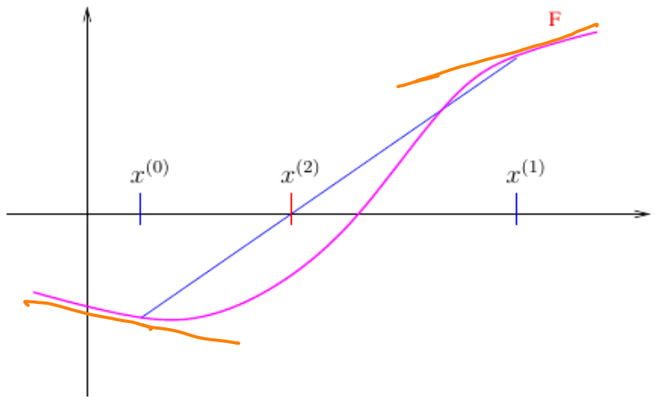
$$f'(x^{(k)}) \approx \frac{f(x^{(k)}) - f(x^{(k-1)})}{x^{(k)} - x^{(k-1)}}$$

Update: $x^{(k+1)} = x^{(k)} - s \qquad (-s = x^{(k+1)} - x^{(k)})$

$$s = \frac{x^{(k)} - x^{(k-1)}}{f(x^{(k)}) - f(x^{(k-1)})} f(x^{(k)})$$

Man braucht 2 Startwerte: $x^{(0)}$ und $x^{(1)}$

Achtung: so einfach nur in 1D.



Beispiel 3.6.3. $F(x) = xe^x - 1\,, \quad x^{(0)} = 0\,,\ x^{(1)} = 5\,.$

k	$x^{(k)}$	$F(x^{(k)})$	$e^{(k)} := x^{(k)} - x^*$	$\frac{\log\lvert e^{(k+1)}\rvert - \log\lvert e^{(k)}\rvert}{\log\lvert e^{(k)}\rvert - \log\lvert e^{(k-1)}\rvert}$
2	0.00673794699909	-0.99321649977589	-0.56040534341070	
3	0.01342122983571	-0.98639742654892	-0.55372206057408	24.43308649757745
4	0.98017620833821	1.61209684919288	0.41303291792843	2.70802321457994
5	0.38040476787948	-0.44351476841567	-0.18673852253030	1.48753625853887
6	0.50981028847430	-0.15117846201565	-0.05733300193548	1.51452723840131
7	0.57673091089295	0.02670169957932	0.00958762048317	1.70075240166256
8	0.56668541543431	-0.00126473620459	-0.00045787497547	1.59458505614449
9	0.56713970649585	-0.00000990312376	-0.00000358391394	1.62641838319117
10	0.56714329175406	0.00000000371452	0.00000000134427	
11	0.56714329040978	-0.00000000000001	-0.00000000000000	

Wir beobachten eine fraktionale Konvergenzordnung p! Man kann $p = \frac{1}{2}(1 + \sqrt{5}) \approx 1.62$ beweisen. Statt Konvergenzordnung 2, wie Newton!

Beispiel 3.6.4. (Lokale Konvergenz des Sekantenverfahrens)

$$F(x) = \arctan(x)$$

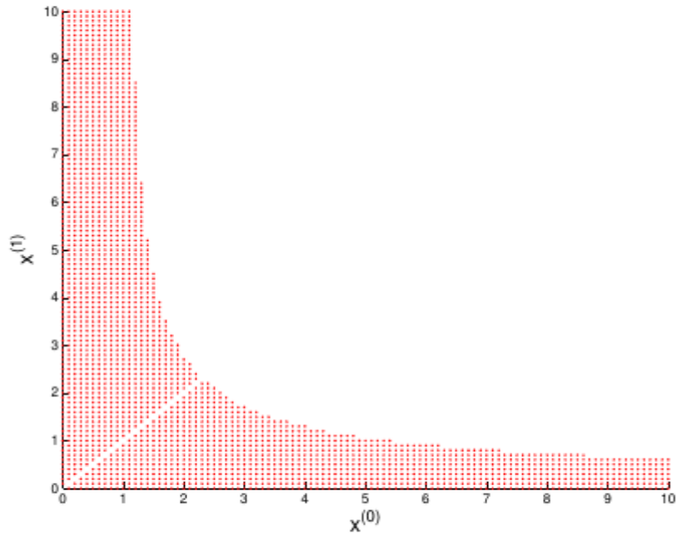


Abb. 3.6.5. $\hat{=}$ Sekantenverfahren konvergiert für ein Paar $(x^{(0)}, x^{(1)})$ als Startwert.

Bem Wahl vom Startwert sehr wichtig.

Beispiel 3.8.1. (Lokale Konvergenz vom Newton-Verfahren)

$$F(x) = xe^x - 1 \implies F'(-1) = 0$$

$$x^{(0)} < -1 \implies x^{(k)} \to -\infty$$
$$x^{(0)} > -1 \implies x^{(k)} \to x^* .$$

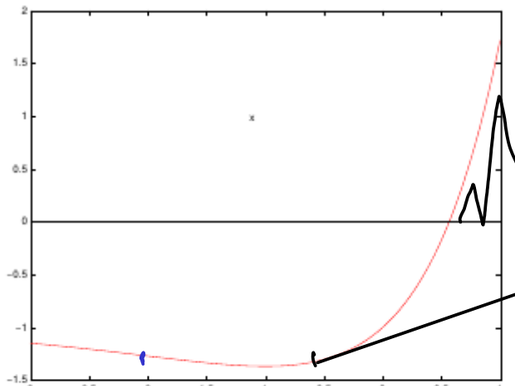
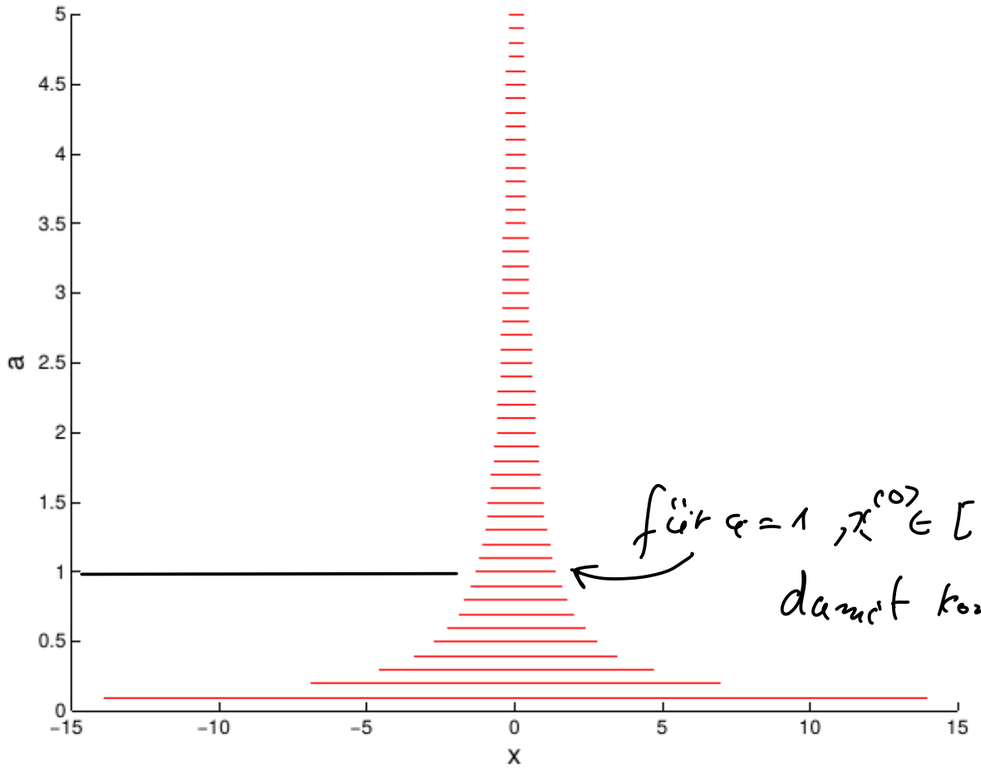


Abb. 3.8.2. Verhalten von $F(x) = xe^x - 1$

(C) V. Gradinaru

$f(x) = \arctan(ax)$ mit $a > 0$ Parameter



Rote Zone $= \{x^{(0)} \in \mathbb{R},\ x^{(k)} \to 0\}$

$x^{(0)}$ muss bereits sehr nah an der Lösung sein. ☹

Bem Bei Divergenz: das Verfahren dämpfen:
erlaube nur kurze Schritte in der Richtung der Tangente

$$x^{(k+1)} = x^{(k)} - \lambda s \quad \text{mit } |\lambda| < 1$$

↳ Wahl von λ ist heuristisch.

Bem Um Rechenzeit zu sparen (in d Dimensionen) kann man auf das vereinfache Newton-Verfahren ausweichen:

$x^{(0)}$; $\underline{\underline{J}} = \underline{\underline{D}} f(x^{(0)})$

$\underline{\underline{J}} = \underline{\underline{L}}\,\underline{\underline{U}}$ kostet $O(d^3)$

verwende für die nächsten Schritte diesen $\underline{\underline{L}}, \underline{\underline{U}}$ ($\Leftrightarrow$ $Df(x^{(0)})$ auch in $x^{(1)}, x^{(2)}, \dots$)

für $k = 0, 1, 2, \dots$

löse $\underline{\underline{L}}\,\underline{\underline{U}}\,\underline{s} = \underline{f}(\underline{x}^{(k)})$ nur $O(d^2)$

$\underline{x}^{(k+1)} = \underline{x}^{(k)} - \underline{s}$

$\underline{\underline{L}}\,\underline{\underline{U}}\,\underline{x} = \underline{b} \Leftrightarrow \underline{\underline{L}}\,\underline{y} = \underline{b}$ mit $\underline{y} = \underline{\underline{U}}\,\underline{x}$

$$\begin{bmatrix} x & x & \dots & x \\ 0 & x & \dots & x \\ 0 & 0 & x \dots & x \\ \vdots & \vdots & \ddots & x \\ 0 & 0 & 0 \dots & x \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{d-1} \\ y_d \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_d \end{bmatrix}$$

$\frac{d^2}{2}$

Rückwärtssubstitution: $y_d = \dots$ $O(d^2)$

$y_{d-1} =$

$\dots$

Dasselbe für $\underline{\underline{U}}\,\underline{x} = \underline{y}$ mit Vorwärtssubstitution.

Typischerweise nur lineare Konvergenz.

Bem Was ist wenn wir für $d>1$ die Ableitung nicht kennen?

1D: $f'(x^{(0)}) \approx \dfrac{f(x^{(k)}) - f(x^{(k-1)})}{x^{(k)} - x^{(k-1)}} = \partial_k$

dD: $\underline{\underline{\partial_k}} \left(\underline{x}^{(k)} - \underline{x}^{(k-1)} \right) = \underline{f}(\underline{x}^{(k)}) - \underline{f}(\underline{x}^{(k-1)})$

$\underline{x}^{(k)} - \underline{x}^{(k-1)} \in \mathbb{R}^d$

$\underline{\underline{\partial_k}}$: $d \times d$ Matrix

Bem Viele mögliche $\underline{\underline{\partial_k}}$!

Broyden: baue $\underline{\underline{\partial_k}}$ iterativ

$$\underline{\underline{\partial_k}} = \underline{\underline{\partial_{k-1}}} + \frac{1}{\|x^{(k)} - x^{(k-1)}\|_2^2} \underline{f}(\underline{x}^{(k)}) \left(\underline{x}^{(k)} - \underline{x}^{(k-1)} \right)^T$$

Vorteil: in jedem Schritt nur ein Mal $\underline{f}$ auszuwerten

Broyden-Verfahren:

$x^{(0)}$ gegeben

$\underline{\underline{\partial_0}} = \underline{\underline{Df}}(x^{(0)})$

für $k = 0, 1, 2, \ldots$

Löse $\underline{\underline{\partial_k}}\, \underline{s} = \underline{f}(\underline{x}^{(k)})$ — teuer

$\underline{x}^{(k+1)} = \underline{x}^{(k)} - \underline{s}$

$\underline{\underline{\partial_{k+1}}} = \underline{\underline{\partial_k}} + \dfrac{1}{\|\underline{s}\|_2^2} \underline{f}(\underline{x}^{(k+1)}) (-\underline{s})^T$

Man kann beweisen:

$$\lim_{k \to \infty} \frac{\|x^{(k+1)} - x^*\|}{\|x^{(k)} - x^*\|} = 0$$

d.h. superlineare Konvergenz

Bessere Implementierung via Sherman-Morrison-Formel

$$\underline{\underline{J}}_{k+1} = \underline{\underline{J}}_k + \frac{1}{\|\underline{s}\|_2^2} \cdot \underline{\underline{R}}$$

$\underline{\underline{R}}$: Rang 1 - Matrix

$$\left(\underline{\underline{A}} + \underline{u}\,\underline{v}^T\right)^{-1} = \underline{\underline{A}}^{-1} - \frac{\underline{\underline{A}}^{-1}\,\underline{u}\,\underline{v}^T\,\underline{\underline{A}}^{-1}}{1+\underline{v}^T\underline{\underline{A}}^{-1}\underline{u}}$$

(einfach überprüfen!)

Rang-1-update braucht nur $\underline{\underline{A}}^{-1}$ (falls $\underline{\underline{A}}$ invertierbar)

In jedem Schritt: $$\underline{\underline{J}}_{k+1}^{-1} = \underline{\underline{J}}_k^{-1} + \frac{\underline{\underline{J}}_k^{-1} f(\underline{x}^{(k+1)})\,\underline{s}^T\,\underline{\underline{J}}_k^{-1}}{\|\underline{s}\|^2 - \underline{s}^T\,\underline{\underline{J}}_k^{-1} f(\underline{x}^{(k+1)})}$$

Iteration: $\underline{\underline{J}}_0 = Df(\underline{x}^{(0)})$

zerlege $\underline{\underline{J}}_0 = \underline{\underline{L}}\,\underline{\underline{U}}$

Löse $\underline{\underline{L}}\,\underline{\underline{U}}\,\underline{s}^{(0)} = f(\underline{x}^{(0)})$

$\underline{x}^{(1)} = \underline{x}^{(0)} - \underline{s}^{(0)}$

$$\underline{\underline{J}}_1^{-1} = \underline{\underline{J}}_0^{-1} + \frac{\underline{\underline{J}}_0^{-1} f(\underline{x}^{(1)})\,\underline{s}^{(0)T}\,\underline{\underline{J}}_0^{-1}}{\|\underline{s}\|^2 - \underline{s}^{(0)T}\,\underline{\underline{J}}_0^{-1} f(\underline{x}^{(1)})}$$

$\underline{x}^{(2)} = \underline{x}^{(1)} - \underline{\underline{J}}_1^{-1} f(\underline{x}^{(1)})$

$\underline{s}^{(1)}$

$\underline{s}^{(1)} = \underline{\underline{J}}_1^{-1} f(\underline{x}^{(1)}) =$

$$= \underline{\underline{J}}_0^{-1} f(\underline{x}^{(1)}) + \frac{\underline{\underline{J}}_0^{-1} f(\underline{x}^{(1)})\, \underline{s}^{(0)T} \underline{\underline{J}}_0^{-1} f(\underline{x}^{(1)})}{\|\underline{s}^{(0)}\|^2 - \underline{s}^{(0)T} \underline{\underline{J}}_0^{-1} f(\underline{x}^{(1)})}$$

$\underline{w}$ $\qquad$ z

$\underline{w} = \underline{\underline{J}}_0^{-1} f(\underline{x}^{(1)})$ berechnet als Lösung von LGS $\underline{\underline{J}}_0 \underline{w} = f(\underline{x}^{(1)})$

$z = \underline{s}^{(0)T} \underline{w} \in \mathbb{R}$

$$\underline{s}^{(1)} = \underline{w} + \frac{\underline{w} z}{\|\underline{s}^{(0)}\|^2 - z} = \left(1 + \frac{z}{\|\underline{s}^{(0)}\|^2 - z}\right) \underline{w}$$

$$\underline{x}^{(2)} = \underline{x}^{(1)} - \underline{s}^{(1)}$$

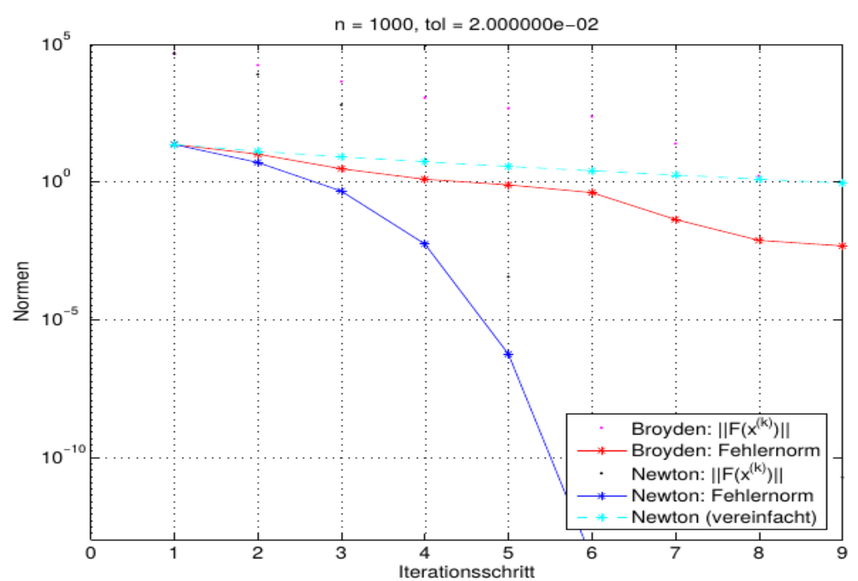
Code 3.9.4: Broyden-Verfahren

```
from numpy.linalg import lu_solve, lu_factor, norm, solve
from numpy import dot, zeros

def fastbroyd(x0, F, J, tol=1e-12, maxit=20):
    x = x0.copy()
    lup = lu_factor(J)
    k = 0; s = lu_solve(lup,F(x))
    x -= s; f = F(x); sn = dot(s,s)
    dx = zeros((maxit,len(x)))
    dxn = zeros(maxit)
    dx[k] = s; dxn[k] = sn
    k += 1; tol *= tol
    while sn > tol and k < maxit:
        w = lu_solve(lup,f)
        for r in range(1,k):
            w +=  dx[r]*( dot(dx[r-1],w) )/dxn[r-1]
        z = dot(s,w)
        s = (1+z/(sn-z))*w
        sn = dot(s,s)
        dx[k] = s; dxn[k] = sn
        x -= s; f = F(x); k+=1

    return x, k
```

$f(x^{(0)})$; $J = Df(\underline{x}^{(0)}) = LU$ wieder verwende! bildig!



Software: optimize.fsolve

↳ auf eine andere Methode.

```
In [1]: import numpy as np
        from scipy import optimize
        import matplotlib.pyplot as plt

        optimize.bisect(lambda x: np.exp(x) - 2, -2, 2)
```

```
In [4]: x_root_guess = 2
        f = lambda x: np.exp(x) - 2
        fprime = lambda x: np.exp(x)
        print(optimize.newton(f, x_root_guess))
        print(optimize.newton(f, x_root_guess, fprime=fprime))
```

```
0.69314718056
0.69314718056
```

```
In [6]: def f(x):
            return [x[1] - x[0]**3 - 2 * x[0]**2 + 1, x[1] + x[0]**2 - 1]
        optimize.fsolve(f, [1, 1])
```

```
Out[6]: array([ 0.73205081,  0.46410162])
```

```
In [7]: import sympy
        x, y = sympy.symbols("x, y")
        f_mat = sympy.Matrix([y - x**3 -2*x**2 + 1, y + x**2 - 1])
        f_mat.jacobian(sympy.Matrix([x, y]))
        def f_jacobian(x):
            return [[-3*x[0]**2-4*x[0], 1], [2*x[0], 1]]
        optimize.fsolve(f, [1, 1], fprime=f_jacobian)
```

```
Out[7]: array([ 0.73205081,  0.46410162])
```



```
In [9]: def f(r):
            return 2 * np.pi * r**2 + 2 / r
        r_min = optimize.brent(f, brack=(0.1, 4))
        r_min, f(r_min)
```

```
Out[9]: (0.54192607725571351, 5.5358104459320856)
```

```
In [10]: def f(X):
             x, y = X
             return (4 * np.sin(np.pi * x) + 6 * np.sin(np.pi * y)) + (x - 1)**2 + (y - 1)**2
```

```
In [11]: x_start = optimize.brute(f, (slice(-3, 5, 0.5), slice(-3, 5, 0.5)), finish=None)
         x_start, f(x_start)
```

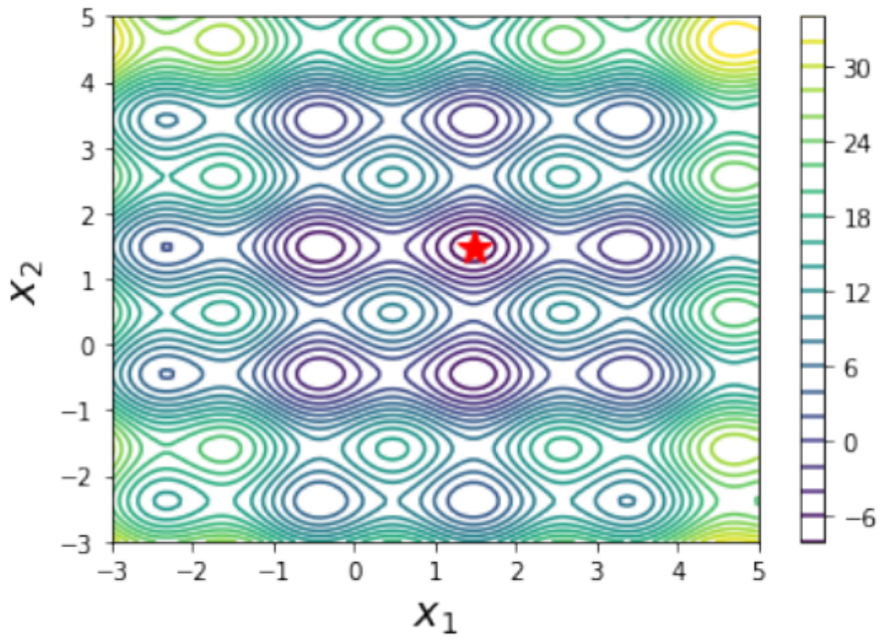
```
Out[11]: (array([ 1.5,  1.5]), -9.5)
```

```
In [12]: x_opt = optimize.fmin_bfgs(f, x_start)
         x_opt, f(x_opt)
```

```
Optimization terminated successfully.
         Current function value: -9.520229
         Iterations: 4
         Function evaluations: 28
         Gradient evaluations: 7
```

```
Out[12]: (array([ 1.47586906,  1.48365787]), -9.5202292730550155)
```

```
In [13]: def func_X_Y_to_XY(f, X, Y):
             """
             Wrapper for f(X, Y) -> f([X, Y])
             """
             s = np.shape(X)
             return f(np.vstack([X.ravel(), Y.ravel()])).reshape(*s)
         fig, ax = plt.subplots(figsize=(6, 4))
         x_ = y_ = np.linspace(-3, 5, 100)
         X, Y = np.meshgrid(x_, y_)
         c = ax.contour(X, Y, func_X_Y_to_XY(f, X, Y), 25)
         ax.plot(x_opt[0], x_opt[1], 'r*', markersize=15)
         ax.set_xlabel(r"$x_1$", fontsize=18)
         ax.set_ylabel(r"$x_2$", fontsize=18)
         plt.colorbar(c, ax=ax)
         plt.show()
```




```
In [14]: result = optimize.minimize(f, x_start, method='BFGS')
         x_opt = result.x
         x_opt, f(x_opt)

Out[14]: (array([ 1.47586906,  1.48365787]), -9.5202292730550155)
```

§7 Intermezzo LA

$$\underline{\underline{A}} = \begin{bmatrix} | & | & \dots & | \\ \underline{a}_1 & \underline{a}_2 & \dots & \underline{a}_n \\ | & | & \dots & | \end{bmatrix}$$

$\underline{a}_i$; $\underline{\underline{A}}_{:,i}$

$$\underline{\underline{A}}\,\underline{x} = \sum_{i=1}^{n} x_i \, \underline{\underline{A}}_{:,i} \qquad \underline{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_n \end{bmatrix}$$

$$\underline{q}^T = [\, q_1 \dots q_n \,], \qquad \underline{q}^H = [\, \bar{q}_1 \dots \bar{q}_n \,]$$

$$\underline{\underline{Q}} = [\, \underline{q}_1 \; \underline{q}_2 \dots \underline{q}_n \,] \quad \text{orthogonal / unitär}$$

$$\underline{\underline{Q}}^H \underline{\underline{Q}} = \underline{\underline{I}} \qquad \underline{q}_1, \dots, \underline{q}_n \text{ orthonormal}$$

Unitäre Transformationen erhalten Winkel und Längen

$$\|\underline{\underline{Q}}\,\underline{x}\|_2^2 = (\underline{\underline{Q}}\,\underline{x})^H (\underline{\underline{Q}}\,\underline{x}) = \underline{x}^H \underline{\underline{Q}}^H \underline{\underline{Q}}\,\underline{x} = \underline{x}^H \underline{x} = \|\underline{x}\|_2^2$$

Orthogonale Transformationen in $\mathbb{R}^n$:

Rotationen, Spiegelungen, Permutation

Ben Gegeben $\underline{a}_1, \underline{a}_2, \dots, \underline{a}_n$ lin. unabhängige Vektoren $\Rightarrow$ ONB in $\operatorname{span}\{\underline{a}_1, \underline{a}_2, \dots, \underline{a}_n\}$ mit Gram-Schmidt

↳ Wichtig: modifizierte Version.

§7.1. Wichtigsten Matrixzerlegungen

1) LU- oder LR-Zerlegung: Gauss-Elimination

$$\underline{\underline{A}} = \underline{\underline{L}} \cdot \underline{\underline{R}} = \begin{bmatrix} 1 & & 0 \\ * & \ddots & \\ & & 1 \end{bmatrix} \cdot \begin{bmatrix} * & & \\ & \ddots & \\ 0 & & * \end{bmatrix}$$

Ben Falls $\underline{\underline{A}}$ symmetrisch $\Rightarrow$

$$\underline{\underline{A}} = \underbrace{\begin{bmatrix} 1 & & 0 \\ * & \ddots & \\ & & 1 \end{bmatrix}}_{\underline{\underline{L}}} \underbrace{\begin{bmatrix} * & & 0 \\ & \ddots & \\ 0 & & * \end{bmatrix}}_{\underline{\underline{D}}} \underbrace{\begin{bmatrix} 1 & & * \\ & \ddots & \\ 0 & & 1 \end{bmatrix}}_{\underline{\underline{L}}^T} = \underline{\underline{L}}\,\underline{\underline{D}}\,\underline{\underline{L}}^T$$

$\underline{\underline{A}}$ symmetrisch, pos. definit :

$$\underline{\underline{A}} = \underline{\underline{L}}\,\sqrt{\underline{\underline{D}}}\;\sqrt{\underline{\underline{D}}}\,\underline{\underline{L}}^T = \underline{\underline{M}}\,\underline{\underline{M}}^T \quad \text{Cholesky-Zerlegung.}$$

Gauss-Elimination $\underline{\underline{P}}\,\underline{\underline{A}} = \underline{\underline{L}}\,\underline{\underline{R}}$

2) $\underline{\underline{Q}}\,\underline{\underline{R}}$-Zerlegung. $\underline{\underline{A}} = \underline{\underline{Q}}\,\underline{\underline{R}}$ → obere Dreiecksmatrix

↳ orthogonale Matrix.

$\underline{\underline{A}} : \mathbb{R}^n \to \mathbb{R}^m \qquad \underline{\underline{A}}\,\underline{x} \in \mathbb{R}^m$

ass. Norm fü Matrizen : $\|\underline{\underline{A}}\| = \sup_{\underline{x} \neq 0} \frac{\|\underline{\underline{A}}\underline{x}\|}{\|\underline{x}\|}$

$\underline{\underline{A}}$ quadratisch

Def $\operatorname{cond}(\underline{\underline{A}}) = \|\underline{\underline{A}}^{-1}\|\,\|\underline{\underline{A}}\|$

↳ Indikator ob das Lösen von LGS $\underline{\underline{A}}\,\underline{x} = \underline{b}$ mit Gauss-Elimination anfällig an Rundungsfehler ist oder nicht.

$\underline{\underline{A}}\,\underline{x} = \underline{b} \Leftrightarrow \underline{\underline{Q}}\,\underline{\underline{R}}\,\underline{x} = \underline{b} \Rightarrow$ (↑ keine Rundungsfehler-akumulation)

$\underline{\underline{Q}}^H$ |

$\Rightarrow \underline{\underline{R}}\,\underline{x} = \underline{\underline{Q}}^H \underline{b}$ und Rückwärtssubstitution.

→ viel stabilere Art LGS zu lösen (kostet auch circa 3 Mal mehr als via LU).

3) Singulärwertzerlegung (SVD) $\underline{\underline{A}}$ beliebig

$$\underline{\underline{A}} = \underline{\underline{U}}\,\underline{\underline{\Sigma}}\,\underline{\underline{V}}^H$$

$\underline{\underline{U}}$: orthonormale Spalten, $\underline{\underline{V}}^H$: orth. Zeilen

$$\underline{\underline{\Sigma}} = \begin{bmatrix} \sigma_1 & & 0 \\ & \sigma_2 & \\ & & \ddots \\ 0 & & 0 \end{bmatrix}$$

$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$

mit $r = \operatorname{Rang}(\underline{\underline{A}})$

4) Schur-Zerlegung. $\underline{\underline{A}}$ beliebig quadratisch

$\Rightarrow$ es gibt $\underline{\underline{U}}$ unitär, so dass

$$\underline{\underline{U}}^H \underline{\underline{A}}\, \underline{\underline{U}} = \underline{\underline{T}} \quad \text{obere Dreiecksmatrix.}$$

5) Polarzerlegung $\underline{\underline{A}} = \underline{\underline{Q}}\, \underline{\underline{H}}$

$\underline{\underline{Q}}$: orthonormal; $\underline{\underline{H}}$: symmetrisch, pos. definit

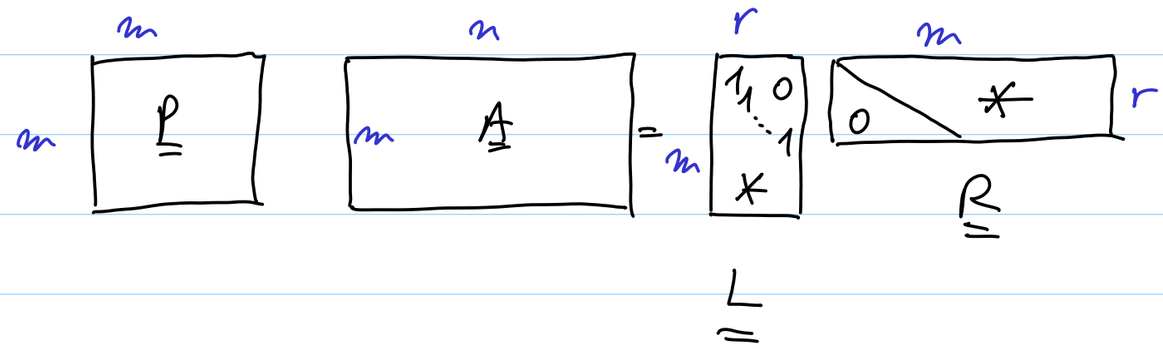
6) $\underline{\underline{A}}$ $n \times n$ Matrix, n lin. unabhängigen $\in V \Rightarrow$

$\underline{\underline{A}} = \underline{\underline{S}}\, \underline{\underline{\Lambda}}\, \underline{\underline{S}}^{-1}$, $\underline{\underline{\Lambda}} = \operatorname{diag}(\in \mathbb{K})$

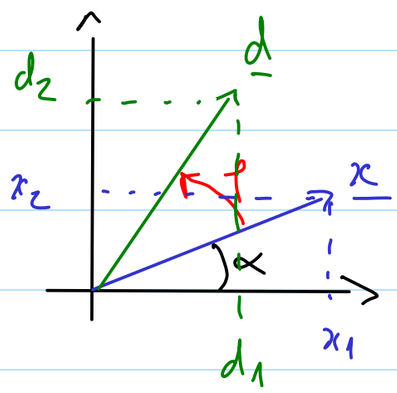
$\hookrightarrow \in V$ als Spalten.

7) $\underline{\underline{A}}$ $n \times n$ symmetrisch $\Rightarrow \underline{\underline{A}} = \underline{\underline{Q}}\, \underline{\underline{\Lambda}}\, \underline{\underline{Q}}^H$ ($\underline{\underline{Q}}$ orthogonal)

LU-Zerlegung: Zeilenstufenform für $m \times n$ Matrix vom Rang $r < m$:



§ 7.2. Orthogonale Matrizen



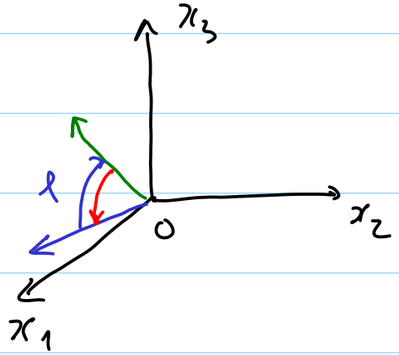
$$\underline{\underline{D}}(\ell) = \begin{bmatrix} \cos\ell & -\sin\ell \\ \sin\ell & \cos\ell \end{bmatrix}$$

$\underline{d} = \underline{\underline{D}}(\ell)\, \underline{x}$

Nützlich $\underline{\underline{D}}(-\ell) = \begin{bmatrix} \cos\ell & \sin\ell \\ -\sin\ell & \cos\ell \end{bmatrix} = \underline{\underline{D}}(\ell)^T$

$G(\ell) =$ Givens-Rotation

Drehung in der $x_1 O x_3$ - Ebene?



$$\underline{\underline{D}}(\ell) = \begin{bmatrix} \cos\ell & 0 & +\sin\ell \\ 0 & 1 & 0 \\ -\sin\ell & 0 & \cos\ell \end{bmatrix}$$

i j

$$\underline{\underline{D}}_{ij}(-\ell) = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \cos\ell & \cdots & +\sin\ell & \\ & & \vdots & 1 & \vdots & \\ & & -\sin\ell & \cdots & \cos\ell & \\ & & & & & 1 \end{bmatrix} \begin{matrix} \\ \\ \leftarrow i \\ \\ \leftarrow j \\ \\ \end{matrix}$$

i j

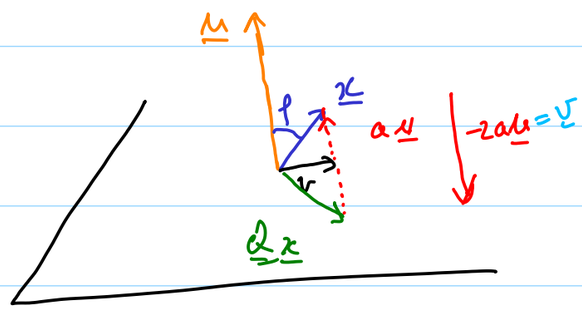
Permutationsmatrix.

$$\begin{bmatrix} 1 & & & & 0 \\ & \ddots & & & \\ & & 1 & & \\ & & & \ddots & \\ 0 & & & & 1 \end{bmatrix} \xrightarrow{\underline{\underline{P}}} \begin{bmatrix} 1 & & & & 0 \\ & \ddots & & & \\ & & 0 & 1 & \\ & & 1 & 0 & \\ & & & & \ddots \\ 0 & & & & 1 \end{bmatrix} \begin{matrix} \\ \\ \leftarrow \\ \leftarrow j \\ \\ \\ \end{matrix}$$

Bsp Spiegelung

$\underline{u} \perp$ Ebene, $\|\underline{u}\|_2 = 1$

ℓ = Winkel zwischen $\underline{x}, \underline{u}$



$$\cos\ell = \frac{a}{\|x\|}$$

$$a = \|x\| \cos\ell$$

$$\underline{Q}\underline{x} = \underline{x} + (-2a\underline{u}) = \underline{x} - 2a\underline{u}$$

$$\underline{u}^T\underline{x} = \langle \underline{u}, \underline{x} \rangle = \|\underline{u}\| \cdot \|\underline{x}\| \cos\ell = \|\underline{u}\| \cdot a = a$$

$$\underline{\underline{Q}}\underline{x} = \underline{x} - 2(\underline{u}^T\underline{x})\underline{u} = \underline{x} - 2\underline{u}(\underline{u}^T\underline{x}) = \underline{x} - 2\underline{u}\underline{u}^T\underline{x}$$

$$= \left(\underline{\underline{I}} - 2\underline{u}\underline{u}^T\right)\underline{x}$$

$$\underline{\underline{P}}_{\underline{u}}$$

$$\underline{\underline{Q}} = \underline{\underline{I}} - 2\underline{\underline{P}}_{\underline{u}}$$

Householdermatrix.

§7.3. QR-Zerlegung

(I)

Erste Möglichkeit: via (modifizierten) Gram-Schmidt.

$\underline{\underline{A}} = [\underline{v}_1 \ \underline{v}_2 \dots \underline{v}_n] \rightsquigarrow \underline{q}_1, \dots \underline{q}_n$ ONB

$\underline{q}_1 := \frac{1}{\|\underline{v}_1\|} \underline{v}_1 \Rightarrow \underline{v}_1 = \|\underline{v}_1\| \underline{q}_1 = \langle \underline{q}_1, \underline{v}_1 \rangle \underline{q}_1$

$\underline{v}_1 = \langle q_1, v_1 \rangle q_1 + 0 q_2 + \dots + 0 q_n$

$c_2 = v_2 - P_{q_1} v_2 = v_2 - \langle q_1, v_2 \rangle q_1$

$v_2 = \langle q_1, v_2 \rangle q_1 + \langle q_2, v_2 \rangle q_2 + 0 \cdot q_3 + \dots + 0 q_n$

$q_2 := \frac{1}{\|c_2\|} c_2$

$c_3 := v_3 - P_{\text{span}\{q_1, q_2\}} v_2 = v_3 - \langle q_1, v_3 \rangle q_1 - \langle q_2, v_3 \rangle q_2$

$v_3 = \langle q_1, v_3 \rangle q_1 + \langle q_2, v_3 \rangle q_2 + \langle q_3, v_3 \rangle q_3 + 0 \cdot q_4 + \dots + 0 q_n$

$q_3 := \frac{1}{\|c_3\|} c_3$

...

...

$c_k = v_k - P_{\text{span}\{q_1, \dots, q_{k-1}\}} v_k = v_k - \langle q_1, v_k \rangle q_1 - \dots - \langle q_{k-1}, v_k \rangle q_{k-1}$

$v_k = \langle q_1, v_k \rangle q_1 + \dots + \langle q_k, v_k \rangle q_k + 0 q_{k+1} + \dots + 0 q_n$

$q_k = \frac{1}{\|c_k\|} c_k$ ...

Im k-ten Schritt:

$$\underline{v}_k = (\underline{q}_1^H \underline{v}_k)\,\underline{q}_1 + (\underline{q}_2^H \underline{v}_k)\,\underline{q}_2 + \dots + (\underline{q}_k^H \underline{v}_k)\,\underline{q}_k$$

$\underline{v}_k$ = lineare Kombination von $\underline{q}_1, \underline{q}_2, \dots, \underline{q}_k$

mit Koeff $(\underline{q}_1^H \underline{v}_k), \dots, (\underline{q}_k^H \underline{v}_k) \in \mathbb{C}$

$$\begin{bmatrix} \\ \underline{v}_k \\ \\ \end{bmatrix} = \begin{bmatrix} \underline{q}_1 & \cdots & \underline{q}_k \end{bmatrix} \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

das für jedes $k = 1, 2, \dots, n$

$$n \begin{bmatrix} \underline{v}_1 \, \underline{v}_2 \cdots \underline{v}_k \end{bmatrix} = \begin{bmatrix} \underline{q}_1 \cdots \underline{q}_k \end{bmatrix} \begin{bmatrix} \underline{q}_1^H \underline{v}_1 & \underline{q}_1^H \underline{v}_2 & \cdots & \underline{q}_1^H \underline{v}_k \\ 0 & \underline{q}_2^H \underline{v}_2 & \cdots & \underline{q}_2^H \underline{v}_k \\ \vdots & \ddots & & \vdots \\ 0 & 0 & & \underline{q}_k^H \underline{v}_k \end{bmatrix} \begin{matrix} \} k \\ \\ \\ \\ \end{matrix}$$

(k über der linken Matrix, k über der rechten Matrix, k unter der rechten Matrix)

$$n \underbrace{\begin{bmatrix} & & \\ & A_k & \\ & & \end{bmatrix}}_{k} = \underbrace{\begin{bmatrix} & & \\ & Q_k & \\ & & \end{bmatrix}}_{k} \begin{bmatrix} * & & * \\ & \ddots & \\ 0 & & * \end{bmatrix} \} k$$

↳ Spalten sind orthonormale Vektoren.

Nach n Schritten.

$$\begin{bmatrix} \underline{v}_1 \cdots \underline{v}_n \end{bmatrix} = \begin{bmatrix} \underline{q}_1 \cdots \underline{q}_n \end{bmatrix} \begin{bmatrix} * & & * \\ & \ddots & \\ 0 & & * \end{bmatrix} \leftarrow j$$

(i über der rechten Matrix, $\underline{q}_j^H \underline{v}_i$ rechts daneben)

Bem Algorithmus entspricht zu Multiplikationen mit oberen Dreiecksmatrizen.:

$$\underline{\underline{A}}\,\underbrace{\underline{\underline{R}}_1\,\underline{\underline{R}}_2 \cdots \underline{\underline{R}}_n}_{\underline{\underline{R}}^{-1}} = \underline{\underline{Q}} \quad \Rightarrow \quad \underline{\underline{A}} = \underline{\underline{Q}}\,\underline{\underline{R}}$$

$$r_{ij} = q_i^H v_j$$

$$\underline{\underline{R}}_1 = \begin{bmatrix} \frac{1}{r_{11}} & -\frac{r_{12}}{r_{11}} & -\frac{r_{13}}{r_{11}} & \cdots \\ & 1 & & \\ & & 1 & \\ 0 & & & \ddots \\ & & & & 1 \end{bmatrix}$$

$$\underline{\underline{R}}_2 = \begin{bmatrix} 1 & & & \\ & \frac{1}{r_{22}} & -\frac{r_{23}}{r_{22}} & \cdots \\ & & 1 & \\ & & & \ddots \\ & & & & 1 \end{bmatrix} \qquad \underline{\underline{R}}_3 = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \frac{1}{r_{33}} & \cdots \\ & & & 1 \\ & & & & \ddots \end{bmatrix}$$

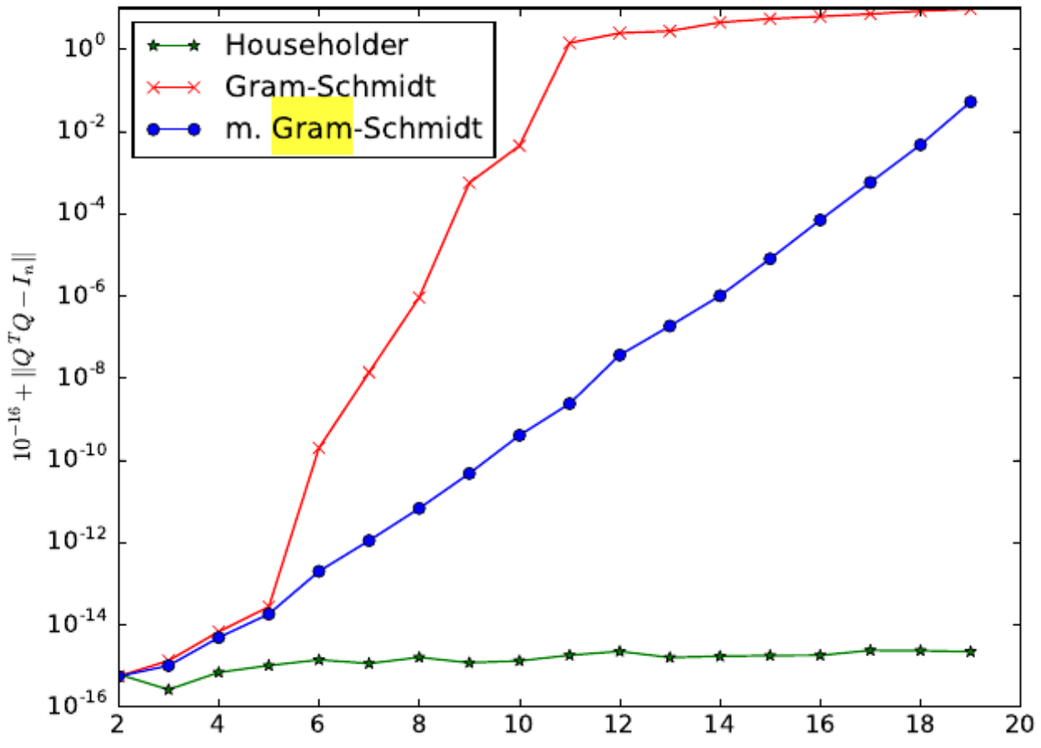
Bem Gram-Schmidt produziert eine orthogonale Matrix $\underline{\underline{Q}}$ indem man in jeden Schritt mit einer oberen Dreiecksmatrix multipliziert.

$\Longrightarrow$ anfällig an Rundungsfehler !

Bem Ein Vorteil von Gram-Schmidt ist, wenn man nach k Schritte aufhört, hat man bereits $\underline{q}_1, \ldots, \underline{q}_k$ orthogonal

$$\text{span}\{\underline{q}_1, \ldots, \underline{q}_k\} = \text{span}\{\underline{v}_1, \ldots, \underline{v}_k\}$$

$$\mathbf{A} = \begin{bmatrix} t_0^{n-1} & \dots & t_0^1 & 1 \\ t_1^{n-1} & \dots & t_1^1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ t_{19}^{n-1} & \dots & t_{19}^1 & 1 \end{bmatrix}, \text{ mit } \quad t_i = \frac{i}{(m-1)}.$$



classisches GS.:

$P_{q_{k-1}} v_k = \left(P_{q_{k-1}} \cdots P_{q_2} P_{q_1} v_k \right)$ → alle gleichzeitig

↓ nur auf q_k projeziert.

sofort $q_1 \dots, q_{k-1}$ verwenden.

Projektion aller folgenden Spalten auf das neu gefundene q_i

classisch. $a_1 \dots a_n$

für $j = 1, \dots n$:

$v_j = a_j$

für $i = 1, 2 \dots, j-1$:

$r_{ij} = q_i^H a_j$

$v_j = v_j - r_{ij} q_i$

$r_{jj} = \|v_j\|$

$q_j = \frac{v_j}{r_{jj}}$

für $j = 1, \dots, n$:

$v_j = a_j$

für $i = 1, \dots, n$

$r_{ii} = \|v_i\|$; $q_i = \frac{1}{r_{ii}} v_i$

für $j = i+1, \dots, n$:

$r_{ij} = q_i^H v_j$

$v_j = v_j - r_{ij} q_i$

Fazit:

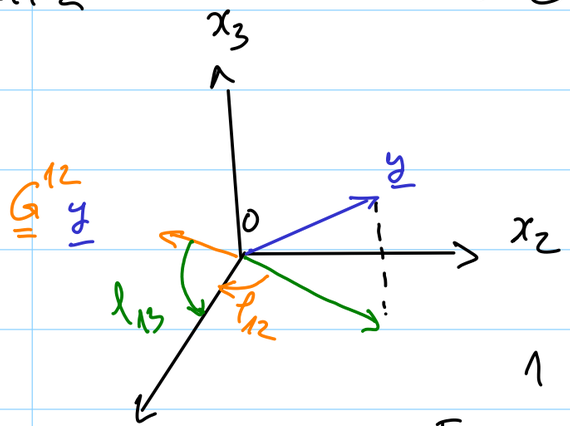
1) classisches Gram-Schmidt ist nur für analytische Zwecke zu verwenden!
2) modifiziertes Gram-Schmidt ist nur für "kleine" Matrizen oder nur für einige Vektoren zu verwenden zu verewenden!
3) sonst II, III die folgen!

II QR via Rotationen

$$\underset{m}{\underline{\underline{A}}} = \underset{m}{\underline{\underline{Q}}} \; \underset{m}{\begin{bmatrix} * \\ & \ddots \\ & & * \\ 0 & & \end{bmatrix}} = \underline{\underline{Q}}\,\underline{\underline{R}}$$

$$\underset{m}{\underline{\underline{A}}} = \underset{m}{\underline{\underline{Q}}} \; \underset{m}{\begin{bmatrix} * & \cdots & * \\ & \ddots & \vdots \\ 0 & & * \end{bmatrix}} = \underline{\underline{Q}}\,\underline{\underline{R}}$$

Idee: Erzeuge die 0 unterhalb der Hauptdiagonale mittels orthogonale Matrizen → Drehungen!



$$\underline{\underline{A}} = \begin{bmatrix} y_1 & \tilde{z}_1 \\ y_2 & \tilde{z}_2 \\ y_3 & \tilde{z}_3 \end{bmatrix}$$

$$\underline{\underline{G}}^{12}(\ell_{12}) = \begin{bmatrix} \cos\ell_{12} & \sin\ell_{12} & 0 \\ -\sin\ell_{12} & \cos\ell_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{matrix} 1 \\ 2 \end{matrix}$$

Givens-Drehung, die y_2 zu 0 macht

$$\underline{\underline{G}}^{12}(\ell_{12})\,\underline{\underline{A}} = \begin{bmatrix} * & * & \cdots \\ 0 & * & \vdots \\ y_3 & \tilde{z}_3 & \cdots \end{bmatrix}$$

$$\underline{\underline{G}}^{13}(\vartheta_{13}) = \begin{bmatrix} \cos\vartheta_{13} & 0 & \sin\vartheta_{13} \\ 0 & 1 & 0 \\ -\sin\vartheta_{13} & 0 & \cos\vartheta_{13} \end{bmatrix} \begin{matrix} 1 \\ \\ 3 \end{matrix}$$

(1 3)

→ Givens-Drehung, die y_3 zu 0 macht

$$\underbrace{\underline{\underline{G}}^{13}\,\underline{\underline{G}}^{12}}_{=:\underline{\underline{Q}}^{1}} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} * \\ 0 \\ 0 \end{bmatrix}$$

orthogonale Matrizen

$$\underline{\underline{Q}}^{1}\underline{\underline{A}} = \begin{bmatrix} * & x & x & \cdots & x \\ 0 & x & x & \cdots & x \\ 0 & x & x & \cdots & x \\ \vdots & \vdots & - & - & - \\ 0 & x & x & \cdots & x \end{bmatrix}$$

$$\underline{\underline{Q}}^{2} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & & & & \\ \vdots & & \text{Drehungen um } x \text{ zu erzeugen} & & \\ 0 & & & & \end{bmatrix}$$

$\underline{y} \in \mathbb{R}^2$

$$\underbrace{\underline{\underline{G}}^{1n} \cdots \underline{\underline{G}}^{13}\,\underline{\underline{G}}^{12}}_{=:\underline{\underline{Q}}^{1}}\,\underline{y} = \begin{bmatrix} * \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Dann kommt die zweite Spalte von $\underline{\underline{A}}$ dran:

$$\begin{bmatrix} x & z_1 & x \\ 0 & z_2 & x \\ 0 & z_3 & x \end{bmatrix}$$

$\underline{\underline{G}}^{23}$ um z_3 zu 0 zu machen

$$\underbrace{\underline{\underline{Q}}^{n-1} \cdots \underline{\underline{Q}}^{2}\,\underline{\underline{Q}}^{1}}_{\text{orthogonale Matrizen}}\,\underline{\underline{A}} = \underline{\underline{R}} \quad \Rightarrow$$

$$\underline{\underline{A}} = \underbrace{\left(\underline{\underline{Q}}^{1}\right)^T \cdots \left(\underline{\underline{Q}}^{n-1}\right)^T}_{\text{orthogonale Matrix}}\,\underline{\underline{R}} = \underline{\underline{Q}}\,\underline{\underline{R}}$$

$$\underline{\underline{G}}^{ij}(\ell) = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & \cos\ell & & \sin\ell & & \\ & & & 1 & & & \\ & & & & \ddots & & \\ & & & & & 1 & \\ & & -\sin\ell & & \cos\ell & & \\ & & & & & & \ddots \\ & & & & & & & 1 \end{bmatrix} \begin{matrix} \\ \\ \leftarrow i \\ \\ \\ \\ \leftarrow j \\ \\ \end{matrix}$$

$$\begin{matrix} & & \uparrow & & \uparrow \\ & & i & & j \end{matrix}$$

$$\underline{\underline{G}}^{ij}(\ell) \begin{bmatrix} x_1 \\ \vdots \\ x_{i-1} \\ x_i \\ x_{i+1} \\ \vdots \\ x_{j-1} \\ x_j \\ x_{j+1} \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_{i-1} \\ r \\ x_{i+1} \\ \vdots \\ x_{j-1} \\ 0 \\ x_{j+1} \\ \vdots \\ x_n \end{bmatrix} \begin{matrix} \\ \\ \\ \leftarrow i \\ \\ \\ \\ \leftarrow j \\ \\ \\ \\ \end{matrix}$$

$$r = \sqrt{x_i^2 + x_j^2}$$

$$\cos\ell = \frac{x_i}{r}$$

$$\sin\ell = \frac{x_j}{r}$$

Bsp $\begin{bmatrix} 4 \\ -3 \\ 1 \end{bmatrix} \rightsquigarrow \begin{bmatrix} * \\ 0 \\ 0 \end{bmatrix}$

$r = \sqrt{4^2 + (-3)^2} = 5$

$\cos\ell = \frac{4}{5}$, $\sin\ell = \frac{-3}{5}$

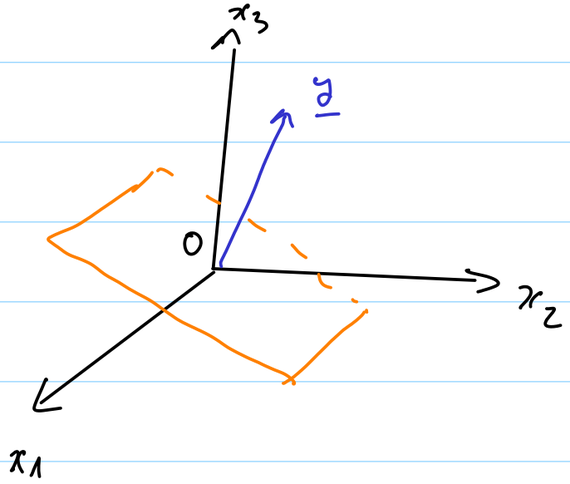
$$\underline{\underline{G}}^{12} \begin{bmatrix} 4 \\ -3 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{4}{5} & -\frac{3}{5} & 0 \\ 3/5 & 4/5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ -3 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \\ 1 \end{bmatrix}$$

$$\underline{\underline{G}}^{13} \begin{bmatrix} 5 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{5}{\sqrt{26}} & 0 & \frac{1}{\sqrt{26}} \\ 0 & 1 & 0 \\ -\frac{1}{\sqrt{26}} & 0 & \frac{5}{\sqrt{26}} \end{bmatrix} \begin{bmatrix} 5 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \sqrt{26} \\ 0 \\ 0 \end{bmatrix}$$

$$\underline{\underline{G}} = \underline{\underline{G}}^{13} \underline{\underline{G}}^{12}$$

Bem Diese Methode (Givens-Drehungen) ist günstig falls $\underline{\underline{A}}$ sehr viele 0-Einträge hat. (dünnbesetzte Matrizen.

III QR-Zerlegung mit Spiegelungen (standard-Software).



$$Q = I - 2uu^T$$

u Einheitsvektor, $\|u\|_2 = 1$

Q^1 spiegelt y so dass $Q^1 y$ auf Ox_1 liegt

$$Q^1 y = \begin{bmatrix} * \\ 0 \\ \vdots \\ 0 \end{bmatrix} \Rightarrow Q^1 A = \begin{bmatrix} * & x \dots x \\ 0 & x \dots \\ \vdots & \vdots \\ 0 & x \dots x \end{bmatrix}$$

z ← Spiegelung in x_2Ox_3

Q^2 spiegelt z so dass $Q^2 z$ auf Ox_2 liegt

$$Q^2 Q^1 A = \begin{bmatrix} * & x & x & \dots & x \\ 0 & x & x & & x \\ \vdots & 0 & x & & x \\ 0 & 0 & x & \dots & x \end{bmatrix} \quad \text{usw.}$$

Bsp

$$x = \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix} \xrightarrow{Q} 3e_1 = \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix} \qquad e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\|x\| = \sqrt{2^2+2^2+1^2} = 3$$

$$x = \|x\| e_1 + v \Rightarrow v = x - \|x\| e_1 = \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix}$$

$$u = \frac{1}{\|v\|} v = \frac{1}{\sqrt{6}} \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix} \Rightarrow$$

$$\Rightarrow Q = I - 2uu^T = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} - \frac{2}{6} \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 2 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} - \frac{2}{6} \begin{bmatrix} 1 & -2 & -1 \\ -2 & 4 & 2 \\ -1 & 2 & 1 \end{bmatrix} =$$

$$= \frac{1}{3} \begin{bmatrix} 2 & 2 & 1 \\ 2 & -1 & -2 \\ 1 & -2 & 2 \end{bmatrix} \Rightarrow Qx = \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix}.$$

$$\underline{\underline{Q}}^{n-1}\ \underline{\underline{Q}}^{n-2}\cdots\underline{\underline{Q}}^{2}\underline{\underline{Q}}^{1}\underline{\underline{A}}=\underline{\underline{R}} \quad\Rightarrow$$

$$\underline{\underline{A}}=\underline{\underline{Q}}\,\underline{\underline{R}}\,,\qquad \underline{\underline{Q}}=\underline{\underline{Q}}^{1T}\underline{\underline{Q}}^{2T}\cdots\left(\underline{\underline{Q}}^{n-1}\right)^{T}$$

Bsp

$$\underline{\underline{A}}=\begin{bmatrix}1&1\\2&0\\2&0\end{bmatrix}\Rightarrow \underline{\underline{Q}}^{1}\underline{\underline{A}}=\begin{bmatrix}\underline{\underline{Q}}^{1}\begin{bmatrix}1\\2\\2\end{bmatrix} & \underline{\underline{Q}}^{1}\begin{bmatrix}1\\0\\0\end{bmatrix}\end{bmatrix}$$

$$\underline{\underline{Q}}^{1}\begin{bmatrix}1\\2\\2\end{bmatrix}=\begin{bmatrix}3\\0\\0\end{bmatrix}\quad da\quad \|\begin{bmatrix}1\\2\\2\end{bmatrix}\|=3$$

$$\underline{v}=\underline{x}-3\underline{e}_1=\begin{bmatrix}1\\2\\2\end{bmatrix}-\begin{bmatrix}3\\0\\0\end{bmatrix}=\begin{bmatrix}-2\\2\\2\end{bmatrix}$$

$$\|\underline{v}\|=\sqrt{(-2)^2+2^2+2^2}=\sqrt{12}$$

$$\underline{u}=\frac{1}{\sqrt{12}}\begin{bmatrix}-2\\2\\2\end{bmatrix}\rightarrow \underline{\underline{Q}}^{1}=\underline{\underline{I}}-2\underline{u}\,\underline{u}^{T}=\underline{\underline{I}}-\frac{2}{12}\begin{bmatrix}-2\\2\\2\end{bmatrix}\begin{bmatrix}-2&2&2\end{bmatrix}=$$

$$=\frac{1}{6}\begin{bmatrix}2&4&4\\4&2&-4\\4&-4&2\end{bmatrix}$$

$$\underline{\underline{Q}}^{1}\underline{\underline{A}}=\begin{bmatrix}3&\frac{1}{3}\\0&\frac{2}{3}\\0&\frac{2}{3}\end{bmatrix}$$

MSW.

Theorem $\underline{\underline{A}} \in \mathbb{R}^{m \times n}$, $m \geq n$ voller Rang ($\operatorname{rang}(\underline{\underline{A}}) = n$).

Dann gibt es eine eindeutige Matrix $\underline{\underline{Q}} \in \mathbb{R}^{m \times m}$ orthogonal, so dass

$$\underline{\underline{A}} = \underline{\underline{Q}} \begin{bmatrix} \underline{\underline{R}} \\ 0 \end{bmatrix} \begin{matrix} n \\ m \end{matrix}$$

mit $\underline{\underline{R}}$ obere Dreiecksmatrix und die Elemente auf der Diagonale von $\underline{\underline{R}}$ sind ≥ 0.

Bem Falls die Elemente auf der Diagonale von $\underline{\underline{R}}$ > 0 sind, dann ist $\underline{\underline{R}}$ der Cholesky-Faktor von $\underline{\underline{A}}^T \underline{\underline{A}}$

Beweis

$$\underline{\underline{A}}^T \underline{\underline{A}} = \begin{bmatrix} \underline{\underline{R}}^T & \underline{\underline{0}} \end{bmatrix} \underline{\underline{Q}}^T \underline{\underline{Q}} \begin{bmatrix} \underline{\underline{R}} \\ \underline{\underline{0}} \end{bmatrix} = \underline{\underline{R}}^T \underline{\underline{R}}$$

Theorem Falls $\operatorname{rang}(\underline{\underline{A}}) = r < n$, dann gibt es eine Permutation $\underline{\underline{P}}$

$$\underline{\underline{A}}\,\underline{\underline{P}} = \underline{\underline{Q}} \begin{bmatrix} \underline{\underline{R}}_{11} & \underline{\underline{R}}_{12} \\ \underline{\underline{0}} & \underline{\underline{0}} \end{bmatrix} \begin{matrix} r \\ m-r \end{matrix}$$

mit $\underline{\underline{R}}_{11}$ obere Dreiecksmatrix mit Elementen auf der Diagonale ≥ 0.

$\underline{\underline{R}}_{11}$ ist eindeutig, $\underline{\underline{R}}_{12}$ nicht!

Beweis wähle r lin. unabhängige Spalten, permutiere sie nach vorne:

$$\underline{\underline{A}}\,\underline{\underline{P}} = \left[\underline{\underline{A}}_1 \middle| \underline{\underline{A}}_2 \right]$$ und wende voriges Theorem. (Givens-Drehung) an.

Bem QR-Zerlegung mit orthogonalen Transformationen darf nicht vorzeitig abgebrochen werden.

§7.4. Singulärwertzerlegung: siehe LA, Skript

Bsp $5 \geq 3 \geq 0.1 \geq 10^{-5} \geq 10^{-8} \geq 10^{-12} \geq 5\cdot 10^{-16} \geq 2\cdot 10^{-16} \geq 10^{-42} \geq 0 = 0$

$\sigma_1 \quad \sigma_2 \quad \sigma_3 \quad \sigma_4 \quad \sigma_5 \quad \sigma_6 \quad \sigma_7 \quad \sigma_8 \quad \sigma_9$

mathematische Rang: $r = 9$

numerische Rang: dort wo $\sigma_p >> \sigma_{p+1}$ $p=6$, oder 7] Anwendungsabhängig!

oder auch. $p=4$ $p=5$?

Bem Reduzierte SVD: $\underline{\underline{A}} = \begin{bmatrix} \underline{u}_1 \dots \underline{u}_r \end{bmatrix} \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_r \end{bmatrix} \begin{bmatrix} \underline{v}_1^T \\ \vdots \\ \underline{v}_r^T \end{bmatrix}$

$\underline{u}_1, \underline{u}_2, \dots, \underline{u}_r, \underline{u}_{r+1}, \dots, \underline{u}_m \in \mathbb{R}^m$ ONB

$\underline{v}_1, \underline{v}_2, \dots, \underline{v}_r, \underline{v}_{r+1}, \dots, \underline{v}_n \in \mathbb{R}^n$ ONB

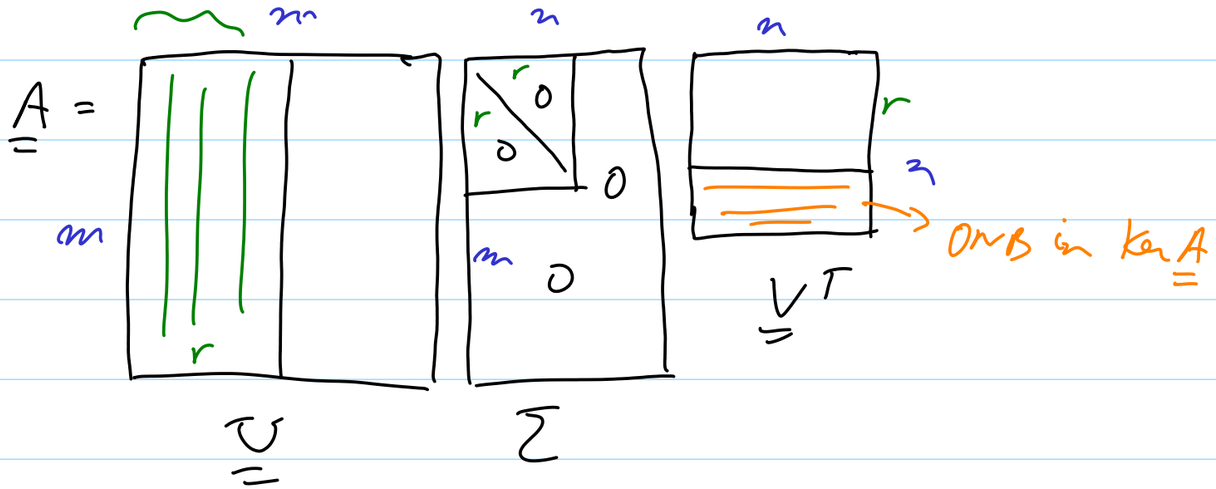
$$\underline{u}_i^T \underline{\underline{A}}\, \underline{v}_j = \begin{cases} \sigma_i, & \text{falls } i=j\le r \\ 0, & \text{sonst} \end{cases} \quad \Leftrightarrow$$

$$\underline{\underline{U}}^T \underline{\underline{A}}\, \underline{\underline{V}} = \underline{\underline{\Sigma}} = \begin{pmatrix} \sigma_1 & & & & \\ & \ddots & & 0 & \\ & & \sigma_r & & \\ & & & & \\ & 0 & & 0 & \end{pmatrix} \begin{matrix} \\ \\ r \\ \\ m \end{matrix}$$

$\begin{matrix} & & n & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{matrix}$

q.e.d

ONB Bild $\underline{\underline{A}}$



$\underline{\underline{U}}_r = [\underline{u}_1 \dots \underline{u}_r]$, $\underline{\underline{V}}_r = [\underline{v}_1 \dots \underline{v}_r]$

$\underline{v}_1, \dots, \underline{v}_r$ = rechte singularvektoren = EV von $\underline{\underline{A}}^T\underline{\underline{A}}$

$\underline{u}_1, \dots, \underline{u}_r$ = linke —— , —— = EV von $\underline{\underline{A}}\underline{\underline{A}}^T$

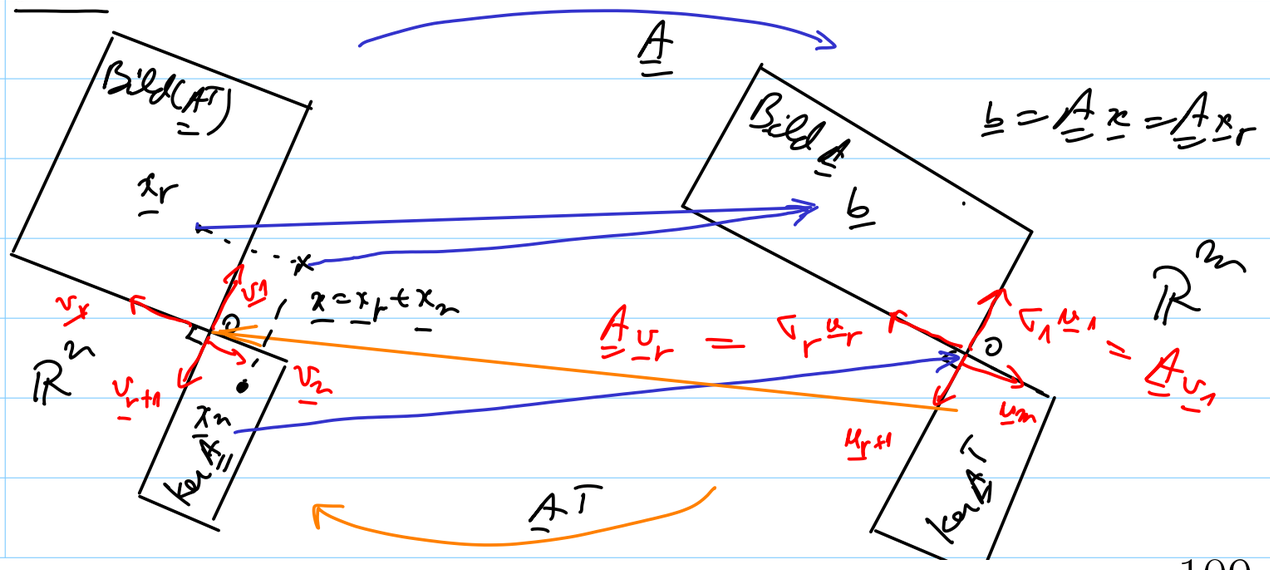
$\underline{\underline{A}}\,\underline{v}_1 = \sigma_1 \underline{u}_1, \dots, \underline{\underline{A}}\,\underline{v}_r = \sigma_r \underline{u}_r$

$\underline{\underline{A}}\,\underline{v}_{r+1} = 0, \dots, \underline{\underline{A}}\,\underline{v}_n = 0$

$\ker \underline{\underline{A}} = \operatorname{span}\{\underline{v}_{r+1}, \dots, \underline{v}_n\}$ ONB

Bild $\underline{\underline{A}} = \operatorname{span}\{\underline{u}_1, \dots, \underline{u}_r\}$ ONB

Bem Erinnern: Fundamentalsatz der LA:



Anwendungen der SVD

1) Speicherplatzreduktion:

$\underline{\underline{A}}$ braucht $m \cdot n$ Plätze

SVD: $rm + r \cdot n + r = r(m+n+1)$ Plätze!

2) Berechnung der Norm einer Matrix

$$\|\underline{\underline{A}}\|_2 = \max_{\|\underline{x}\|_2 = 1} \|\underline{\underline{A}}\,\underline{x}\|_2 \qquad \underline{\underline{U}} \text{ orthogonal}$$

$$\|\underline{\underline{A}}\,\underline{x}\|_2^2 = \|\underline{\underline{U}}\,\underline{\underline{\Sigma}}\,\underline{\underline{V}}^T\underline{x}\|_2^2 = \|\underline{\underline{\Sigma}}\,\underline{\underline{V}}^T\underline{x}\|_2^2 .$$

$$\|\underline{\underline{A}}\| = \max_{\|\underline{x}\|_2=1} \|\underline{\underline{A}}\,\underline{x}\|_2 = \max_{\|x\|_2=1} \|\underline{\underline{\Sigma}}\,\underline{\underline{V}}^T\underline{x}\|_2 =$$

$\underline{y} := \underline{\underline{V}}^T\underline{x} \Rightarrow \underline{x} = \underline{\underline{V}}\,\underline{y}$

$$= \max_{\|\underline{\underline{V}}\,\underline{y}\|_2=1} \|\underline{\underline{\Sigma}}\,\underline{y}\|_2 = \max_{\|\underline{y}\|_2=1} \|\underline{\underline{\Sigma}}\,\underline{y}\|_2 =$$

$$= \max_{\|\underline{y}\|_2=1} \sqrt{\sigma_1^2|y_1|^2 + \sigma_2^2|y_2|^2 + \dots + \sigma_r^2|y_r|^2} \le$$

$(\sigma_1 \ge \sigma_2 \ge \dots \ge \sigma_r > 0)$

$$\le \max_{\|\underline{y}\|_2=1} \sqrt{\sigma_1^2\|\underline{y}\|_2^2} = \sigma_1$$ erreicht für $\underline{y} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$.

$\Rightarrow \quad \|\underline{\underline{A}}\|_2 = \sigma_1$

3) $\underline{\underline{A}} = \sigma_1 \underline{u}_1 \underline{v}_1^T + \sigma_2 \underline{u}_2 \underline{v}_2^T + \dots + \sigma_r \underline{u}_r \underline{v}_r^T$

= Summe von Matrizen von Rang 1.

$\{\underline{u}_1 \dots, \underline{u}_r\}$, $\{\underline{v}_1 \dots, \underline{v}_r\}$

SVD

⇒ wird verwendet um eine Approximation von $\underline{\underline{A}}$ mit einer Matrix von niedrigen Rang zu bauen!

Daten mit Rausch / Fehlern:

$$\underline{\underline{A}} = \sum_{j=1}^{r} \sigma_j \underline{u}_j \underline{v}_j^T \approx \sum_{j=1}^{k} \sigma_j \underline{u}_j \underline{v}_j^T \quad \text{mit } k \ll r.$$

≠ Diagonalisierung verwenden wurde.

Falls $\underline{\underline{A}}$ symmetrisch: $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$

$$\underline{\underline{A}} = \sum_{j=1}^{n} \lambda_j \underline{s}_j \hat{\underline{s}}_j^T \qquad \underline{\underline{A}} = \underline{\underline{S}} \underline{\underline{\Lambda}} \underline{\underline{S}}^{-1}$$

Aber: → das geht so nur für quadratische Matrix $\underline{\underline{A}}$ ≠ SVD

→ keine Ordnung für λ_j, falls λ_j komplex ≠SVD

→ keine Orthogonalität $\underline{s}_j \not\perp \hat{\underline{s}}_j$

Bsp $\underline{\underline{A}} = \begin{bmatrix} 2 & 100 \\ 0 & 1 \end{bmatrix}$ $\lambda_1 = 2$, $\lambda_2 = 1$

$$\underline{\underline{A}} = \underline{\underline{S}} \underline{\underline{\Lambda}} \underline{\underline{S}}^{-1} = 2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 100 \end{bmatrix} + 1 \begin{bmatrix} 1 \\ -\frac{1}{100} \end{bmatrix} \begin{bmatrix} 0 & -100 \end{bmatrix}$$

$\lambda_1 \underline{s}_1 \hat{\underline{s}}_1^T$ $\lambda_2 \underline{s}_2 \hat{\underline{s}}_2^T$

Versuch der Approximation mit einer Rang 1 Matrix:
mit $\in W, \in V$:

Fehler $= \underline{\underline{A}} - \lambda_1 \underline{s}_1 \hat{\underline{s}}_1^T = \begin{bmatrix} 0 & -100 \\ 0 & 1 \end{bmatrix}$ > Fehler ist sehr gross

$\underline{\underline{A}} - \lambda_2 \underline{s}_2 \hat{\underline{s}}_2^T = \begin{bmatrix} 2 & 200 \\ 0 & 0 \end{bmatrix}$ also so gehts NICHT ☹

Mit SVD:

$$\underline{\underline{A}} = 100.025 \begin{bmatrix} 0.9995 \\ \\ 0.001 \end{bmatrix} \begin{bmatrix} 0.01999 & 0.9998 \end{bmatrix} +$$

$\sigma_1 \underline{u}_1 \underline{v}_1^T$

$$+ 0.02 \begin{bmatrix} 0.01 \\ \\ 0.9995 \end{bmatrix} \begin{bmatrix} -0.9998 & 0.01999 \end{bmatrix}$$

$\sigma_2 \quad \underline{u}_2 \underline{v}_2^T$

Fehler $\underline{\underline{A}} - \sigma_1 \underline{u}_1 \underline{v}_1^T = \begin{bmatrix} 2 \cdot 10^{-4} & 0 \\ \\ -2 \cdot 10^{-2} & 4 \cdot 10^{-4} \end{bmatrix}$ kleiner Fehler ☺

Theorem [Eckart-Young]

Sei $\underline{\underline{A}} \in \mathbb{C}^{m \times n}$. Für jedes $k \leq \text{Rang}(\underline{\underline{A}})$ gibt es eine abgebrochene SVD

$$\underline{\underline{A}}_k = \sum_{j=1}^{k} \sigma_j \underline{u}_j \underline{v}_j^H$$

ist die beste Approximation von Rang k an $\underline{\underline{A}}$ im Sinne von:

$$\| \underline{\underline{A}} - \underline{\underline{A}}_k \| = \min_{\text{Rang}(\underline{\underline{X}}) \leq k} \| \underline{\underline{A}} - \underline{\underline{X}} \| = \sigma_{k+1}$$

(Euklidische & Frobenius Norm).

4) PCA = principal component analysis.

Zwei Signale, gemessen an $m = 50$ Orte.

Messfehler, messen 10 Mal jedes Signal $\Rightarrow n = 20$ Messungen.

Daten in einer $m \times n$ Matrix $\underline{\underline{A}}$ (durcheinander)

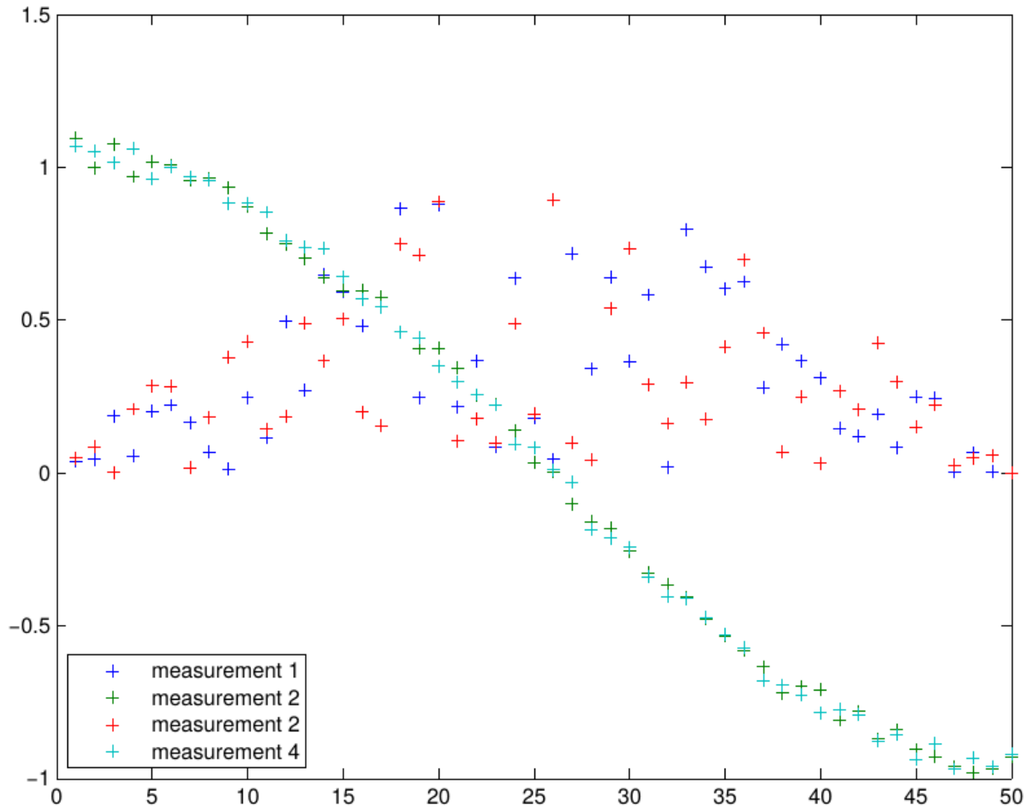
Frage: Wie viele Signale gemessen worden sind?

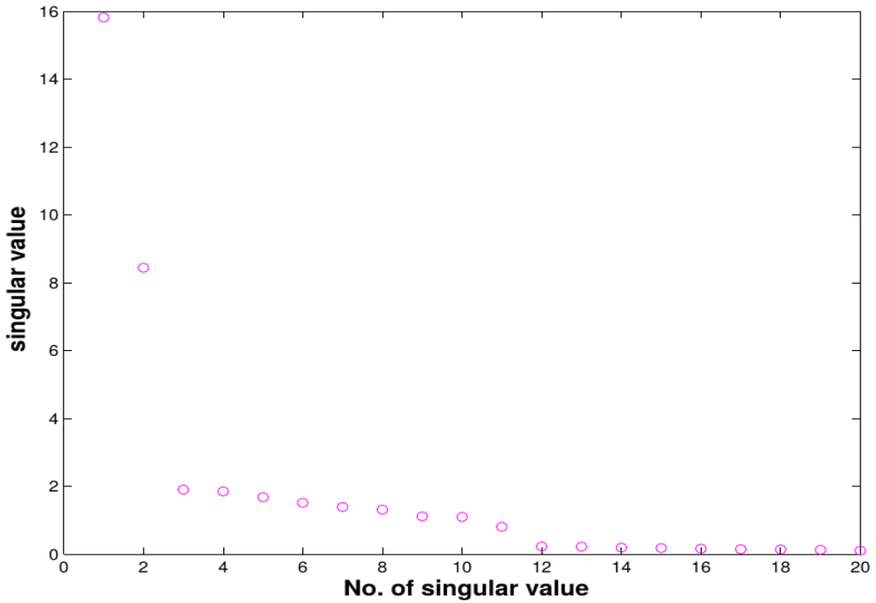
Wie sehen die echte Signale ungefähr aus?

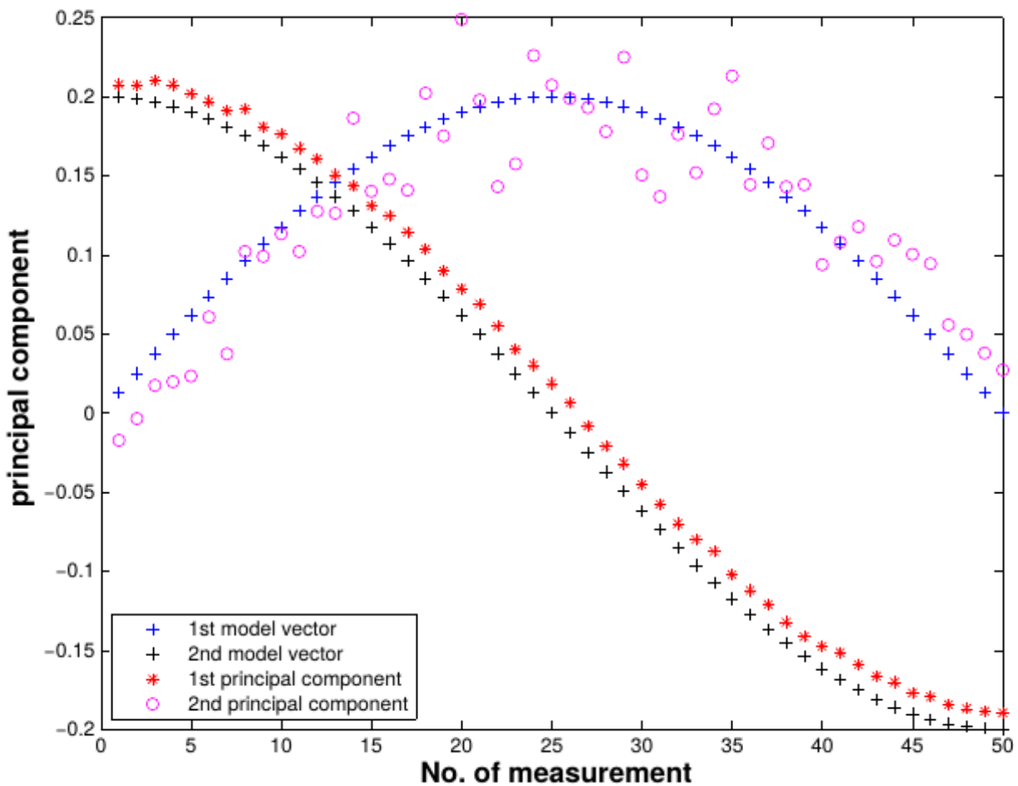
In der Sprache der LA: Was ist $\text{Rang}(\underline{\underline{A}})$?

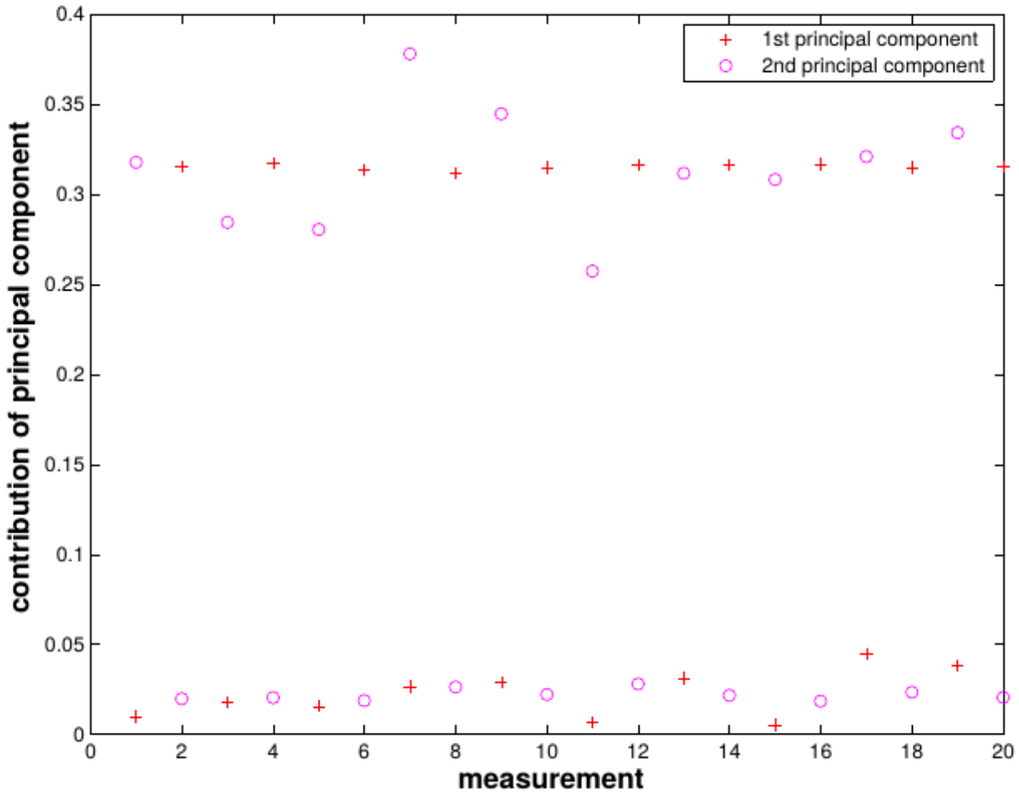
Wie viele unabhängige Vektoren / Messungen sind in $\underline{\underline{A}}$?

```
n = 10; m = 50
r = linspace(1,m,m)
x = sin(pi*r/m)
y = cos(pi*r/m)
A = zeros((2*n,m))
for k in range(n):
    A[2*k] = x*rand(m)
    A[2*k+1] = y+0.1*rand(m)
```




singular value
No. of singular value


principal component
No. of measurement
1st model vector
2nd model vector
1st principal component
2nd principal component


contribution of principal component
measurement
1st principal component
2nd principal component

§8 Ausgleichsrechnung

§8.1. Motivation und Normalengleichung

Bsp Modell $f(t) = c \cdot e^{-at^2+bt}$ mit physikalischen Parameter a, b, c die aus 200 Messungen zu bestimmen.

Lineares Modell:

$\underline{y} = \underline{a}^T \underline{t} + c \cdot 1$ mit $\underline{a} \in \mathbb{R}^n$, $c \in \mathbb{R}$ Parameter

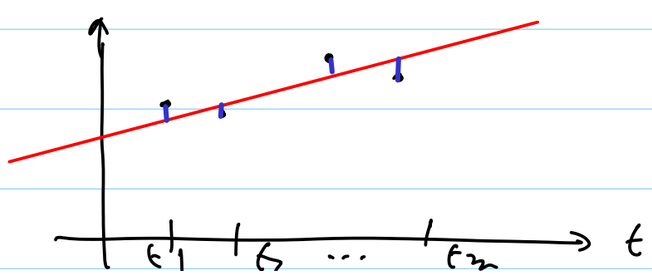
m Messungen

$\underline{t}_1, \underline{t}_2, \dots, \underline{t}_m \in \mathbb{R}^n$ Messpunkte

$\downarrow \quad \downarrow \quad \dots \quad \downarrow$

$y_1 \quad y_2 \quad y_m$ Gemessene Werte

Für $n = 1$



Vorschlag: $\underline{a}, c$, die $\underline{q}, \underline{p}$ des Minimum realisieren:

$$\min_{\substack{\underline{p} \in \mathbb{R}^n \\ q \in \mathbb{R}}} \sum_{i=1}^{m} \left| y_i - \underline{p}^T \underline{t}_i - 1 \cdot q \right|^2 = \min_{\underline{x} \in \mathbb{R}^{1+n}} \| \underline{A}\underline{x} - \underline{b} \|_2$$

$$\underline{x} = \begin{bmatrix} q \\ \underline{p} \end{bmatrix} \in \mathbb{R}^{1+n}, \quad \underline{A} = \begin{bmatrix} 1 & \underline{t}_1^T \\ 1 & \underline{t}_2^T \\ \vdots & \vdots \\ 1 & \underline{t}_m^T \end{bmatrix} \in \mathbb{R}^{m \times (1+n)}$$

$$\underline{b} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

Methode der kleinsten Quadrate (least squares Method).

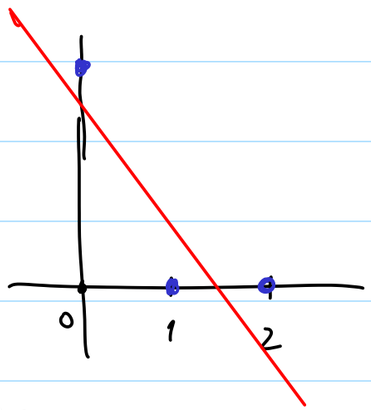
Bsp $n=1$ $\quad t_1=0,\ t_2=1,\ t_3=2$

$y_1=6,\ y_2=0,\ y_3=0$

Modell $y = dt + c \cdot 1$.

Messpunkte $\begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$, Messwerte $\begin{bmatrix} 6 \\ 0 \\ 0 \end{bmatrix}$

$$\underline{\underline{A}} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 6 \\ 0 \\ 0 \end{bmatrix}$$



$$\min_{x \in \mathbb{R}^2} \| \underline{\underline{A}} \underline{x} - \underline{b} \|_2$$

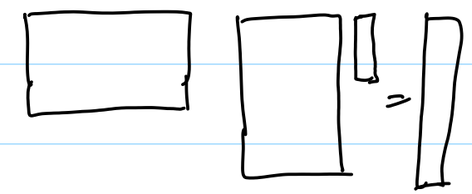
alg. Problem: $\begin{cases} b_1 = dt_1 + c_1 \\ b_2 = dt_2 + c_2 \\ b_3 = dt_3 + c_3 \end{cases} \Rightarrow \underline{\underline{A}}\,\underline{x} = \underline{b}$ hat keine Lösung.

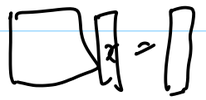
Möchte aber "Lösung" im Sinne.

$$\| \underbrace{\underline{\underline{A}} x - \underline{b}}_{\text{Residuum}} \|_2 \overset{!}{=} \text{Min}!$$

I Algebraische Methode:

$$\underline{\underline{A}}^T \mid \quad \underline{\underline{A}}\,\underline{x} = \underline{b} \quad \Rightarrow \quad \underbrace{\underline{\underline{A}}^T \underline{\underline{A}}}_{\text{quadratisch, symmetrisch}}\,\underline{x} = \underline{\underline{A}}^T \underline{b}$$





Falls $\boxed{\text{Rang}(\underline{\underline{A}}) = n}$ $\Rightarrow$ $\underline{\underline{A}}^T \underline{\underline{A}}$ invertierbar $\Rightarrow$

kann das LGS eindeutig lösen, am besten mit QR-Zerlegung!

↳ weil stabiler als LU-Zerlegung!

$\text{cond}(\underline{\underline{A}})$ gross $\Rightarrow$ $\text{cond}(\underline{\underline{A}}^T \underline{\underline{A}}) = \text{cond}(\underline{\underline{A}})^2$

$\text{cond}(A) = 10^2 \rightarrow \text{cond}(\underline{\underline{A}}^T \underline{\underline{A}}) = 10^4$ ⇝ LU!

$\underline{\underline{A}}^T \underline{\underline{A}}\, x = \underline{\underline{A}}^T \underline{b}$ Normalengleichung.

Ben Das ist in der Tat, auch die Lösung von Min.-Problem:

$\underline{\underline{A}} : \mathbb{R}^n \to \mathbb{R}^m$; $b \in \mathbb{R}^m \Rightarrow \underline{b} = \underline{p} + \underline{e}$

$\underline{x} \mapsto \underline{\underline{A}}\underline{x}$

$\underline{p} \in$ Bild($\underline{\underline{A}}$), $\underline{e} \in$ ker($\underline{\underline{A}}^T$) = Bild($\underline{\underline{A}}$)$^\perp$

es gibt eine Lösung $\hat{\underline{x}}$ von $\underline{\underline{A}}\hat{\underline{x}} = \underline{p}$

Fehler $\underline{b} - \underline{\underline{A}}\hat{\underline{x}} = \underline{b} - \underline{p} = \underline{e} \in \ker(\underline{\underline{A}}^T) \perp \text{Bild}(\underline{\underline{A}}) \Rightarrow$

$\|\underline{b} - \underline{\underline{A}}\hat{\underline{x}}\|_2^2 = \min_{\underline{x} \in \mathbb{R}^n} \|\underline{b} - \underline{\underline{A}}\underline{x}\|_2^2 \Rightarrow \hat{\underline{x}}$ ist die Lösung des Min.-Problems.

$\underline{\underline{A}}^T \cdot \Rightarrow \quad \underline{\underline{A}}^T\underline{b} = \underline{\underline{A}}^T\underline{p} + \underline{\underline{A}}^T\underline{e} = \underline{\underline{A}}^T\underline{p} = \underline{\underline{A}}^T\underline{\underline{A}}\hat{\underline{x}}$

$\underline{e} \in \ker(\underline{\underline{A}}^T) \Leftarrow \underline{\underline{A}}^T\underline{e} = 0$

II Analytische Methode

$f(\underline{x}) = \|\underline{\underline{A}}\underline{x} - \underline{b}\|_2^2$, $f: \mathbb{R}^{1+n} \to [0, \infty[$

Min=?

D $f(\underline{x}) = 0$

$f(\underline{x}) = \|\underline{\underline{A}}\underline{x} - \underline{b}\|_2^2 = (\underline{\underline{A}}\underline{x} - \underline{b})^T(\underline{\underline{A}}\underline{x} - \underline{b}) =$

$= \underline{x}^T\underline{\underline{A}}^T\underline{\underline{A}}\underline{x} - \underline{b}^T\underline{\underline{A}}\underline{x} - \underline{x}^T\underline{\underline{A}}^T\underline{b} + \underline{b}^T\underline{b} =$

$\underline{\underline{M}}$ symmetrisch.

$= \underline{x}^T\underline{\underline{M}}\underline{x} - 2\underline{x}^T\underline{\underline{A}}^T\underline{b} + \underline{b}^T\underline{b} =$

$\underline{w}$

$= \sum_{i=1}^{n} x_i \sum_{j=1}^{n} m_{ij} x_j - 2\sum_{i=1}^{n} x_i w_i + \underline{b}^T\underline{b} =$

$$= x_1 \sum_{j=1}^{n} m_{1j} x_j + \sum_{i=2}^{n} x_i \sum_{j=1}^{n} m_{ij} x_j - 2 \sum_{i=1}^{n} x_i w_i + \underline{b}^T \underline{b} =$$

$$\sum_{j=1}^{n} x_j \sum_{i=2}^{n} x_i m_{ij}$$

$$= x_1 m_{11} x_1 + x_1 \sum_{j=2}^{n} m_{1j} x_j + x_1 \sum_{i=2}^{n} x_i m_{i1} + \sum_{j=2}^{n} x_j \sum_{i=2}^{n} x_i m_{ij} -$$

$$- 2 \sum_{i=1}^{n} x_i w_i + \underline{b}^T \underline{b} \quad \overset{M \text{ symmetrisch.}}{=}$$

$$= m_{11} x_1^2 + 2 x_1 \sum_{i=2}^{n} x_i m_{i1} + \sum_{i,j=2}^{n} x_i m_{ij} x_j - 2 \sum_{i=1}^{n} x_i w_i + \underline{b}^T \underline{b}$$

$$\frac{\partial}{\partial x_1} f(\underline{x}) = 2 m_{11} x_1 + 2 \sum_{i=2}^{n} x_i m_{i1} - 2 w_1 \quad \Rightarrow$$

$$D f(\underline{x}) = 2 \underline{M} \underline{x} - 2 \underline{A}^T \underline{b} =$$

$$= 2 \underline{A}^T \underline{A} x - 2 \underline{A}^T \underline{b}.$$

kritische Punkte $Df(\underline{x}) = 0 \Leftrightarrow$,

$$\underline{A}^T \underline{A} \underline{x} = \underline{A}^T \underline{b} \quad \text{Normalengleichung.}$$

Minimum nur wenn $D^2 f(\underline{x}) = 2 \underline{A}^T \underline{A}$

symmetrisch positiv definit

$\uparrow$

nur wenn $\operatorname{Rang}(\underline{A}) = n$.

Bem $\operatorname{cond}_2 \underline{\underline{M}} = \|\underline{\underline{M}}^{-1}\|_2 \, \|\underline{\underline{M}}\|_2$

↳ wie sehr sich die Rundungsfehler sich auf das Lösen vom $\underline{\underline{M}}\underline{x} = \underline{b}$ auswirken.

$\operatorname{cond}(\underline{\underline{M}}) >> 1 \Rightarrow$ grosse Rundungsfeheler ☹

$$\operatorname{cond}_2(\underline{\underline{A}}^T\underline{\underline{A}}) = \operatorname{cond}_2(\underline{\underline{V}}\underline{\underline{\Sigma}}^T \underbrace{\underline{\underline{U}}^T\underline{\underline{U}}}_{I}\underline{\underline{\Sigma}}\underline{\underline{V}}^T) = \operatorname{cond}(\underline{\underline{V}}\underline{\underline{\Sigma}}^2\underline{\underline{V}}^T)$$

$$= \operatorname{cond}(\underline{\underline{\Sigma}}^2) = \frac{\sigma_1^2}{\sigma_r^2} = \operatorname{cond}(\underline{\underline{A}})^2$$

wobei $\sigma_1, \dots, \sigma_r$ die Singulärwerte von $\underline{\underline{A}}$ sind.

Bsp $f \in L^2 = \{f : I \to \mathbb{R} \,;\, \int_I |f(t)|^2 dt < \infty\}$

$\mathbb{R}$ $\quad \cup$

$f_n \in V_n$, $\dim V_n < \infty$

↳ Basis $b_1, \dots, b_n$

$$f_n = \sum_{j=1}^{n} x_j b_j \Rightarrow f_n(t) = \sum_{j=1}^{n} x_j b_j(t)$$

verwende Messungen $y_i \approx f(t_i)$

Aufgabe: gegeben (t_i, y_i) für $i = 1, 2, \dots, m$ Messungen, finde die beste Approximation in V_n an f, d.h. finde $x_1, x_2, \dots, x_n$ so dass

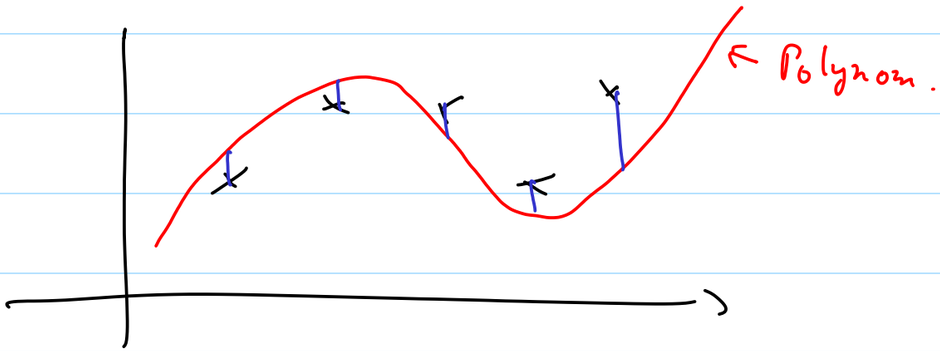
$$\sum_{i=1}^{m} |f_n(t_i) - y_i|^2 = \text{minimal !}$$

$$\begin{cases} \sum_{j=1}^{n} b_j(t_i)\, x_j = y_i \\ \longrightarrow \; i = 1, 2, \dots, m \end{cases} \qquad \underline{\underline{A}} = \left[b_j(t_i) \right]_{\substack{i=1,2,\dots,m \\ j=1,2,\dots,n}}$$

Bem Spezielle Wahl: $V_n = P_n$ = Polynome vom Grad Maximal $n-1$

Basis in V_n: Monome $b_j(t) = t^{j-1}$

$\longrightarrow$ polynomiales fit ; polyfit.



§8.2 Lösung mittels orthogonalen Transformationen

Fall 1 $\operatorname{rang}(\underline{\underline{A}}) = n$

$$\underline{\underline{A}} = \underline{\underline{Q}}\,\underline{\underline{R}} = \underline{\underline{Q}} \begin{bmatrix} \tilde{\underline{\underline{R}}} \\ 0 \end{bmatrix} \begin{matrix} \} n \\ \} m-n \end{matrix}$$

$\underline{\underline{Q}}$ orthogonal

$$\min_{\underline{x}\in\mathbb{R}^n} \|\underline{\underline{A}}\underline{x} - \underline{b}\|_2^2 = \min_{x\in\mathbb{R}^n} \|\underline{\underline{Q}}\underline{\underline{R}}\underline{x} - \underline{\underline{I}}\underline{b}\|_2^2 =$$

$$= \min_{\underline{x}\in\mathbb{R}^n} \|\underline{\underline{Q}}\underline{\underline{R}}\underline{x} - \underline{\underline{Q}}\underline{\underline{Q}}^H\underline{b}\|_2^2 =$$

$$= \min_{\underline{x}\in\mathbb{R}^n} \|\underline{\underline{Q}}(\underline{\underline{R}}\underline{x} - \underline{\underline{Q}}^H\underline{b})\|_2^2 =$$

$$= \min_{\underline{x}\in\mathbb{R}^n} \|\underline{\underline{R}}\underline{x} - \underbrace{\underline{\underline{Q}}^H\underline{b}}_{\underline{\beta}}\|_2^2 =$$

$$= \min_{\underline{x}\in\mathbb{R}^n} \left\| \begin{bmatrix} \tilde{\underline{\underline{R}}} \\ 0 \end{bmatrix} [x] - \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \\ \beta_{n+1} \\ \vdots \\ \beta_m \end{bmatrix} \right\|_2^2 = \min_{x\in\mathbb{R}^n} \left\| \begin{bmatrix} \tilde{\underline{\underline{R}}}\underline{x} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \\ \beta_{n+1} \\ \vdots \\ \beta_m \end{bmatrix} \right\|_2^2 =$$

($\tilde{\beta}$ = $\beta_1, \dots, \beta_n$)

$$= \min_{\underline{x}\in\mathbb{R}^n} \underbrace{\|\underline{\tilde{R}}\underline{x} - \underline{\tilde{\beta}}\|}_{\text{für } \underline{x} \text{ die Lösung von } \underline{\tilde{R}}\underline{x}=\underline{\tilde{\beta}}} + |\beta_{n+1}|^2 + \dots + |\beta_m|^2$$

$\|$
0

$$\begin{bmatrix} * & & \\ & \ddots & \\ 0 & & * \end{bmatrix} \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} \tilde{\beta} \end{bmatrix}$$

Bsp $\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 0 \\ 0 \end{bmatrix} \Rightarrow$ Normalengleichung.

$$\begin{bmatrix} 3 & 3 \\ 3 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 0 \end{bmatrix} \Rightarrow \begin{matrix} x_1 = 5 \\ x_2 = -3 \end{matrix}$$

QR-Zerlegung

$$\underline{\underline{A}} = \underline{\underline{Q}}\,\underline{\underline{R}} = \begin{bmatrix} -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & ? \\ -\frac{1}{\sqrt{3}} & 0 & ? \\ -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & ? \end{bmatrix} \begin{bmatrix} -\frac{3}{\sqrt{3}} & -\frac{3}{\sqrt{3}} \\ 0 & -\frac{2}{\sqrt{2}} \\ 0 & 0 \end{bmatrix}$$

numpy.linalg.qr($\underline{\underline{A}}$)

LGS $\underline{\underline{\tilde{R}}}\,\underline{x} = \underline{\underline{Q}}^T \underline{b}$

$$\begin{bmatrix} -\frac{3}{\sqrt{3}} & -\frac{3}{\sqrt{3}} \\ 0 & -\frac{2}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -\frac{6}{\sqrt{3}} \\ 6/\sqrt{2} \end{bmatrix} \Rightarrow$$

$\Rightarrow x_1 = 5,\ x_2 = -3.$

Fall 2 $\operatorname{rang}(\underline{\underline{A}}) < n \Rightarrow$ SVD!

$r \quad m-r \quad r$

$$\underline{\underline{A}} = \begin{bmatrix} \underline{\underline{U}}_1 & \underline{\underline{U}}_2 \end{bmatrix} \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ \hline V_2^T \end{bmatrix} \begin{matrix} r \\ n-r \end{matrix}$$

$$\Sigma_r = \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_r \end{bmatrix} \quad \text{mit } \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$$

$\underline{\underline{U}}$ orthogonal

$$\|\underline{\underline{A}}\underline{x} - \underline{b}\|_2 = \left\| \begin{bmatrix} \underline{\underline{\Sigma}}_r \underline{\underline{V}}_1^T \underline{x} \\ 0 \end{bmatrix} - \begin{bmatrix} \underline{\underline{U}}_1^T \underline{b} \\ \underline{\underline{U}}_2^T \underline{b} \end{bmatrix} \right\|_2 \Rightarrow$$

Minimal für $\underline{x}$ Lösung von $\underline{\Sigma}_r \underline{V}_1^T \underline{x} = \underline{U}_1^T \underline{b}_1$

$$\underline{x} = \underline{V}_1 \underline{\Sigma}_r^{-1} \underline{U}_1^T \underline{b}_1$$

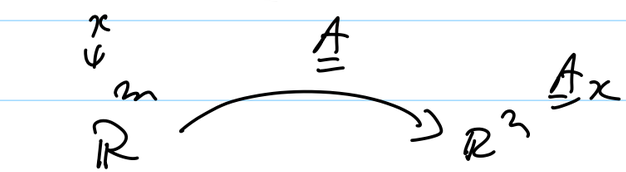
und das Minimum ist $\| \underline{U}_2^T \underline{b}_2 \|_2$

→ Standard in Codes $\mathrm{lsq}(\underline{A}, \underline{b})$

Def Pseudoinverse $\underline{A}^+ := \underline{V} \underline{\Sigma}^+ \underline{U}^T$

$$\underline{\Sigma}^+ = \begin{bmatrix} \sigma_1^{-1} & & & 0 \\ & \sigma_2^{-1} & & \\ & & \ddots & \\ & & & \sigma_r^{-1} \\ 0 & & 0 \cdots 0 & \end{bmatrix}$$

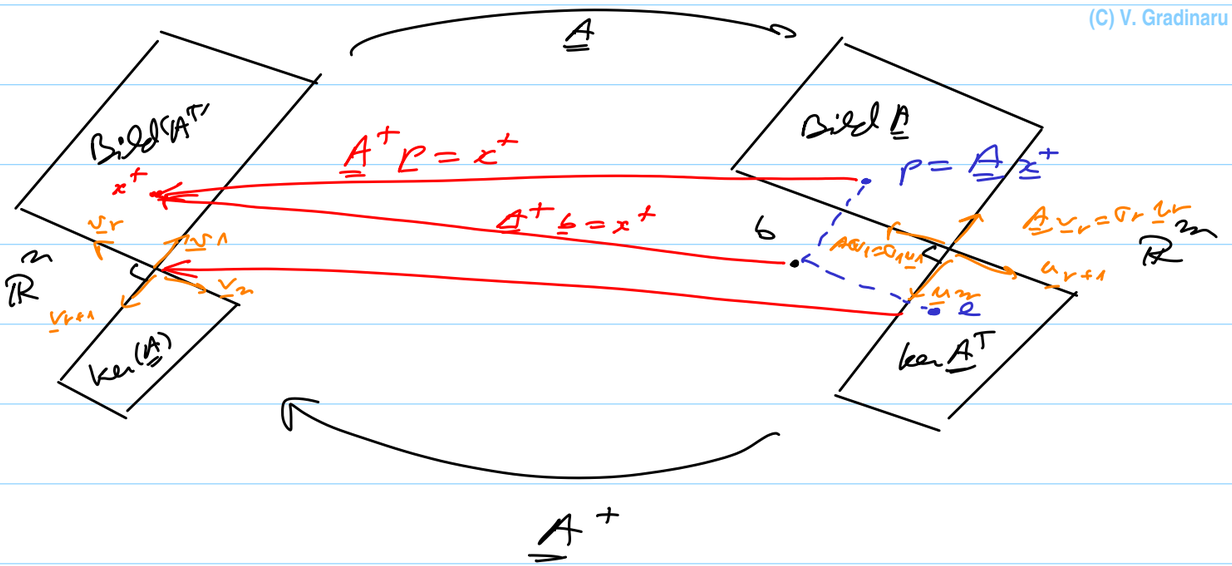
Moore-Penrose inverse.



$\|\underline{A}x^+ - b\|_2^2$ min!

x^+ $\underline{b}$

$\underline{A}^+$ die Ausgleichslösung.



Bem Was ist wenn wir später entdecken,

1) bestimmte Messungen falsch waren?

$\underline{Q}\underline{R}$-Methode ist da besser.

scipy. qr_delete(row)

2) bestimmte Basisfunktionen gar nicht dabei sein sollen?

qr_delete (column).

§8.3. Lineare Ausgleichsrechnung mit Lin. Nebenbedingungen

Finde $\underline{x} \in \mathbb{R}^n$ bei gegebenen $\underline{\underline{A}} \in \mathbb{R}^{m \times n}$

$m \geq n = \operatorname{rang}(\underline{\underline{A}})$

$\underline{b} \in \mathbb{R}^m$

$$\begin{cases} \|\underline{\underline{A}}\underline{x} - \underline{b}\|_2^2 = \text{minimal!} \\ \underline{\underline{C}}\underline{x} = \underline{d} \end{cases}$$

mit $\underline{\underline{C}} \in \mathbb{R}^{p \times n}$ mit $p < n$

$\operatorname{rang}(\underline{\underline{C}}) = p$

$\underline{d} \in \mathbb{R}^p$

Methode 1 Lagrange-Multiplicatoren $\underline{m} \in \mathbb{R}^p$

$$L(\underline{x}, \underline{m}) := \frac{1}{2}\|\underline{\underline{A}}\underline{x} - \underline{b}\|_2^2 + \underline{m}^T(\underline{\underline{C}}\underline{x} - \underline{d})$$

$\min \max L(\underline{x}, \underline{m})$

$\underline{x} \in \mathbb{R}^n \quad \underline{m} \in \mathbb{R}^p$

$$\begin{cases} \frac{\partial L}{\partial \underline{x}}(\underline{x}, \underline{m}) = 0 \\ \frac{\partial L}{\partial \underline{m}}(\underline{x}, \underline{m}) = 0 \end{cases}$$

$$\begin{cases} \underline{\underline{A}}^T(\underline{\underline{A}}\underline{x} - \underline{b}) + \underline{\underline{C}}^T\underline{m} = 0 \\ 0 + \underline{\underline{C}}\underline{x} - \underline{d} = 0 \end{cases}$$

$$\begin{bmatrix} \underline{\underline{A}}^T\underline{\underline{A}} & \underline{\underline{C}}^T \\ \underline{\underline{C}} & \underline{\underline{0}} \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{m} \end{bmatrix} = \begin{bmatrix} \underline{\underline{A}}^T\underline{b} \\ \underline{d} \end{bmatrix}$$

Block-LR-Zerlegung

1) Cholesky-Zerlegung von $\underline{\underline{A}}^T\underline{\underline{A}} = \underline{\underline{R}}^T\underline{\underline{R}}$

2) berechne $\underline{\underline{G}}$ aus $\underline{\underline{R}}^T\underline{\underline{G}}^T = \underline{\underline{C}}^T$

3) $\underline{\underline{S}}$ aus Cholesky-Zerlegung.

$$\underline{\underline{S}}^T\underline{\underline{S}} = -\underline{\underline{G}}\,\underline{\underline{G}}^T$$

$$\begin{bmatrix} A^TA & C^T \\ C & 0 \end{bmatrix} = \begin{bmatrix} R^T & 0 \\ G & S \end{bmatrix} \begin{bmatrix} R & G^T \\ 0 & S^T \end{bmatrix}$$

Cholesky-Zerlegung!

Nachteil: $\mathrm{cond}(\underline{\underline{A}}^T\underline{\underline{A}}) = \mathrm{cond}(\underline{\underline{A}})^2$ ☹

Vorteil: falls $\underline{\underline{A}}$, $\underline{\underline{C}}$ eine Struktur haben, dann kann man Algorithmen nehmen, die diese Struktur erhalten.

Methode 2 SVD besser für die Kondition, verliert die Struktur

$$\underline{\underline{C}} = \underline{\underline{U}} \begin{bmatrix} \underline{\underline{\Sigma}} & \underline{\underline{0}} \end{bmatrix} \begin{bmatrix} \underline{\underline{V}}_1^H \\ \underline{\underline{V}}_2^H \end{bmatrix} \begin{matrix} p \\ n-p \end{matrix}$$

$\mathrm{Ker}(\underline{\underline{C}}) = \mathrm{Bild}(\underline{\underline{V}}_2)$.

Trick: Definiere $\underline{x}_0 = \underline{\underline{V}}_1 \underline{\underline{\Sigma}}_1^{-1} \underline{\underline{U}}^H \underline{d}$

dann suche $\underline{x} = \underline{x}_0 + \underline{\underline{V}}_2 \underline{y}$ mit $\underline{y} \in \mathbb{R}^{n-p}$

$\underline{\underline{V}}_2 \underline{y} \in \mathrm{Ker}(\underline{\underline{C}}) = \mathrm{Bild}(\underline{\underline{V}}_2)$

$(\Rightarrow \underline{\underline{C}}\underline{x}_0 = \underline{d})$.

$$\|\underline{\underline{A}}\underline{x} - \underline{b}\|_2^2 = \|\underline{\underline{A}}\underline{x}_0 + \underline{\underline{A}}\underline{\underline{V}}_2\underline{y} - \underline{b}\|_2^2 = \|\underline{\underline{A}}\underline{\underline{V}}_2\underline{y} - (\underline{b} - \underline{\underline{A}}\underline{x}_0)\|_2^2$$

$\underline{\underline{A}}\underline{\underline{V}}_2\underline{y}$: min noch $\underline{y}$; $\underline{b} - \underline{\underline{A}}\underline{x}_0$: fest

Das ist ein standard Ausgleichsproblem!

§8.4. Totales lineares Ausgleichsproblem

total = Messfehler auch in den Messorte $\underline{\underline{t}}_i$

↔ $\underline{\underline{A}}\,\underline{x} = \underline{b}$ mit $\underline{b}$ und $\underline{\underline{A}}$ Fehlerhaft.

Bem $\underline{\underline{A}}\,\underline{x} = \underline{b}$ hat keine Lösung falls $\underline{b} \notin \text{Bild}(\underline{\underline{A}})$.

Ziel: finde $\hat{\underline{\underline{A}}}$, $\hat{\underline{x}}$, $\hat{\underline{b}}$ so dass

$\hat{\underline{\underline{A}}}\,\hat{\underline{x}} = \hat{\underline{b}}$ $\quad (\hat{\underline{b}} \in \text{Bild}(\hat{\underline{\underline{A}}}))$ $\quad \hat{\underline{\underline{A}}}$ $m \times n$, $\hat{b} \in \mathbb{R}^m$

"so nah wie möglich" an $\underline{\underline{A}}\,\underline{x} = \underline{b}$

$\underline{\underline{C}} = [\,\underline{\underline{A}}\ \underline{b}\,]$; $\hat{\underline{\underline{C}}} = [\,\hat{\underline{\underline{A}}}\ \hat{\underline{b}}\,]$

$\| \underline{\underline{C}} - \hat{\underline{\underline{C}}} \|_F = \min!$ Frobeniusnorm

$\| \underline{\underline{M}} \|_F^2 = \sum_{ij} |M_{ij}|^2$

mit $\hat{\underline{b}} \in \text{Bild}(\hat{\underline{\underline{A}}})$

$\text{rang}(\underline{\underline{A}}) = n$ auch $\text{Rang}(\hat{C}) = n$.

Lösung: das ist die Niedrigrangapproximation von $\underline{\underline{C}}$

$$\underline{\underline{C}} = \underline{\underline{U}}\,\underline{\underline{\Sigma}}\,\underline{\underline{V}}^H = \sum_{j=1}^{n+1} \sigma_j \underline{u}_j \underline{v}_j^H$$

Definiere $\hat{\underline{\underline{C}}} = \sum_{j=1}^{n} \sigma_j \underline{u}_j \underline{v}_j^H$

↳ optimale Approximation an $\underline{\underline{C}}$ in der Menge der Matrizen von Rang n ist.

Noch dazu wissen wir: $\underline{v}_j$ orthogonal $\Rightarrow \hat{\underline{\underline{C}}}\,\underline{v}_{n+1} = 0$

Falls $v_{n+1,n+1} \neq 0$

$$\hat{\underline{\underline{A}}}\,\hat{\underline{x}} = \hat{\underline{b}} \Rightarrow \quad \hat{\underline{x}} = -\frac{1}{v_{n+1,n+1}}\, \underline{v}_{n+1}.$$

Beweis

$$\hat{\underline{\underline{A}}}\,\hat{\underline{x}} = \hat{\underline{b}} \Leftrightarrow \hat{\underline{\underline{A}}}\,\hat{\underline{x}} - \hat{\underline{b}} = 0 \Leftrightarrow \left[\hat{\underline{\underline{A}}} \;\; \hat{\underline{b}}\right] \begin{bmatrix} \hat{\underline{x}} \\ -1 \end{bmatrix} = 0$$

$$\Leftrightarrow \begin{bmatrix} \hat{\underline{x}} \\ -1 \end{bmatrix} \in \ker\left(\hat{\underline{\underline{C}}}\right) \Rightarrow$$

$$\dim \ker\left(\hat{\underline{\underline{C}}}\right) = 1, \quad \underline{v}_{n+1} \in \ker \hat{\underline{\underline{C}}}$$

$$\begin{bmatrix} \hat{\underline{x}} \\ -1 \end{bmatrix} = k\, \underline{v}_{n+1}$$

Definiere $k = -\frac{1}{v_{n+1,n+1}} \Rightarrow \hat{\underline{x}} = -\frac{1}{v_{n+1,n+1}}\, \underline{v}_{n+1}.$

§ 8.5 Nichtlineare Ausgleichsrechnung

Modell $f(t, \underline{x}) = y$

↳ Parameters, aus Messungen zu bestimmen.

$$f(t_i, \underline{x}) \approx y_i \quad , i = 1, 2, \ldots, m$$

$$\underline{F}(\underline{x}) = \begin{bmatrix} f(t_1, \underline{x}) - y_1 \\ f(t_2, \underline{x}) - y_2 \\ \vdots \\ f(t_m, \underline{x}) - y_m \end{bmatrix}; \quad \underline{F}: \mathbb{R}^n \to \mathbb{R}^m$$

Finde $\underline{x}^* \in \mathbb{R}^n$

$\|\underline{F}(\underline{x}^*)\|_2^2$ Minimal!

$$\Phi(\underline{x}) = \frac{1}{2} \|\underline{F}(\underline{x})\|_2^2, \quad \Phi: \mathbb{R}^n \to [0, \infty[$$

Minimum von Φ:

Finde $x^*: D\Phi(\underline{x}^*) = 0$ ← Nullstellensuche!

mit Newton-Verfahren ⇒ quadratische Konvergenz

lokal.

$$D\Phi(x) = D\left(\tfrac{1}{2} F(x)^T F(x)\right) = DF(x)^T F(x)$$

$$D\Phi(x^*) = 0 \iff DF(x^*)^T F(x^*) = 0$$

dafür Newton!

In Newton brauchen wir $D(D\Phi(x)) = H\Phi(x)$

Hesse Matrix von Φ

$$H\Phi(x) = D\left(DF(x)^T F(x)\right) = DF(x)^T DF(x) + \sum_{j=1}^{m} F_j(x) D^2 F_j(x)$$

$\to$ Matrix

$$\left(H\Phi(x)\right)_{ik} = \sum_{j=1}^{m} \left(\frac{\partial F_j}{\partial x_k}(x) \frac{\partial F_j}{\partial x_i}(x) + F_j(x) \frac{\partial^2 F_j(x)}{\partial x_i \partial x_k} \right)$$

Newton-Schritt $x^{(k+1)} := x^{(k)} + s$

Newton-Korrektur s aus LGS

$$H\Phi(x^{(k)})\, s = - DF(x^{(k)})^T F(x^{(k)}) .$$

Alternative: Gauss-Newton-Verfahren.

Idee: linearisiere Lokal in jeder Schritt einer Iteration. $\Rightarrow$ eine Folge von lin. Ausgleichsprobleme

Linearisierung.

$$\operatorname*{argmin}_{x \in \mathbb{R}^n} \|F(x)\|_2^2 \approx \operatorname*{argmin}_{x \in \mathbb{R}^n} \|F(x^0) + DF(x^{(0)})(x - x^0)\|_2^2$$

$$= \operatorname*{argmin}_{x \in \mathbb{R}^n} \|Ax - b\|_2^2$$

mit $A = DF(x^0)$ und $b = F(x^0) - DF(x^0)\, x^0$

$\underline{x}^{(0)} \xrightarrow{\text{lin LSP}} \underline{x}^{(1)} \xrightarrow{\text{lin LSP}} \underline{x}^{(2)} \to \dots \to \underline{x}^{(k)}$

$\underline{x}^{(k+1)} = \underline{x}^{(k)} + \underline{s}^{(k)}$, $\underline{s}^{(k)} = \underset{\underline{s}}{\arg\min} \left\| F(x^{(k)}) + DF(x^{(k)})\underline{s} \right\|_2^2$

Nachteil: lineare Konvergenz.

Professionele Software: noch raffinierter Alg.

Levenberg-Marquardt-Verfahren.

$\min \left\{ \left\| \underline{F}(x^{(k)}) + \underline{\underline{D}}F(x^{(k)})\underline{s} \right\|_2^2 + \lambda^{(k)} \|\underline{s}\|_2^2 \right\}$

$\lambda^{(k)} > 0$, heuristisch gewählt.

Totales Ausgleichsproblem: Software:

scipy.odr

§9 Eigenwerte

Software: $\underline{\underline{A}} \in \mathbb{R}^{N\times N}$

- `eig(A)` kostet $O(N^3)$ für $\underline{\underline{A}} \in \mathbb{R}^{N\times N}$
- `eigh(A)` kostet $O(N^2)$ für $\underline{\underline{A}}^H = \underline{\underline{A}}$.

Für "kleines" $N < 20$: $e^{At} = \text{expm}(At)$

[Algorithmus Padé-Approximation $e^x \simeq \frac{p_m(x)}{q_n(x)}$ etc.]

Bem Die Eigenvektoren (EV) werden typischerweise nicht so gut approximiert)

Bsp
$$\underline{\underline{A}} = \begin{bmatrix} 5 & 1 & 0 \\ 0 & 5 & 1 \\ 0 & 0 & 5 \end{bmatrix}$$
EW = 5 dreifach
$$EV = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$\Rightarrow$ nicht diagonalisierbar

Bsp
$$\begin{cases} \dot{\underline{y}} = \underline{\underline{A}}\,\underline{y} & \underline{\underline{A}} \in \mathbb{R}^{N\times N} \\ \underline{y}(0) = y_0 \end{cases}$$

$N=2, N=3$ ok: $y(t) = e^{\underline{\underline{A}}t}\,\underline{y}(0)$

$$e^{\underline{\underline{A}}t} = \sum_{n=0}^{\infty} \frac{(\underline{\underline{A}}t)^n}{n!}$$

$N \approx 10$ $\underline{\underline{A}} = \underline{\underline{S}}\,\underline{\underline{D}}\,\underline{\underline{S}}^{-1}$ falls $\underline{\underline{A}}$ diagonalisierbar

$\Rightarrow e^{\underline{\underline{A}}t} = \underline{\underline{S}}\, e^{\underline{\underline{D}}t}\, \underline{\underline{S}}^{-1}$ N gross

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \end{bmatrix} = \begin{bmatrix} 5 & 1 & 0 \\ 0 & 5 & 1 \\ 0 & 0 & 5 \end{bmatrix}$$

$\dot{y}_1 = 5y_1 + y_2$

$\dot{y}_2 = 5y_2 + y_3$

$\dot{y}_3 = 5y_3 \Rightarrow y_3(t) = y_3(0)\, e^{5t}$

$\dot{y}_2 = 5y_2 + y_3(0)\, e^{5t} \Rightarrow y_2(t) = \left(y_2(0) + t y_3(0)\right) e^{5t}$

$\Rightarrow y_1(t) = \left(y_1(0) + t y_2(0) + \frac{t^2}{2} y_3(0)\right) e^{5t}$.

Falls $\underline{\underline{A}} = \begin{bmatrix} & & * \\ & \ddots & \\ 0 & & \end{bmatrix}$ dann können wir

$$\underline{y} = \underline{\underline{A}}\,\underline{z}$$

exakt lösen!

Bem

Wenn numerische Berechnungen zu machen sind, dann NIEMALS Jordan's Normalform, da sehr instabil.

Numerik: Schur-Zerlegung.

§9.2. Potenzmethoden.

$\underline{\underline{A}}$ nxn Matrix

– 1904 Nobel-Preis Physik

Def Rayleigh-Quotienten:

$$\rho_A(\underline{x}) = \frac{\underline{x}^H \underline{\underline{A}}\,\underline{x}}{\underline{x}^H \underline{x}}, \quad \underline{x} \neq \underline{0}$$

Bem $\underline{x}$ EV von $\underline{\underline{A}}$: $\underline{\underline{A}}\,\underline{x} = \lambda \underline{x}$

$$\Rightarrow \rho_A(\underline{x}) = \frac{\underline{x}^H \lambda \underline{x}}{\underline{x}^H \underline{x}} = \lambda$$

Bem $\rho_A(\underline{x}) = \operatorname{argmin}_{\alpha} \| \underline{\underline{A}}\,\underline{x} - \alpha \underline{x} \|_2$

Stationärpunkte von $\rho_A(\underline{x}) \Rightarrow$ EV.

Taylor für $\rho_A(x)$ um EV $\Rightarrow$

$$\rho_A(x) - \rho_A(EV) = O(\|x - EV\|_2^2) \quad \text{für } x \text{ Nähe von EV.}$$

Direkte Potenzmethode.

Ziel: finde das betragsgrösste EW von $\underline{\underline{A}}$ und ein EV dazu.

Annahme $\underline{\underline{A}}$ diagonalisierbar

$$\underline{\underline{S}}^{-1}\underline{\underline{A}}\,\underline{\underline{S}} = \mathrm{diag}(\lambda_1,\dots,\lambda_n)$$

EV stehen in den Spalten von $\underline{\underline{S}}$, $\|\underline{s}_j\|_2 = 1$

Annahme:

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|$$

$\mathbb{R}^n \ni \underline{x} = \sum_{j=1}^{n} a_j \underline{s}_j$ mit $a_1 \neq 0$, sonst beliebig

$$\underline{\underline{A}}\,\underline{x} = \sum_{j=1}^{n} a_j \underline{\underline{A}}\,\underline{s}_j = \sum_{j=1}^{n} a_j \lambda_j \underline{s}_j$$

$$\underline{\underline{A}}^2 \underline{x} = \sum_{j=1}^{n} a_j \lambda_j^2 \underline{s}_j \quad \dots$$

$$\underline{\underline{A}}^k \underline{x} = \sum_{j=1}^{n} a_j \lambda_j^k \underline{s}_j = \lambda_1^k \left(a_1 \underline{s}_1 + \sum_{j=2}^{n} \left(\frac{\lambda_j}{\lambda_1}\right)^k a_j \underline{s}_j \right)$$

$$\left|\frac{\lambda_j}{\lambda_1}\right| < 1 \Rightarrow \left|\frac{\lambda_j}{\lambda_1}\right|^k \ll 1 \longrightarrow \downarrow$$

für grosses k zeigt $\underline{\underline{A}}^k \underline{x}$ in der Richtung von $\underline{s}_1$

Somit $\dfrac{\underline{\underline{A}}^k \underline{x}}{\|\underline{\underline{A}}^k \underline{x}\|_2} \longrightarrow \pm \underline{s}_1$

Notiere $\underline{x}_k = \underline{\underline{A}}^k \underline{x} \Rightarrow \rho_{\underline{\underline{A}}}(\underline{x}_k) = \dfrac{\underline{x}_k^H \underline{\underline{A}}\,\underline{x}_k}{\underline{x}_k^H \underline{x}_k} =$

$$= \frac{1}{\underline{x}_k^H \underline{x}_k} \left(\underline{x}_k^H \sum_{j=1}^{n} a_j \lambda_j^{k+1} \underline{s}_j \right) = \lambda_1 + O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)$$

Potenzmethode: Wähle x_0 zufällig, $\|x_0\|=1$

für $k=1,2,\dots$

$$w = A x_{k-1}$$

$$x_k = \frac{w}{\|w\|}$$

$$\lambda = x_k^H A x_k$$

Theorem Potenzmethode liefert eine Iteration, die linear konvergiert gegen λ_1 mit der Rate $\left|\frac{\lambda_2}{\lambda_1}\right|$

Bem Falls A normal $\Rightarrow$ EV orthogonal $\Rightarrow$ Fehler $O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right)$

$\Rightarrow$ quadratische Konvergenz!

Bem Das war die erste Idee für Page Rank - Alg. von Google!

Bew $A = A^H$ ONB. von EV.

$$x = \sum_{j=1}^{n} c_j u_j = U c$$

$$\frac{x^H A x}{x^H x} = \frac{c^H U^H A U c}{c^H U^H U c} = \frac{c^H \Lambda c}{c^H c} =$$

$$= \frac{\lambda_1 |c_1|^2 + \lambda_2 |c_2|^2 + \dots + \lambda_n |c_n|^2}{|c_1|^2 + \dots + |c_n|^2}$$

$$\rho_A(x^{(k)}) = \lambda_1 + c\left(\frac{\lambda_2}{\lambda_1}\right)^{2k} + \dots)$$

Ziel: finde das betragskleinste $\in \mathbb{W}$:

Annahme: $\underline{\underline{A}}$ invertierbar:

$\underline{\underline{A}}\,\underline{x} = \lambda \underline{x} \Rightarrow \underline{x} = \lambda \underline{\underline{A}}^{-1}\underline{x}$

$\underline{\underline{A}}^{-1}|$

$\frac{1}{\lambda}\underline{x} = \underline{\underline{A}}^{-1}\underline{x}$

$\Rightarrow$ betragskleinste $\in \mathbb{W}$: λ_n : $\frac{1}{\lambda_n}\underline{x} = \underline{\underline{A}}^{-1}\underline{x}$

$\frac{1}{\lambda_n}$ betragsgrösste $\in \mathbb{W}$ von $\underline{\underline{A}}^{-1}$ und

dieselben EV!

Darum "inverse Potenzmethode" = Potenzmethode für $\underline{\underline{A}}^{-1}$.

Bem Wir berechnen nicht $\underline{\underline{A}}^{-1}$ sondern nur eine LU-Zerlegung von $\underline{\underline{A}}$ (strukturerhaltend) nur ein Mal

Dann Löse LGS mit Matrizen L, U um $\underline{\underline{A}}^{-1}\underline{x}$ zu implementieren (Alg. in Skript)

Ziel: Gegeben $\alpha \in \mathbb{C}$, finde $\in \mathbb{W}$ nah an α:

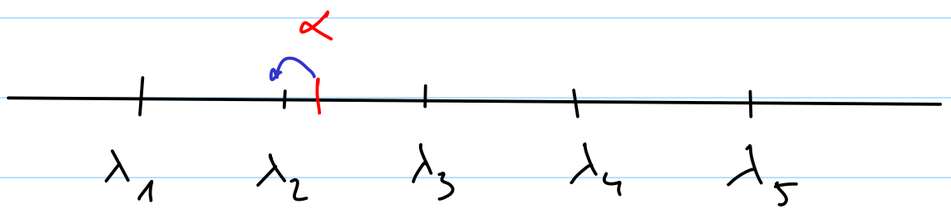
$|\alpha - \lambda| = \min\{|\alpha - \mu|;\ \mu \in \mathbb{W} \text{ von } \underline{\underline{A}}\}$

$\underline{\underline{A}}\,\underline{x} = \lambda \underline{x} \Leftrightarrow \underline{\underline{A}}\,\underline{x} - \alpha \underline{\underline{I}}\,\underline{x} = (\lambda - \alpha)\underline{x} \Leftrightarrow$

$(\underline{\underline{A}} - \alpha \underline{\underline{I}})\,\underline{x} = (\lambda - \alpha)\,\underline{x} \Leftrightarrow \frac{1}{\lambda - \alpha}\underline{x} = (\underline{\underline{A}} - \alpha \underline{\underline{I}})^{-1}\underline{x}$

$\Rightarrow$ Iteration mit $(\underline{\underline{A}} - \alpha \underline{\underline{I}})^{-1} \Rightarrow \frac{1}{\lambda - \alpha} \Rightarrow \lambda$

"shifted inverse Iteration"



Bem Die Potenzmethode ist schneller wenn $\alpha \approx \lambda_j$.

Darum die Idee: wähle α adaptiv, z.B.

$\alpha = \rho_A(\underline{x}^{(k-1)})$ im k-ten Schritt.

$\Rightarrow$ beschleunigte Konvergenz

Rayleigh-Quotienten-Iteration

(Skript Code & Bsp)

Bem Die Potenzmethoden brauche einen guten Start-Vektor

Für RQI bekommt man einen guten Start-Vektor mit einige Schritte von shifted inverse Iteration.

$\Rightarrow$ Konvergenzordnung 3 !!!

Für $\underline{\underline{A}} = \underline{\underline{A}}^H$; $\underline{x} = \sum_{j=1}^{n} c_j \underline{u}_j = \underline{\underline{U}}\,\underline{c}$, $\underline{u}_1, \dots, \underline{u}_n \in V$

$$\rho_{\underline{\underline{A}}}(\underline{x}) = \frac{\underline{x}^H \underline{\underline{A}} \underline{x}}{\underline{x}^H \underline{x}} = \frac{\lambda_1 |c_1|^2 + \dots + \lambda_n |c_n|^2}{\|x\|_2^2} \Rightarrow$$

$$\lambda_1 \le \rho_{\underline{\underline{A}}}(\underline{x}) \le \lambda_n \quad \text{falls} \quad \lambda_1 \le \lambda_2 \le \dots \le \lambda_n$$

$$\rho_{\underline{\underline{A}}}(\underline{x}) \in [\lambda_1, \lambda_n]$$

$\lambda_1 = \min_{\underline{x} \in \mathbb{C}^n} \rho_{\underline{\underline{A}}}(x)$ erreicht für $\underline{c} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

$\lambda_n = \max_{\underline{x} \in \mathbb{C}^n} \rho_{\underline{\underline{A}}}(\underline{x})$ $\quad \underline{c} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$

Bem $\underline{x} \perp \underline{u}_1 \Rightarrow c_1 = 0$, $\underline{x} = c_2 \underline{u}_2 + \dots + c_n \underline{u}_n$

$\rho_A(\underline{x}) \ge \lambda_2$ erreicht für $\underline{c} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

$$\lambda_2 = \min_{\substack{\underline{x} \in \mathbb{C}^n \\ \underline{x} \perp \underline{u}_1}} \rho_{\underline{\underline{A}}}(\underline{x})$$

$$\lambda_{n-1} = \max_{\substack{\underline{x} \in \mathbb{C}^n \\ \underline{x} \perp \underline{u}_n}} \rho_{\underline{\underline{A}}}(\underline{x}) \qquad \text{und so weiter}$$

$$\lambda_k = \min_{\underline{x} \perp \operatorname{span}\{\underline{u}_1, \dots, \underline{u}_{k-1}\}} \rho_{\underline{\underline{A}}}(\underline{x}) = \min_{\underline{x} \in \operatorname{span}\{\underline{u}_k, \underline{u}_{k+1}, \dots, \underline{u}_n\}} \rho_{\underline{\underline{A}}}(\underline{x}) =$$

$$= \max_{\underline{x} \perp \{\underline{u}_{k+1}, \dots, \underline{u}_n\}} \rho_A(x) = \max_{\underline{x} \in \operatorname{span}\{\underline{u}_1, \dots, \underline{u}_k\}} \rho_A(x)$$

Theorem [Courant-Fisher]

$$\lambda_k = \min_{\dim U = k} \max_{x \in U} \frac{x^H A x}{x^H x} = \max_{\dim U = n-k+1} \min_{x \in U} \frac{x^H A x}{x^H x}$$

Theorem [Cauchy]

$\underline{\underline{H}} \in \mathbb{C}^{m \times m}$ mit EW $\theta_1, \dots, \theta_m$

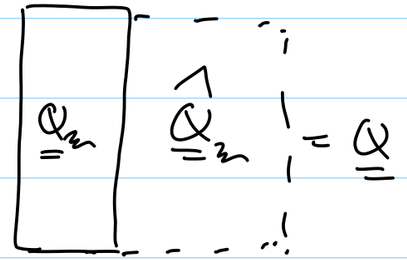
$$\underline{\underline{A}} = \begin{bmatrix} \underline{\underline{H}} & \underline{\underline{B}}^H \\ \underline{\underline{B}} & \underline{\underline{R}} \end{bmatrix} \qquad m \ll n$$

Dann $\lambda_k \le \theta_k \le \lambda_{k+n-m}$

Konsequenz

$\underline{\underline{Q}}_m \in \mathbb{C}^{n \times m}$ mit $\underline{\underline{Q}}_m^H \underline{\underline{Q}}_m = \underline{\underline{I}}$

erweitere $\underline{\underline{Q}}_m$ auf ONB in $\mathbb{C}^n$



$$\underline{\underline{Q}}^H \underline{\underline{A}}\, \underline{\underline{Q}} = \begin{bmatrix} \underline{\underline{Q}}_m^H \underline{\underline{A}}\, \underline{\underline{Q}}_m & \underline{\underline{Q}}_m^H \underline{\underline{A}}\, \hat{\underline{\underline{Q}}}_m \\ \hat{\underline{\underline{Q}}}_m^H \underline{\underline{A}}\, \underline{\underline{Q}}_m & \hat{\underline{\underline{Q}}}_m^H \underline{\underline{A}}\, \hat{\underline{\underline{Q}}}_m \end{bmatrix}$$

hat dieselbe EW wie $\underline{\underline{A}}$

Idee Für n gross und m klein, wähle $\underline{\underline{Q}}_m$

so dass Bild $\underline{\underline{Q}}_m \approx \operatorname{span}\{\underline{u}_1, \dots, \underline{u}_m\}$ ← EV von $\underline{\underline{A}}$

$\Rightarrow$ gute Approximation $\theta_k \approx \lambda_k$ für $\lambda = 1, 2, \dots, m$

Wie ? modifiziertes Gram-Schmidt für

$\{ \underline{v}, \underline{A}\underline{v}, \underline{A}^2\underline{v}, \dots, \underline{A}^{m-1}\underline{v} \}$

Krylov-Verfahren (Arnoldi, Lanczos)

§ 9.3 Krylov-Verfahren

für "kleine" Matrizen: "eig" (QR-Alg.) gut

für "grosse" Matrizen ist eig zu langsam,

ausserdem QR-Alg. zerstört die Struktur von $\underline{A}$

Krylov-Verfahren: grosse, dünnbesetzte Matrizen.

Def Sei $0 \neq \underline{z} \in \mathbb{C}^n$, $\underline{A} \in \mathbb{C}^{n \times n}$

$\mathcal{K}_\ell(\underline{A}, \underline{z}) := \operatorname{span}\{ \underline{z}, \underline{A}\underline{z}, \underline{A}^2\underline{z}, \dots, \underline{A}^{\ell-1}\underline{z} \} =$

$= \{ p(\underline{A})\underline{z} \; ; \; p = \text{Polynom vom Grad} \leq \ell-1 \}$

Krylov-Raum

Suche eine ONB in $\mathcal{K}_\ell(\underline{A}, \underline{z})$

$\mathcal{K}_1 = \operatorname{span}\{\underline{z}\} \subset \operatorname{span}\{\underline{z}, \underline{A}\underline{z}\} = \mathcal{K}_2 \subset \dots \subset \mathcal{K}_\ell$

Iterativ: gegeben $\underline{v}_1, \dots, \underline{v}_\ell$ ONB in $\mathcal{K}_\ell(\underline{A}, \underline{z})$

$\operatorname{span}\{\underline{v}_1, \dots, \underline{v}_j\} = \mathcal{K}_j(\underline{A}, \underline{z})$ für $j = 1, 2, \dots, \ell$

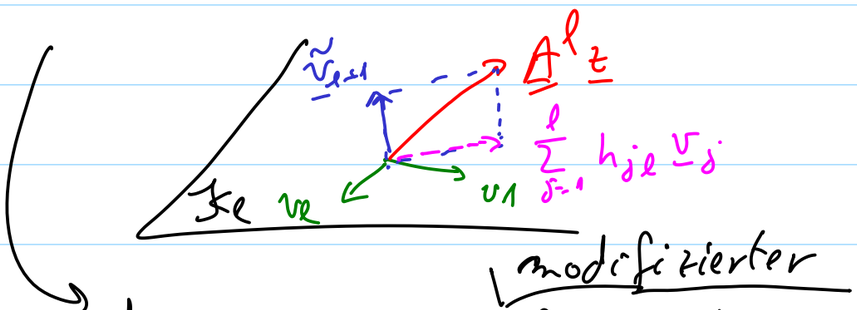
baue $\underline{v}_1, \dots, \underline{v}_{\ell+1}$ ONB in $\mathcal{K}_{\ell+1}(\underline{A}, \underline{z})$

$\mathcal{K}_{\ell+1}(\underline{A}, \underline{z}) = \operatorname{span}\{ \underline{z}, \underline{A}\underline{z}, \dots, \underline{A}^{\ell-1}\underline{z}, \underline{A}^{\ell}\underline{z} \}$

entweder $\underline{\underline{A}}^l \underline{z} \in \mathcal{K}_l(\underline{\underline{A}}, \underline{z})$, d.h. $\underline{\underline{A}}^l \underline{z}$ (lin. abhängig von $\underline{z}, \dots, \underline{\underline{A}}^{l-1}\underline{z}$

oder

$\underline{\underline{A}}^l \underline{z} \notin \mathcal{K}_l(\underline{\underline{A}}, \underline{z}) \Rightarrow \underline{\underline{A}}^l \underline{z} \in \mathcal{K}_{l+1} \setminus \mathcal{K}_l$



modifizierter

dann v_{l+1} aus Gram-Schmidt Schritt

für $\underline{z}, \underline{\underline{A}}\underline{z}, \dots, \underline{\underline{A}}^{l-1}\underline{z}, \underline{\underline{A}}^l \underline{z}$

$$\underline{\tilde{v}}_{l+1} = \underline{\underline{A}}^l \underline{z} - \sum_{j=1}^{l} h_{jl} \underline{v}_j$$

$$\underline{v}_l = \frac{\underline{\tilde{v}}_{l+1}}{\|\underline{\tilde{v}}_{l+1}\|}$$; daßei sind $h_{jl} = \underline{v}_j^H \underline{\underline{A}} \underline{v}_l$

(orth. in $\mathcal{K}_l$)

Algorithmus [Arnoldi Prozess]

$\underline{z}$ = beliebig.

$\underline{v}_1 = \underline{z} / \|\underline{z}\|$

für $l = 1, 2, \dots, k-1$:

$\underline{\tilde{v}} = \underline{\underline{A}} \underline{v}_l$

für $j = 1, 2, \dots, l$:

$h_{jl} = \underline{v}_j^H \underline{\tilde{v}}$

mod. GS: $\underline{\tilde{v}} = \underline{\tilde{v}} - h_{jl} \underline{v}_j$

$h_{l+1,l} = \|\underline{\tilde{v}}\|$

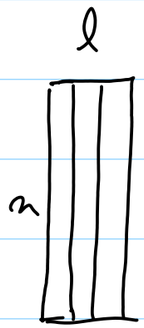
$v_{l+1} = \tilde{v} / h_{l+1,l}$

$l=1$	$l=2$	$l=3$
h_{11}	h_{12}	h_{13}
h_{21}	h_{22}	h_{23}
	h_{32}	h_{33}
		h_{43}

Bem Falls $h_{l+1,l} = 0 \Rightarrow$ Abbruch der Iteration

$\hookrightarrow \underline{\underline{A}}\, \underline{v}_l \in \mathcal{K}_l(\underline{\underline{A}}, z)$

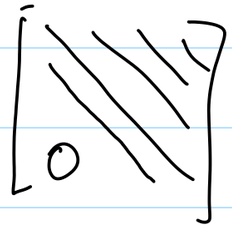
$$\underline{\underline{V}}_l = [\underline{v}_1\ \underline{v}_2 \dots \underline{v}_l] \in \mathbb{C}^{n \times l}$$



$$\underline{\underline{\tilde{H}}}_l = \begin{bmatrix} h_{11} & h_{12} & h_{13} & \\ h_{21} & h_{22} & h_{23} & \\ 0 & h_{32} & h_{33} & \\ & 0 & 0 & h_{43} \end{bmatrix} \in \mathbb{C}^{l+1,\,l}$$

$\underline{\underline{H}}_l \in \mathbb{C}^{l,l}$

$\underline{\underline{H}}_l$ obere Hessenbergmatrix $l \times l$

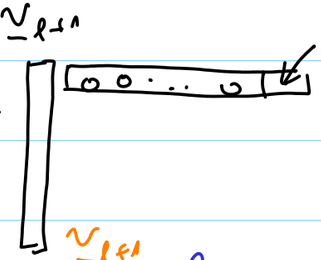


Aus Konstruktion: $\underline{\underline{A}}\, \underline{v}_k = h_{k+1,k}\, \underline{v}_{k+1} + \sum_{j=1}^{k} h_{jk}\, \underline{v}_j$

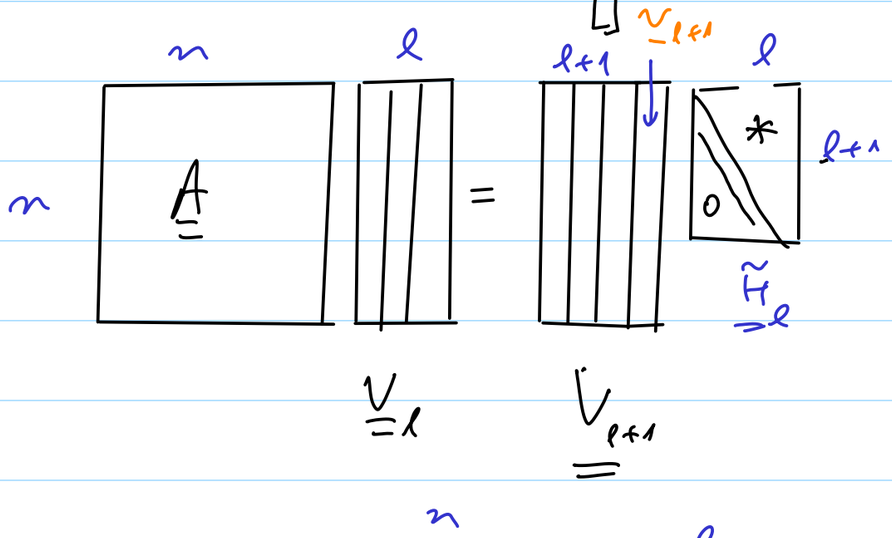
für $k = 1, 2 \dots, l$

das heisst:

$$\underline{\underline{A}}\, \underline{\underline{V}}_l = \underline{\underline{V}}_{l+1}\, \underline{\underline{\tilde{H}}}_l =$$



$$+ \underline{\underline{V}}_l\, \underline{\underline{H}}_l$$



Bem 1)

$$l \begin{bmatrix} \underline{\underline{V}}_l^H \end{bmatrix} \begin{matrix} \\ \\ \\ n \end{matrix} \begin{bmatrix} \\ \underline{\underline{V}}_l \\ \\ \end{bmatrix} \begin{matrix} \\ \\ \\ n \end{matrix} = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix} \begin{matrix} \\ \\ l \end{matrix}$$

$$\underline{\underline{V}}_l^H\, \underline{\underline{V}}_l = \underline{\underline{I}}_l$$

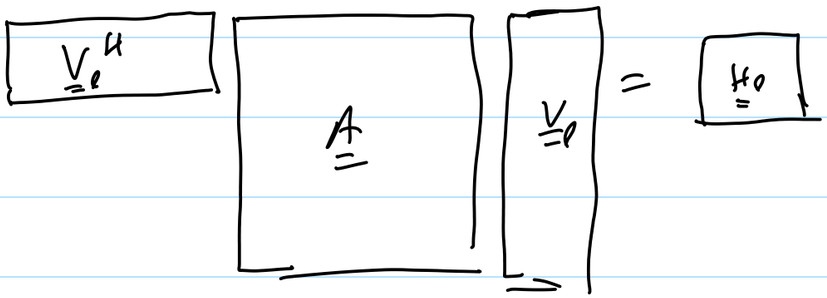
2)

$\underline{V}_{\ell}^H$ |

$$\underline{V}_{\ell}^H \underline{A}\,\underline{V}_{\ell} = \underline{V}_{\ell}^H \underline{V}_{\ell+1} \begin{pmatrix} h_{\ell+1,\ell} \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \underline{V}_{\ell}^H \underline{V}_{\ell}\,\underline{H}_{\ell}$$

$\underbrace{\qquad}_{0}$ $\qquad$ $\underbrace{\qquad}_{\underline{I}_{\ell}}$

$$\Rightarrow \quad \underline{V}_{\ell}^H \underline{A}\,\underline{V}_{\ell} = \underline{H}_{\ell}$$



3) falls $h_{\ell+1,\ell} = 0 \Rightarrow \mathcal{K}_{\ell+1} = \mathcal{K}_{\ell}$ und

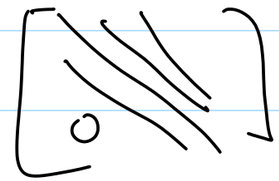
$$\underline{A}\,\underline{V}_{\ell} = \underline{V}_{\ell}\,\underline{H}_{\ell}$$

4) falls $\underline{A}$ Hermite-symmetrisch: $\underline{A}^H = \underline{A}$

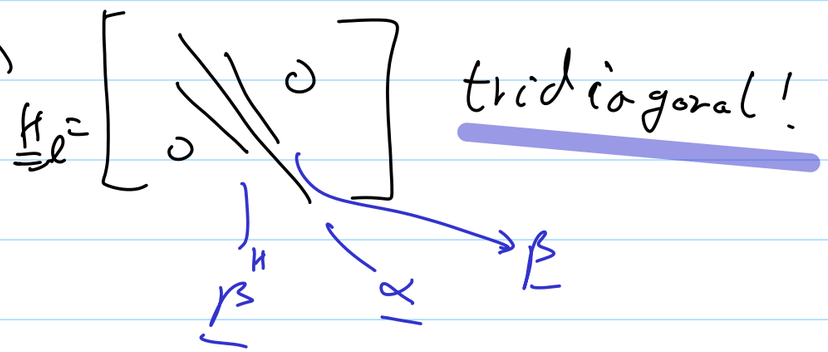
$$\underline{H}_{\ell}^H = \left(\underline{V}_{\ell}^H \underline{A}\,\underline{V}_{\ell}\right)^H = \underline{V}_{\ell}^H \underline{A}^H \underline{V}_{\ell} = \underline{V}_{\ell}^H \underline{A}\,\underline{V}_{\ell} =$$

$= \underline{H}_{\ell} \Rightarrow \underline{H}_{\ell}$ auch. Hermite-symmetrisch!

$\underline{H}_{\ell}$ Hessenberg



$\Rightarrow$



$\Rightarrow$ es reichen die Vektoren α, β um $\underline{H}_{\ell}$ zu speichern

und die innere Schleife im Arnoldi Prozess hat konstante Länge $\ell = 2$.

$$\tilde{v}_{\ell+1} = A v_\ell - h_{\ell\ell} v_\ell - h_{\ell-1,\ell} v_{\ell-1}$$

Lanczos-Verfahren $O(nk)$

Arnoldi-Verfahren $O(nk^2)$

Allgemeinen-Namen: Krylov-Raum-Verfahren.

Theorem Falls $h_{\ell+1,\ell} = 0$ und $h_{j+1,j} \neq 0$ für $j = 1, 2, \ldots, \ell-1$, dann

(1) jeder EW von H_ℓ ist auch EW von A

(2) falls A regulär, dann gibt es $y \in \mathbb{C}^\ell$ so dass

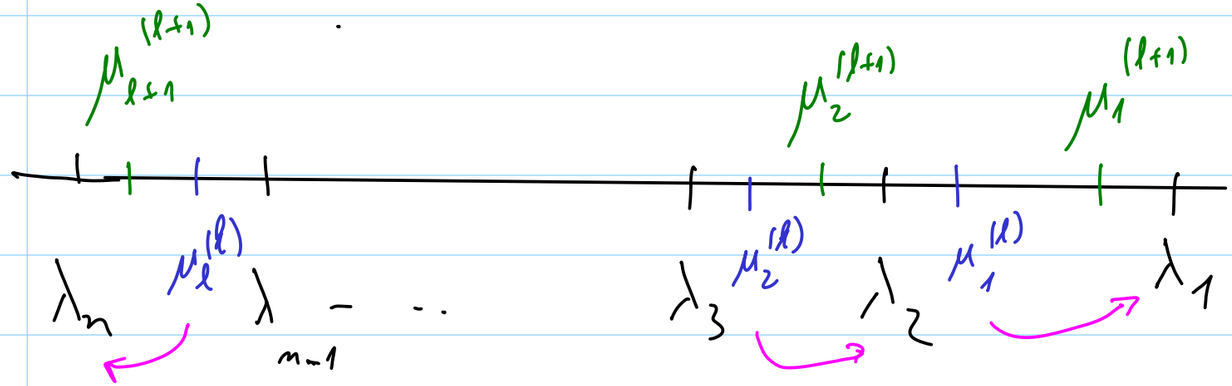
$Ax = b$ mit $x = V_\ell y$

Theorem 7.4.12. Seien $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ und $\mu_1^{(\ell)} \geq \mu_2^{(\ell)} \geq \cdots \geq \mu_\ell^{(\ell)}$ die Eigenwerte der Hermite-symmetrischen Matrix $\mathbf{A} \in \mathbb{C}^{n\times n}$, bzw. von $\mathbf{H}_\ell = \mathbf{V}_\ell^H \mathbf{A} \mathbf{V}_\ell$ für $\ell = 1, 2, \ldots$. Dann gelten für $1 \leq j \leq \ell$ die Ungleichungsketten

$$\lambda_{n-j+1} \leq \mu_{\ell+1-j+1}^{(\ell+1)} \leq \mu_{\ell-j+1}^{(\ell)}$$

und

$$\mu_j^{(\ell)} \leq \mu_j^{(\ell+1)} \leq \lambda_j .$$



Skript → einfache Implementierung

ARPACK ← eigvals

Bsp im Skript.

Lanczos ⇝ "ghost EW" da Orthogonalität verloren!

⇒ re-orthogonalisiere

§ 10 Lineare Anfangswertprobleme

1. Fall

$$\begin{cases} \dot{\underline{y}} = \underline{\underline{A}}\,\underline{y} \\ \underline{y}(0) = \underline{y}_0 \end{cases} \qquad \underline{\underline{A}} = \underline{\underline{S}}^{-1}\underline{\underline{D}}\,\underline{\underline{S}}$$

$\hat{\underline{y}} = \underline{\underline{S}}^{-1}\underline{y} \Rightarrow$ entkoppeln

$$\begin{cases} \dot{\hat{y}}_1 = \lambda_1 \hat{y}_1 \\ \dots \\ \dot{\hat{y}}_d = \lambda_d \hat{y}_d \end{cases} \Rightarrow \hat{y}_i(t) = \left(\underline{\underline{S}}^{-1}\underline{y}_0\right)_i e^{\lambda_i t} \qquad t \in \mathbb{R}$$

$$\underline{y}(t) = \underline{\underline{S}} \begin{bmatrix} e^{\lambda_1 t} & & \\ & \ddots & \\ & & e^{\lambda_d t} \end{bmatrix} \underline{\underline{S}}^{-1} \underline{y}_0$$

ok für kleine d, aber für $d \geq 5$ instabil.

Für $d = 5, 6, \dots, 20$ besser `expm`

$\hookrightarrow$ $\underline{y}(t) \approx \mathrm{expm}(\underline{\underline{A}}t)\,\underline{y}_0$

Padé-Approximation $d \geq 21$ zulangsam

d gross, $\underline{\underline{A}}$ dünnbesetzt: Krylov-Verfahren

Für $\underline{\underline{A}}$: es gibt $\underline{\underline{V}}_m \in \mathbb{C}^{d \times m}$ mit orthonormalen Spalten.

$$\underline{\underline{V}}_m^H \underline{\underline{A}}\,\underline{\underline{V}}_m = \underline{\underline{H}}_m$$

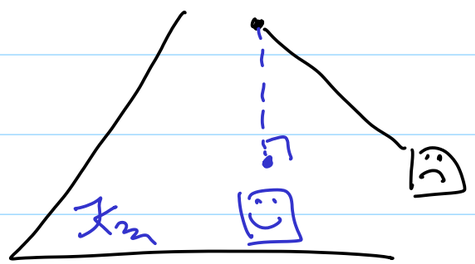
$m \times m$-Matrix (klein)

$\hookrightarrow$ Hessenberg $m \ll d$.

$$\begin{cases} \text{Finde } \underline{y}(t) \in \mathbb{R}^d \supset \mathcal{K}_m(\underline{\underline{A}}, \underline{y}_0) = \mathrm{span}\{\underline{y}_0, \underline{\underline{A}}\underline{y}_0, \dots, \underline{\underline{A}}^{m-1}\underline{y}_0\} \\ \dot{\underline{y}}(t) - \underline{\underline{A}}\,\underline{y}(t) = 0 \\ \underline{y}(0) = \underline{y}_0 \end{cases}$$

$= \mathrm{span}\{\underline{v}_1, \dots, \underline{v}_m\}$

ONB!



Darum ersetzen wir das Problem durch

Finde $\underline{u}_m \in \mathcal{K}_m(\underline{\underline{A}}, \underline{y}_0)$ so dass

Residuum $\left(\dot{\underline{u}}_m(t) - \underline{\underline{A}}\,\underline{u}_m(t)\right) \perp \mathcal{K}_m(\underline{\underline{A}}, \underline{y}_0)$

$\underline{u}_m(0) = \underline{y}_0$

$\Leftrightarrow$

Finde $\underline{u}_m \in \mathcal{K}_m(\underline{\underline{A}}, \underline{y}_0)$

$\langle \underline{w}, \dot{\underline{u}}_m(t) - \underline{\underline{A}}\,\underline{u}_m(t) \rangle = 0$ für alle $w \in \mathcal{K}_m(\underline{\underline{A}}, \underline{y}_0)$

$\underline{u}_m(0) = \underline{y}_0$

$\underline{u}_m(t) \in \mathcal{K}_m(\underline{\underline{A}}, \underline{y}_0) = \operatorname{span}\{\underline{v}_1, \dots, \underline{v}_m\} \Rightarrow$

$$\underline{u}_m(t) = \sum_{k=1}^{m} c_k(t)\,\underline{v}_k \quad ; \quad \underline{c}(t) = \begin{bmatrix} c_1(t) \\ \vdots \\ c_m(t) \end{bmatrix} \in \mathbb{C}^m$$

Einsetzen:

$$\langle \underline{w}, \sum_{k=1}^{m} \dot{c}_k(t)\,\underline{v}_k - \sum_{k=1}^{m} c_k(t)\,\underline{\underline{A}}\,\underline{v}_k \rangle = 0$$

für alle $\underline{w} \in \mathcal{K}_m(\underline{\underline{A}}, \underline{y}_0)$

Wähle $\underline{w} = \underline{v}_1$:

$$\langle \underline{v}_1, \sum_{k=1}^{m} \dot{c}_k(t)\,\underline{v}_k \rangle = \langle \underline{v}_1, \sum_{k=1}^{m} c_k(t)\,\underline{\underline{A}}\,\underline{v}_k \rangle$$

$$\sum_{k=1}^{m} \dot{c}_k(t) \underbrace{\langle \underline{v}_1, \underline{v}_k \rangle}_{\underline{v}_1^H \underline{v}_k} = \sum_{k=1}^{m} c_k(t) \langle \underline{v}_1, \underline{\underline{A}}\,\underline{v}_k \rangle$$

$\underline{v}_1^H \underline{v}_k = 0$ falls $k \neq 1$ da ONB

1 falls $k = 1$

$$\underline{v}_1^H \underline{\underline{A}}\,\underline{v}_k = (\underline{\underline{H}}_m)_{1k}$$

$$\dot{c}_1(t) = \sum_{k=1}^{m} (\underline{\underline{H}}_m)_{1k}\, c_k(t)$$

Für $w = \underline{v}_2, \underline{v}_3, \dots, \underline{v}_m$ analog $\Rightarrow$

$$\begin{cases} \dot{\underline{c}}(t) = \underline{\underline{H}}_m\, \underline{c}(t) \\ \underline{y}(0) = \underline{y}_0 \Rightarrow \underline{c}(0) = \|\underline{y}_0\| \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^m \end{cases}$$

m klein! $\Rightarrow$ Padé geht schnell

$$\underline{c}(t) = \|\underline{y}_0\|\, e^{\underline{\underline{H}}_m t} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$\Rightarrow$ Lösung in $\mathcal{K}_m(\underline{\underline{A}}, \underline{y}_0)$

$$\underline{u}_m(t) = \sum_{k=1}^{m} c_k(t)\, \underline{v}_k = \|\underline{y}_0\|\, \underline{\underline{V}}_m\, e^{\underline{\underline{H}}_m t} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

2. Fall $\dot{\underline{y}}(t) = \underline{\underline{A}}\,\underline{y}(t) + g(t)$ inhomogene Fall.

Variation der Konstanten

$$\underline{y}(t) = e^{\underline{\underline{A}}(t-t_0)}\, \underline{y}_0 + \int_{t_0}^{t} e^{\underline{\underline{A}}(t-s)}\, g(s)\, ds$$

3. Fall $\dot{\underline{y}}(t) = \underline{\underline{A}}(t)\,\underline{y}(t)$: Magnus-Integratoren

Unter bestimmte Voraussetzungen:

$$\underline{y}(t) = e^{\underline{\underline{\Omega}}(t,\, t_0)}\, \underline{y}_0$$ Magnus Entwicklung

mit $\underline{\underline{\Omega}} = \sum_{k=1}^{\infty} \underline{\underline{\Omega}}_k$; Kommutator $[A,B] = AB - BA$

$$\Omega_1 = \int_{t_0}^{t} \mathbf{A}(\tau_1)d\tau_1 ,$$

$$\Omega_2 = \frac{1}{2}\int_{t_0}^{t}\int_{t_0}^{\tau_1} [\mathbf{A}(\tau_1), \mathbf{A}(\tau_2)]d\tau_2 d\tau_1 ,$$

$$\Omega_3 = \frac{1}{12}\int_{t_0}^{t}\int_{t_0}^{\tau_1}\int_{t_0}^{\tau_2} [[\mathbf{A}(\tau_1), \mathbf{A}(\tau_2)], \mathbf{A}(\tau_3)] + [\mathbf{A}(\tau_1), [\mathbf{A}(\tau_2), \mathbf{A}(\tau_3)]]d\tau_3 d\tau_2 d\tau_1 .$$

Idee: Statt $\dot{\underline{y}} = \underline{\underline{A}}(t)\,\underline{y}$ löse $\dot{\hat{\underline{y}}} = \hat{\underline{\underline{A}}}(t)\,\hat{\underline{y}}$

wobei $\hat{\underline{\underline{A}}}(t) = \sum_{i=1}^{p} l_i(t)\,\underline{\underline{A}}(t_n + c_i h)$

$l_i(t)$ = Lagrange Polynom in $t_n + c_i h$

c_i = Quadraturknoten zwischen $[0,1]$

$t_n + c_i h \in [t_n, t_n + h]$

$\Rightarrow \hat{\underline{\underline{A}}}$ = Polynom vom Grad s in t

$\hat{\underline{\underline{A}}}(t_n + c_i h) = \underline{\underline{A}}(t_n + c_i h)$ für $i = 1, 2, \dots, p$.

Dann Magnus-Entwicklung für

$$\dot{\hat{\underline{y}}} = \hat{\underline{\underline{A}}}(t)\,\hat{\underline{y}}$$

Vorteil: Integration nur von Polynome in t $\Rightarrow$ einfach und exakt berechnen.

$\hat{\underline{\underline{A}}}(t)$ glatt; man kann zeigen,

Rest in der Magnus-Entwicklung ($\hat{\underline{\underline{\Omega}}}$) ist

$O(t^5)$ nach 4 Terme.

Theorem (b_i, c_i) $i=1,\dots,s$ Quadraturformel der Ordnung $p \geq s$

$$y(t_n+h) - \hat{y}(t_n+h) = O(h^{p+1})$$

Beispiel 2.8.1 (Magnus-Verfahren 2. Ordnung)**.** Die Mittelpunktsregel auf $[0,1]$ ergibt den Schritt von t_n zu t_{n+1} als:

$$\mathbf{y}_{n+1} = e^{\mathbf{\Omega}^{[2]}}\mathbf{y}_n \text{ mit } \mathbf{\Omega}^{[2]} = h\mathbf{A}(t_n + \frac{h}{2}) \,,$$

mit dem lokalen Fehler $\mathcal{O}(h^{2+1})$ und dem globalen Fehler $\mathcal{O}(h^2)$.

Beispiel 2.8.2 (Magnus-Verfahren 4. Ordnung)**.** Die Gauss-Quadratur mit 2 Punkten auf $[0,1]$ ergibt den Schritt von t_n zu t_{n+1} als:

$$\mathbf{y}_{n+1} = e^{\mathbf{\Omega}^{[4]}}\mathbf{y}_n \text{ mit } \mathbf{\Omega}^{[4]} = \frac{h}{2}(\mathbf{A}_1 + \mathbf{A}_2) - h^2\frac{\sqrt{3}}{12}[\mathbf{A}_1, \mathbf{A}_2]$$

wobei

$$\begin{aligned} \mathbf{A}_1 &= \mathbf{A}\left(t_n + \left(\frac{1}{2} - \frac{\sqrt{3}}{6}\right)h\right) \,, \\ \mathbf{A}_2 &= \mathbf{A}\left(t_n + \left(\frac{1}{2} + \frac{\sqrt{3}}{6}\right)h\right) \,. \end{aligned}$$

Der lokale Fehler ist $\mathcal{O}(h^{4+1})$ und der globale Fehler ist $\mathcal{O}(h^4)$.

Beispiel 2.8.3 (Magnus-Verfahren 4. Ordnung - Simpson-Regel)**.** Die Simpson-Regel mit 3 Punkten auf $[0,1]$ ergibt den Schritt von t_n zu t_{n+1} als:

$$\mathbf{y}_{n+1} = e^{\mathbf{\Omega}^{[4]}}\mathbf{y}_n \text{ mit } \mathbf{\Omega}^{[4]} = \frac{h}{6}(\mathbf{A}_1 + 4\mathbf{A}_2 + \mathbf{A}_3) - h^2\frac{1}{72}[\mathbf{A}_1 + 4\mathbf{A}_2 + \mathbf{A}_3, \mathbf{A}_3 - \mathbf{A}_1]$$

wobei

$$\begin{aligned} \mathbf{A}_1 &= \mathbf{A}(t_n) \,, \\ \mathbf{A}_2 &= \mathbf{A}\left(t_n + \frac{h}{2}\right) \,, \\ \mathbf{A}_3 &= \mathbf{A}\left(t_n + h\right) \,. \end{aligned}$$

Der lokale Fehler ist $\mathcal{O}(h^{4+1})$ und der globale Fehler ist $\mathcal{O}(h^4)$. Eine andere Methode derselben Ordnung ergibt sich mit

$$\mathbf{\Omega}^{[4]} = \frac{h}{6}(\mathbf{A}_1 + 4\mathbf{A}_2 + \mathbf{A}_3) - h^2\frac{1}{12}[\mathbf{A}_1, \mathbf{A}_3]$$

Ordnung 6
ohne Kommutatoren → Skript
Bsp.

§11 Exponentielle Integratoren

$$\begin{cases} \dot{\underline{y}} = \underline{f}(\underline{y}) & \text{mit } \underline{f} \text{ stetig differenzierbar} \\ \underline{y}(0) = \underline{y}_0 \end{cases}$$

autonom

Idee der Linearisierung: $\underline{\underline{J}} := \underline{\underline{D}}\underline{f}(\underline{y}_0)$

$$\dot{\underline{y}} = \underbrace{\underline{\underline{J}}\,\underline{y}}_{\text{linear}} + \underbrace{\underline{f}(\underline{y}) - \underline{\underline{J}}\,\underline{y}}_{\underline{g}(\underline{y})} \qquad \dot{\underline{y}} = \underline{\underline{J}}\,\underline{y} + \underline{g}(\underline{y})$$

Variation der Konstanten:

$$\underline{y}(h) = e^{\underline{\underline{J}}h}\,\underline{y}_0 + \int_0^h e^{\underline{\underline{J}}(h-\tau)}\,\underline{g}(\underline{y}(\tau))\,d\tau$$

Für das Integral ersetze die Unbekannte $\underline{y}(\tau)$ durch $\underline{y}_0 \Rightarrow$

$$\int_0^h e^{\underline{\underline{J}}(h-\tau)}\,\underline{g}(\underline{y}(\tau))\,d\tau \approx \int_0^h e^{\underline{\underline{J}}(h-\tau)}\,\underline{g}(\underline{y}_0)\,d\tau =$$

$$= h\,\varphi(h\underline{\underline{J}})\,\underline{g}(\underline{y}_0)$$

$$\varphi(z) \overset{\text{def}}{=} \frac{e^z - 1}{z} = \sum_{n=1}^{\infty} \frac{1}{n!} z^{n-1} = \sum_{n=0}^{\infty} \frac{z^n}{(n+1)!}$$

Beweis

$$\int_0^h e^{\underline{\underline{J}}(h-\tau)}\,\underline{g}(\underline{y}_0)\,d\tau = \int_0^h \sum_{n=0}^{\infty} \frac{1}{n!} \left(\underline{\underline{J}}(h-\tau)\right)^n \underline{g}(\underline{y}_0)\,d\tau =$$

Umtauschen

$$\int_0^h (h-\tau)^n\,d\tau$$

$$= \sum_{n=0}^{\infty} \frac{1}{n!}\,\underline{\underline{J}}^n (-1) \frac{(h-\tau)^{n+1}}{n+1}\Bigg|_{\tau=0}^{\tau=h} \underline{g}(\underline{y}_0) =$$

$$= \sum_{n=0}^{\infty} \frac{h}{(n+1)!}\,\underline{\underline{J}}^n h^n \underline{g}(\underline{y}_0) = h\varphi(h\underline{\underline{J}})\,\underline{g}(\underline{y}_0)$$

Bem Definition von g: $g(y_0) = f(y_0) - J y_0$ $\Rightarrow$

$$h\,\varphi(hJ)\,g(y_0) = h\,\varphi(hJ)\,f(y_0) - h\,\varphi(hJ)\,J y_0$$

$$h\,\varphi(hJ)\,J\,y_0 = h\sum_{n=0}^{\infty} \frac{1}{(n+1)!}(hJ)^n\,J\,y_0 = e^{Jh}y_0 - y_0$$

$$\Rightarrow h\,\varphi(hJ)\,g(y_0) = h\,\varphi(hJ)\,f(y_0) - e^{Jh}y_0 + y_0$$

Somit:

$$y(h) \approx e^{Jh}y_0 + h\,\varphi(hJ)f(y_0) - e^{Jh}y_0 + y_0 \Rightarrow$$

$$y(h) \approx y_0 + h\,\varphi(hJ)f(y_0) \qquad O(h^2)$$

exponentielles Eulerverfahren.

Hier ist $J = Df(y_0)$

$$\varphi(hJ) = \left(e^{hJ} - I\right)(hJ)^{-1} \qquad O(d^3)$$

teuer für grosses d.

Bem Für grosses d kann man das Krylov-Verfahren verwenden.

$$\varphi(A)b = V_m\,\varphi(H_m)\begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix}$$ aus Krylov-Verfahren $\mathcal{K}_m(A,b)$

Bem Stabilitätsfunktion $S(z) = e^z$ $\Rightarrow$
das ideale Stabilitätsgebiet

da exakt für $\dot{y} = Ay + g$
$\hookrightarrow$ konstant.

Verallgemeinerung: exponentiele RK-Verfahren

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

semi-implizite Euler: $\underline{y}_1 = \underline{y}_0 + \left(\underline{\underline{I}} - h\underline{\underline{J}}\right)^{-1} h\, \underline{f}(\underline{y}_0)$

exponentiele Euler: $\underline{y}_1 = \underline{y}_0 + \varphi(h\underline{\underline{J}})\, h\, \underline{f}(\underline{y}_0)$

Idee: ersetze $\frac{1}{1-z}$ durch $\varphi(z) = \frac{e^z - 1}{z}$

$$\Rightarrow \begin{cases} \underline{k}_i = \varphi(\gamma h \underline{\underline{J}})\left(\underline{f}(\underline{u}_i) + h\underline{\underline{J}} \sum_{j=1}^{i-1} \gamma_{ij}\, \underline{k}_j\right) \\ \underline{u}_i = \underline{y}_0 + h \sum_{j=1}^{i-1} \alpha_{ij}\, \underline{k}_j \\ \underline{y}_1 = \underline{y}_0 + h \sum_{i=1}^{s} b_i\, \underline{u}_i \end{cases}$$

explizite RK: $\varphi(z) = 1$, $\underline{\underline{J}} = \underline{\underline{0}}$

RoW $\varphi(z) = \frac{1}{1-z}$

exp. RK: $\varphi(z) = \frac{e^z - 1}{z}$.

ODE	nicht steif	steif	oszillierend
Methode	expl. RK; oder	impl. RK;	exponentiele RK
Stabilität	$h < \frac{1}{\lambda}$	alle h	alle h
Implementierung	$\underline{f}(\underline{y})$	$\underline{\underline{D}}\underline{f}(\underline{y}_0)$ Lösen nicht-lin. Gleichungssystem	$\underline{\underline{D}}\underline{f}(\underline{y}_0)$ $\varphi(\underline{\underline{A}})\,\underline{b}$ Schnell mit Krylov.
			exp4

⊕ Erhaltungseigenschaften wichtig?

⊕ Autonom?

⊕ Ordnung der Dgl.?

Die Methode des Konjugierten Gradienten

$\underline{\underline{A}} \in \mathbb{R}^{n \times n}$ symmetrisch, pos. definit.

Ziel $\underline{\underline{A}}\,\underline{x} = \underline{b}$: x geschickt approximiert.

Definiere "Energie" $Q(\underline{x}) = \frac{1}{2}\underline{x}^T \underline{\underline{A}}\,\underline{x} - \underline{x}^T \underline{b}$

$\left(\underline{\underline{A}}\,\underline{x} = \underline{b} \Leftrightarrow \underline{\underline{A}}\underline{x} - \underline{b} = \underline{0} \Leftrightarrow \underline{x}^T \underline{\underline{A}}\,\underline{x} - \underline{x}^T \underline{b} = 0. \right)$

Statt Nährung an Lösung von $\underline{\underline{A}}\,\underline{x} = \underline{b}$

suche Nährung an Lösung von $\operatorname{argmin}_{\underline{x} \in \mathbb{R}^n} Q(\underline{x})$

1) Wähle Suchrichtung $\underline{p}_k \neq \underline{0}$

2) optimiere in dieser Richtung.: $Q(\underline{x}_k + \alpha_k \underline{p}_k) = \min!$

$$\alpha_k = \frac{\underline{p}_k^T \underline{r}_k}{\underline{p}_k^T \underline{\underline{A}}\,\underline{p}_k} \quad ; \quad \underline{r}_k = \underline{b} - \underline{\underline{A}}\,\underline{x}_k \text{ Residuum.}$$

Wie wählt man die Suchrichtung?

Analysis: Gradienten-Richtung.

Bessere Idee: Variablen wechseln

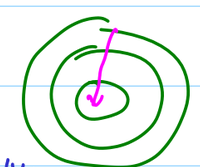
$\underline{x} = \underline{\underline{P}}\,\underline{y} \Rightarrow Q(\underline{x}) = \frac{1}{2}\underline{y}^T \underline{\underline{D}}\,\underline{y} - (\underline{\underline{P}}^T \underline{b})^T \underline{y}$

So das $\underline{\underline{P}}^T \underline{\underline{A}}\,\underline{\underline{P}} = \underline{\underline{D}} = \text{diagonal}$

↓

das ist nicht eine $\in W$-Zerlegung!

da $\underline{\underline{P}}$ nicht orthogonal!



Stiefel: wie man effizient die Richtungen baut:

$\underline{x}_0$ gegeben, $\underline{p}_0 = \underline{r}_0 = \underline{b} - \underline{\underline{A}}\,\underline{x}_0$

für $k = 0, 1, \ldots, n-1$

$$\alpha_k = \frac{\underline{r}_k^T \underline{p}_k}{\underline{p}_k^T \underline{\underline{A}}\, \underline{p}_k}$$

$$\underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{p}_k$$

$$\underline{r}_{k+1} = \underline{r}_k - \alpha_k \underline{\underline{A}}\, \underline{p}_k$$

$$\beta_k = - \frac{\underline{p}_k^T \underline{\underline{A}}\, \underline{r}_{k+1}}{\underline{p}_k^T \underline{\underline{A}}\, \underline{p}_k}$$

$$\underline{p}_{k+1} = \underline{r}_{k+1} + \beta_k \underline{p}_k$$

effizienter:

x_0 gegeben, $\underline{p}_0 = \underline{r}_0 = \underline{b} - \underline{\underline{A}}\,\underline{x}_0$

$or = \|r_0\|^2$

für $k = 0, 1, 2, \ldots,$

$\underline{w}_k = \underline{\underline{A}}\, \underline{p}_k$

$$\alpha_k = \frac{or}{\underline{p}_k^T \underline{w}_k}$$

$$\underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{p}_k$$

$$\underline{r}_{k+1} = \underline{r}_k - \alpha_k \underline{w}_k$$

$r_0 = \|r_{k+1}\|^2$

$$\beta_k = \frac{r_0}{or} \qquad \left(\text{also } \frac{\underline{r}_{k+1}^T \underline{r}_k}{\underline{r}_k^T \underline{r}_k} \right)$$

$$\underline{p}_{k+1} = \underline{r}_{k+1} + \beta_k \underline{p}_k$$

CG = conjugate gradient

Theorem 1) $r_l \in \mathcal{K}_{l+1}(\underline{\underline{A}}, p_0) = \mathcal{K}_{l+1}(\underline{\underline{A}}, r_0)$

2) CG optimal: $\|x - x_{l+1}\| = \min_{\tilde{x} \in \mathcal{J}_k} \|x - \tilde{x}\|_{\underline{\underline{A}}}$

mit $\mathcal{J}_k = x_0 + \mathcal{K}_{l+1}(\underline{\underline{A}}, r_0)$

$\|x\|_A^2 = x^T \underline{\underline{A}} x$

3) Falls $\underline{\underline{A}}$ nur $m \le n$ verschiedene EW hat, dann konvergiert CG in maximal m Schritte

4) $e_k = x - x_k \Rightarrow \|e_k\|_{\underline{\underline{A}}} \le 2 \left(\frac{\sqrt{\mathrm{cond}(\underline{\underline{A}})} - 1}{\sqrt{\mathrm{cond}(\underline{\underline{A}})} + 1} \right)^k \|e_0\|_{\underline{\underline{A}}}$

Begründung der Wahl der Richtung.

Idee: Suche x_k die optimale Approximation in $x_0 + \mathcal{K}_{l+1}(\underline{\underline{A}}, r_0)$

$r_k = b - \underline{\underline{A}} x_k \perp \mathcal{K}_k(\underline{\underline{A}}, r_0)$

$0 = V_k^T r_k = V_k^T (b - \underline{\underline{A}} x_k) = V_k^T (b - \underline{\underline{A}} V_k y)$ (mit $-Ax_0$)

$x_k = x_0 + V_k y$

$= V_k^T (b - Ax_0) - V_k^T A V_k y \Rightarrow$

($b - Ax_0 = r_0$, $V_k^T A V_k = H_k$)

$H_k y = V_k^T r_0 = \|r_0\| e_1$ $\qquad H_k = \begin{bmatrix} \times & \cdots & \times \\ & \ddots & \vdots \\ 0 & & \times \end{bmatrix}$

Lanczos macht die Iteration kurz!!

Bem Bei schlecht konditionierte Matrizen ist CG noch zu langsam!

* CG in Prinzip $O(n)$ aber wegen Rundungsfehler $O(n \log n)$
* beschleunige das Verfahren indem man die Matrix verändert : vorkonditioniert!

$$\underline{\underline{A}}\,\underline{x} = \underline{b} \qquad \hat{\underline{\underline{A}}}\,\hat{\underline{x}} = \hat{\underline{b}}$$

$$\hat{\underline{\underline{A}}} = \underline{\underline{S}}\,\underline{\underline{A}}\,\underline{\underline{S}}^T \qquad \hat{\underline{x}} := \underline{\underline{S}}^{-T}\underline{x} \qquad \hat{\underline{b}} = \underline{\underline{S}}\,\underline{b}$$

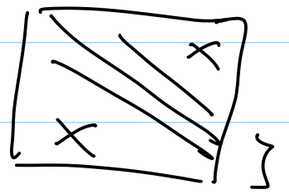
nehme $\underline{\underline{S}}$ so dass 1) $\mathrm{cond}(\hat{\underline{\underline{A}}}) < \mathrm{cond}(\underline{\underline{A}})$

2) $\underline{\underline{S}}\,\hat{\underline{x}} = \underline{c}$ soll einfach sein.

v.CG : CG mit $\underline{\underline{S}}^{-1}\underline{\underline{A}}$ statt $\underline{\underline{A}}$!!

typischerweise: $\mathrm{diag}(\underline{\underline{A}})$

oder



Bem Noch bessere Methoden:

incomplete LU decomposition,

* algebraisch ↔ sparse approx. inverse
  ↔ alg. multigrid. $O(n)$

* geometrisch : das Physik. des Problems verwendet.
  → domain decomposition method.
  → geometric multigrid. $O(n)$