FS19

§1 Quadratur

§1.1. Motivation

Bsp $\begin{cases} \dot{y} = f(t,y) & t \in \mathbb{R} \quad y(t) \in \mathbb{R} \\ y(t_0) = y_0 \end{cases}$

Gesucht: $y : [t_0, T] \to \mathbb{R}$

Bescheidener: $y(T)$ ist gesucht

$$\int_{t_0}^{T} dt \Rightarrow y(T) = y(t_0) + \int_{t_0}^{T} f(t, y(t))\, dt$$

Bsp Was ist die Periode eines Pendulums?

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

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

Bsp Plank' Gesetz für die Strahlung eines schwarzen Körpers

T = Temperatur in K, λ = Wellenlänge in μm

c_1, c_2 = phys. Konst.

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

↳ Energieverlust pro Flächen einheit.

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

↙

σ = Stefan-Boltzmann-konstante.

Ziel: berechne σ

$$\sigma T^4 = \int_0^{\infty} E(\lambda, T)\, d\lambda \Rightarrow \sigma = \frac{1}{T^4} \int_0^{\infty} E(\lambda, T)\, d\lambda$$

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

w_j: Gewichte, x_j: Knoten.

Ziel: Wähle Knoten, Gewichte, Strategie für diese Berechnung um den Fehler $|J-Q|$ aber auch die Kosten klein zu halten.

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

z.B.

$f \approx$ Polynom $\quad \alpha_0 + \alpha_1 x + \dots + \alpha_n x^n$

$f \approx$ trigonometrischen Polynom.

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

Gegeben Knoten $x_0, x_1, \dots, x_n$

kann man $\alpha_0, \alpha_1, \dots, \alpha_n$ berechnen, sodass

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

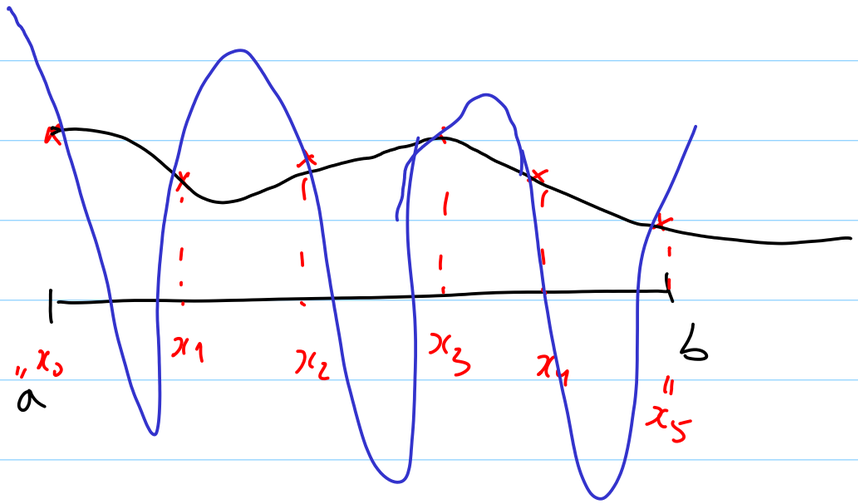
$p_n(x_j) = f(x_j)$ für $j = 0, 1, \dots, n$

Im Prinzip ist das das LGS:

$$\begin{bmatrix} 1 & x_0 & \dots & x_0^n \\ 1 & x_1 & & x_1^n \\ 1 & x_2 & & \vdots \\ \vdots & \vdots & & \vdots \\ 1 & x_n & & x_n^n \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} f(x_0) \\ \vdots \\ f(x_n) \end{bmatrix}$$

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

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



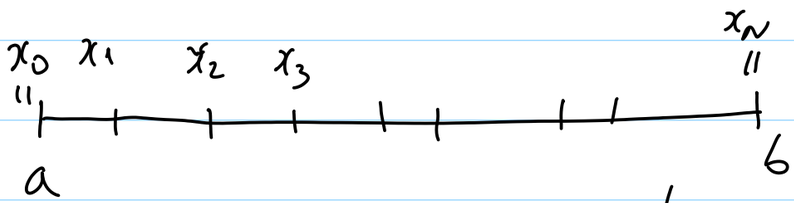
Man kann beweisen:

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

Länge des Intervalls ist wichtig.

Glattheit 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 jedem kleine Intervall der Länge $h \Rightarrow$

zusammengesetzte Quadraturformel

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

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

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

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

$$\leq \max_{z\in[a,b]} \left|f^{(n)}(z)\right| = C \qquad h = \frac{b-a}{N} \Rightarrow N = \frac{b-a}{h}$$

$$\Rightarrow \quad |J-Q| \leq C \cdot \frac{h^{n+1}}{n!} \cdot \frac{b-a}{h} = C \frac{h^n}{n!}(b-a)$$

Def Quadraturformel hat Ordnung $n+1$

wenn sie Polynome von Grad maximal n exakt integriert.

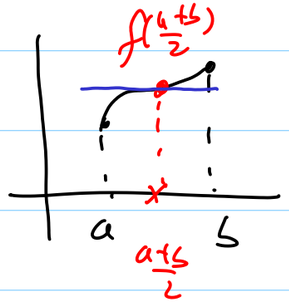
(das erste falsche Ergebnis: x^{n+1})

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

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.



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

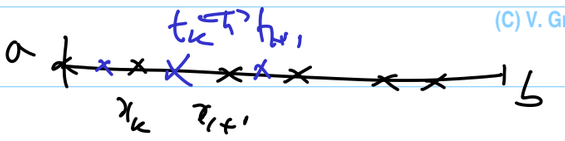
Bem 1) Polynome vom Grad 0 und 1 werden exakt integriert

$\Rightarrow$ MPR hat Quadraturordnung 2

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

Bem 2) offene Quadraturformel: Enden $[a,b]$ sind keine Knoten.

MPR:



$$\int_a^b f(x)\,dx \approx \sum_{k=0}^{N-1} (x_{k+1} - x_k)\, f\Big(\underbrace{\frac{x_k + x_{k+1}}{2}}_{t_k}\Big)$$

$t_k = \frac{x_k + x_{k+1}}{2} = x_k + \frac{h}{2}$ mit $h = x_{k+1} - x_k = \frac{b-a}{N}$

$$\int_a^b f(x)\,dx \approx \frac{b-a}{N} \sum_{k=0}^{N-1} f(t_k)$$

Implementierung.

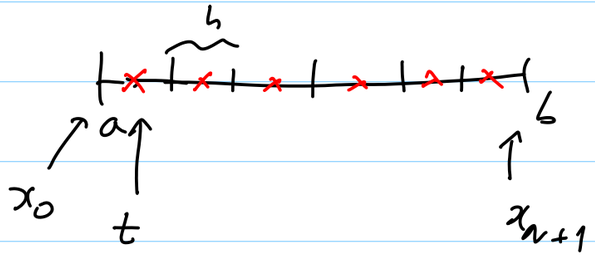
gegeben a, b;
wähle N; $S = 0$
für $k = 0, 1, 2, \ldots, N-1$: $S = S + f(t_k)$; $t_k = t_k + h$
$S = S \cdot h$
return S

$h = \frac{b-a}{N}$; $t_k = a + \frac{h}{2}$

Gegeben: f, a, b

wähle: N

$h = (b-a)/N$



$t = a + h/2$

$S = 0$

für $k = 0, 1, 2, \ldots, N-1$:

$S = S + f(t)$

$t = t + h$

$\sum_{k=0}^{N-1} f(t_k)$

$S = h \cdot S$

$x[:] = [x[0], x[1], \ldots, x[N]]$

return. S

$x[:-1] = [x[0], x[1], \ldots x[N-1]]$

Python: * numpy → array

* scipy

* matplotlib. /...

$x[x:-2] = [\ldots \quad x[N-2]]$

$\vdots$

$x[2:-2) =$

$= [x[2], \ldots, x[N-2]]$

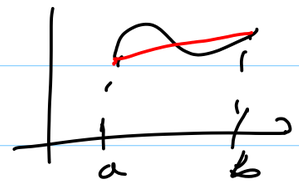
$t = \text{linspace}(a + \frac{h}{2}, b, N-1)$

oder:

$x = \text{linspace}(a, b, N+1)$

$t = \frac{h}{2} + x[:-1]$; $ft = f(t)$; $\text{linsum}(ft)$

Bsp 2) Trapezregel



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

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

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

2) geschlossene Quadraturformel

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

$$\text{TR:} \quad \int_a^b f(x)\,dx = \sum_{k=1}^{N} \int_{t_{k-1}}^{t_k} f(x)\,dx \approx$$

$$= \sum_{k=1}^{N} \left(\frac{h}{2} f(x_{k-1}) + \frac{h}{2} f(x_k) \right) =$$

$$= \frac{h}{2} f(x_0) + \underbrace{\frac{h}{2} f(x_1) + \frac{h}{2} f(x_1)}_{h f(x_1)} + \underbrace{\frac{h}{2} f(x_2)}_{h f(x_2)} + \dots + \underbrace{\frac{h}{2} f(x_{N-1})}_{h f(x_{N-1})} + \frac{h}{2} f(x_N)$$

$$= \frac{h}{2} f(x_0) + h \sum_{k=1}^{N-1} f(x_k) + \frac{h}{2} f(x_N)$$

```
x = linspace(a, b, N+1)
h*sum(f(x[1:-1])) + h/2*(f(x[0]) + f(x[N]))
```

Bsp 3) Simpson Regel

(Polynom von Grad ≤ 3 wird exakt integriert)

$\Rightarrow$ Ordnung 4

$$Q^S(f,a,b) = \underbrace{\frac{b-a}{6}}_{w_1} \underbrace{f(a)}_{x_1} + \underbrace{\frac{b-a}{6}\cdot 4}_{w_2} \underbrace{f\left(\frac{a+b}{2}\right)}_{x_2} + \underbrace{\frac{b-a}{6}}_{w_3} \underbrace{f(b)}_{x_3}$$

Fehler: $\frac{1}{90}\left(\frac{b-a}{2}\right)^5 f^{(4)}(t)$ mit $t \in [a,b]$

Aufgabe: MPR, TR, Simpson. implementieren.

Anwendg.:

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

Stefan-Boltzmann Konstante

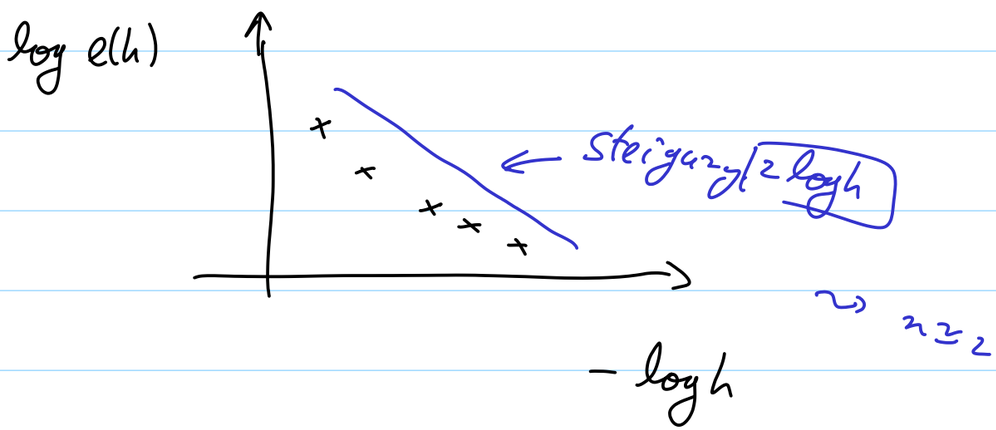
(Planck' Gesetz)

```
from numpy import linspace, array, sum, ...
from scipy import integrate
```

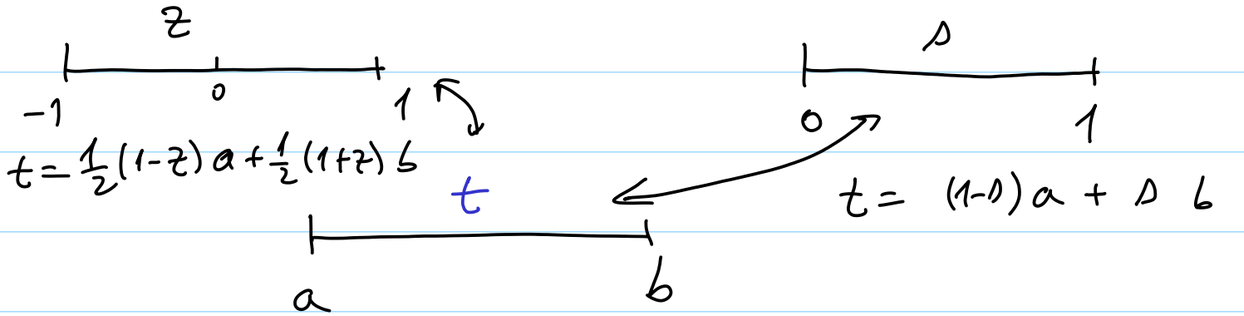
`integrate.quad(f, a, b)`

Glattheit $\Rightarrow$ Fehler $c \cdot h^n = e(h)$

$\log e(h) = n \log h + c$



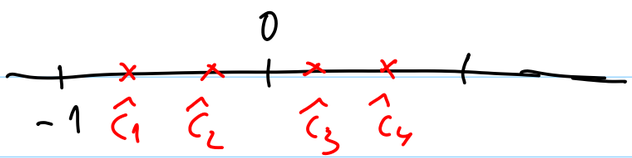
§1.2. Referenzintervale und Symmetrische Quadraturformel



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

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

Definition QF auf $[-1,1]$ heisst **symmetrisch**, falls $\hat{c}_k = -\hat{c}_{n+1-k}$ / $\hat{\omega}_k = \hat{\omega}_{n+1-k}$



Theorem Die Quadraturordnung einer symmetrischen QF ist gerade.

Beweis

Annahme: QF exakt für Polynome vom Grad $2n-2$.

Nehme $f(x) = a x^{2n-1}$

$$\int_{-1}^{1} f(x)\,dx = a\int_{-1}^{1} x^{2n-1}\,dx = 0$$

Die QF ist exakt für ax^{2n-1} falls $Q(f,-1,1) = 0$

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

$$= -a\sum_{k=1}^{n} \hat{\omega}_{n+1-k}\,\hat{c}_{n+1-k}^{2n-1} =$$

$$= -a\sum_{j=n}^{1} \hat{\omega}_j \hat{c}_j^{2n-1} = -a\sum_{j=1}^{n} \hat{\omega}_j \hat{c}_j^{2n-1} = -Q(f,-1,1)$$

$\Rightarrow Q(f,-1,1) = 0 = \int_{-1}^{1} f(x)\,dx$ qed.

§1.3. Fehler für Quadratur auf $[0,1]$

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

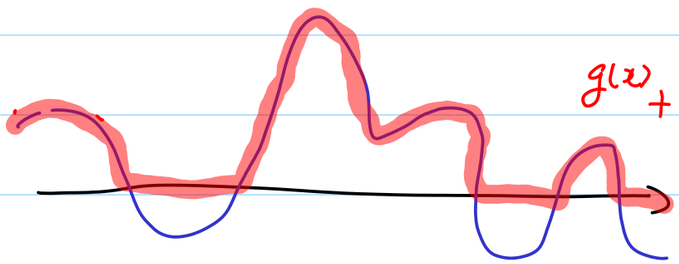
c_j ↳ Knoten in $[0,1]$

b_j ↳ Gewichte

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

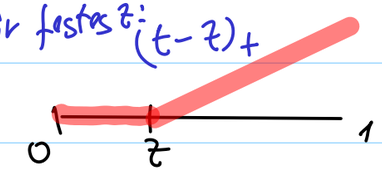
$A, B \in \mathbb{R}$ und g, f Funktionen

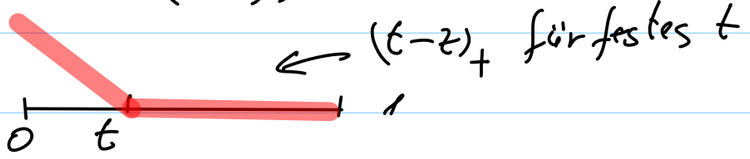
das positive Teil $g(x)_+ = \begin{cases} g(x), & \text{falls } g(x) > 0 \\ 0, & \text{sonst} \end{cases}$



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

für festes z: $(t-z)_+$





als Funktion von der Variable t

$$K_n(z) = E\big(\alpha(z,\cdot)\big) =$$

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

$$= \frac{(1-z)^n}{n!} - \sum_{j=1}^{n} b_j \frac{(c_j-z)_+^{n-1}}{(n-1)!}$$

Theorem

Sei Q eine QF mit Quadraturordnung n und sei g n-mal stetig differentierbar.

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

Beweis Taylor am Punkt 0:

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

$g(t)$ Polynom vom Grad $n-1$

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

Linearität von E:

$$E(g) = \underbrace{E(q)}_{0} + \int_0^1 \underbrace{E(\alpha(z,\cdot))}_{K_n(z)} g^{(n)}(z)\,dz$$

Bemerkung Wenden wir den Satz auf $g(t) = f(x_0 + th)$:

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

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

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

$$= h\,E(g) = h \cdot h^n \int_0^1 K_n(z) f^{(n)}(x_0 + hz)\,dz$$

da:

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

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

$g''(t) = f''(x_0+ht)\,h^2$

$\dots$

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

Fehler auf $[x_0, x_0+h]$ ist

$$h^{n+1} \int_0^1 k_n(t)\, f^{(n)}(x_0+ht)\,dt$$

Zusammengesetzt:

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

$$\le c_0 \cdot h^{2} \max |f^{(n)}(x)| \longrightarrow 0 \quad \text{für } h \to 0$$

$x \in [a,b]$ → Glattheit

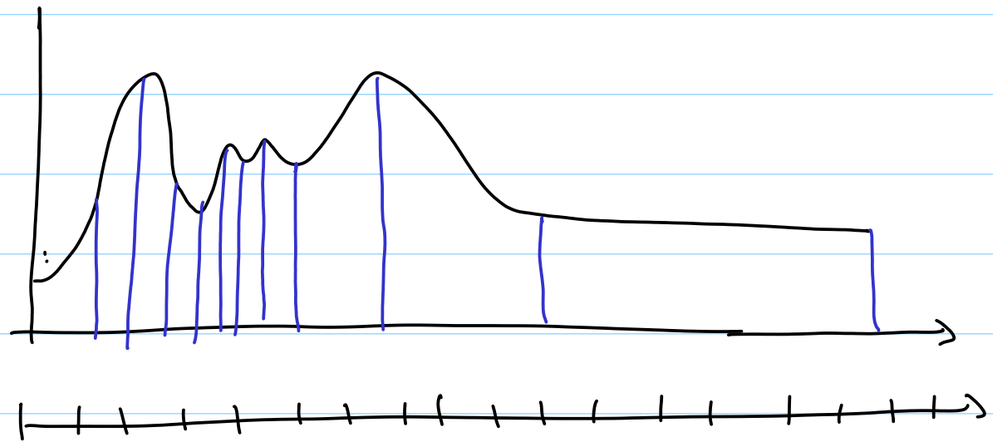
Ordnung der lokalen QF

kommt aus dem Peano kern

$$\int_0^1 |k_n(t)|\,dt = \text{konstante}$$

MPR, TR = $\frac{1}{12}$ Simpson $\frac{1}{2880}$.

§1.4. Adaptive Quadratur



Optimiere: Anzahl Funktionsauswertungen

Lokalen Interval $[x_{k-1}, x_k]$ hat lokalen Fehler ε_k

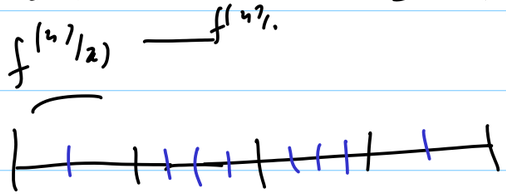
$$\varepsilon_k = \left| \int_{x_{k-1}}^{x_k} f(x)\,dx - \sum_{j=1}^{r} b_j f(x_{k-1} + c_j h_k) \right)$$

$h_k = x_k - x_{k-1}$ lokale Intervalllänge

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

IDEE: Wähle nur dort ein kleines h, wo $|f^{(n)}(x)|$ gross

Wie?



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

(möchte möglichst gleiche lokale Fehler haben)?

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

$$\approx \left| Q^S(f, x_{k-1}, x_k) - Q^T(f, x_{k-1}, x_k) \right| = \tilde{\varepsilon}_k$$

Schätzung des lokalen Fehlers.

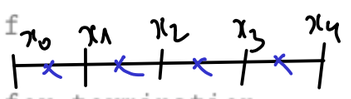
Verfeinere die Intervalle, wo $\tilde{\varepsilon}_k$ gross ist.

Verwendet wird allerdings das Wert, das Q^S gibt

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

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

```python
def adaptquad(f,M,rtol,abstol):
    """
    adaptive quadrature using trapezoid and simpson rules
    Arguments:
    f       handle to function f
    M       initial mesh
    rtol    relative tolerance for termination
    abstol  absolute tolerance for termination, necessary in case the exact
integral value = 0, which renders a relative tolerance meaningless.
    """
    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
    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
```



$M = \text{array}([x_0, x_1, x_2, x_3, x_4])$

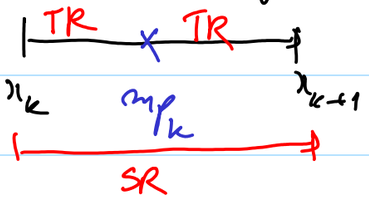
$h_k = x_{k+1} - x_k \Rightarrow h = [h_0, h_1, h_2, h_3]$ array

$\frac{1}{2}(x_k + x_{k+1}) \Rightarrow [mp_0, mp_1, mp_2, mp_3]$ array.

fx = array der Länge 5

fm = array de Länge 4

auf $[x_k, x_{k+1}]$



Mittleren Fehler

$Loc > \frac{9}{10} \cdot \frac{tot}{\#\,Intervale}$

array.

Zahl $\Rightarrow$ [False, True, True, False]

[0, 1, 1, 0]

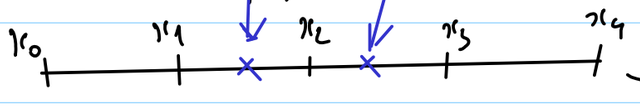
nonzero → welche Indicies sind $\neq 0 \Rightarrow 1, 2$

→ Lese help für nonzero um [0] zu verstehen!

Sort([$x_0, x_1, x_2, x_3, x_4, mp_1, mp_2$])

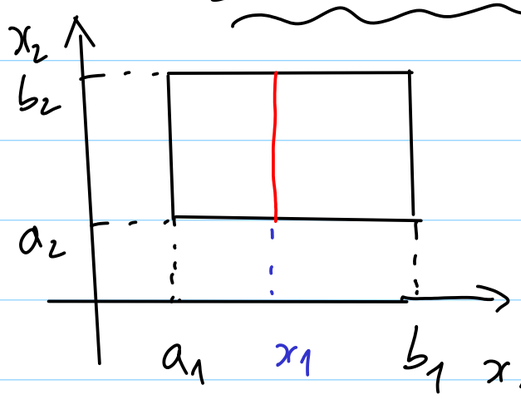
↳ [$x_0, x_1, mp_1, x_2, mp_2, x_3, x_4$]

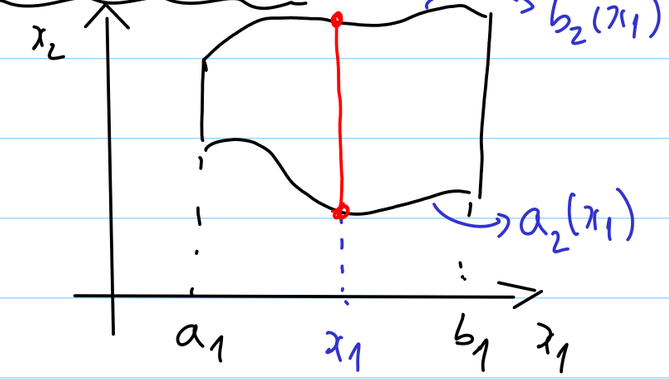
neues Gitter



→ Rekursion!

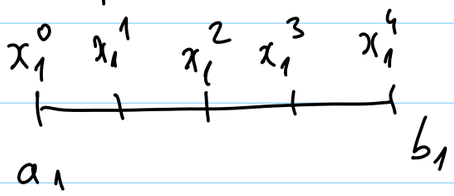
§1.5. Quadratur in $\mathbb{R}^d$ $d=2,3$





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

$$=\int_{a_1}^{b_1} F(x_1)\,dx_1=\sum_{k_1=1}^{N_1}\int_{x_1^{k_1-1}}^{x_1^{k_1}} F(x_1)\,dx_1 \approx$$



$$=\frac{b_1-a_1}{N_1}\sum_{k_1=1}^{N_1}\sum_{j_1=1}^{S_1} F\left(x_1^{k_1-1}+h_1 c_{j_1}^1\right)\cdot w_{j_1}^1$$

$x_1^{k_1-1}+h_1 c_{j_1}^1$ ↑ knoten in Ox_1-Richtung auf dem Unterinterval $\left[x_1^{k_1-1};x_1^{k_1}\right]$

$$=\frac{b_1-a_1}{N_1}\sum_{k_1=1}^{N_1}\sum_{j_1=1}^{S_1}\int_{a_2\left(x_1^{k_1-1}+h_1 c_{j_1}^1\right)}^{b_2\left(x_1^{k_1-1}+h_1 c_{j_1}^1\right)} f\left(x_2, x_1^{k_1-1}+h_1 c_{j_1}^1\right)\cdot w_{j_1}^1\,dx_2 \qquad k_1=1,2,\dots,N_1$$

$\int_{a_2}^{b_2} f\,dx_2 \approx$ QF

$$=\frac{b_1-a_1}{N_1}\sum_{k_1=1}^{N_1}\sum_{j_1=1}^{S_1}\frac{b_2-a_2}{N_2}\sum_{k_2=1}^{N_2}\sum_{j_2=1}^{S_2} f\left(x_2^{k_2-1}+h_2 c_{j_2}^2, x_1^{k_1-1}+h_1 c_{j_1}^1\right)\cdot w_{j_2}^2\, w_{j_1}^1$$

$$(b_1-a_1)(b_2-a_2)\,\frac{1}{N_1 N_2}\sum_{k_1=1}^{N_1}\sum_{k_2=1}^{N_2}\sum_{j_1=1}^{D_1}\sum_{j_2=1}^{D_2} f\left(x_2^{k_2-1}+h_2 c_{j_2}^{2},\, x_1^{k_1-1}+h_1 c_{j_1}^{1}\right) w_{j_2}^{2}\, w_{j_1}^{1}$$

im d-Dimensionen $N_1 \cdot N_2 \cdots N_d$ Auswertungen von f.

$$h_1 h_2 \cdots h_d \sum_{k_1}\sum_{k_2}\cdots\sum_{k_d} f(\ldots)\, w_{j_d}^{d}\, w_{j_{d-1}}^{d-1} \cdots w_{j_1}^{1}$$

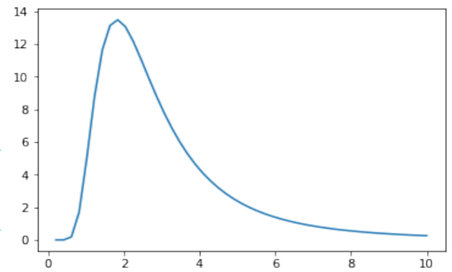
Für grosse d:

* f glatt $\Rightarrow$ dünne Gitter / sparse grids
* Monte Carlo; quasi-Monte Carlo
* hochoszillatorische Funktionen.

```
In [1]: from numpy import linspace, exp, inf
        from pylab import plot, show

In [2]: # Plank's law for blackbody radiation
        c1 = 3.7413*10**8 #W \mu m^4/m^2
        c2 = 1.4388*10**4 # \mu m K
        T = 1000.# T = temperature in K
        # lam = wavelength in um
        # monochromatic emissive power of a blackbody:
        def Eblam(lam, T=1000.):
            return c1/(lam**5*( exp(c2/(lam*T)) -1.))

In [3]: print('------------ plot ---------------')
        y = linspace(0,10)
        plot(y, Eblam(y,1600.)/10**4)
        show()
```




```
In [4]: print('------------ integrate: quad ---------------')
        from scipy import integrate
        # total energy lost per unit area
        Q, abserr, info = integrate.quad(Eblam,0,inf, full_output=True)
        neval = info['neval']
        # Q = sigma * T^4; sigma = Stefan-Boltzmann constant
        print(Q/T**4, abserr, 'neval=', neval)

------------ integrate: quad ---------------
5.669294910232968e-08 9.879197389039973e-07 neval= 105
```

```
In [8]: def ga(x):
            return exp(-x**2)

        val, err, info = integrate.quad(ga,-1,1, full_output=True)
        print(val)
        print(err)
        print(info['neval'])

1.493648265624854
1.6582826951881447e-14
21
```

```
In [9]: def gaabc(x,a,b,c):
            return a*exp(-((x-b)/c)**2)

        val, err, info = integrate.quad(gaabc,-1,1,args=(2,0.5,4), full_output=True)
        print(val)
        print(err)
        print(info['neval'])

        val, err, info = integrate.quad(gaabc,-inf,inf, args=(2,0.5,4), full_output=True)
        print(val)
        print(err)
        print(info['neval'])

3.8599302795534305
4.2853834698080357e-14
21
14.17963080724413
2.5011577807838655e-08
330
```

```
from scipy.special import jv
f0 = lambda x: jv(0,x)
f1 = lambda x: jv(1,x)
val, err, info = integrate.quad(f0,0,5, full_output=True)
print(val)
print(err)
print(info['neval'])
val, err, info = integrate.quad(f1,0,5, full_output=True)
print(val)
print(err)
print(info['neval'])
```

```
0.7153119177847678
2.47260738289741e-14
21
1.177596771314338
1.8083362065765924e-14
21
```

In [11]:
```
from numpy import sqrt
f = lambda x:1/sqrt(abs(x))
integrate.quad(f,-1,1)
```

Out[11]: (inf, inf)

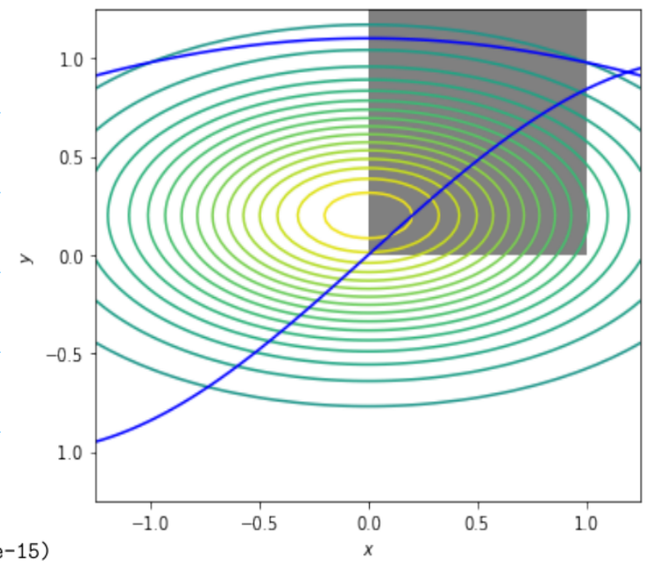
In [12]:
```
integrate.quad(f,-1,1, points=[0]) # can deal with singularity if you tell it
```

Out[12]: (3.9999999999999813, 5.684341886080802e-14)

In [13]:
```
from pylab import subplots
import matplotlib.patches as patches
from numpy import *
def f(y,x):
    'y must be the first argument, and x the second.'
    return exp(-x**2 - 3*(y-0.2)**2)

fig, ax = subplots(figsize=(6,5))
x = y = linspace(-1.25,1.25,75)
X, Y = meshgrid(x,y)
c = ax.contour(X,Y,f(Y,X), 16, vmin=-1, vmax=1)
bound = patches.Rectangle((0,0),1,1.25, facecolor='grey')

ax.add_patch(bound)
ax.set_xlabel('$x$')
ax.set_ylabel('$y$')
ay = lambda x: sin(x)
by = lambda x: 0.1+ cos(0.5*x)
plot(x,ay(x),'-b')
plot(x,by(x),'-b')
show()
```



In [14]:
```
integrate.dblquad(f,0,1,ay,by)
```

Out[14]: (0.24918585141988464, 7.637755883693233e-15)

§1.6. Quadratur mit erhöhter Ordnung

s Knoten auf Referenzinterval, s Gewichte — so bestimmt, dass Polynome höchstes Grades exakt mit dieser QF integriert werden.

Wie hoch kann so eine Quadraturordnung sein?

$2s$ Unbekannte (Knoten, Gewichte);

$2s$ Gleichungen:

$$\begin{cases} \int_0^1 1\,dt = Q_s(1,0,1) \\ \int_0^1 t\,dt = Q_s(t,0,1) \\ \dots \quad \left(\frac{1}{3} = b_1 c_1^2 + b_2 c_2^2 + \dots\right) \\ \int_0^1 t^{2s-1}\,dt = Q_s(t^{2s-1},0,1) \end{cases}$$

Da QF exakt

Möchte Ordnung $p = s+m$

Jedes Polynom vom Grad $\le p-1 = s+m-1$ soll mit QF exakt integriert werden.

Sei f Polynom vom Grad $\le p-1 = s+m-1$

IDEE: teile das Polynom f durch

$$M(x) = (x-c_1)(x-c_2)\dots(x-c_s)$$

$$\text{Grad}(M) = s$$

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

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

$\Vert$ ← da QF exakt

$\to \Vert$

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

$M(c_j) = 0$

$= 0$

$\Rightarrow \int_0^1 M(t)\, g(t)\, dt = 0$ für alle Polynome g mit $\mathrm{Grad}(g) \le m-1$.

f = Polynom vom Grad $\le s+m-1$

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

Theorem

Ordnung der QF ist $s+m$ $\Longleftrightarrow$

$\langle M, g \rangle = 0$ für alle Polynome vom Grad $\le m-1$.

$P_m = \mathrm{span}\{1, t, t^2, \dots, t^{m-1}\}$: $M \perp P_m$

Theorem Ordnung einer QF ist höchstens $2s$.

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

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

Nehme $g = M$

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

$M(t) \equiv 0$ ↯

Widerspruch $\Rightarrow$ wahr: $p \le 2s$.

Orthogonale Polynome

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

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

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

Betachte den linearen Raum:

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

Bem Alle Polynome liegen in V

V: Skalarproduct

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

Theorem [Gram-Schmidt in LA]

Es existiert eine eindeutige Folge von Polynomen $p_0, p_1, \dots$ mit

$$p_k(x) = x^k + \text{Polynom vom Grad} \le k-1$$

so dass

$$p_k \perp \text{span}\{p_0, p_1, \dots, p_{k-1}\}.$$

Diese Polynome baut man so: (3-Term Rekursion)

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

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: oft werden sie aber zu $p_k(1) = 1$ "normiert". Darum. $p_2(x) = \frac{3}{2}(x^2 - \frac{1}{3})$

Bem $c_1, c_2, \dots, c_n$ die Nullstellen von (M) von p_n.

$M = p_n$

Bsp 1) $\omega(x) \equiv 1$, $a = 0, b = 1$: $\int_0^1 f(x)\,g(x)\,dx$

Gram-Schmidt $\Rightarrow$ orthogonale Polynome

Legendre Polynome

QF: Gauss-Quadratur

2)

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

QF $\Rightarrow$ Hermite-Quadratur.

Bsp Gauss-Quadratur

$[0,1]$ Notiere Knoten γ_j, $j = 1\dots,n$

$[-1,1]$ Notiere Knoten c_j, $j = 1\dots,n$.

Beweis des Theorems [Gram-Schmidt-Orthogonalisierung]

Mathematische Induktion:

Annahme: $p_0, p_1, \dots, p_k$ bereits bekannt

Ziel: Konstruiere: p_{k+1}

Nehme $p_{k+1} = x\,p_k + \sum_{j=0}^{k} \alpha_j p_j$

und berechne α_j, so dass $p_{k+1} \perp p_j$ für $j = 0, \dots, k$.

$p_{k+1} \perp p_k \iff 0 = \langle p_{k+1}, p_k \rangle =$

$= \langle x p_k, p_k \rangle + \sum_{j=0}^{k-1} \alpha_j \langle p_j, p_k \rangle + \alpha_k \langle p_k, p_k \rangle$

$\langle p_j, p_k \rangle = 0$ ← Induktionsannahme

$\iff \alpha_k = - \dfrac{\langle x p_k, p_k \rangle}{\langle p_k, p_k \rangle} = -\beta_{k+1}$

$p_{k+1} \perp p_{k-1} \iff 0 = \langle x p_k, p_{k-1} \rangle + \alpha_{k-1} \langle p_{k-1}, p_{k-1} \rangle + 0$

aber $\langle x p_k, p_{k-1} \rangle = \int_{-1}^{1} x\, p_k(x)\, p_{k-1}(x)\, dx = \langle p_k, x p_{k-1} \rangle$

Schreibe $x p_{k-1} = p_k + q$ mit $\operatorname{Grad} q \le k-1 \Rightarrow$

$\Rightarrow \langle x p_k, p_{k-1} \rangle = \langle p_k, p_k \rangle + \langle p_k, q \rangle$, wobei $\langle p_k, q \rangle = 0$

Somit $p_{k+1} \perp p_{k-1} \iff \alpha_{k-1} = - \dfrac{\langle p_k, p_k \rangle}{\langle p_{k-1}, p_{k-1} \rangle} = -\gamma_{k+1}^2$

Für $j \le k-2$:

$p_{k+1} \perp p_j \iff 0 = \langle p_{k+1}, p_j \rangle = \langle x p_k, p_j \rangle + \alpha_j \langle p_j, p_j \rangle =$

$= \langle p_k, \underbrace{x p_j}_{\text{Grad } j+1 \le k-1} \rangle + \alpha_j \langle p_j, p_j \rangle \Rightarrow$

$\langle p_k, x p_j \rangle = 0$

$\Rightarrow \alpha_j = 0.$

1) $n=1$ auf $[0,1]$ $\Rightarrow$ $p_1(x) = x - \frac{1}{2}$ $\Rightarrow$ $\gamma_1 = \frac{1}{2}$, $b_1 = 1$

auf $[-1,1]$ $\Rightarrow$ $p_1(x) = x$ $\Rightarrow$ $c_1 = 0$, $b_1 = 1$

↳ Gauss-Quadratur mit $n=1$ Knoten $\equiv$ MPR.

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

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

$\gamma_{1,2} = \pm \frac{1}{\sqrt{3}}$, $c_1 = \frac{1}{2} - \frac{\sqrt{2}}{6}$, $c_2 = \frac{1}{2} + \frac{\sqrt{2}}{6}$.

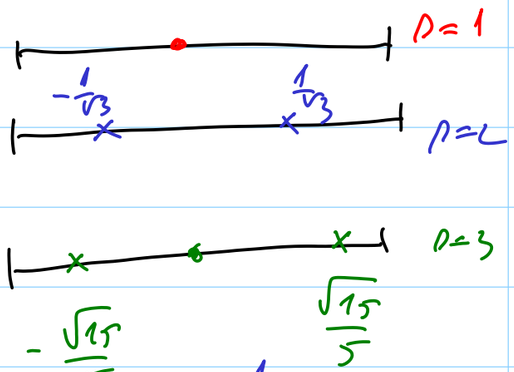
Ordnung $2n = 2 \cdot 2 = 4$

zu $p_3(1) = 1$ "normiert"

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

$\gamma_2 = 0$, $\gamma_{1,3} = \pm \frac{\sqrt{15}}{5}$

$b_2 = \frac{8}{18}$, $b_1 = b_3 = \frac{5}{18}$



Ordnung $2n = 2 \cdot 3 = 6$

Ben: Gewichte wurden hier "Hand"-Berechnet via $b_i = \int_0^1 l_i(t)^2 dt$ (l_i = Lagrange Polynom)

Ben Gewichte der Gauss-QF sind positiv

Def Lagrange Polynome zu Stützstellen $x_0, x_1, \dots, x_n$

für $i = 0, 1, 2, \dots, n$

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

Bsp x_0, x_1, x_2:

$$l_0(x) = \frac{x - x_1}{x_0 - x_1} \cdot \frac{x - x_2}{x_0 - x_1}$$

$$l_1(x) = \frac{x - x_0}{x_1 - x_0} \quad \frac{x - x_2}{x_1 - x_2}$$

$$l_2(x) = \frac{x - x_0}{x_2 - x_0} \quad \frac{x - x_1}{x_2 - x_1}$$

Bem 1) $\ell_i(x_j) = 0$ für alle $i \neq j$

2) $\ell_i(x_i) = 1$

3) Grad $\ell_i = n$

4) $\sum_{i=0}^{n} \ell_i(x) = 1$ für alle $x \in \mathbb{R}$.

5) $\sum_{i=0}^{n} \ell_i^{(m)}(x) = 0$ für $m \geq 1$.

6) $\ell_0, \ell_1, \dots \ell_n$ bilden eine Basis im Raum der Polynom von Grad $\leq n$

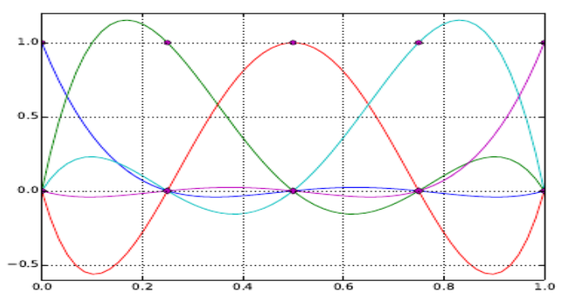


Abb. 8.3.2. Die 5 Lagrange Polynome zu Knoten $0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1$.

Seite 289 im Skript

Theorem Die Gewichte der Gauss QF sind positiv.

Beweis Verwende die Lagrange-Polynome zu den Stützstellen. $c_1, c_2, \dots, c_p$ = Knoten der QF

Grad $\ell_i = p-1$ für $i = 1, 2, \dots p$ } exakt $\Rightarrow$

$$\int_0^1 f(t)\,dt \approx \sum_{i=1}^{p} b_i f(c_i)$$

$f(t) = \ell_i(t)^2$ Polynom vom Grad $2(p-1)$ }

$$0 < \int_0^1 \ell_i(t)^2 dt = \sum_{j=1}^{p} b_j \ell_i(c_j)^2 = b_i \cdot 1$$

$\ell_i(c_j) = 0$ für $j \neq i$

1 für $j = i$

Im Prinzip $b_i = \int_0^1 \ell_i(t)^2 dt$ könnte man verwenden um Gewichte zu rechnen

Es gibt eine bessere Methode.

3-Term Rekurenz für Polynome:

(Achtung $\underline{a}, \underline{b}, \underline{c}$ sind allgemein, $\underline{c} \neq$ Knoten $t_1, t_2, \dots, t_n$)

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

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

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

$\underline{p}(x)$

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

3-Term Rekursion $\Leftrightarrow$

$$x\, \underline{p}(x) = \underline{\underline{A}}\, \underline{p}(x) + \frac{1}{a_n} p_n(x)\, \underline{e}_n$$

Knoten der Gauss-QF sind die Nullstellen von $p_n(x)$

$\hookrightarrow t_1, t_2, \dots, t_n \Rightarrow p_n(t_j) = 0$ für $j = 1, 2, \dots, n$

$$t_j\, \underline{p}(t_j) = \underline{\underline{A}}\, \underline{p}(t_j) \quad +0$$

$\Rightarrow$ t_j Eigenwert der Matrix $\underline{\underline{A}}$

$\underline{p}(t_j)$ Eigenvektor der Matrix $\underline{\underline{A}}$

$$\underline{p}(t_j) = \begin{bmatrix} p_0(t_j) \\ p_1(t_j) \\ \vdots \\ p_{n-1}(t_j) \end{bmatrix}$$

$\underline{\underline{A}}$ symmetrisch.

$\boxed{\text{eigh}(\underline{\underline{A}})}$

$\text{eig}(\underline{\underline{A}})$

Bem Es gibt ziemlich gute numerische Verfahren um EW-Probleme zu lösen

```python
from numpy import arange, diag, sqrt
from numpy.linalg import eigh

def gaussquad(n):
    r"""
        Compute nodes and weights for Gauss-Legendre quadrature.

        n:  Number of node-weight pairs
        """
    i = arange(n)   #i = array([0,1,...,n-1])
    b = (i+1) / sqrt(4*(i+1)**2 - 1)
    # now we generate the matrix; it is symmetric
    J = diag(b, -1) + diag(b, 1)
    # in order to find the eigenvalues we can use eigh since J is symmetric
    x, ev = eigh(J)
    # finally, we apply the formula for the weights
    w = 2 * ev[0,:]**2
    return x, w
```

auf $[-1,1]$

```python
from sympy import *

x = Symbol("x")
n = Symbol("n", positive=True)
# First we generate two streams of symbols corresponding to knots and weights
xigen = numbered_symbols(prefix="xi", start=1)
omegagen = numbered_symbols(prefix="omega", start=1)
# Now we specify how many parameters we need.  In this case, two knots and two
weights.  Therefore we set N = 2
N = 2  # Choose <= 3
xis = [ next(xigen) for i in range(N) ]
wis = [ next(omegagen) for i in range(N) ]
# Using sympy one can perform simple symbolic integrations, which deliver the
exact result.
# In order to do this we use the function "integrate"
# For every n in the interval [0, 2*N-1] we set the condition that the result of
the numerical integration
#be equal to the analytical value
eqns = [ integrate(x**n, (x, -1, 1)) - (sum([wi*xi**n for xi,wi in
zip(xis,wis)])) for n in range(2*N)]
pprint(eqns)
# Therefore we get a system of equations which we need to solve:
#we set eqns = 0 and find the corresponding knots and weights
sols = solve(eqns, numerical=True)
pprint(sols)
```

Berechnung der Gewichte für Gauss-Quadratur auf $[-1,1]$

$$\langle p_i, p_k \rangle = \int_{-1}^{1} \underbrace{p_i(x)\, p_k(x)}_{\text{Grad} \le 2n-2}\, dx = \sum_{j=1}^{n} p_i(t_j)\, p_k(t_j)\, w_j .$$

$$= \begin{cases} 0, & i \neq k \\ 1, & i = k \end{cases}$$

Konstruktion: Gram-Schmidt

QF exakt bis Grad $2n-1$.

$$\underline{f}(x) = \begin{bmatrix} p_0(x) \\ p_1(x) \\ \vdots \\ p_{n-1}(x) \end{bmatrix} \quad \text{in } t_1, \dots, t_n \text{ ausgewertet}$$

Notiere

$$\underline{\underline{M}} = \left[\underline{f}(t_1) \;\; \underline{f}(t_2) \dots \underline{f}(t_n) \right] \Longrightarrow$$

$$\underline{\underline{I}} = \underline{\underline{M}}\, \mathrm{diag}(w_1, \dots, w_n)\, \underline{\underline{M}}^T \Bigg\} \Rightarrow$$

$\Rightarrow \underline{\underline{M}}$ invertierbar

$\underline{\underline{M}}^{-1}$ | | $\underline{\underline{M}}^{-T}$

$$\mathrm{diag}(w_1, \dots, w_n) = (\underline{\underline{M}}^T \underline{\underline{M}})^{-1} \Rightarrow$$

$$\mathrm{diag}(w_1, \dots, w_n)^{-1} = \underline{\underline{M}}^T \underline{\underline{M}} \Rightarrow$$

$$\frac{1}{w_j} = \underline{f}(t_j)^T \underline{f}(t_j) = \| \underline{f}(t_j) \|^2 =$$

$$= \sum_{k=0}^{n-1} p_k(t_j)^2$$

oder $$w_j = \frac{1}{\| \underline{f}(t_j) \|^2}$$

ABER Die Eigenvektoren sind nicht eindeutig. ausserdem liefert eig normierte EV.

Sei $\underline{v}^j$ ein Eigenvektor $\Rightarrow$ es gibt c Konstante so dass

$$\underline{v}^j = c\,\underline{f}(t_j) = c \begin{bmatrix} p_0(t_j) \\ \vdots \\ p_{n-1}(t_j) \end{bmatrix}$$

$$1 = \langle p_0, p_0 \rangle = \int_{-1}^{1} p_0(t_1)\, p_0(t_1)\, dx \quad \Rightarrow \quad p_0(t_1) = \frac{1}{\sqrt{2}}$$

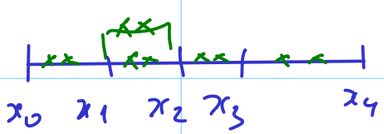
$$(\underline{v}^j)_1 = c \cdot p_0(t_1) = c \cdot \frac{1}{\sqrt{2}} \quad \Rightarrow \quad c = \sqrt{2}\,(\underline{v}^j)_1$$

$$\underline{f}(t_j) = \frac{1}{c}\,\underline{v}^j = \frac{1}{\sqrt{2}\,(\underline{v}^j)_1}\,\underline{v}^j \quad \Rightarrow$$

$$\omega_j^{-1} = \| \underline{f}(t_j) \|^2 = \frac{1}{2} \frac{\|\underline{v}^j\|^2}{(v_1^j)^2} \quad \Rightarrow$$

$$w_j = \frac{2\,(\underline{v}^j)_1^2}{\|\underline{v}^j\|^2} \; ; \text{ eig liefert normiertes } v \Rightarrow$$

$$\Rightarrow w_j = 2\,(\underline{v}_1^j)^2$$



Bem 1) Gauss-Knoten sind nicht verschachtelt
$\Rightarrow$ Teuer bei Adaptivität, ...

2) Endpunkte sind keine Knoten $\Rightarrow$
Gauss-QF ist offen.

3) Manchmal braucht man ein oder beide Endpunkte des Integrationsintervals als Knoten $\Rightarrow$
1 Knoten fest $\Rightarrow$ max Ordnung $2N-1$ (Radau)
2 Knoten fest $\Rightarrow$ max Ordnung $2N-2$ (Lobatto)

4) Fehler bei Gauss-Quadratur:

$$\int_a^b f(x)\,dx - \sum_{j=1}^{N} b_j f(c_j) = \frac{f^{(2N)}(\xi)}{(2N)!} \quad \text{mit } a < \xi < b$$

$G_N(f, a, b)$ | kombiniert:

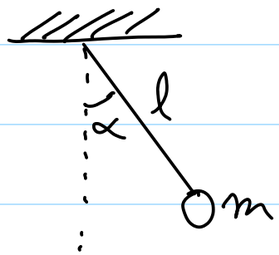
$$\left| \int_a^b f(x)\,dx - \sum_{k=1}^{M} G_N(f, x_{k-1}, x_k) \right| \le C \cdot h^{2N} \max_{\xi \in [a,b]} |f^{(2N)}(\xi)| .$$

§2 Einfache Verfahren für ODEs 1. Ordnung

ODE = ordinary differential equations

gewöhnliche Differentialgleichungen

§2.1. Linearisierung: global und lokal



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

Physik: $\sin\alpha(t) \approx \alpha(t)$

↳ gut nur für $\alpha(t)$ klein.

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

⊛ ODE 2. Ordnung.
da zweite Ableitung der unbekannten Funktion α

⊛ autonom, da t nicht explizit erscheint

$$\dot{\alpha}(t) = \frac{d}{dt}\alpha(t)$$

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

"global": linearisiere die rechte Seite:

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

↳ klein für α klein.

Neues Model:

$$(1)\quad \begin{cases} \ddot{\beta}(t) = -\frac{g}{l}\beta(t) & \text{hat exakte Lösung.} \\ \beta(0) = \alpha_0 & \omega^2 = \frac{g}{l} \\ \dot{\beta}(0) = \dot{\alpha}_0 \end{cases}$$

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

Trick: Reduziere die Ordnung der ODE.

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

$$\dot{p}(t) = \ddot{\alpha}(t) = -mg\sin\alpha(t)$$

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

$$\begin{cases} \dot{\alpha} = p \\ \dot{p} = -mg\sin\alpha(t) \\ \alpha(0) = \alpha_0 \\ p(0) = \dot{\alpha}_0 \end{cases}$$

$$\underline{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \alpha \\ \rho \end{bmatrix} \qquad \underline{y} : [0,T] \to \mathbb{R}^2 \text{ Unbekannte Funktion.}$$

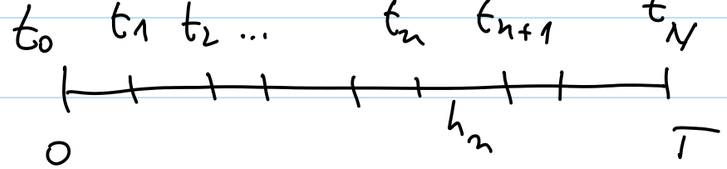
$$\underline{f}(\underline{y}) = \begin{bmatrix} y_2 \\ -\frac{g}{l} \sin y_1 \end{bmatrix} \qquad \underline{f} : \mathbb{R}^2 \to \mathbb{R}^2$$

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

Ziel: Möchte Approximation von

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

Zeitgitter:



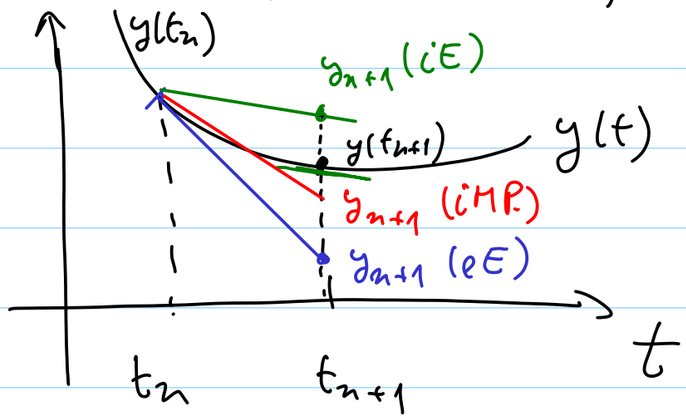
$0 = t_0 < t_1 < t_2 < \dots < t_n < t_{n+1} < \dots < t_N = T$

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

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

Idee: lokal: linearisiere die Lösung lokal.

Taylor um t_n für $\underline{y}(t)$



Taylor um t_n: $\underline{y}(t_n + h) = \underline{y}(t_n) + h\,\dot{\underline{y}}(t_n) + O(h^2)$

$\underline{y}(t_n) = \underline{y}_n$, $\underline{y}(t_n + h) = \underline{y}_{n+1}$, $\dot{\underline{y}}(t_n) \overset{(2)}{=} \underline{f}(t_n, \underline{y}(t_n)) = \underline{f}(t_n, \underline{y}_n)$ $\Longrightarrow$

$$(eE) \begin{cases} \underline{y}_{n+1} = \underline{y}_n + h\,\underline{f}(t_n, \underline{y}_n) & \text{expliziter Euler} \\ \underline{y}_0 = \underline{y}(t_0) \text{ gegeben.} & \text{lokaler Fehler } O(h^2) \end{cases}$$

$n = 0, 1, 2, \dots$ $\qquad |\underline{y}(t_{n+1}) - \underline{y}_{n+1}| \le C \cdot h^2$

andere Herleitung:

$$\underline{f}(t_n, \underline{y}(t_n)) = \dot{\underline{y}}(t_n) = \lim_{h\to 0} \frac{\underline{y}(t_n+h) - \underline{y}(t_n)}{h} \approx \frac{y_{n+1}-y_n}{h}$$

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

$\Rightarrow$ (e.E)

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

$$\underline{y}(t_{n+1}-h) = \underline{y}(t_{n+1}) - h\dot{\underline{y}}(t_{n+1}) + O(h^2) \quad \Rightarrow$$

$\underline{y}(t_{n+1}) \approx y_n$, $\underline{y}(t_{n+1}) \approx y_{n+1}$, $\dot{\underline{y}}(t_{n+1}) = f(t_{n+1}, \underline{y}(t_{n+1})) \approx f(t_{n+1}, y_{n+1})$

$$(iE)\begin{cases} y_n = y_{n+1} - h\, f(t_{n+1}, y_{n+1}) & \text{impliziter Euler} \\ n = 0,1,2,\dots \quad y_0 = y(t_0) \text{ gegeben} & \text{lokaler Fehler } O(h^2) \end{cases}$$

Bem $f(t,y) = -\frac{g}{l}\sin y$

$y_n = y_{n+1} - h\left(-\frac{g}{l}\right)\sin y_{n+1}$ algebraische Gleichung für y_{n+1}

Neue Idee: linearisiere lokal um die Mitte des Intervalls:

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

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

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

$\ominus \Rightarrow$

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

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

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

$\underline{y}(t_{n+1}) \approx y_{n+1}$, $\underline{y}(t_n) \approx y_n$, $\underline{y}(t^*)$?

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

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

($\underline{y}(t_n) \approx \underline{y}_n$, $\underline{y}(t_{n+1}) \approx \underline{y}_{n+1}$)

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

implizites Mittelpunktsregel / lokaler Fehler $O(h^3)$

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

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

implizites Trapezregel / lokaler Fehler $O(h^3)$

Bem.

Das geht so nur wenn $\underline{y}(t)$ glatt genug ist.

Bemerkung Addieren wir die 2 Gleichungen:

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

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

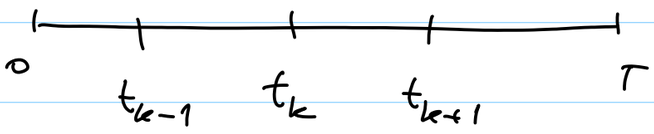
$$\Rightarrow \ddot{\underline{y}}(t_n + \tfrac{h}{2}) \approx \frac{\underline{y}_{n+1} - 2\underline{y}_{n+\frac{1}{2}} + \underline{y}_n}{\left(\frac{h}{2}\right)^2}$$ mit lokaler Fehler $O(h^2)$

Bemerkung Selbe Rechnung um $t_n \Rightarrow$

$$\ddot{\underline{y}}(t_n) \approx \frac{\underline{y}_{n+1} - 2\underline{y}_n + \underline{y}_{n-1}}{h^2}$$ mit lokaler Fehler $O(h^2)$

§2.2. Störmer-Verlet Verfahren

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



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

$$\Rightarrow \boxed{\underline{y}_{k+1} = -\underline{y}_{k-1} + 2\underline{y}_k + h^2 \underline{f}(t_k, \underline{y}_k)} \quad (St\text{-}V)$$

explizites, 2-Schritt-Verfahren

$\Rightarrow$ brauche Startwerte $\underline{y}(t_0) = \underline{y}_0$

$\underline{y}_1 \approx \underline{y}(t_1)$

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

zweite Möglichkeit: Taylor mit 3 Terme:

$$\underline{y}(t_1) = \underbrace{\underline{y}(t_0)}_{\underline{y}_0} + h\underbrace{\dot{\underline{y}}(t_0)}_{\underline{v}_0} + \frac{h^2}{2}\underbrace{\ddot{\underline{y}}(t_0)}_{\underline{f}(t_0, \underline{y}_0)} + O(h^3) \Rightarrow$$

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

(St-V): lokal $O(h^3)$ und global $O(h^2)$

$$|\underline{y}(T) - \underline{y}(t_N)| \leq c \cdot h^2$$

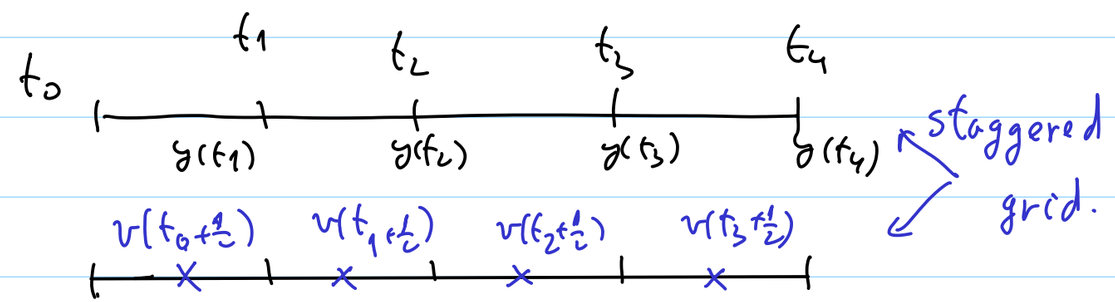
Notiere $\underline{v}_{k+\frac{1}{2}} = \frac{\underline{y}_{k+1} - \underline{y}_k}{h}$ entspricht $\underline{v}(t_k + \frac{1}{2}h)$

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

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: $\underline{v}_k = \dfrac{\underline{y}_{k+1} - \underline{y}_{k-1}}{2h}$ (SE-V) $\Longrightarrow$

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

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

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

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

§2.3. Vorgehensweise bei Implementierung

Ziel:
Pendelgleichung "lösen": Approximation der Lösung und der Energie mittels vorigen Methoden.

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

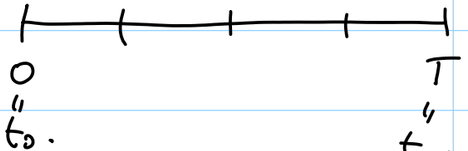
1. Schritt: Umschreiben in ODE 1. Ordnung.

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

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

Möchte: Lösung & Energie der Lösung zu Zeitpunkten.

$0 = t_0 < t_1 < t_2 < \dots < t_N = T$



Gegeben: $\alpha_0, \dot\alpha_0, g, l$ $\left(\omega^2 = \frac{g}{l}\right)$, $t_0 = 0, T$

Wähle N!

$\underline{y}(t_0), \underline{y}(t_1) \dots, \underline{y}(t_N)$

Potentiele Energie: $V(\underline{y}) = -g\cos y_1 + C$

kinetische Energie: $K(\underline{y}) = \frac{1}{2} y_2^2$

Totale Energie $E(\underline{y}) = V(\underline{y}) + K(\underline{y})$

```
def potE(y):
    return -g cos y[:,0]
```

```
def kinE(y)
    return 1/2 y[:,1]^2
```

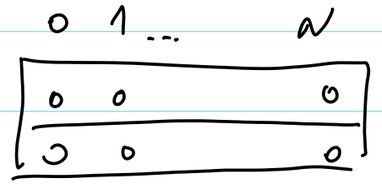
```
def totE(y)
    return potE(y) + kinE(y)
```

$\underline{y} = \begin{bmatrix} x \\ \alpha \end{bmatrix}$ $\underline{y}[0]$ $\underline{y}[1]$

Speicherplatz vorbereiten!

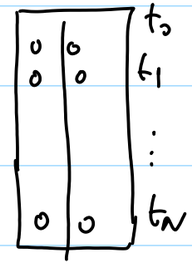
$t_0 \quad t_1 \dots \quad t_N$

$\begin{bmatrix} y[0] \\ y[1] \end{bmatrix} \begin{bmatrix} y[0] \\ y[1] \end{bmatrix} \dots \begin{bmatrix} y[0] \\ y[1] \end{bmatrix} \Rightarrow$ array $(2, N+1)$



oder

```
y = zeros((N+1,2))
```




```
y[0] = array([α_0, α̇_0])
```

(nA) $y[n,0] = \frac{\alpha}{\omega}\sin(\omega t_n) + \alpha_0\cos(\omega t_n) = g(t_n)$

$y[n,1]$ = Ableitung davon $= \dot\alpha\cos(\omega t_n) - \alpha_0\omega\sin(\omega t_n)$
$= dg(t_n)$

```
für n = 0, 1, 2..., N-1:
    y[n+1,0] = g(t_{n+1})
    y[n+1,1] = dg(t_{n+1})
```

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

→ implementiere die rechte Seite $\underline{f}$.

```
def f(y):
    return ( y[1]            )   ( -ω² sin α(t) )
           ( -g/l sin y[0]   )
```

für $n = 0, 1, 2 \ldots, N-1$

$y[n+1,0] = y[n,0] + h f(\underline{y})[0]$

$y[n+1,1] = y[n+1,0] + h f(\underline{z})[1]$

$\underline{y}[n+1] = \underline{y}[n] + h\underline{f}(\underline{z})$

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

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.

Sei die Unbekannte $\underline{z}$ ⇒ (iE) $\underline{z} = \underline{a} + h\underline{f}(\underline{z})$

$\Leftrightarrow \underbrace{\underline{z} - \underline{a} - h\underline{f}(\underline{z})}_{\underline{F}(\underline{z})} = 0 \Leftrightarrow \underline{F}(\underline{z}) = 0$

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

Finde die Nullstelle(n) von $\underline{F}$!

scipy.optimise.fsolve :

fsolve($\underline{F}$, $\underline{z}_0$) ← Startwert (nah der gesuchten Nullstelle)

Startwert für fsolve in einem Schritt von (EE)?

$\underline{z}_0 = \underline{y}_n$ oder Lösung mit lin. Rechte Seite

oder was (EE) vorschlägt

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

für $n = 0, 1, 2, \ldots, N-1$:

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

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

zweite Möglichkeit:

nutze die Form der Gleichung des Pendels.

alg. Problem:

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

$$\Rightarrow \quad z_1 - a_1 - h\left(a_2 + h\frac{g}{l}\sin z_1\right) = 0 \quad \Leftrightarrow$$

$$\Leftrightarrow \quad G(z) = 0 \quad \text{mit} \quad G(z) = z - a_1 - h a_2 - h^2\frac{g}{l}\sin z$$

alg. Gleichung in einer Unbekannten $z \in \mathbb{R}$

↖ billiger, einfacher!

fsolve $\Rightarrow z_1 \Rightarrow z_2$

§2.4 Fehlerschätzung und Konvergenz

Taylor mit Rest als Integral:

$$f(x) = f(a) + \frac{x-a}{1!} f'(a) + \frac{(x-a)^2}{2!} f''(a) + \dots + \frac{(x-a)^{n-1}}{(n-1)!} f^{(n-1)}(a) + \int_a^x \frac{(x-t)^{n-1}}{(n-1)!} f^{(n)}(t)\,dt$$

$\dot{\underline{y}} = \underline{f}(t, \underline{y})$ mit $\underline{y}(t_0) = \underline{y}_0$

$\underline{y}_n \approx \underline{y}(t_n)$ mit $t_n = t_0 + nh$; h = Zeitschrittweite.

↳ vom numerischen Verfahren definiert

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

$\|\underline{f}(t, \underline{y}) - \underline{f}(t, \underline{z})\| \le L \|\underline{y} - \underline{z}\|$ für alle $\underline{y}, \underline{z} \in \mathbb{R}^d$, $t \in [t_0, T]$

($L \to$ Konstante $\in \mathbb{R}$.)

Explizite Euler: $\underline{y}_{n+1} = \underline{y}_n + h \underline{f}(t_n, \underline{y}_n)$

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

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

Beweis: 3 Schritte.

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

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

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

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

Satz von Taylor mit Rest als Integral:

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

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

Möchte aber $\int_0^1$, als wie bei der Fehlerschätzung vom Quadratur: Variablenwechsel:

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

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

$$\leq \max \|\underline{\ddot{y}}(t)\| \quad [t_0, T]$$

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

2) Fehlerfortpflanzung:

(eE) mit Startwert $\underline{z}_n$: $\underline{z}_{n+1} = \underline{z}_n + h\,\underline{f}(t_n, \underline{z}_n)$

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

$$\Rightarrow \|\underline{z}_{n+1} - \underline{w}_{n+1}\| \leq \|\underline{z}_n - \underline{w}_n\| + h\|\underline{f}(t_n, \underline{z}_n) - \underline{f}(t_n, \underline{w}_n)\| \leq$$

$$\leq \|\underline{z}_n - \underline{w}_n\| + hL\|\underline{z}_n - \underline{w}_n\| = \|\underline{z}_n - \underline{w}_n\|(1+hL)$$

3) Fehlerakkumulierung

1.Schritt, 2.Schritt, 3.Schritt

$$\|\underline{y}(t_n) - \underline{y}_n\| \leq ch^2 + ch^2(1+hL) + ch^2(1+hL)^2 + \ldots + ch^2(1+hL)^{n-1} =$$

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

$$\leq ch\,\frac{e^{nhL} - 1}{L} = ch\,\frac{e^{(t_n - t_0)L} - 1}{L}$$

(imp): lokale Fehler: $t^* = \frac{1}{2}(t_n + t_{n+1})$, $\bar{\underline{y}} = \frac{1}{2}(\underline{y}_n + \underline{y}_{n+1})$

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

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

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

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

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

$\underline{y}(t^*) - \underline{\bar{y}} = \frac{1}{2}\underline{y}(t^*) + \frac{1}{2}\underline{y}(t^*) - \frac{1}{2}\underline{y}_n - \frac{1}{2}\underline{y}_{n+1}$

$\underline{y}(t_n)$

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

Taylor $= \frac{h}{2}\dot{\underline{y}}(t^*) + O(h^2) - \frac{h}{2}\dot{\underline{y}}(t^*) + O(h^2) + \frac{1}{2}\left(\underline{y}(t_{n+1}) - \underline{y}_{n+1} \right)$

Somit: $\| \underline{f}(t^*, \underline{y}(t^*)) - \underline{f}(t^*, \underline{\bar{y}}) \| \le L \cdot c \cdot h^2 + \frac{1}{2} \| \underline{y}(t_{n+1}) - \underline{y}_{n+1} \|$

$\| \underline{y}(t_{n+1}) - \underline{y}_{n+1} \| \le h^3 \cdot L \cdot c + \frac{hL}{2} \| \underline{y}(t_{n+1}) - \underline{y}_{n+1} \| \Rightarrow$

$\| \underline{y}(t_{n+1}) - \underline{y}_{n+1} \| \le \frac{2}{2 - hL} L c h^3$

(St.-V) lokaler Fehler:

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

$= \underline{y}(t_{n+1}) + \underline{y}(t_{n-1}) - 2\underline{y}(t_n) + h^2 \underline{f}(t_n, \underline{y}(t_n)) \overset{\text{Taylor}}{=}$

$= h^2 \ddot{\underline{y}}(t_n) - h^2 \underline{f}(t_n, \underline{y}(t_n)) + O(h^4) \Rightarrow$ lokal $O(h^4)$ $\Rightarrow$

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

0

aber die Startwerte ist nur $O(h^3)$

Fehlerfortpflanzung + akumulierung

$\Rightarrow O(h^2)$

§3 Strukturerhaltung.

§3.1. Invariante und Hamilton'sche Systeme

Bsp autonome Lotka-Volterra: $d=2$

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

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

$\alpha, \beta, \gamma, \delta > 0$ konstant, bekannt

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

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

$\Rightarrow$

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

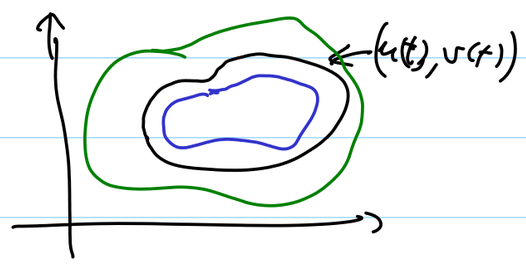
$$\left(\delta - \frac{\gamma}{u}\right)\dot u - \left(\frac{\alpha}{v} - \beta\right)\dot v = 0 \quad \Leftrightarrow$$

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

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

$\Rightarrow I(u(t), v(t)) =$ konstant, wenn u, v die Lösungen des L-V-systems sind.

d.h. die Trajektorien sind Niveaulinien der Funktion $I : \mathbb{R}^2 \to \mathbb{R}$.



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

$I(y(t)) =$ konstant für jede Lösung $y = y(t)$ der ODE.

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

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

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

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

Wie ist $\underline{\text{grad}}\, f(\underline{x}) = ?$

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

$$\underline{\text{grad}}\, f(\underline{x}) = \frac{1}{\|\underline{x}\|_2} \underline{x}$$

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

$\Longleftrightarrow$

$\underline{\text{grad}}\, I(\underline{y}) \cdot \underline{f}(t, \underline{y}) = 0$ für alle $(t, \underline{y}) \in [0,T] \times D$ wo es eine Lösung gibt. (glatt)

Beweis Annahme.

$\Leftarrow$: $0 = \underline{\text{grad}}\, I(\underline{y}(t)) \cdot \underline{f}(t, \underline{y}(t)) =$ ($\underline{y}(t)$ Lösung)

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

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

$\Rightarrow$: Nehme $\underline{z}_0 \in D$, $t_0 \in [0,T]$ beliebig.

Verwende die ODE: $\begin{cases} \dot{\underline{y}} = \underline{f}(t, \underline{y}) \\ \underline{y}(t_0) = \underline{z}_0 \end{cases} \Rightarrow \underline{y}(t)$ (Invariante)

$\Rightarrow I(\underline{y}(t)) = \text{konstant} \overset{\text{ODE}}{\Rightarrow} \underline{\text{grad}}\, I(\underline{y}(t)) \dot{\underline{y}}(t) = 0$ ($= \underline{f}(t, \underline{y}(t))$)

für alle t (nehme $t = t_0$) $\Rightarrow \underline{\text{grad}}\, I(\underline{y}(t_0)) \dot{\underline{y}}(t_0) = 0 \Rightarrow$

$\underline{\text{grad}}\, I(\underline{y}(t_0)) \cdot \underline{f}(t_0, \underline{z}_0) = 0$

$\Rightarrow$ für alle $t_0, \underline{z}_0$.

Notation $H: \underset{\psi}{\mathbb{R}^d} \times \underset{\psi}{\mathbb{R}^d} \to \mathbb{R}$ Hamilton Funktion.

$\underline{p} \quad \underline{q}$

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

Def Hamiltonische Differentialgleichung.

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

für $j = 1, 2, \dots, d$

autonomes Hamilton-System mit der Hamilton-Funktion H.

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

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

$q = \alpha =$ Winkel $\quad p = \dot{q}$

2). Konservatives Kraftfeld:

d.h.

$$\underline{f}(\underline{x}) = -\operatorname{grad} U(\underline{x}) \qquad \underline{x} \in \mathbb{R}^d$$

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

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

$p = m\dot{r} \Rightarrow$ Bewegungsgleichungen eines Massenpunktes m im konservativen Zentralfeld.

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

vergleiche mit Newton' Gleichung. $m\ddot{r}(t) = f(r(t))$.

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

Bem $H(p,q)$ Invariante für das Hamilton'System.

Beweis

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

Hamilton'System:

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

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

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

Überprüfe dies für $I := H$:

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

$$\operatorname{grad} H(\underline{y}) = \begin{bmatrix} \underline{a} \\ \underline{b} \end{bmatrix} \rightleftharpoons [\underline{a} \ \underline{b}] \begin{bmatrix} \underline{\underline{0}} & -\underline{\underline{I}} \\ \underline{\underline{I}} & \underline{\underline{0}} \end{bmatrix} \begin{bmatrix} \underline{a} \\ \underline{b} \end{bmatrix} =$$

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

Theorem $I: \mathbb{R}^d \to \mathbb{R}$, $I(\underline{y}) = \frac{1}{2} \underline{y}^T \underline{\underline{B}} \underline{y}$ mit $\underline{\underline{B}} \in \mathbb{R}^{d \times d}$, I Invariante für $\dot{\underline{y}} = \underline{f}(\underline{y})$ mit $\underline{f}$ differentierbar.

Sei $(\underline{y}_n)$ die numerische Approximation aus (IMP).

Dann

$$I(\underline{y}_n) = I(\underline{y}_0) \quad \text{für alle } n.$$

Beweis

$I(\underline{y}) \in \mathbb{R} \Rightarrow I(\underline{y}) = I(\underline{y})^T = \left(\frac{1}{2} \underline{y}^T \underline{\underline{B}} \underline{y}\right)^T =$

$$= \frac{1}{2} \underline{y}^T \underline{\underline{B}}^T \underline{y} \Rightarrow$$

$$2I(\underline{y}) = I(\underline{y}) + I(\underline{y})^T = \frac{1}{2} \underline{y}^T (\underbrace{\underline{\underline{B}} + \underline{\underline{B}}^T}_{\underline{\underline{A}}}) \underline{y} = \text{konstant.}$$

$\frac{1}{2} \underline{y}^T \underline{\underline{A}} \underline{y}$ Invariante mit $\underline{\underline{A}} = \underline{\underline{B}} + \underline{\underline{B}}^T$ also symmetrisch.

Invariante $\Leftrightarrow \frac{d}{dt} = 0 \Rightarrow$ $\underline{\underline{A}}\,\underline{y}(t) \cdot \underline{f}(\underline{y}(t)) = 0$ für $\underline{y}(t)$ Lösung.

$\underline{\underline{A}}$ symmetrisch.

$$\frac{d}{dt} I(\underline{y}) = \underline{\underline{A}}\,\underline{y}(t) \cdot \dot{\underline{y}}(t) = \underline{\underline{A}}\,\underline{y}(t) \cdot \underline{f}(\underline{y}(t))$$

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

$$\frac{\partial}{\partial y_i}\left(\underline{y}^T \underline{\underline{A}}\,\underline{y}\right) = \dots$$

$$I(\underline{y}_{n+1}) - I(\underline{y}_n) = \frac{1}{2}(\underline{y}_{n+1} + \underline{y}_n)^T \underline{\underline{A}}\,(\underline{y}_{n+1} - \underline{y}_n) =$$

$\left(\underline{\underline{A}}\,\frac{1}{2}(\underline{y}_{n+1} + \underline{y}_n)\right)^T$ // $\leftarrow$ (IMP)

$$= \underline{\underline{A}}\left(\underbrace{\frac{1}{2}(\underline{y}_{n+1} + \underline{y}_n)}_{z}\right) \cdot h\, \underline{f}\left(\underbrace{\frac{1}{2}\underline{y}_n + \underline{y}_{n+1}}_{z}\right) =$$

$$= (\underline{\underline{A}}\,\underline{z}) \cdot h\, \underline{f}(\underline{z}) = 0$$

Da:

$$\underline{y}_{n+1}^T \underline{\underline{A}}\,\underline{y}_{n+1} - \underline{y}_n^T \underline{\underline{A}}\,\underline{y}_n = \underline{y}_{n+1}^T \underline{\underline{A}}\,\underline{y}_{n+1} - \underline{y}_{n+1}^T \underline{\underline{A}}\,\underline{y}_n + \underline{y}_{n+1}^T \underline{\underline{A}}\,\underline{y}_n - \underline{y}_n^T \underline{\underline{A}}\,\underline{y}_n$$

$$\pm\, \underline{y}_{n+1}^T \underline{\underline{A}}\,\underline{y}_n =$$

$$= \underline{y}_{n+1}^T \underline{\underline{A}}\,(\underline{y}_{n+1} - \underline{y}_n) + \underbrace{(\underline{y}_{n+1} - \underline{y}_n)^T \underline{\underline{A}}\,\underline{y}_n}_{=} = (\underline{y}_{n+1}^T + \underline{y}_n^T)\underline{\underline{A}}\,(\underline{y}_{n+1} - \underline{y}_n)$$

$\underline{\underline{A}}$ symmetrisch.

$$\left((\underline{y}_{n+1} - \underline{y}_n)^T \underline{\underline{A}}\,\underline{y}_n\right)^T \leftarrow \underline{y}_n^T \underline{\underline{A}}\,(\underline{y}_{n+1} - \underline{y}_n)$$

§3.2. Splitting Verfahren

autonome ODE: $\dot{y} = f(y)$

Bem Was tun wenn ODE nicht autonom ist?

Trick:

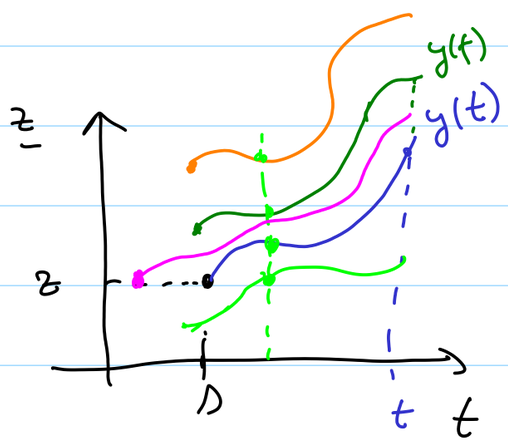
$$z = \begin{bmatrix} t \\ y \end{bmatrix} \Rightarrow \dot{z} = \begin{bmatrix} 1 \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 1 \\ g(t,y) \end{bmatrix} = \begin{bmatrix} 1 \\ g(z) \end{bmatrix} = f(z)$$

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

$\rightarrow \dot{z} = f(z)$

mit $z(0) = \begin{bmatrix} t_0 \\ y(t_0) \end{bmatrix}$

Bem
$$\begin{cases} \dot{y} = g(t,y) \\ y(s) = x \end{cases}$$

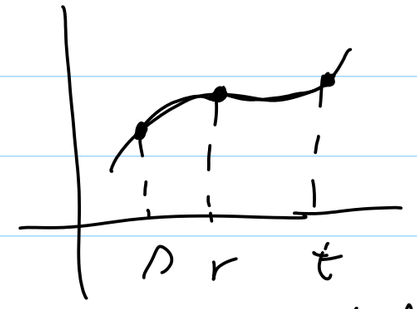


$\Phi^{s,t} x = y(t)$ mit $y(s) = x$

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

1) $\Phi^{t,t} =$ Identität! $\quad \Phi^{t,t} x = x$ für alle $x \in D$.

2) $\Phi^{s,t} x = \Phi^{r,t} \Phi^{s,r} x$



3) Falls 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.

Beweis:

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

$= f(u(t))$

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

Bsp $e^{mh} = \sum_{n=0}^{\infty} \frac{1}{n!}(mh)^n$

↳ auch als Lösung der ODE:

$$\dot{y}(h) = m\,y(h) \;;\; y(0)=1$$

Für mehrere solchen linearen, entkoppelten ODE:

$$\begin{cases} \dot{y}_1(t) = m_1 y_1(t) \Rightarrow y_1(h) = e^{m_1 h} \\ \dot{y}_2(t) = m_2 y_2(t) \Rightarrow y_2(h) = e^{m_2 h} \\ \dots \\ \dot{y}_d(t) = m_d y_d(t) \Rightarrow y_d(h) = e^{m_d h} \end{cases}$$

$$\underline{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_d \end{bmatrix} : \quad \dot{\underline{y}} = \operatorname{diag}(m_1,\dots,m_d)\,\underline{y}$$

$$\underline{y}(h) = \operatorname{diag}(e^{m_1 h},\dots,e^{m_d h}) \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$= e^{h\,\operatorname{diag}(m_1,\dots,m_d)} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$e^{\operatorname{diag}(m_1,\dots,m_d)h} = \sum_{n=0}^{\infty} \frac{1}{n!} \left(\operatorname{diag}(m_1,\dots,m_d)h\right)^n$$

$\dot{\underline{y}} = \underline{\underline{M}}\,\underline{y}$ mit $\underline{\underline{M}}$ eine beliebige $d \times d$-Matrix.

mit Lösung $\underline{y}(t) = e^{\underline{\underline{M}}t}\,\underline{y}_0$ wobei

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

Bsp $\underline{\underline{M}} = \underline{\underline{A}} + \underline{\underline{B}}$

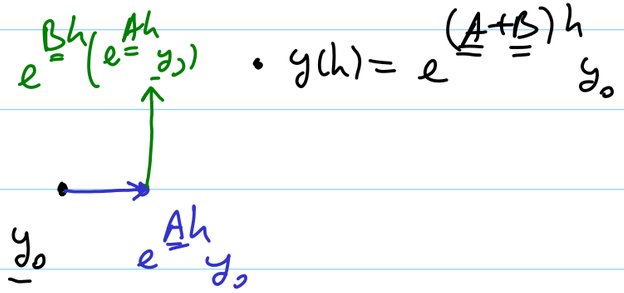
$$e^{(A+B)h} = \underline{\underline{I}} + (\underline{\underline{A}}+\underline{\underline{B}})h + \frac{1}{2}(\underline{\underline{A}}^2 + \underline{\underline{A}}\,\underline{\underline{B}} + \underline{\underline{B}}\,\underline{\underline{A}} + \underline{\underline{B}}^2)h^2 + \dots$$

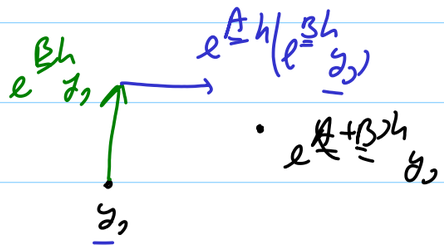
$$e^{\underline{\underline{A}}h}\,e^{\underline{\underline{B}}h} = (\underline{\underline{I}} + \underline{\underline{A}}h + \tfrac{1}{2}\underline{\underline{A}}^2h^2 + \dots)(\underline{\underline{I}} + \underline{\underline{B}}h + \tfrac{1}{2}\underline{\underline{B}}^2h^2 + \dots) =$$

$$= \underline{\underline{I}} + (\underline{\underline{A}}+\underline{\underline{B}})h + (\tfrac{1}{2}\underline{\underline{A}}^2 + \underline{\underline{A}}\,\underline{\underline{B}} + \tfrac{1}{2}\underline{\underline{B}}^2)h^2$$

ungleich wenn $\underline{\underline{A}}\,\underline{\underline{B}} \neq \underline{\underline{B}}\,\underline{\underline{A}}$

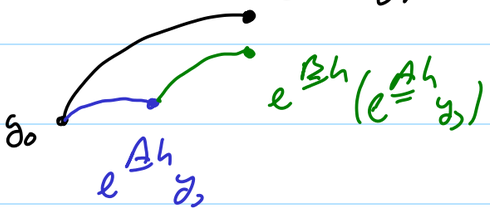
Man kann beweisen dass $e^{(\underline{\underline{A}}+\underline{\underline{B}})h} \approx e^{\underline{\underline{A}}h}\,e^{\underline{\underline{B}}h}$





$$e^{(A+B)h} \approx e^{Ah}\left(e^{Bh}y_0\right)$$

$$e^{(A+B)h} \approx e^{Bh}\left(e^{Ah}y_0\right)$$



$\dot{y} = Ay$ $\quad \dot{y} = By$

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

$\Psi_1^h := \Phi_b^h \circ \Phi_a^h$ Lie-Trotter Splitting

$\tilde{\Psi}_1^h := \Phi_a^h \circ \Phi_b^h$

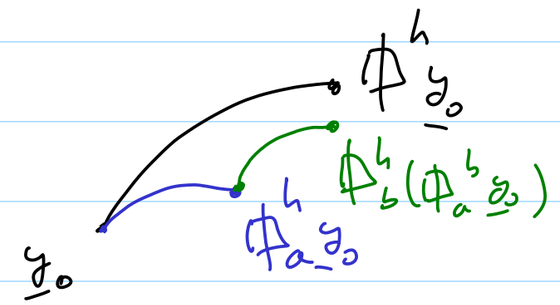
Bem Für nicht-lineare autonome ODE erster Ordnung:

$$\dot{y} = f(y) \quad \text{mit} \quad f(y) = f_a(y) + f_b(y)$$

Idee: Wähle f_a, f_b so dass die ODE:

(a) $\dot{y} = f_a(y)$ und (b) $\dot{y} = f_b(y)$

einfach oder exakt lösbar sind.

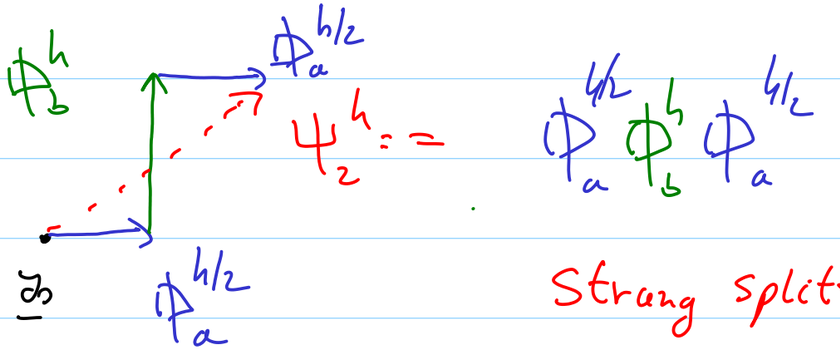


$$\Psi_1^h := \Phi_b^h \circ \Phi_a^h$$

Lie-Trotter Splitting.

Fehler $O(h)$ zur Endzeit T

Bem Symmetrie schenkt uns eine Ordnung mehr:



Strang splitting

Allgemeines Splittingverfahren:

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

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

Bsp $s=1$, $a_1 = b_1 = 1$ 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:

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

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

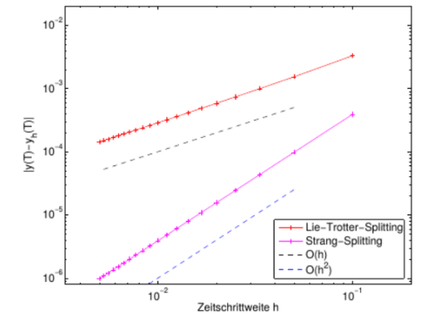
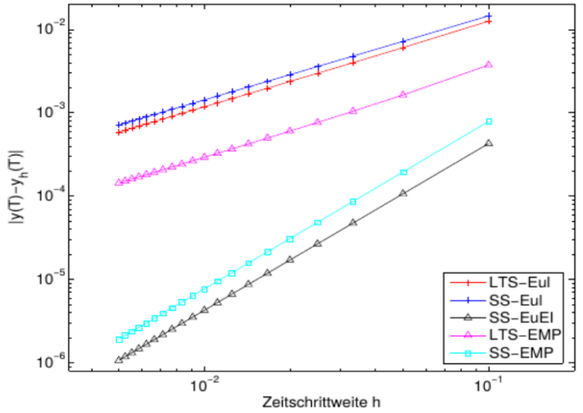


Abb. 2.4.2. Fehler zur Endzeit $T = 1$.



LTS-Eul	explizites Euler als $\Psi_{a,b}^h$, $\Psi_{a,b}^h$ und Lie-Trotter-Splitting
SS-Eul	explizites Euler als $\Psi_{a,b}^h$, $\Psi_{a,b}^h$ und Strang-Splitting
SS-EuEI	Strang-Splitting: explizites Euler als $\Psi_a^{h/2}$, exaktes Φ_b^h und implizites Euler als $\Psi_a^{h/2}$
LTS-EMP	explizite Mittelpunkt-Regel als $\Psi_{a,b}^h$, $\Psi_{a,b}^h$ und Lie-Trotter-Splitting
SS-EMP	explizite Mittelpunkt-Regel als $\Psi_{h,a}^h$, $\Psi_{h,f}^h$ und Strang-Splitting

Bsp Splitting Verfahren für Newton' Gleichung.

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

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

(a) $\begin{cases} \dot{r} = 0 \\ \dot{v} = a(r) \end{cases}$ Löse von Startzeit 0 zu Endzeit h: Exakt!

$\dot{r} = 0 \Rightarrow r(h) = r(0) = r_0$

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

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

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

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

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

kombinieren wir diese 2 exakten Lösungen:

(1) Lie-Trotter-Splitting:

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

$$= \begin{bmatrix} r + h(v + h\,a(r)) \\ v + h\,a(r) \end{bmatrix}$$

symplektische Euler-Verfahren.

(2) Strang-Splitting:

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

$$= \Phi_g^{h/2}\Phi_f^{h}\begin{bmatrix} r+\frac{h}{2}v \\ v \end{bmatrix} = \Phi_g^{h/2}\begin{bmatrix} r+\frac{h}{2}v \\ v+ha(r+\frac{h}{2}v) \end{bmatrix}$$

$$= \begin{bmatrix} r+\frac{h}{2}v+\frac{h}{2}(v+ha(r+\frac{h}{2}v)) \\ v+ha(r+\frac{h}{2}v) \end{bmatrix}$$

Notation:

$$\begin{cases} r_{k+\frac{1}{2}} = r_k + \frac{1}{2}hv_k \\ v_{k+1} = v_k + ha(r_{k+\frac{1}{2}}) \\ r_{k+1} = r_k + \frac{1}{2}hv_{k+1} \end{cases}$$

≡ ein-Schritt-Formulierung des St-V !!

Bem Trick geht auch für separablen Hamilton'system

$$H(p,q) = T(p) + V(q)$$

Lie-Trotter ⇒

$$\begin{cases} p_{n+1} = p_n - h\,\mathrm{grad}\,V(q_n) \\ q_{n+1} = q_n + h\,\mathrm{grad}\,T(p_{n+1}) \end{cases}$$

Symplektische Euler-Verfahren; global $O(h)$.

Strang-Splitting ⇒ (St-V) für (separablen) Hamilton'systeme.

Bem Was tun für zeitabhängige rechte Seiten?

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

Autonomisieren!

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

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

Nicht autonom $\Rightarrow$ autonomisieren

Unbekannte $t \Rightarrow \dot{t} = 1$

$$\underline{u} = \begin{bmatrix} q \\ t \\ p \end{bmatrix} \Rightarrow \dot{\underline{u}} = \underline{f}(\underline{u}) \text{ mit } \underline{f}(\underline{u}) = \begin{bmatrix} p \\ 1 \\ -\frac{g}{l}\sin q + A\cos(\omega t) \end{bmatrix}$$

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

(a) $\dot{q} = 0 \Rightarrow q(h) = q(0) = q_0$

$\dot{t} = 0 \Rightarrow t(h) = t(0) = t_0$

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

$$p(h) = p_0 - \left(\frac{g}{l}\sin q_0\right)h + A\cos(\omega t_0)\,h.$$

exakt.

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

exakt.

(a_c) (b_c) aus Splitting $\xrightarrow{(a) \circ (b)}$ Methode!

Processing

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

Π^h: pre-processor, $(\Pi^h)^{-1}$: post-processor

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

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

Vorteile falls:

+ $\hat\Psi^h$ genauer als Ψ^h

+ Π^h, $(\Pi^h)^{-1}$ günstig

+ keine/wenige Ausgaben der Lösung vor Endzeit gewünscht!

Bsp Strang-Splitting:

$$\Psi_2^h = \Phi_a^{h/2} \circ \Phi_b^h \circ \Phi_a^{h/2} \cdot I = \Phi_a^{h/2} \circ \Phi_b^h \circ \Phi_a^h \circ (\Phi_a^{h/2})^{-1}$$

$\Phi_a^{h/2} \circ (\Phi_a^{h/2})^{-1}$

$\Phi_a^{h/2} \circ \Phi_b^h \circ \Phi_a^h \circ (\Phi_a^{h/2})^{-1}$: Lie-Trotter.

Leicht gestörte Probleme

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

optimierte Splitting-Verfahren, klar

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

mit $r_1 \geq r_2 + 1 \geq \dots$

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

§4 Runge-Kutta-Verfahren

§4.1. Grundidee

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

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

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

QF mit Gewichten b_i, Knoten $c_i \in [0,1] \Rightarrow$

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

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

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

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

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

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

Idee: verwende etwas billigeres für $\underline{y}(t_1)$

$$\underline{y}(t_0 + h) \approx \underline{y}_0 + h \underbrace{f(t_0, \underline{y}_0)}_{k_1} \qquad (eE)$$

$$\begin{cases} \underline{k}_1 := \underline{f}(t_0, \underline{y}_0) \\ \underline{k}_2 := \underline{f}(t_0+h, \underline{y}_0 + h\underline{k}_1) \\ \underline{y}_1 = \underline{y}_0 + h\cdot\frac{1}{2}\underline{k}_1 + h\,\frac{1}{2}\underline{k}_2 \end{cases}$$

explizite Trapezregel

Bspl QF = Mittelpunktsregel

$$\underline{y}_1 = \underline{y}_0 + h\,\underline{f}(t_0 + h\tfrac{1}{2},\ y(t_0 + h\tfrac{1}{2}))$$

(eE): $\underline{y}(t_0 + h\frac{1}{2}) = \underline{y}_0 + h\,\frac{1}{2}\,\underline{f}(t_0, \underline{y}_0)$

$\underline{f}(t_0, \underline{y}_0) = \underline{k}_1$

$$\begin{cases} \underline{k}_1 := f(t_0, \underline{y}_0) \\ \underline{k}_2 = f(t_0 + \frac{h}{2},\ y_0 + \frac{h}{2} k_1) \\ \underline{y}_1 = \underline{y}_0 + h k_2 \end{cases}$$

explizite Mittelpunktsregel.

(E) $\Rightarrow$ implizite MPR ...

Def Runge-Kutta-Verfahren mit s Stufen:

Gegeben Butcher-Schema

c_1 c_2 $\vdots$ c_s	$\underline{A} \in \mathbb{R}^{s\times s}$
1	$b_1\ b_2 \dots\ b_s$

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

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

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

ein $(n\cdot d)\times(n\cdot d)$ nichtlinearen algebraischen Gleichungssystem

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

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

$a_{ij} = 0$ für alle $i \le j$

$\Rightarrow$ RK explizit

$$\underline{k}_i := \underline{f}\left(t_0 + c_i h,\ y_0 + h \sum_{j=1}^{i-1} a_{ij} \underline{k}_j\right)$$

$$\underline{\underline{A}} = \begin{bmatrix} * & & 0 \\ & \ddots & \\ * & & * \end{bmatrix} \Rightarrow$$ diagonal implizite RK

$$\underline{k}_i := \underline{f}\left(t_0 + c_i h,\ y_0 + h \sum_{j=1}^{i-1} a_{ij} \underline{k}_j + h a_{ii} \underline{k}_i\right)$$

↳ $d \times d$ nicht-lin. alg. Gleichung

$\underline{y} \in \mathbb{R}^d$, $\underline{k}_i \in \mathbb{R}^d$, $f : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$

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

Def Konsistenzordnung q wenn

lokale Fehler $\|\underline{y}(t_0 + h) - \underline{z}_1\| \le c \cdot h^{q+1}$

Theorem

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

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

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

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

$$\le c \cdot h^{q+1} \Rightarrow$$

Voraussetzung

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

$$\left| \frac{1}{n+1} - \sum_{j=1}^{s} b_j c_j^n \right| \le c \cdot h^{q-n} \xrightarrow[h \to 0]{} 0 \Rightarrow$$

solange $q > n$

Für $n=0,1,2,\dots,q-1$:

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

$\Longleftrightarrow$ QF exakt für $p_n(t) = t^n$

Konsequenz RK mit s Stufen $\Rightarrow$ max. Konsistenzordnung $2s$

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

2) RK hat mindestens Konsistenzordnung $q=2$

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

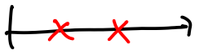
3) $q=3$ brauchen wir

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

usw.

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

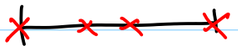
Gauss-Quadratur $\Rightarrow$ $q = 2s$



Radau-Quadratur

↳ Radau-Verfahren für ODEs.

Lobatto-Quadratur



↳ Lobatto-Verfahren für ODEs.

Theorem

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

$$\| \underline{y}(t_i) - \underline{y}_i \| \le C \cdot h^q \quad \text{für alle } i=1,2,\dots,N$$

$(T = Nh)$

§4.2. Kollokation

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

Kollokationspolynom $u(t)$ von Grad s:

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

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

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

mit k so bestimmt 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 $c_1, \dots, c_s$

$\Updownarrow$

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

$b_i := \int_0^1 l_i(\tau)\,d\tau$

wobei

$$l_i(\tau) = \frac{(\tau - c_1)(\tau - c_2)\cdots\big|\cdots(\tau - c_s)}{(c_i - c_1)(c_i - c_2)\cdots\big|\cdots(c_i - c_s)} = \prod_{\substack{j=1 \\ j \neq i}}^{s} \frac{\tau - c_j}{c_i - c_j}$$

Lagrange Polynom.

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

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

$$\dot{u}(t_0 + \tau h) = \sum_{j=1}^{p} k_j \, l_j(\tau)$$

hat Grad $p-1$

$\int_0^{c_i} \Rightarrow$

$$u(t_0 + c_i h) = y_0 + h \sum_{j=1}^{p} k_j \int_0^{c_i} l_j(\tau)\,d\tau$$

zwischen stellen.

a_{ij}

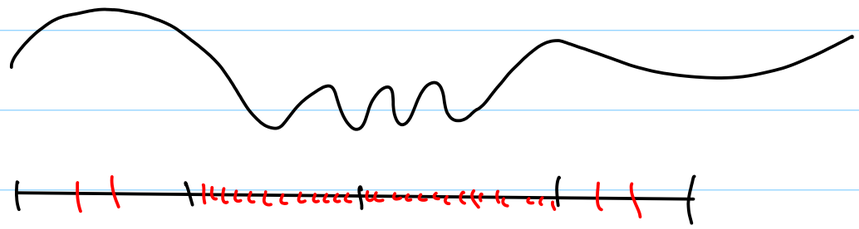
$\int_0^1 \Rightarrow$ $u(t_0 + h) = y_0 + h \sum_{j=1}^{p} k_j \int_0^1 l_j(\tau)\,d\tau$

$\| \; \vdots \; y_1$

b_j

Konsequenz Kollokationsmethode hat dieselbe Ordnung wie die entsprechende QF.

§4.3. Adaptivität



lokaler Fehler abschätzen:

$\Phi - \Psi_h \Rightarrow \tilde{\Psi}_h - \Psi_h$ Schätzung des Fehlers.

$u \longrightarrow O(h^{p+1})$

Ψ_h genauer $\longrightarrow O(h^{p+2})$

$$est_k = \left| \tilde{\Psi}^{t_k, t_k+h}(y(t_k)) - \Psi^{t_k, t_k+h}(y(t_k)) \right| \approx ch^{p+1} = toL$$

$$h^* = h \sqrt[p+1]{\frac{toL}{est_k}}$$

§4.4. Partitionierte RK-Verfahren

System ODE partioniert:

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

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

für $\underline{y}$ $\quad \begin{array}{c|c} \underline{c} & \underline{\underline{A}} \\ \hline 1 & \underline{b} \end{array}$ $\qquad$ für $\underline{z}$ $\quad \begin{array}{c|c} \hat{\underline{c}} & \hat{\underline{\underline{A}}} \\ \hline 1 & \hat{\underline{b}} \end{array}$

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

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

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

Bsp 1)

(eE): $b_1 = 1,\ a_{11} = 1$
(iE) $\quad b_1 = 1,\ \hat{a}_{11} = 0$

] für Newton' Gleichung
⇒ sympleketische Euler-Verfahren.

Bsp 2)

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

PRK für Newton' Gleich. ⇒ Störmer-Verlet

Verallgemeinerung vom Störmer-Verlet:

3-Stufige Lobatto-Paar

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{5}{24} & \frac{1}{3} & -\frac{1}{24} \\ 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array} \qquad \begin{array}{c|ccc} 0 & \frac{1}{6} & -1/6 & 0 \\ \frac{1}{2} & \frac{1}{6} & 1/3 & 0 \\ 1 & 1/6 & 5/6 & 0 \\ \hline 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

$O(h^4)$

Bem Newton' Gleichung.

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

Umschreiben:

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

PRK $\Rightarrow$ RK-Nyström-Verfahren (RKN)

$$\begin{cases} \underline{l}_i = g(c_i h, \underline{y}_0 + c_i h \underline{z}_0 + h^2 \sum_{j=1}^{s} \bar{a}_{ij} \underline{l}_j, \underline{z}_0 + h \sum_{j=1}^{s} \hat{a}_{ij} \underline{l}_j) \\ \underline{y}_1 = \underline{y}_0 + h(\underline{z}_0 + h \sum_{i=1}^{s} \bar{b}_i \underline{l}_i) \quad \text{mit } \bar{b}_i = \sum_{k=1}^{s} b_k \hat{a}_{ki} \\ \underline{z}_1 = \underline{z}_0 + h \sum_{i=1}^{s} \hat{b}_i \underline{l}_i \end{cases}$$

aus den $\underline{k}_i = \dots$

$$\bar{a}_{ij} = \sum_{k=1}^{s} a_{ik} \hat{a}_{kj}$$

Wenn g nicht von $\dot{y}$ abhängt $\Rightarrow$ braucht man $\hat{a}_{kj}$ nicht.

Bem PRK = Splitting mit

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

$\Rightarrow$ klar!

einfacher anzuwenden!

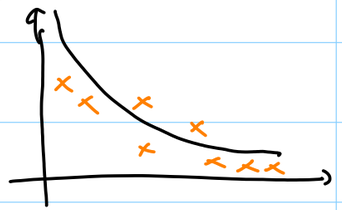
BM42 mit $O(h^4)$

BM63 mit $O(h^6)$

$\Big\}$ symplektische Verfahren.

§5 Steife Differentialgleichungen

§5-1. Einführung



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

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

Interesse: asymptotisches Verhalten der numerischen Lösung sollte qualitativ (zumindest) ähnlich der exakten Lösung sein.

(eE):

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

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

...

$y_N = (1-\lambda h)^N y_0$

$|y_N| \to 0$ nur wenn $|1-\lambda h| < 1 \Leftrightarrow 0 < h < \frac{2}{\lambda}$

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

Bsp

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

$\underline{\dot{y}} = \underline{\underline{B}}\,\underline{y}$

$\underline{\underline{B}}$ symmetrisch $\Rightarrow$ es gibt $\underline{\underline{S}}$ (mit $\underline{\underline{S}}\,\underline{\underline{S}}^T = \underline{\underline{I}}$) dass

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

mit $\underline{\underline{D}}$ = Diagonalmatrix.

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

$\underline{\underline{S}}^T |$

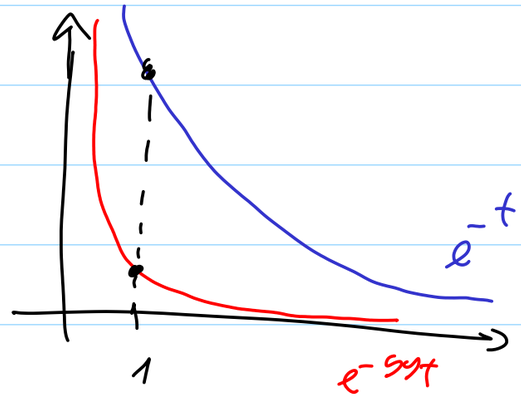
$$\begin{cases} \dot{z}_1 = \lambda_1 z_1 \\ \dots \\ \dot{z}_d = \lambda_d z_d \end{cases}$$

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

$\dot{z}_1 = -z_1 \Rightarrow \quad h < 2$

$\dot{z}_2 = -99 z_2 \Rightarrow \quad h < \frac{2}{99}$

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



zur Zeit $t=1$: $y_1(1) = e^{-1} + e^{-99}$

sehr klein, irrelevant!

aber ein. expl. Verfahren will ein kleines h

↳ bestimmt durch e^{-99t}

(i.E.) $y_1 = y_0 + h f(z_1) = y_0 - \lambda h y_1 \Rightarrow$

$(1+\lambda h) y_1 = y_0 \Rightarrow y_1 = \frac{1}{1+\lambda h} y_0 \Rightarrow y_N = \left(\frac{1}{1+\lambda h}\right)^N y_0 \to 0$

$h>0, \lambda>0$ $\qquad N \to \infty$

Bem Implizite Verfahren stellen keine Bedingung an h.

Bsp explizite Trapezregel: $(\dot{y} = -\lambda y)$

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

$k_1 = -\lambda y_0 - \lambda^2 h y_0$

$y_1 = y_0 + \frac{1}{2} h k_1 + \frac{1}{2} h k_2 = y_0 + (-\lambda h) y_0 + \frac{(\lambda h)^2}{2} y_0$

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

$y_N = (S(\lambda h))^N y_0 \to 0$ nur wenn $|S(\lambda h)| < 1$

§5.2. Stabilität des RK-Verfahrens

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

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

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

$\lambda = -1+5i \Rightarrow y(t) = e^{-t} y_0(\cos 5t + i \sin 5t) \to 0$ $t \to \infty$

Numerisches Verfahren $y_N = S(\lambda h)^N y_0$

Frage: wann $|y_N| \to 0$ für $N \to \infty$?

$S(z)$ = Stabilitätsfunktion.

(eE) $S(z) = 1 + z$

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

(eTR) $S(z) = 1 + z + \frac{1}{2} z^2$

(da wir jetzt $\dot{y} = +\lambda y$ notierten, in §5.1 hatte ich $\dot{y} = -\lambda y$)

RK-Verfahren mit s Stufen:

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

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

$$\Rightarrow \begin{cases} k_i = \lambda y_0 + \underbrace{\lambda h}_{z} \sum_{j=1}^{s} a_{ij} k_j \\ i = 1, 2, \dots, s \end{cases} \qquad \underline{k} = \begin{bmatrix} k_1 \\ \vdots \\ k_s \end{bmatrix}$$

$$\Rightarrow \underline{k} = \lambda y_0 \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + z \underline{\underline{A}}\, \underline{k} \quad \Longleftrightarrow$$

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

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

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

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

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

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

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

Stabilitätsfunktion des s-stufiges RkV.

$y_1 = S(z)\, y_0$ mit $z = \lambda h$

$|y_N| \longrightarrow 0$ für $N \to \infty$ nur wenn $|S(z)| < 1$

$|y_N| = |y_0|$ falls $|S(z)| = 1$

Theorem Die Stabilitätsfunktion eines s-stufiges RkV ist eine (komplexwertige) rationale Funktion

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

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

Falls RkV explizit, dann $Q(z) \equiv 1$.

Beweis explizite RkV: $\underline{\underline{A}} = \begin{bmatrix} 0 & \cdots & 0 \\ * & \ddots & 0 \end{bmatrix}$

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

$$\left(\underline{\underline{I}} - z\underline{\underline{A}}\right)^{-1} = \underline{\underline{I}} + z\underline{\underline{A}} + (z\underline{\underline{A}})^2 + (z\underline{\underline{A}})^3 + \ldots + \underbrace{(z\underline{\underline{A}})^s}_{0} + \underbrace{\ldots}_{0}$$

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

= Polynom vom Grad $s-1$ in z.

$\underline{v} = (\underline{I} - z\underline{A})^{-1} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \Rightarrow \quad v_i(z) = \dfrac{P_D(z)}{\det(\underline{I} - z\underline{A})}$

↑ Cramer Regel.

$Q(z) = 0 \iff \det(\underline{I} - z\underline{A}) = 0 \iff z = \dfrac{1}{\mu}$ mit $\mu \in EW$ von $\underline{A}$.

Konsequenz

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

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

mit wohldefinierten Stabilitätsfunktion $S(z)$.

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

num. Lösung $y_n = S(\lambda h)^n$

RKV hat Konvergenzordnung q:

Fehler $|e^{nh\lambda} - S(\lambda h)^n| \leq O(h^{q+1})$

Somit ist $S(z)$ eine Approximation an e^z

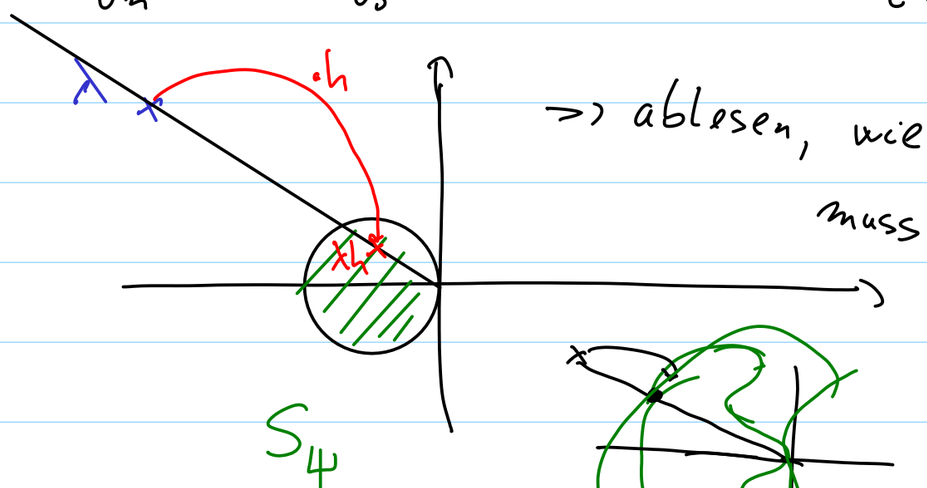
$|e^z - S(z)| \leq O(|z|^{q+1})$

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

Definition Stabilitätsgebiet des num. 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 $z \in S_\Psi$

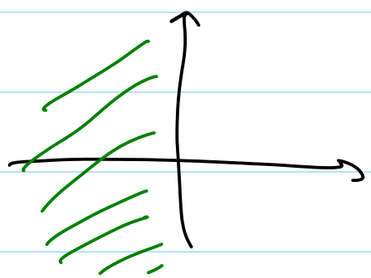


⇒ ablesen, wie klein muss h gewählt werden!

Bem RK explizit $\Rightarrow$ $S(z)$ = Polynom ist $\Rightarrow$
Stabilitätsgebiet beschränkt
$\Rightarrow$ immer eine Schranke an h.

Bem
oder / dopri5 verwenden explizite RK
$\Rightarrow S_\psi$ beschränkt $\Rightarrow$ brauchen kleines h.

Def Ein Verfahren heisst A-stabil falls
$\{z \in \mathbb{C};\ \mathrm{Re}\, z < 0\} \subset S_\psi$



Bsp (iE) ist A-stabil
RK-Gauss Verfahren sind A-stabil.
(z.B. iMP)

Bem $S(z) \approx e^z$; $z \to -\infty \Rightarrow e^z \to 0$
Frage $S(-\infty) = 0$?

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

Bem L-stabil falls $\underline{b}^T = a_{s\cdot}$ = letzte Zeile in $\underline{A}$

$\underline{c}$	$\underline{A}$
1	$\underline{b}^T$
	$\underline{b}^T$

$c_s = 1$ ⇓

$\Rightarrow$ Radau-Verfahren (basieren auf Radau-Quadratur) sind L-stabil und höchste Ordnung haben. $(2s-1)$

Radau-Verfahren von Ordnungen 3, 5 $\to$ Skript

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

$\underline{y}^* \in \mathbb{R}^d$ heisst stationär $\Leftrightarrow \underline{f}(\underline{y}^*) = 0$

linearisiere (Taylor um y^* konf)

$\dot{\underline{z}} = \underline{Df}(\underline{y}^*)\,\underline{z}$ Testproblem!

Bem implizite Rk: in jedem Zeitschritt ein nicht-lineares algebraisches System mit $d \cdot s$ Gleichungen lösen (s Stufen $\underline{k}_1, \dots, \underline{k}_s \in \mathbb{R}^d$)

TEUER mit fsolve $F(\underline{x}) = 0$.

Bem Autonome ODE, in Rk:

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

Idee: linearisiere: ↓ ersetzt durch.

$$\underline{k}_i = \underline{f}(\underline{y}_0) + h \underline{\underline{D}}\underline{f}(\underline{y}_0)\left(\sum_{j=1}^{s} a_{ij} \underline{k}_j\right)$$

$\underline{k}_i \in \mathbb{R}^d$, $\underline{f}(\underline{y}_0) \in \mathbb{R}^d$, $\underline{\underline{D}}\underline{f}(\underline{y}_0) \in \mathbb{R}^{d \times d}$, $\sum_{j=1}^{s} a_{ij} \underline{k}_j \in \mathbb{R}^d$

ein grosses LGS $\mathbb{R}^{ds \times ds}$.

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

Skript: man verliert eine Konvergenzordnung!

(Newton: Ordnung 2)

(linearisierte Rk: Ordnung 1)

Radau ($s=2$) + Newton $\Rightarrow$ $O(h^3)$

Radau ($s=2$) + Linearisierung $\Rightarrow$ $O(h^2)$

Idee: Verwende ein Schritt Newton für

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

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

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

ein Schritt Newton für $\underline{F}$ mit Startwert $z_0 = y_0$

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

$$\underline{y}_1 = \underline{y}_0 + \left[\underline{\underline{I}} - \underline{\underline{Df}}(z_0)\right]^{-1} h\underline{f}(z_0)$$

Wenn wir aber einen besseren Startpunkt für Newton haben, dann vielleicht auf bessere Konvergenz.

Ben Ordnung retten geht bei diagonal-impliziten Verfahren.

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

$$\left.\begin{aligned} \underline{k}_i &= f\Big(\underline{y}_0 + h\sum_{j=1}^{i} a_{ij}\,\underline{k}_j\Big) \\ i &= 1,2,\dots,s \end{aligned}\right\}$$

$$\underline{F}(\underline{k}) = \underline{k} - f\Big(\underline{y}_0 + \underline{z} + h a_{ii}\,\underline{k}\Big) \qquad \underline{k} \in \mathbb{R}^d$$

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

Newton-Schritt:

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

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.

$$\Big(\underline{\underline{I}} - h a_{ii}\,\underline{\underline{\partial}}\Big)\underline{k}_i = f\Big(\underline{y}_0 + h\sum_{j=0}^{i-1}(a_{ij}+d_{ij})\underline{k}_j\Big) - h\,\underline{\underline{\partial}}\sum_{j=1}^{i-1} d_{ij}\,\underline{k}_j$$

mit $\underline{\underline{J}} = \underline{\underline{D}}f\left(\underline{y}_0 + h\sum_{j=1}^{i-1}(a_{ij}+d_{ij})\underline{k}_j\right)$

wie bei vereinfachten Newton

können wir darauf verzichten

→ günstiger.

⇒ Row2 / Row3 → adaptiven impliziten Verfahren.

ode23s

§6 Nichtlineare algebraische Gleichungen

§6.1. Einführung

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

Bsp

$d=1 \quad F(x) = xe^x - 1$

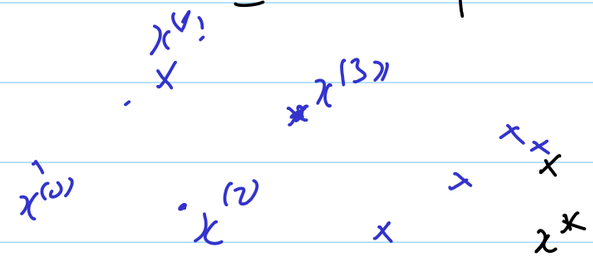
Man baut eine Folge von Approximationen an $\underline{x}^*$

$\underline{x}^{(0)}, \underline{x}^{(1)}, \underline{x}^{(2)} \dots, \underline{x}^{(k)} \longrightarrow \underline{x}^*$

$\uparrow$

Gegeben

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



Def $\underline{x}^{(k+1)} = \Phi(\underline{x}^{(k)})$ heisst linear konvergent nach $\underline{x}^*$ falls

es gibt $L<1$ so dass

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

Bem $\|\underline{x}^{(k+1)} - \underline{x}^*\| \leq L \|\underline{x}^{(k)} - \underline{x}^*\| \leq$

$\leq L^2 \|\underline{x}^{(k-1)} - \underline{x}^*\| \leq \dots$

$\leq L^{k+1} \|\underline{x}^{(0)} - \underline{x}^*\|$ $\Big\}$ $\Rightarrow$ konvergenz

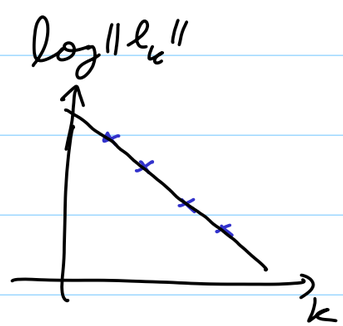
$0 < L < 1$

$\underline{x}^{(k+1)} \to \underline{x}^*$ in $\|\cdot\|$.

Fehler $\underline{e}_k = \underline{x}^{(k)} - \underline{x}^*$

$\log \|\underline{e}_k\| \leq k \log L + \log \|\underline{e}_0\|$

Gerade mit Steigung $\log L$.

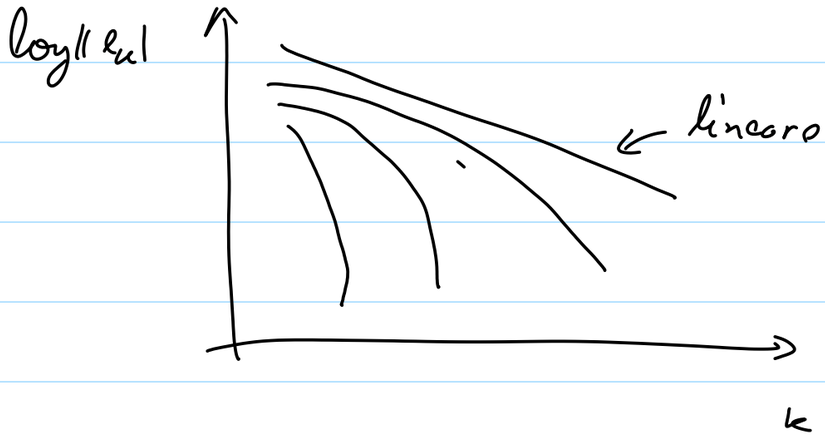


Def konvergent der Ordnung p des iterativen Verfahrens,

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

$F(x^*)=0 \Leftrightarrow x^* = \Phi(x^*)$ Fixpunkt von Φ

Falls (x^k) konvergent $\Rightarrow x^*$ $\quad \Phi(x^*) \Rightarrow x^*$ Lösung

Bsp 1) $xe^x - 1 = 0 \Leftrightarrow xe^x = 1 \Rightarrow x = e^{-x}$

$$\Phi_1(x) = e^{-x}$$

Starte mit $x^{(0)}$, $x^{(k+1)} = \Phi_1(x^{(k)})$

2) $xe^x - 1 = 0 \Leftrightarrow xe^x - 1 + x = x \Leftrightarrow x(e^x+1) = x+1 \Leftrightarrow$

$$\Leftrightarrow x = \frac{x+1}{e^x+1}$$

$$\Phi_2(x) = \frac{x+1}{e^x+1}$$

Starte mit $x^{(0)}$, $x^{(k+1)} = \Phi_2(x^{(k)})$

3) $x e^x - 1 = 0 \Leftrightarrow x e^x - 1 - x = -x \Leftrightarrow x = x + 1 - x e^x$

$\Phi_3(x) = x + 1 - x e^x$

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 ↓ quadratische ↓ keine

k	$\lvert x_1^{(k+1)} - x^* \rvert$	$\lvert x_2^{(k+1)} - x^* \rvert$	$\lvert x_3^{(k+1)} - x^* \rvert$
0	0.067143290409784	0.067143290409784	0.067143290409784
1	0. 039387369302849	0. 000832287212566	0.108496074240152
2	0. 021904078517179	0. 000000125374922	0.219330611898582
3	0. 012559804468284	0. 000000000000003	0.288178118764323
4	0. 007078662470882	0. 000000000000000	0.723649245792953
5	0. 004028858567431	0. 000000000000000	0.410183132337935
6	0. 002280343429460	0. 000000000000000	1.186907542305364
7	0. 001294757160282	0. 000000000000000	0.146569797006362
8	0. 000733837662863	0. 000000000000000	0.310516641279937
9	0. 000416343852458	0. 000000000000000	0.357777386500765
10	0. 000236077474313	0. 000000000000000	0.974565695952037

Bem Hinreichende Bedingung für lokale lineare Konvergenz

U konvex, $\Phi: U \subset \mathbb{R}^d \to \mathbb{R}^d$ stetig differentierbar

$$L = \sup_{x \in U} \| D\Phi(x) \| < 1$$

Wenn $\Phi(x^*) = x^*$ für $x^* \in U$, dann

konvergiert $x^{(k+1)} = \Phi(x^{(k)})$ gegen x^* lokal mindestens linear.

Def $\Phi: U \subset \mathbb{R} \to \mathbb{R}$ Φ $(m+1)$-mal stetig diff'bar

$\Phi(x^*) = x^* \in \bar{U}$

Taylor: $\Phi(y) = \Phi(x) + \sum_{k=1}^{m} \frac{1}{k!} \Phi^{(k)}(x)(y-x)^k + O(|y-x|^{m+1})$

Theorem Falls $\Phi^{(l)}(x^*) = 0$ für $l = 1, 2, \dots, m \geq 1$

dann konvergiert die Fixpunktiteration

$x^{(k+1)} = \Phi(x^{(k)})$ gegen x^* lokal

mit der Ordnung $p \geq m+1$.

Beweis Taylor für $x = x^*$, $y = x^{(k)}$ $\Rightarrow$

$$x^{k+1} - x^* = \Phi(x^k) - \Phi(x^*) = \sum_{j=1}^{m} \frac{1}{j!} \underbrace{\Phi^{(j)}(x^*)}_{=0} (x^{(k)} - x^*)^j + O(|x^{(k)} - x^*|^{m+1})$$

$$\Rightarrow |x^{k+1} - x^*| \leq C |x^{(k)} - x^*|^{m+1}$$

Bsp $\Phi_2(x) = \frac{x+1}{e^x+1}$

$$\Phi_2'(x) = \frac{e^x + 1 - (x+1)e^x}{(e^x+1)^2} = \frac{1 - xe^x}{(e^x+1)^2} \Rightarrow$$

$$\Phi_2'(x^*) = \frac{1 - x^* e^{x^*}}{(e^{x^*}+1)^2} = 0$$ für x^* die Nullstelle von $xe^x - 1$.

$\Rightarrow$ Φ_2 gibt mindestens Konvergenzordnung 2.

§6.3. Abbruchkriterium

Wann brechen wir eine Iteration ab?

1) nach k_{max} Schritte: "a priori" Abbruchskriterium.

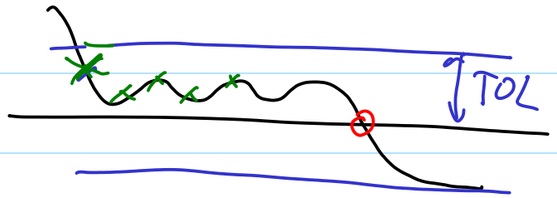
2) "a posteriori" ideal: $\|\underline{x}^{(k)} - \underline{x}^*\| \leq TOL$ ☹

$\|\underline{x}^{(k+1)} - \underline{x}^{(k)}\| < TOL \Rightarrow$ Abbruch

oder verwende Residuum

$\|F(\underline{x}^{(k)})\| < TOL \Rightarrow$ Abbruch

Gefahr:



Wenn wir Informationen über die Konvergenzart der Iteration, dann bessere Kriterien!

Bsp lineare Konvergenz mit der Rate L

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

$\pm \underline{x}^{(k+1)}$

$$\Rightarrow (1-L)\|\underline{x}^{(k+1)} - \underline{x}^*\| \leq L\|\underline{x}^{(k)} - \underline{x}^{(k+1)}\| \Rightarrow$$

$$\|\underline{x}^{(k+1)} - \underline{x}^*\| \leq \frac{L}{1-L}\|\underline{x}^{(k)} - \underline{x}^{(k+1)}\| \quad \text{da } 0 < L < 1$$

$\Rightarrow$ Abbruchkriterium:

$$\|\underline{x}^{(k)} - \underline{x}^{(k+1)}\| \leq \frac{1-L}{L} \cdot TOL \Rightarrow \|\underline{x}^{(k+1)} - \underline{x}^*\| \leq TOL$$

Abbruch!

Falls L unbekannt, verwende ein geschätztes $\Sigma > L$

Bem 1) Bei der Berechnung. Fehler $\varepsilon_k = \|\underline{x}^{(k)} - \underline{x}^*\|$

$$\varepsilon_{k+1} \approx L\varepsilon_k \Rightarrow \log \varepsilon_{k+1} \approx \log L + \log \varepsilon_k$$

$$\Rightarrow \log L \approx \frac{\log \varepsilon_{k+1}}{\log \varepsilon_k}$$

oder. $$\approx \frac{\|x^{(k)} - x^{(N)}\|}{\|x^{(k-1)} - x^{(N)}\|}$$

2) Konvergenzordnung ρ aus Experiment erraten:

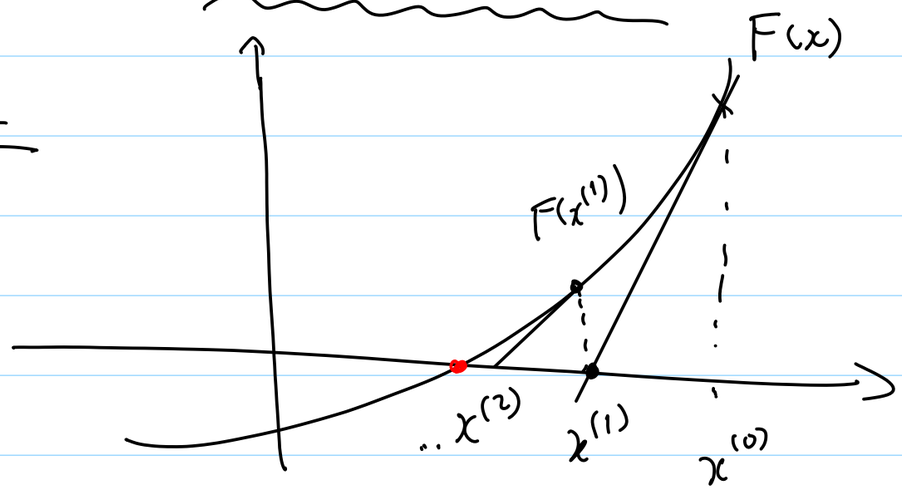
$$\varepsilon_{k+1} \approx C \varepsilon_k^{\rho} \Rightarrow \log \varepsilon_{k+1} \approx \log C + \rho \log \varepsilon_k$$
$$\log \varepsilon_k \approx \log C + \rho \log \varepsilon_{k-1} \quad \Big\} \Leftrightarrow$$

$$\Rightarrow \log \varepsilon_{k+1} - \log \varepsilon_k \approx \rho (\log \varepsilon_k - \log \varepsilon_{k-1})$$

$$\Rightarrow \rho \approx \dots$$

§ 6.4 Newton-Verfahren

Idee



Ersetze die komplizierte Funktion F durch eine einfachere, z.B. durch eine lineare Funktion ($\Rightarrow$ Linearisierung!)

Linearisiere lokal! $\Rightarrow$ Tangente + Neustart

$F \approx \tilde{F}$ mit $\tilde{F}(x)$ einfacher

Newton

$$\underline{\tilde{F}}(\underline{x}) = \underline{F}(\underline{x}^0) + \underline{\underline{DF}}(\underline{x}^{(0)})(\underline{x} - \underline{x}^{(0)})$$

statt $F(x)=0$ löse $\underline{\tilde{F}}(\underline{x}) = 0$ $\Leftrightarrow$

$$\underline{F}(\underline{x}^0) + \underline{\underline{DF}}(\underline{x}^{(0)})(\underline{x} - \underline{x}^{(0)}) = 0 \quad \Leftrightarrow$$

{ LGS in Unbekannte $\underline{x}$

LGS: $\underline{\underline{DF}}(\underline{x}^{(0)})\,\underline{x}^{(1)} = -\underline{F}(\underline{x}^{(0)}) + \underline{\underline{DF}}(\underline{x}^{(0)})\,\underline{x}^{(0)}$

Und starte von diesen $x^{(1)}$ wieder $\Rightarrow$

Newton-Iteration:

gegeben $\underline{x}^{(0)}$

für $k = 0, 1, 2, \dots$:

$$\underline{x}^{(k+1)} = \underline{x}^{(k)} - \underline{\underline{DF}}(\underline{x}^{(k)})^{-1}\,\underline{F}(\underline{x}^{(k)})$$

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 $d=1$ $\quad x^{(k+1)} = x^{(k)} - f'(x^{(k)})^{-1} f(x^{(k)})$

Das ist eine Fixpunktiteration

$$x^* = \Phi(x^*) \text{ mit } \Phi(x) = x - f'(x)^{-1} f(x)$$

Newton Iteration für $f(x^*)=0$

$$\Phi'(x) = 1 - \frac{f'(x)f'(x) - f(x)f''(x)}{(f'(x))^2} = 1 - 1 + \frac{f(x)f''(x)}{(f'(x))^2} \Rightarrow$$

$$\Rightarrow \Phi'(x^*) = \frac{f(x^*)f''(x^*)}{(f'(x^*))^2} = 0 \quad \text{falls } f'(x^*) \neq 0. \Rightarrow p=2.$$

* sympy + lambdify

* Wahl vom Startwert + gedämpftes Newton-Verfahren.

* vereinfachtes Newton-Verfahren.

Bem In jeden Schritt muss man ein LGS lösen:

$$\underline{x}^{(k+1)} = \underline{x}^{(k)} - \underline{\underline{D}} f(\underline{x}^{(k)})^{-1} \underline{f}(\underline{x}^{(k)})$$

$$\underline{x}^{(k+1)} = \underline{x}^{(k)} - \underline{\Delta}^{(k)} \qquad \underline{\Delta}^{(k)} = \underline{x}^{(k+1)} - \underline{x}^{(k)}$$

$$\underline{\underline{D}} f(\underline{x}^{(k)})\, \underline{\Delta}^{(k)} = \underline{f}(\underline{x}^{(k)}) \qquad \underline{f}(\underline{x}^*) = 0$$

$d \times d$

$\underline{\underline{D}} f(\underline{x}^{(k)}) \in \mathbb{R}$ → teuer nach k Schritten

$\underline{\underline{\partial}}(\underline{x}^{(k)}) = \underline{\underline{D}} f(\underline{x}^{(k)})$ $\qquad \underline{\underline{\partial}} = \underline{\underline{L}}\,\underline{\underline{U}}$ kostet $O(d^3)$ Schritte

Idee: Wiederverwendung von $\underline{\underline{J}}$ für mehrere Newton-Schritte

$\underline{\underline{J}} = \underline{\underline{L}}\,\underline{\underline{U}}$ ein mal, dann verwende die Faktoren $\underline{\underline{L}}$ und $\underline{\underline{U}}$ auch für die nächsten Schritte

(0) $k = 1, 2, 3, 4.$

Vereinfachte Newton:

$\underline{\underline{J}} = \underline{\underline{D}} f(\underline{x}^{(0)})$ → $O(d^3)$

$\underline{\underline{J}} = \underline{\underline{L}}\,\underline{\underline{U}}$ $\left(\underline{\underline{L}}, \underline{\underline{U}} = \text{Lu\_factor}(\underline{\underline{J}})\right)$

für $k = 0, 1, 2, \ldots$

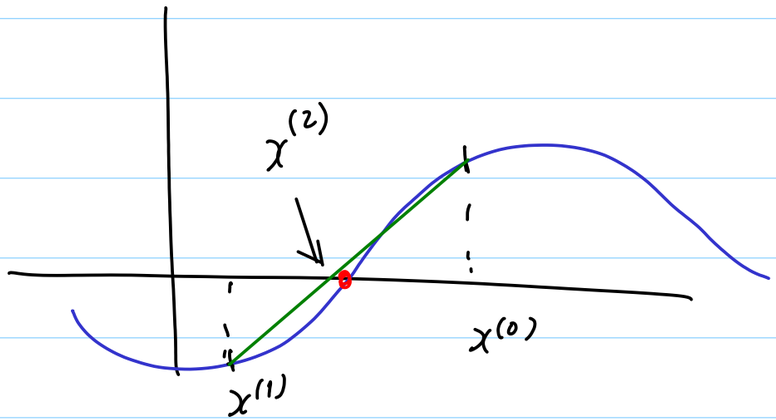
löse $\underline{\underline{L}}\,\underline{\underline{U}}\,\underline{\Delta} = \underline{f}(\underline{x}^{(k)})$ nur $O(d^2)$

$\underline{x}^{(k+1)} = \underline{x}^{(k)} + \underline{\Delta}^{(k)}$

Typischerweise bekommt man nur lineare Konvergenz

Bem Was tun wenn $\underline{\underline{D}} f(\underline{x})$ nicht bekannt?

1D



Idee: verwende statt Tangente eine andere Gerade, z.B. eine Sekante.

Brauchen jetzt 2 Startwerte: $x^{(0)}, x^{(1)}$

$$f'(x^{(k)}) \approx \frac{f(x^{(k)}) - f(x^{(k-1)})}{x^{(k)} - x^{(k-1)}}$$

update: $x^{(k+1)} = x^{(k)} - s$ mit

$$s = \frac{x^{(k)} - x^{(k-1)}}{f(x^{(k)}) - f(x^{(k-1)})} f(x^{(k)})$$

Allgemein: d Dimensionen:

$$\underline{\underline{J}}_k \left(\underline{x}^{(k)} - \underline{x}^{(k-1)} \right) = f(x^{(k)}) - f(x^{(k-1)})$$

$\underline{\underline{J}}_k$: $d \times d$ — $\underline{x}^{(k)} - \underline{x}^{(k-1)} \in \mathbb{R}^d$

Bem Viele mögliche $\underline{\underline{J}}_k$ erfüllen das!

Broyden: baue $\underline{\underline{J}}_k$ iterativ

$$\underline{\underline{J}}_k = \underline{\underline{J}}_{k-1} + \frac{1}{\| x^{(k)} - x^{(k-1)} \|_2^2} f(x^{(k)}) \left(\underline{x}^{(k)} - \underline{x}^{(k-1)} \right)^T$$

Vorteil: man muss f nur ein Mal auswerten!

Broyden-Verfahren:

$\underline{x}^{(0)}$ gegeben

$\underline{\underline{J}}_0 = \underline{\underline{D}} f(x^{(0)})$

für $k = 0, 1, 2, \ldots$:

Löse $\underline{\underline{J}}_k \underline{s} = f(\underline{x}^{(k)})$ teuer

$\underline{x}^{(k+1)} = \underline{x}^{(k)} - \underline{s}$

$\underline{\underline{J}}_{k+1} = \underline{\underline{J}}_k + \frac{1}{\| \underline{s} \|_2^2} f(\underline{x}^{(k+1)}) (-\underline{s})^T$

Man kann beweisen $\lim_{k \to \infty} \frac{\| x^{(k+1)} - x^* \|}{\| x^{(k)} - x^* \|} = 0$

d.h. superlineare Konvergenz.

Bessere Implementierung via Shermann- Morisson - Formel

$$\underline{\underline{J}}_{k+1} = \underline{\underline{J}}_k + \text{Rang 1-Matrix.}$$

$$\left(\underline{\underline{A}} + \underline{u}\,\underline{v}^T\right)^{-1} = \underline{\underline{A}}^{-1} - \frac{\underline{\underline{A}}^{-1}\,\underline{u}\,\underline{v}^T\,\underline{\underline{A}}^{-1}}{1+\underline{v}^T\underline{\underline{A}}^{-1}\underline{u}}$$

(einfach überprüfen!)

Rang-1-update braucht nur $\underline{\underline{A}}^{-1}$

In jedem Schritt:

$$\underline{\underline{J}}_{k+1}^{-1} = \underline{\underline{J}}_k^{-1} + \frac{\underline{\underline{J}}_k^{-1} f(x^{(k+1)})\,\underline{s}^T\,\underline{\underline{J}}_k^{-1}}{\|\underline{s}\|^2 - \underline{s}^T\,\underline{\underline{J}}_k^{-1} f(x^{(k+1)})}$$

Iteration: $\underline{\underline{J}}_0 = \underline{\underline{Df}}(\underline{x}^{(0)})$

zerlege $\underline{\underline{J}}_0 = \underline{\underline{L}}\,\underline{\underline{U}}$

löse $\underline{\underline{L}}\,\underline{\underline{U}}\,\underline{s}^{(0)} = \underline{f}(\underline{x}^{(0)})$

$$\underline{x}^{(1)} = \underline{x}^{(0)} - \underline{s}^{(0)}$$

$$\underline{\underline{J}}_1^{-1} = \underline{\underline{J}}_0^{-1} + \frac{\underline{\underline{J}}_0^{-1} f(\underline{x}^{(1)})\,\underline{s}^{(0)T}\,\underline{\underline{J}}_0^{-1}}{\|\underline{s}^{(0)}\|^2 - \underline{s}^{(0)T}\,\underline{\underline{J}}_0^{-1} f(\underline{x}^{(1)})}$$

$$\underline{x}^{(2)} = \underline{x}^{(1)} - \underline{\underline{J}}_1^{-1} f(\underline{x}^{(1)})$$

$\underline{s}^{(1)}$

$$\underline{s}^{(1)} = \underline{\underline{J}}_1^{-1} f(x^{(1)}) =$$

$$= \underline{\underline{J}}_0^{-1} f(\underline{x}^{(1)}) + \frac{\underline{\underline{J}}_0^{-1} f(\underline{x}^{(1)})\,\underline{s}^{(0)T}\,\underline{\underline{J}}_0^{-1} f(\underline{x}^{(1)})}{\|\underline{s}^{(0)}\|^2 - \underline{s}^{(0)T}\,\underline{\underline{J}}_0^{-1} f(\underline{x}^{(1)})}$$

w

z

$\underline{w} = \underline{\underline{\partial}}_0^{-1} \underline{f}(\underline{x}^{(1)})$ berechnet man als Lösung

von $\underline{\underline{\partial}}_0 \underline{w} = \underline{f}(\underline{x}^{(1)})$ [LGS]

$z = \underline{s}^{(0)T} \underline{w} \in \mathbb{R}$

$$\underline{s}^{(1)} = \underline{w} + \frac{\underline{w}\, z}{\|\underline{s}^{(0)}\|^2 - z} = \left(1 + \frac{z}{\|\underline{s}^{(0)}\|^2 - z}\right) \underline{w}$$

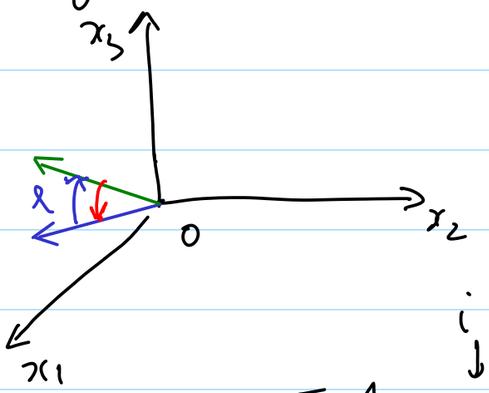
$$\underline{x}^{(2)} = \underline{x}^{(1)} - \underline{s}^{(1)}$$

* code

§7. Intermezzo über Lineare Algebra

§7.1 Orthogonale Matrizen

Drehung in der $x_1 O x_3$ - Ebene



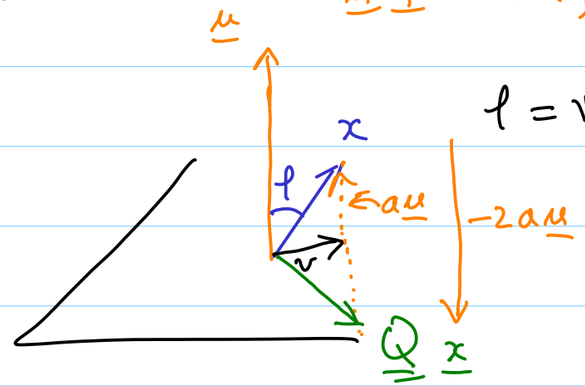
$$\underline{\underline{D}}(\varphi) = \begin{bmatrix} \cos\varphi & 0 & +\sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi \end{bmatrix}$$

i j

$$\underline{\underline{D}}_{ij}(-\varphi) = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \cos\varphi & & +\sin\varphi & \leftarrow i \\ & & & 1 & & \\ & & -\sin\varphi & & \cos\varphi & \leftarrow j \\ & & & & & 1 \end{bmatrix}$$

Spiegelung

$\underline{u} \perp$ Ebene, $\|\underline{u}\|_2 = 1$



φ = Winkel zwischen $\underline{x}, \underline{u}$

$$\cos\varphi = \frac{a}{\|\underline{x}\|} \Rightarrow$$

$$a = \|\underline{x}\| \cos\varphi$$

$$\underline{\underline{Q}}\,\underline{x} = \underline{x} + (-2a\,\underline{u}) = \underline{x} - 2a\,\underline{u}$$

$$\underline{u}^T \underline{x} = \langle \underline{u}, \underline{x} \rangle = \|\underline{u}\|\,\|\underline{x}\| \cos\varphi = \|\underline{u}\| \cdot a = a$$

$$\underline{\underline{Q}}\,\underline{x} = \underline{x} - 2(\underline{u}^T\underline{x})\,\underline{u} = \underline{x} - 2\,\underline{u}(\underline{u}^T\underline{x}) = \underline{x} - 2(\underline{u}\,\underline{u}^T)\underline{x}$$

$$= \left(\underline{\underline{I}} - 2\underbrace{\underline{u}\,\underline{u}^T}_{\underline{\underline{P}}_u}\right)\underline{x}$$

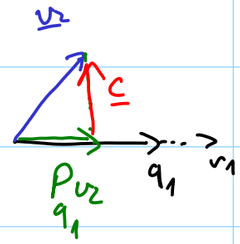
$$\underline{\underline{Q}} = \underline{\underline{I}} - 2\underline{u}\,\underline{u}^T = \underline{\underline{I}} - 2\underline{\underline{P}}_u$$ Householdermatrix

§7.2 QR-Zerlegung

(I) via (modifizierten) Gram-Schmidt

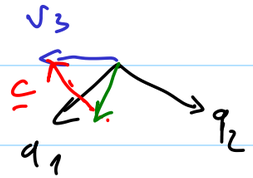
$\underline{\underline{A}} = [\underline{v}_1\ \underline{v}_2\ \dots\ \underline{v}_n] \rightsquigarrow \underline{q}_1 \dots \underline{q}_n$ ONB

$\underline{q}_1 = \frac{1}{\|\underline{v}_1\|}\underline{v}_1 \Rightarrow \underline{v}_1 = \|\underline{v}_1\|\,\underline{q}_1 = \langle \underline{q}_1, \underline{v}_1\rangle \underline{q}_1$



$\underline{c}_2 = \underline{v}_2 - P_{\underline{q}_1}\underline{v}_2 = \underline{v}_2 - \langle \underline{q}_1, \underline{v}_2\rangle \underline{q}_1$

$\underline{q}_2 = \frac{1}{\|\underline{c}_2\|}\underline{c}_2$



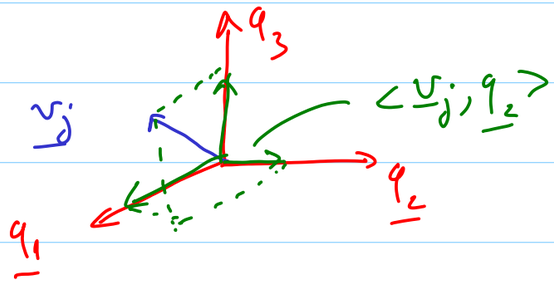
$\underline{c}_3 = \underline{v}_3 - P_{\mathrm{span}\{\underline{q}_1,\underline{q}_2\}}\underline{v}_2 = \underline{v}_3 - \langle \underline{q}_1, \underline{v}_3\rangle \underline{q}_1 - \langle \underline{q}_2, \underline{v}_3\rangle \underline{q}_2$

$\underline{q}_3 = \frac{1}{\|\underline{c}_3\|}\underline{c}_3$

$\dots$

$\underline{c}_k = \underline{v}_k - P_{\mathrm{span}\{\underline{q}_1,\dots,\underline{q}_{k-1}\}}\underline{v}_k = \underline{v}_k - \langle \underline{q}_1, \underline{v}_k\rangle \underline{q}_1 - \dots - \langle \underline{q}_{k-1}, \underline{v}_k\rangle \underline{q}_{k-1}$

$\underline{q}_k = \frac{1}{\|\underline{c}_k\|}\underline{c}_k$



$\underline{v}_1 = \langle \underline{q}_1, \underline{v}_1\rangle \underline{q}_1 + 0\,\underline{q}_2 + \dots + 0\,\underline{q}_n$

$\underline{v}_2 = \langle \underline{q}_1, \underline{v}_2\rangle \underline{q}_1 + \langle \underline{q}_2, \underline{v}_2\rangle \underline{q}_2 + 0\cdot\underline{q}_3 + \dots + 0\cdot\underline{q}_n$

$\underline{v}_3 = \langle \underline{q}_1, \underline{v}_3\rangle \underline{q}_1 + \langle \underline{q}_2, \underline{v}_3\rangle \underline{q}_2 + \langle \underline{q}_3, \underline{v}_3\rangle \underline{q}_3 + 0\cdot\underline{q}_4 + \dots + 0\cdot\underline{q}_n$

$\dots$

$\underline{v}_k = \langle \underline{q}_1, \underline{v}_k\rangle \underline{q}_1 + \langle \underline{q}_2, \underline{v}_k\rangle \underline{q}_2 + \dots + \langle \underline{q}_k, \underline{v}_k\rangle \underline{q}_k + 0\,\underline{q}_{k+1} + \dots + 0\cdot\underline{q}_n$

Im k-ten Schritt:

$$\underline{v}_k = (\underline{q}_1^H \underline{v}_k)\,\underline{q}_1 + (\underline{q}_2^H \underline{v}_k)\,\underline{q}_2 + \ldots + (\underline{q}_k^H \underline{v}_k)\,\underline{q}_k$$

$\underline{v}_k$ = (lineare Kombination von $\underline{q}_1, \ldots, \underline{q}_k$ mit Koeff $(\underline{q}_1^H \underline{v}_k), \ldots (\underline{q}_k^H \underline{v}_k) \in \mathbb{C}$

$$\underline{v}_k = \begin{bmatrix} \underline{q}_1 \ldots \underline{q}_k \end{bmatrix} \begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix}$$

$$\begin{bmatrix} \underline{v}_1 \ \underline{v}_2 \end{bmatrix} = \begin{bmatrix} \underline{q}_1 \ \underline{q}_2 \end{bmatrix} \begin{bmatrix} \underline{q}_1^H \underline{v}_2 \\ \underline{q}_2^H \underline{v}_2 \end{bmatrix}$$

$$n \begin{bmatrix} \underline{v}_1 \ \underline{v}_2 \ \ldots \ \underline{v}_k \end{bmatrix}_{k} = \begin{bmatrix} \underline{q}_1 \ldots \underline{q}_k \end{bmatrix}_{k} \begin{bmatrix} \underline{q}_1^H \underline{v}_1 & \underline{q}_1^H \underline{v}_2 & \ldots & \underline{q}_1^H \underline{v}_k \\ 0 & \underline{q}_2^H \underline{v}_2 & \ldots & \underline{q}_2^H \underline{v}_k \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & \underline{q}_k^H \underline{v}_k \end{bmatrix}_{k}$$

$$n \begin{bmatrix} \underline{A}_k \end{bmatrix}_{k} = \begin{bmatrix} \underline{Q}_k \end{bmatrix}_{k} \begin{bmatrix} * & \\ & \ddots \\ 0 & * \end{bmatrix}_{k}$$

↳ Spalten sind orthonormale Vektoren

Nach n Schritte:

$$r_{ij} := \underline{q}_i^H \underline{v}_j$$

$$\begin{bmatrix} \underline{v}_1 \ldots \underline{v}_n \end{bmatrix} = \begin{bmatrix} \underline{q}_1 \ldots \underline{q}_n \end{bmatrix} \begin{bmatrix} * & \\ & \ddots \\ 0 & * \end{bmatrix} \leftarrow j \quad (i \downarrow)$$

$$\begin{bmatrix} \underline{v}_1 & \underline{v}_2 \end{bmatrix} = \begin{bmatrix} \underline{q}_1 & \underline{q}_2 \end{bmatrix} \underbrace{\begin{bmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{bmatrix}}_{\underline{\underline{R}}} \qquad | \underline{\underline{R}}^{-1}$$

$$\begin{bmatrix} \underline{v}_1 & \underline{v}_2 \end{bmatrix} \begin{bmatrix} \frac{1}{r_{11}} & -\frac{r_{12}}{r_{11}} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{r_{22}} \end{bmatrix} = \begin{bmatrix} \underline{q}_1 & \underline{q}_2 \end{bmatrix}$$

$$\begin{bmatrix} \underline{v}_1 & \underline{v}_2 & \underline{v}_3 \end{bmatrix} = \begin{bmatrix} \underline{q}_1 & \underline{q}_2 & \underline{q}_3 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix} \qquad | \underline{\underline{R}}^{-1}$$

$$\begin{bmatrix} \underline{v}_1 & \underline{v}_2 & \underline{v}_3 \end{bmatrix} \begin{bmatrix} \frac{1}{r_{11}} & -\frac{r_{12}}{r_{11}} & -\frac{r_{13}}{r_{11}} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{r_{22}} & -\frac{r_{23}}{r_{22}} \\ 0 & 0 & 1 \end{bmatrix} \cdot$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{r_{33}} \end{bmatrix} = \begin{bmatrix} \underline{q}_1 & \underline{q}_2 & \underline{q}_3 \end{bmatrix}$$

Bem GS ~ Multiplikation mit oberen Dreiecksmatrizen

$$\underline{\underline{A}} \underbrace{\underline{\underline{R}}_1 \underline{\underline{R}}_2 \cdots \underline{\underline{R}}_n}_{\underline{\underline{R}}^{-1}} = \underline{\underline{Q}} \Rightarrow \underline{\underline{A}} = \underline{\underline{Q}} \underline{\underline{R}}$$

$$r_{ij} = \underline{q}_i^H \underline{v}_i$$

$$\underline{\underline{R}}_1 = \begin{bmatrix} \frac{1}{r_{11}} & -\frac{r_{12}}{r_{11}} & \cdots & -\frac{r_{1n}}{r_{11}} \\ 0 & 1 & 0 & 0 \\ 0 & & 1 & \\ \vdots & & & \\ 0 & 0 & \cdots 0 & 1 \end{bmatrix}$$

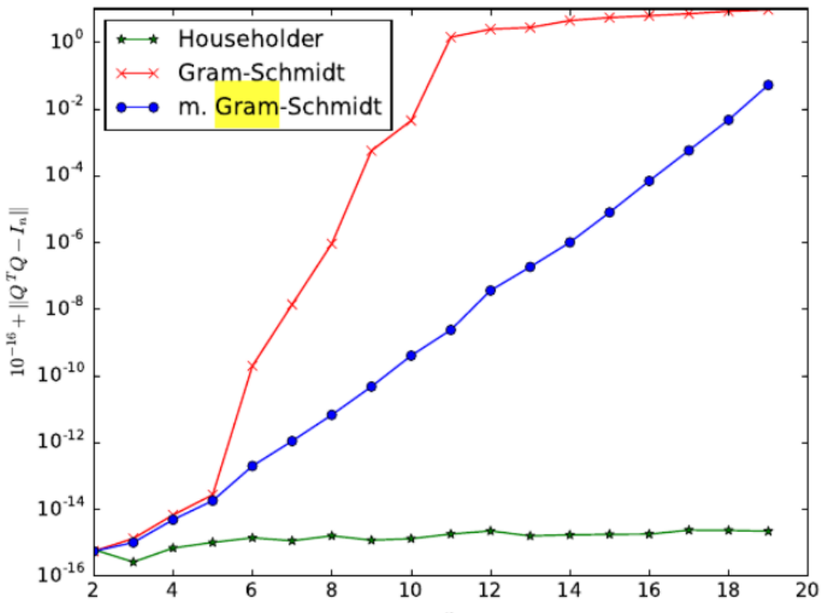
$$\underline{\underline{R}}_2 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \frac{1}{r_{22}} & -\frac{r_{23}}{r_{22}} & \cdots -\frac{r_{2n}}{r_{22}} \\ 0 & 0 & 1 & \cdots 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots 1 \end{bmatrix}$$

$$\underline{\underline{R}}_3 = \begin{bmatrix} 1 & & & 0 \\ & 1 & & \\ & & \frac{1}{r_{33}} & \cdots \\ & & & 1 \\ 0 & & & \ddots \end{bmatrix}$$

Bem Gram-Schmidt produziert nach n Schritte eine orthogonale Matrix Q in dem man in jedem Schritt mit einer oberen Dreiecksmatrix multipliziert

$\Rightarrow$ anfällig an Rundungsfehler!

$$\mathbf{A} = \begin{bmatrix} t_0^{n-1} & \cdots & t_0^1 & 1 \\ t_1^{n-1} & \cdots & t_1^1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ t_{19}^{n-1} & \cdots & t_{19}^1 & 1 \end{bmatrix}, \text{ mit } \quad t_i = \frac{i}{(m-1)}$$



Bem Ein Vorteil vom Gram-Schmidt Algorithmus ist: wenn man nach $k < n$ Schritte aufhört hat man bereits $q_1, \ldots, q_k$ orthogonal so dass

$\text{span}\{q_1, \ldots, q_k\} = \text{span}\{v_1, \ldots, v_k\}$

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 projiziert.

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 geeignet!
2) modifiziertes Gram-Schmidt ist nur für kleine Matrizen zu verwenden!
3) sonst II, III die folgen!

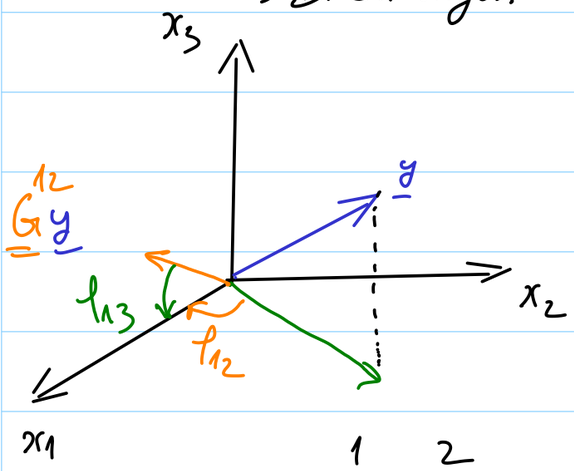
II QR via Rotationen.

$$\underset{m}{\underset{n}{A}} = \underset{m}{\underset{m}{Q}} \; \underset{m}{\underset{n}{\begin{bmatrix} * & & * \\ & \ddots & \\ 0 & & * \end{bmatrix}}} = \underset{m}{\underset{n}{Q R}}$$

$$\underset{m}{\underset{n}{\begin{bmatrix} A \end{bmatrix}}} = \underset{m}{\underset{m}{\begin{bmatrix} Q \end{bmatrix}}} \underset{m}{\underset{n}{\begin{bmatrix} * & & * \\ & \ddots & \\ 0 & & * \end{bmatrix}}} = \underset{m}{\underset{n}{Q R}}$$

Idee: Erzeuge die 0 unterhalb der Hauptdiagonale mittels orthogonale Matrizen

→ Drehungen



$$\underline{\underline{A}} = \begin{bmatrix} y_1 & \tilde{z}_1 \\ y_2 & \tilde{z}_2 \\ y_3 & \tilde{z}_3 \end{bmatrix}$$

$$\underline{\underline{G}}^{12}(\varphi_{12}) = \begin{bmatrix} \cos\varphi_{12} & \sin\varphi_{12} & 0 \\ -\sin\varphi_{12} & \cos\varphi_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ \\ \end{matrix}$$

↳ Givens Drehung, die y_2 zu 0 macht

$$\underline{\underline{G}}^{12}(\varphi_{12})\,\underline{\underline{A}} = \begin{bmatrix} * & * \\ 0 & x \\ y_3 & \tilde{z}_3 \end{bmatrix}$$

$$\underline{\underline{G}}^{13}(\varphi_{13}) = \begin{bmatrix} \cos\varphi_{13} & 0 & \sin\varphi_{13} \\ 0 & 1 & 0 \\ -\sin\varphi_{13} & 0 & \cos\varphi_{13} \end{bmatrix} \begin{matrix} 1 \\ \\ 3 \end{matrix}$$

↳ Givens Drehung, die y_3 zu 0 macht

$$\underline{\underline{G}}^{13}\,\underline{\underline{G}}^{12} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} * \\ 0 \\ 0 \end{bmatrix}$$

orthogonale Matrix

$$\underline{y} \in \mathbb{R}^n \qquad \underbrace{\underline{\underline{G}}^{1n} \cdots \underline{\underline{G}}^{13}\,\underline{\underline{G}}^{12}}_{\underline{\underline{Q}}^{1}}\,\underline{y} = \begin{bmatrix} * \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Dann kommt die zweite Spalte von $\underline{\underline{A}}$ dran:

$$\begin{bmatrix} * & z_1 & x \\ 0 & z_2 & x \\ 0 & z_3 & x \end{bmatrix}$$

$\underline{\underline{G}}^{23}$ usw. z_3 zu 0 zu machen

$$\underline{\underline{Q}}^1 \underline{\underline{A}} = \begin{bmatrix} x & x & x & \dots & x \\ 0 & x & x & \dots & x \\ 0 & x & x & \dots & x \\ \vdots & - & - & - & - \\ 0 & x & x & \dots & x \end{bmatrix}$$

$$\Rightarrow \underline{\underline{Q}}^2 = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & \text{Drehungen um} & \\ 0 & & \begin{bmatrix} x \\ 0 \\ \vdots \\ 0 \end{bmatrix} \text{ zu erteugen} & \end{bmatrix}$$

$$\underline{\underline{G}}^{ij}(\ell) = \begin{bmatrix} 1 & & & & & & & 0 \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & & \cos\ell & & \sin\ell & & \\ & & & & 1 & & & \\ & & & -\sin\ell & & \cos\ell & & \\ & & & & & & \ddots & \\ 0 & & & & & & & 1 \end{bmatrix} \begin{matrix} \\ \\ \\ \leftarrow i \\ \\ \leftarrow j \\ \\ \\ \end{matrix}$$

$$\underline{\underline{G}}^{ij}(\ell) \begin{bmatrix} x_1 \\ \vdots \\ x_{i-1} \\ x_i \\ x_{i+1} \\ \vdots \\ x_{j-1} \\ x_j \\ x_{j+1} \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_{i-1} \\ r \\ x_{i+1} \\ \vdots \\ x_{j-1} \\ 0 \\ x_{j+1} \\ \vdots \\ x_n \end{bmatrix}$$

$$r = \sqrt{x_i^2 + x_j^2}$$

$$\cos\ell = \frac{x_i}{r}$$

$$\sin\ell = \frac{x_j}{r}$$

$$\underbrace{\underline{\underline{Q}}^{n-1} \dots \underline{\underline{Q}}^2 \underline{\underline{Q}}^1}_{\text{orthogonale Matrizen}} \underline{\underline{A}} = \underline{\underline{R}}$$

$$\left(\underline{\underline{Q}}^1\right)^T \dots \left(\underline{\underline{Q}}^{n-1}\right)^T$$

$$\Rightarrow \underline{\underline{A}} = \underbrace{\left(\underline{\underline{Q}}^1\right)^T \left(\underline{\underline{Q}}^2\right)^T \dots \left(\underline{\underline{Q}}^{n-1}\right)^T}_{\underline{\underline{Q}} \text{ orthogonal}} \underline{\underline{R}}$$

$$\underline{\underline{A}} = \underline{\underline{Q}}\,\underline{\underline{R}}$$

Bsp $\begin{bmatrix} 4 \\ -3 \\ 1 \end{bmatrix} \rightsquigarrow \begin{bmatrix} * \\ 0 \\ 0 \end{bmatrix}$ $\quad r = \sqrt{4^2 + (-3)^2} = 5$

$\cos\varphi = \frac{4}{5}$, $\sin\varphi = -\frac{3}{5}$

$$\underline{\underline{G}}^{12} \begin{bmatrix} 4 \\ -3 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{4}{5} & -\frac{3}{5} & 0 \\ 3/5 & 4/5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ -3 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \\ 1 \end{bmatrix}$$

$$\underline{\underline{G}}^{13} \begin{bmatrix} 5 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{5}{\sqrt{26}} & 0 & \frac{1}{\sqrt{26}} \\ 0 & 1 & 0 \\ -\frac{1}{\sqrt{26}} & 0 & \frac{5}{\sqrt{26}} \end{bmatrix} \begin{bmatrix} 5 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \sqrt{26} \\ 0 \\ 0 \end{bmatrix}$$

$$\underline{\underline{G}} = \underline{\underline{G}}^{13} \underline{\underline{G}}^{12}$$

Bem Diese Methode (Givens-Drehungen) ist günstig falls $\underline{\underline{A}}$ sehr viele 0-Einträge hat (dünnbesetzte Matrix)

III QR-Zerlegung mit Spiegelungen (Standard)

$\underline{\underline{Q}}^1$ spiegelt $\underline{y}$ so dass $\underline{\underline{Q}}^1 \underline{y}$ auf Ox_1 liegt

$$\underline{\underline{Q}}^1 \underline{y} = \begin{bmatrix} * \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \Rightarrow \underline{\underline{Q}}^1 \underline{\underline{A}} = \begin{bmatrix} * & x & x \dots x \\ 0 & x & \dots \\ 0 & \vdots & \\ 0 & x & \dots \end{bmatrix}$$

$\underline{z}$ ↑ Spiegelung in $\mathbb{R}^{m-1}$ $x_2 O x_3$

$\underline{\underline{Q}}^2$ spiegelt $\underline{z}$ so dass $\underline{\underline{Q}}^2 \underline{z}$ auf Ox_2 liegt

$$\underline{\underline{Q}}^2 \underline{\underline{Q}}^1 \underline{\underline{A}} = \begin{bmatrix} * & x & x & x \dots \\ 0 & x & x & x \dots \\ 0 & 0 & x & x \dots \\ \vdots & \vdots & \vdots & \\ 0 & 0 & x & \dots \end{bmatrix} \quad \text{usw}$$

$$\underline{\underline{Q}}^{n-1} \underline{\underline{Q}}^{n-2} \dots \underline{\underline{Q}}^2 \underline{\underline{Q}}^1 \underline{\underline{A}} = \underline{\underline{R}} \Rightarrow$$

$$\underline{\underline{A}} = \underline{\underline{Q}} \underline{\underline{R}} \quad \text{mit} \quad \underline{\underline{Q}} = \underline{\underline{Q}}^{1T} \underline{\underline{Q}}^{2T} \dots \underline{\underline{Q}}^{n-1\,T}.$$

Bsp

$$\underline{x} = \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix} \Longrightarrow \underline{\underline{Q}}\,\underline{x} = \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix} = 3\,\underline{e}_1$$

$$\|\underline{x}\| = \sqrt{2^2+2^2+1} = 3$$

$$\underline{x} = \|x\|\,\underline{e}_1 + \underline{v} \Rightarrow \underline{v} = \underline{x} - \|x\|\,\underline{e}_1 = \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix}$$

$$\underline{u} = \frac{1}{\|v\|}\,\underline{v} = \frac{1}{\sqrt{6}} \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix}$$

$$\underline{\underline{Q}} = \underline{\underline{I}} - 2\,\underline{u}\,\underline{u}^T = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} - \frac{2}{6} \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix} \begin{pmatrix} -1 & 2 & 1 \end{pmatrix} =$$

$$= \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} - \frac{2}{6} \begin{bmatrix} 1 & -2 & -1 \\ -2 & 4 & 2 \\ -1 & 2 & 1 \end{bmatrix} =$$

$$= \frac{1}{3} \begin{bmatrix} 2 & 2 & 1 \\ 2 & -1 & -2 \\ 1 & -2 & 2 \end{bmatrix}$$

$$\underline{\underline{A}} = \begin{bmatrix} 1 & 1 \\ 2 & 0 \\ 2 & 0 \end{bmatrix} \Rightarrow \underline{\underline{Q}}^1\,\underline{\underline{A}} = \begin{bmatrix} \underline{\underline{Q}}^1 \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} & \underline{\underline{Q}}^1 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix}$$

$$\underline{\underline{Q}}^1 = \underline{\underline{I}} - \frac{2}{12} \begin{bmatrix} -2 \\ 2 \\ 2 \end{bmatrix} \begin{pmatrix} -2 & 2 & 2 \end{pmatrix} \Rightarrow$$

$$\underline{\underline{Q}}^1\,\underline{\underline{A}} = \begin{bmatrix} 3 & \frac{1}{3} \\ 0 & 2/3 \\ 0 & 2/3 \end{bmatrix}$$

usw.

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

$$\underline{\underline{A}} = \underline{\underline{Q}} \begin{bmatrix} \underline{\underline{R}} \\ 0 \end{bmatrix}_m^n$$

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 $\underline{\underline{R}}$ die Cholesky-Zerlegung von $\underline{\underline{A}}^T\underline{\underline{A}}$

Beweis $\underline{\underline{A}}^T\underline{\underline{A}} = [\underline{\underline{R}}^T \; \underline{\underline{0}}]\,\underline{\underline{Q}}^T\underline{\underline{Q}}\begin{bmatrix}\underline{\underline{R}}\\ \underline{\underline{0}}\end{bmatrix} = \underline{\underline{R}}^T\underline{\underline{R}}$

$\underline{\underline{A}}^T\underline{\underline{A}}$ symm. positiv. $\underline{\underline{Q}}^T\underline{\underline{Q}} = \underline{\underline{I}}$

Theorem Falls $\operatorname{rang}\underline{\underline{A}} = r < n$ dann gibt es eine Permutation $\underline{\underline{P}}$ so dass

$$\underline{\underline{A}}\,\underline{\underline{P}} = \underline{\underline{Q}}\begin{bmatrix}\underline{\underline{R}}_{11} & \underline{\underline{R}}_{12}\\ \underline{\underline{0}} & \underline{\underline{0}}\end{bmatrix}\begin{matrix}r\\ m-r\end{matrix}$$

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 Spalten lin. unabhängig permutiere sie noch vorne

$\underline{\underline{A}}\underline{\underline{P}} = [\underline{\underline{A}}_1 \; \underline{\underline{A}}_2]$ und wende das vorige Theorem. ($\underline{\underline{A}}_1$: r Spalten)

(Givens-Drehungen) an.

Bem QR-Zerlegung mit orthogonalen Transformationen darf nicht vorzeitig abgebrochen werden.

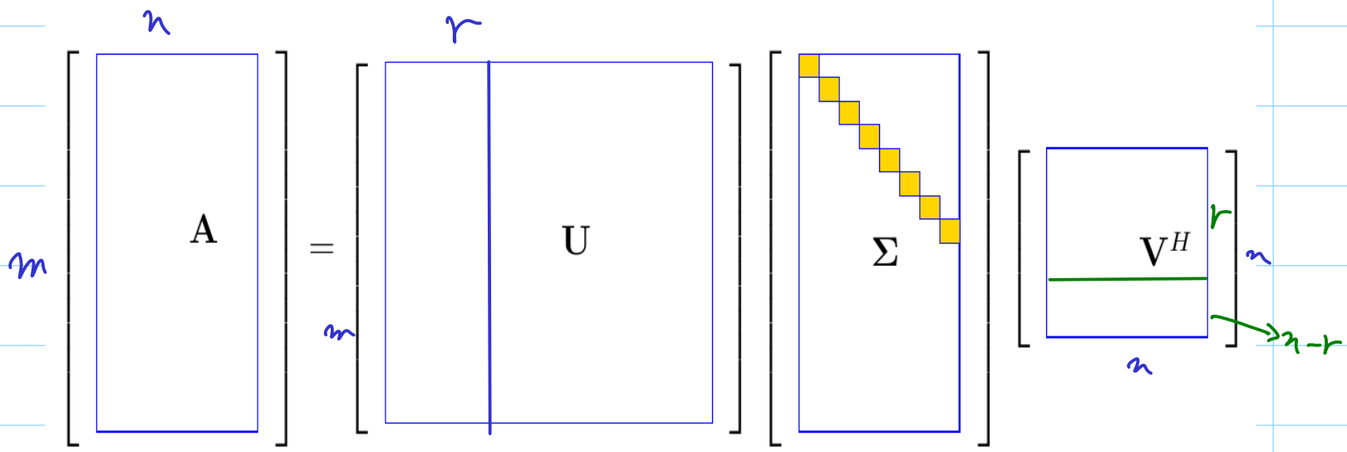
Bem $\underline{\underline{A}} = \underline{\underline{Q}}\underline{\underline{R}}$; $\underline{\underline{A}}\underline{x} = \underline{b}$

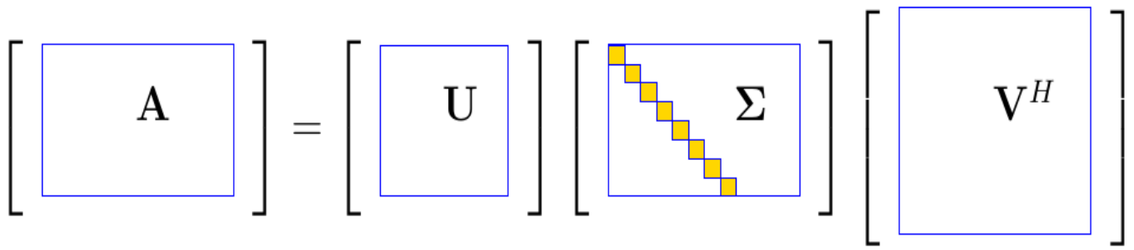
$\underline{\underline{Q}}^T|$ $\underline{\underline{Q}}\underline{\underline{R}}\,\underline{x} = \underline{b} \Rightarrow \underline{\underline{R}}\,\underline{x} = \underline{\underline{Q}}^T\underline{b}$

Rückwärtssubstitution!

§7.3. Singulärwertzerlegung.

$A \in \mathbb{R}^{m \times n}$; $\mathbb{C}^{n \times n}$ $\qquad r = \operatorname{Rang}(A)$





Es gibt $u_1, u_2, \dots, u_r, u_{r+1}, \dots, u_m \in \mathbb{R}^m$ ONB

$v_1, v_2, \dots, v_r, v_{r+1}, \dots, v_n \in \mathbb{R}^n$ ONB

$$u_i^T A v_j = \begin{cases} \sigma_i, & \text{falls } i=j \le r \\ 0, & \text{sonst} \end{cases}$$

$$U^T A V = \Sigma = \begin{matrix} \\ m \end{matrix} \overset{n}{\begin{bmatrix} \sigma & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 0 \end{bmatrix}} \begin{matrix} \\ \}r \end{matrix}$$

$U_r = [u_1 \dots u_r]$, $V_r = [v_1 \dots v_r]$

$v_1, \dots, v_r$ = rechte Singulärvektoren = EV von A^TA

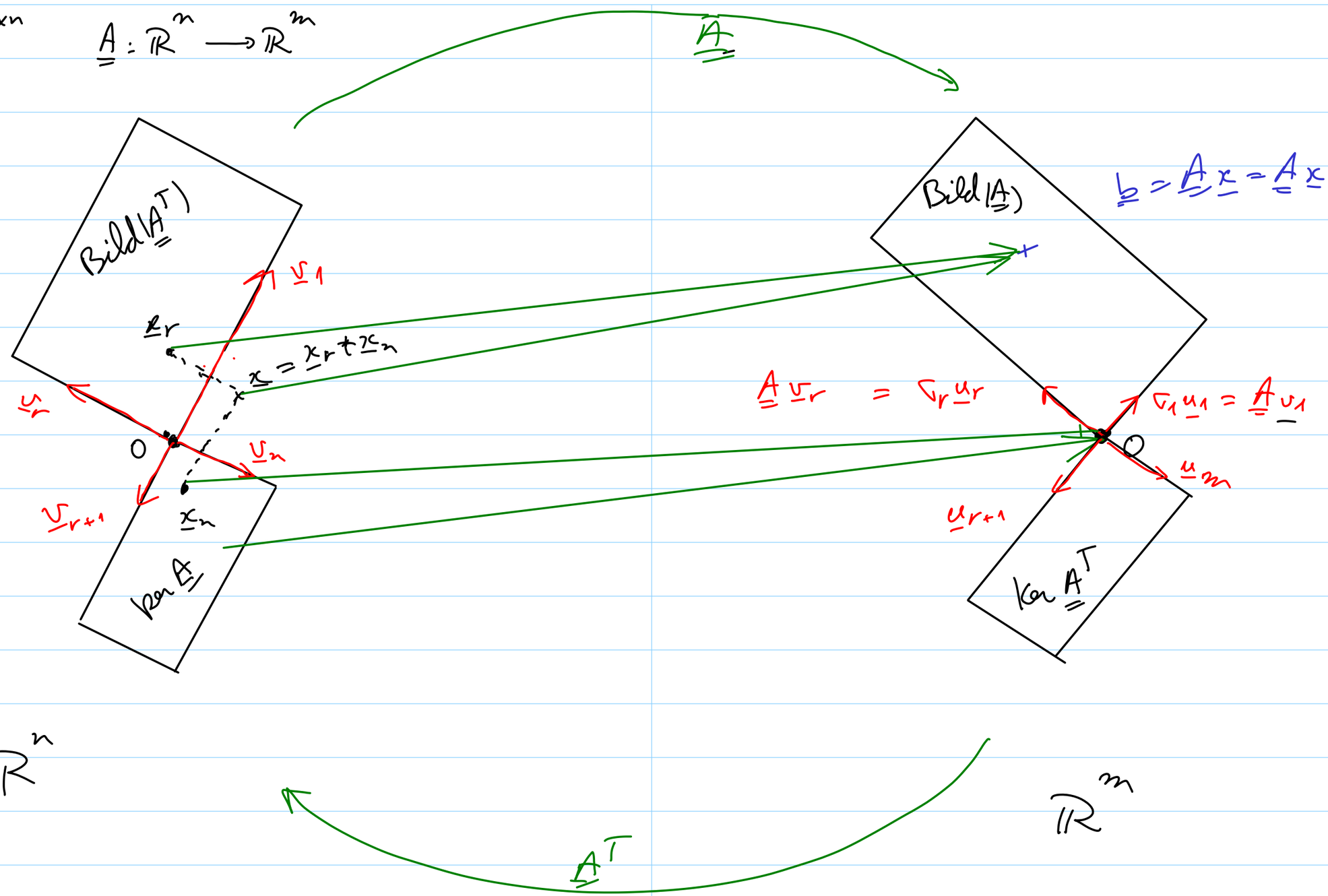
$u_1, \dots, u_r$ = linke Singulärvektoren = EV von AA^T

$A v_1 = \sigma_1 u_1, \dots, A v_r = \sigma_r u_r$

$A v_{r+1} = 0, \dots A v_n = 0$

$\operatorname{Ker} A = \operatorname{span}\{v_{r+1}, \dots, v_n\}$; $\operatorname{Bild} A = \operatorname{span}\{u_1, \dots, u_r\}$.

$\underline{\underline{A}} \in \mathbb{R}^{m \times n}$ $\quad \underline{\underline{A}} : \mathbb{R}^n \longrightarrow \mathbb{R}^m$



Bsp $5 \geq 3 \geq 0.1 \geq 10^{-5} \geq 10^{-8} \geq 10^{-12} \geq 5\cdot 10^{-16} \geq 2\cdot 10^{-16} \geq 10^{-42} \geq 0 = 0$

$\sigma_1 \quad \sigma_2 \quad \sigma_3 \quad \sigma_4 \quad \sigma_5 \quad \sigma_6 \quad \sigma_7 \quad \sigma_8 \quad \sigma_9 \quad \sigma_{10} = \sigma_{11} = \dots = \sigma_{42}$

Numerischer Rang = ?

Anwendung

1) Speicherplatz reduktion.

SVD: $rm + rn + r = r(m+n+1)$ Plätze

$$\underline{\underline{A}} = \sum_{k=1}^{r} \underbrace{\underline{u}_k \sigma_k \underline{v}_k^T}_{\text{Rang 1 Matrizen}} \approx \sum_{k=1}^{r_{num}} \underline{u}_k \sigma_k \underline{v}_k^T$$

2) Berechnung der Norm einer Matrix

$$\|\underline{\underline{A}}\|_2 = \max_{\|\underline{x}\|_2 = 1} \|\underline{\underline{A}}\underline{x}\|_2$$

$$\|\underline{\underline{A}}\underline{x}\|_2^2 = \|\underline{\underline{U}}\,\underline{\underline{\Sigma}}\,\underline{\underline{V}}^T\underline{x}\|_2^2 = \|\underline{\underline{\Sigma}}\,\underline{\underline{V}}^T\underline{x}\|_2^2$$

$\underline{\underline{U}}$ orthogonal $\Rightarrow$ erhält Längen

$$\|\underline{\underline{A}}\| = \max_{\|x\|_2=1} \|\underline{\underline{A}}\underline{x}\|_2 = \max_{\|\underline{x}\|_2=1} \|\underline{\underline{\Sigma}}\underbrace{\underline{\underline{V}}^T\underline{x}}_{\underline{y}}\| = \max_{\|\underline{\underline{V}}\underline{y}\|_2=1} \|\underline{\underline{\Sigma}}\underline{y}\|_2 =$$

$\underline{x} = \underline{\underline{V}}\underline{y}$

$$= \max_{\|\underline{y}\|_2=1} \|\underline{\underline{\Sigma}}\underline{y}\|_2 = \max_{\|\underline{y}\|_2=1} \sqrt{\sigma_1^2|y_1|^2 + \sigma_2^2|y_2|^2 + \dots + \sigma_r^2|y_r|^2} \leq$$

$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$

$$\leq \max_{\|\underline{y}\|_2=1} \sqrt{\sigma_1^2\|\underline{y}\|^2} = \sigma_1 \text{ erreicht für } \underline{y} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Somit $\|\underline{\underline{A}}\|_2 = \sigma_1$

3) Kompression.

Daten mit Rausch / Fehler:

$$\underline{\underline{A}} = \sum_{j=1}^{r} \sigma_j \underline{u}_j \underline{v}_j^T \approx \sum_{j=1}^{k} \sigma_j \underline{u}_j \underline{v}_j^T \quad \text{mit } k \ll r.$$

$\neq$ Diagonalisierung für $m=n$

Bsp $\underline{\underline{A}} = \begin{bmatrix} 2 & 100 \\ 0 & 1 \end{bmatrix}$ $\quad \lambda_1 = 2, \quad \lambda_2 = 1$

$$\underline{\underline{A}} = \underline{\underline{S}}\,\underline{\underline{\Lambda}}\,\underline{\underline{S}}^{-1} = 2 \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\underline{s}_1} \underbrace{\begin{bmatrix} 1 & 100 \end{bmatrix}}_{\hat{s}_1^T} + 1 \underbrace{\begin{bmatrix} 1 \\ -\frac{1}{100} \end{bmatrix}}_{\underline{s}_2} \underbrace{\begin{bmatrix} 0 & -100 \end{bmatrix}}_{\hat{s}_2^T}$$

(λ_1, λ_2)

Versuch der Approximation mit einer Rang 1-Matrix

Fehler $\underline{\underline{A}} - \lambda_1 \underline{s}_1 \hat{s}^T = \begin{bmatrix} 0 & -100 \\ 0 & 1 \end{bmatrix}$ ☹

$$\underline{\underline{A}} - \lambda_2 \underline{s}_2 \underline{s}_2^T = \begin{bmatrix} 2 & 200 \\ 0 & 0 \end{bmatrix}$$ ☹

also so geht es nicht!

Mit SVD:

$$\underline{\underline{A}} = 100.025 \underbrace{\begin{bmatrix} 0.9995 \\ 0.001 \end{bmatrix}}_{=\underline{u}_1} \underbrace{\begin{bmatrix} 0.01999 & 0.9998 \end{bmatrix}}_{\underline{v}_1^T} +$$

(σ_1)

$$+ 0.02 \underbrace{\begin{bmatrix} 0.01 \\ 0.9997 \end{bmatrix}}_{=\underline{u}_2} \underbrace{\begin{bmatrix} -0.9998 & 0.01999 \end{bmatrix}}_{\underline{v}_2^T}$$

(σ_2)

Fehler $\underline{\underline{A}} - \sigma_1 \underline{u}_1 \underline{v}_1^T = \begin{bmatrix} 2 \cdot 10^{-4} & 0 \\ -2 \cdot 10^{-2} & 4 \cdot 10^{-4} \end{bmatrix}$ klein ☺

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

$$\underline{\underline{A}}_k = \sum_{j=1}^{k} \sigma_j \, \underline{u}_j \, \underline{v}_j^H$$

die, die beste Approximation von Rang k an $\underline{\underline{A}}$ ist, 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 → Skript.

§8 Ausgleichsrechnung

§8.1. Motivation und Normalengleichung

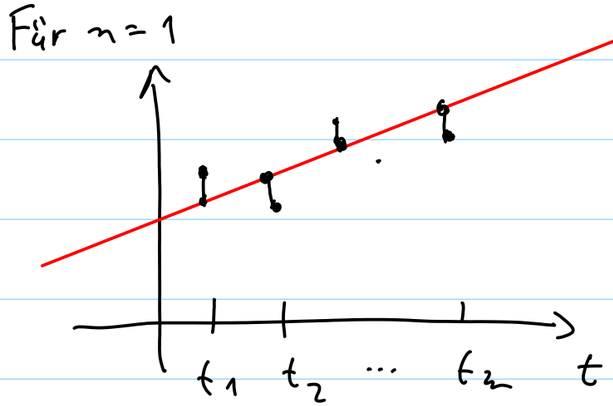
Lineares Modell:

$y = \underline{a}^T \underline{t} + c \cdot 1$ 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

$y_1 \quad y_2 \quad \dots \quad y_m$ Gemessene Werte (mit Fehlern)

Für $n = 1$



Vorschlag: $\underline{a}, c$ die $\underline{p}, q$, die das Minimum realisieren:

$$\min_{\substack{\underline{p} \in \mathbb{R}^n \\ q \in \mathbb{R}}} \sum_{i=1}^{m} \left| y_i - \underline{p}^T \underline{t}_i - 1 \cdot q \right|^2 = \min_{\underline{x} \in \mathbb{R}^{1+n}} \| \underline{A}\underline{x} - \underline{b} \|_2$$

$$\underline{b} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} \qquad \underline{x} = \begin{bmatrix} q \\ \underline{p} \end{bmatrix} \in \mathbb{R}^{1+n}, \quad \underline{A} = \begin{bmatrix} 1 & \underline{t}_1^T \\ 1 & \underline{t}_2^T \\ 1 & \vdots \\ 1 & \underline{t}_m^T \end{bmatrix} \in \mathbb{R}^{m \times (1+n)}$$

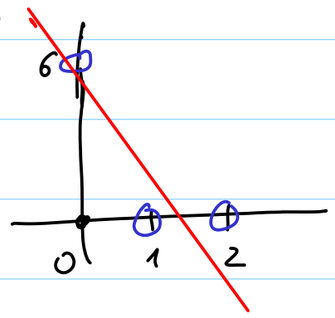
$\mathbb{R}^{m \times (1+n)}$

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 $\quad y=dt+c1$

Messwerte: $\begin{bmatrix}6\\0\\0\end{bmatrix}$, $\underline{\underline{A}}=\begin{bmatrix}1&0\\1&1\\1&2\end{bmatrix}$

$\min \| \underline{\underline{A}}\underline{x}-\underline{b} \|_2$

$\underline{x}\in\mathbb{R}^2$

alg. Problem:
$$\begin{cases} b_1 = df_1+c \\ b_2 = df_2+c \\ b_3 = df_3+c \end{cases}$$

$\underline{\underline{A}}\underline{x}=\underline{b}$ hat keine Lösung.

I Algebraische Methode.

$\underline{\underline{A}}\underline{x}=\underline{b} \Rightarrow \underline{\underline{A}}^T\underline{\underline{A}}\,\underline{x}=\underline{\underline{A}}^T b$ Normalengleichung

$\underline{\underline{A}}^T\,|$

$\underline{\underline{A}}^T\underline{\underline{A}}$ quadratisch, symmetrisch.

Falls $\operatorname{Rang}(\underline{\underline{A}})=n \Rightarrow \underline{\underline{A}}^T\underline{\underline{A}}$ invertierbar

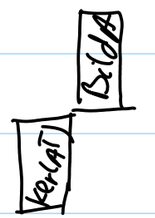
kann das LGS eindeutig lösen

$\operatorname{cond}(\underline{\underline{A}}) = \|\underline{\underline{A}}^{-1}\|\,\|\underline{\underline{A}}\|$ gross $\Rightarrow$ $\operatorname{cond}(\underline{\underline{A}}^T\underline{\underline{A}})$ riesig.

100 $\qquad 10^4$

$\operatorname{cond}(\underline{\underline{A}}^T\underline{\underline{A}})=\operatorname{cond}(\underline{\underline{A}})^2$ $\qquad$ LU $\rightsquigarrow$

$\underline{\underline{A}}: \mathbb{R}^n \to \mathbb{R}^m$; $\underline{b}\in\mathbb{R}^m =$



$\underline{x} \mapsto \underline{\underline{A}}\underline{x}$

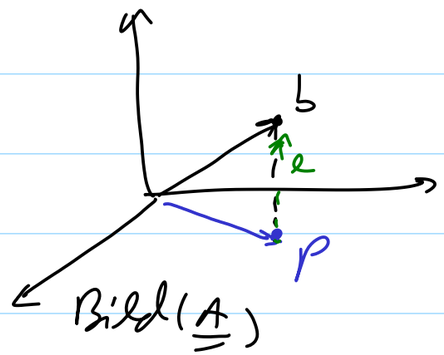
$\underline{b}=\underline{p}+\underline{e}$ $\qquad \underline{p}\perp\underline{e}$

$\underline{p}\in$ Bild$(\underline{\underline{A}})$ $\qquad \underline{e}\in$ Kern$(\underline{\underline{A}}^T)$

$p\in$ Bild$(\underline{\underline{A}})$ $\Rightarrow$ es gibt $\hat{\underline{x}}\in\mathbb{R}^2$ so dass $\underline{\underline{A}}\hat{\underline{x}}=\underline{p}$

Fehler $\underline{b} - \underline{\underline{A}}\underline{\hat{x}} = \underline{b} - \underline{p} = \underline{e} \in \ker(\underline{\underline{A}}^T) \perp \text{Bild}(\underline{\underline{A}})$

$$\Rightarrow \quad \|\underline{b} - \underline{\underline{A}}\underline{\hat{x}}\|_2^2 = \min_{\underline{x} \in \mathbb{R}^n} \|\underline{b} - \underline{\underline{A}}\underline{x}\|_2^2$$



$$\underline{\underline{A}}^T \underline{b} = \underline{\underline{A}}^T (\underline{p} + \underline{e}) =$$

$$= \underline{\underline{A}}^T \underline{p} + \underline{\underline{A}}^T \underline{e} = \underline{\underline{A}}^T \underline{p}$$

$$\Rightarrow \underline{\underline{A}}^T \underline{\underline{A}}\underline{\hat{x}} = \underline{\underline{A}}^T \underline{p}$$

II $\quad f(\underline{x}) = \|\underline{\underline{A}}\underline{x} - \underline{b}\|_2^2 \; ; \quad f: \mathbb{R}^{1+n} \to [0, \infty[\quad \text{Min ?}$

Finde $\underline{x}$:

1) $f(\underline{x}) = 0$

$$f(\underline{x}) = (\underline{\underline{A}}\underline{x} - \underline{b})^T (\underline{\underline{A}}\underline{x} - \underline{b}) = \underline{x}^T \underline{\underline{A}}^T \underline{\underline{A}}\underline{x} - \underline{b}^T \underline{\underline{A}}\underline{x} - \underline{x}^T \underline{\underline{A}}^T \underline{b} + \underline{b}^T \underline{b}$$

($\underline{\underline{A}}^T \underline{\underline{A}} = \underline{\underline{M}}$, $\underline{b}^T \underline{\underline{A}}\underline{x} = \underline{x}^T \underline{\underline{A}}^T \underline{b}$, $\underline{\underline{A}}^T \underline{b} = \underline{w}$)

$$= \underline{x}^T \underline{\underline{M}}\underline{x} - 2\underline{x}^T \underline{w} + \underline{b}^T \underline{b} =$$

$$= \sum_{i=1}^n x_i \sum_{j=1}^n m_{ij} x_j - 2 \sum_{i=1}^n x_i w_i + \underline{b}^T \underline{b} =$$

$$= x_1 \sum_{j=1}^n m_{1j} x_j + \sum_{i=2}^n x_i \sum_{j=1}^n m_{ij} x_j - 2 \sum_{i=1}^n x_i w_i + \underline{b}^T \underline{b}$$

$$\sum_{j=1}^n x_j \sum_{i=2}^n m_{ij} x_i$$

$$= x_1 m_{11} x_1 + x_1 \sum_{j=2}^n m_{1j} x_j + x_1 \sum_{i=2}^n m_{i1} x_i + \sum_{j=2}^n x_j \sum_{i=2}^n m_{ij} x_i -$$

$$- 2 \sum_{i=1}^n x_i w_i + \underline{b}^T \underline{b} =$$

(M symmetrisch)

$$= m_{11} x_1^2 + 2 x_1 \sum_{i=2}^n x_i m_{i1} + \sum_{i,j=2}^n x_i m_{ij} x_j -$$

$$- 2 \sum_{i=1}^n x_i w_i + \underline{b}^T \underline{b}$$

$$\frac{\partial}{\partial x_1} f(\underline{x}) = 2 m_{11} x_1 + 2 \sum_{i=2}^{n} x_i \cdot m_{i1} - 2 w_1 \quad \Rightarrow$$

$$Df(\underline{x}) = 2 \underline{\underline{M}}\, \underline{x} - 2 \underline{\underline{A}}^T \underline{b}$$

kritische Punkte $Df(\underline{x}) = 0 \Leftrightarrow 2 \underline{\underline{A}}^T \underline{\underline{A}} \underline{x} - 2 \underline{\underline{A}}^T \underline{b} = 0$

$$\Leftrightarrow \quad \underline{\underline{A}}^T \underline{\underline{A}} x = \underline{\underline{A}}^T \underline{b}$$

Minimum nur wenn $D^2 f(\underline{x}) = 2 \underline{\underline{A}}^T \underline{\underline{A}}$

symmetrische pos. def.

nur wenn $\operatorname{Rang}(\underline{\underline{A}}) = n$.

$$\operatorname{cond}_2(\underline{\underline{A}}^T \underline{\underline{A}}) = \operatorname{cond}_2(\underbrace{\underline{\underline{V}} \underline{\underline{\Sigma}}^T \underline{\underline{U}}^T}_{I} \underbrace{\underline{\underline{U}} \underline{\underline{\Sigma}} \underline{\underline{V}}^T}_{I}) =$$

$$= \operatorname{cond}_2(\underline{\underline{V}} \underline{\underline{\Sigma}}^2 \underline{\underline{V}}^T) = \operatorname{cond}_2(\underline{\underline{\Sigma}}^2) = \frac{\sigma_1^2}{\sigma_r^2} = \operatorname{cond}_2(\underline{\underline{A}})^2$$

wobei $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$ die Singulärwerte von $\underline{\underline{A}}$ sind.

Bsp $f \in L^2 = \{ f : I \to \mathbb{R};\ \int_I |f(t)|^2 dt \}$

$\mathbb{R}$ $\quad$ U

$f_n \in V_n$, $\dim V_n < \infty$

↳ Basis $b_1, \dots, b_n$

$$f_n = \sum_{j=1}^{n} x_j b_j \Rightarrow f_n(t) = \sum_{j=1}^{n} x_j \cdot b_j(t)$$

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

$$\sum_{i=1}^{m} | f_n(t_i) - y_i |^2 = \text{minimal!}$$

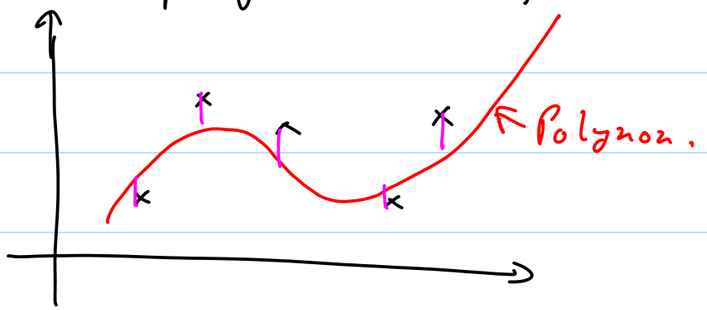
$$\begin{cases} \sum_{j=1}^{n} b_j(t_i)\, x_j = y_i \\ \hookrightarrow i = 1, 2, \dots, m \end{cases} \qquad A = [\, b_j(t_i) \,]_{\substack{i=1,\dots,m \\ j=1,\dots,n}}$$

Bem Spezielle Wahl:

$V_n = P_n$ = Polynome vom Grad maximal $n-1$

Basis in V_n: Monome $b_j(t) = t^{j-1}$

→ polynomials fit ; polyfit



§8.2. Lösung mittels orthogonalen Transformationen

Fall 1 $\operatorname{Rang}(\underline{\underline{A}}) = n$ (voller Rang)

$$\underline{\underline{A}} = \underline{\underline{Q}}\,\underline{\underline{R}} = \underline{\underline{Q}} \begin{bmatrix} \tilde{\underline{\underline{R}}} \\ 0 \end{bmatrix} \begin{matrix} n \\ \} m-n \end{matrix} \qquad \underline{\underline{Q}} \text{ orthogonal}$$

$$\min_{\underline{x} \in \mathbb{R}^n} \| \underline{\underline{A}}\,\underline{x} - \underline{b} \|_2^2 = \min_{\underline{x} \in \mathbb{R}^n} \| \underline{\underline{Q}}\,\underline{\underline{R}}\,x - \underline{b} \|_2^2 =$$

$\underline{\underline{I}} = \underline{\underline{Q}}\,\underline{\underline{Q}}^H$

$$= \min_{\underline{x} \in \mathbb{R}^n} \| \underline{\underline{Q}} (\underline{\underline{R}}\,\underline{x} - \underline{\underline{Q}}^H \underline{b}) \|_2^2 =$$

$$= \min_{\underline{x} \in \mathbb{R}^n} \| \underline{\underline{R}}\,\underline{x} - \underbrace{\underline{\underline{Q}}^H \underline{b}}_{\underline{\beta}} \|_2^2 =$$

$$\min_{\underline{x} \in \mathbb{R}^n} \left\| \begin{bmatrix} \tilde{\underline{\underline{R}}} \\ 0 \end{bmatrix} [x] - \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \\ \beta_{n+1} \\ \vdots \\ \beta_m \end{bmatrix} \right\|_2^2 = \min_{\underline{x} \in \mathbb{R}^n} \left\| \begin{bmatrix} \tilde{\underline{\underline{R}}}\,\underline{x} \\ 0 \end{bmatrix} - \begin{bmatrix} \tilde{\underline{\beta}} \\ \beta_{n+1} \\ \vdots \\ \beta_m \end{bmatrix} \right\|_2^2 =$$

$$= \min_{\underline{x} \in \mathbb{R}^n} \left\{ \| \tilde{\underline{\underline{R}}}\,\underline{x} - \tilde{\underline{\beta}} \|_2^2 + |\beta_{n+1}|^2 + \dots + |\beta_m|^2 \right\}$$

$$\boxed{\begin{matrix} * \\ 0 \end{matrix}} [\underline{x}] = \tilde{\underline{\beta}}$$

$$= \min_{\underline{x} \in \mathbb{R}^n} \| \tilde{\underline{\underline{R}}}\,\underline{x} - \tilde{\underline{\beta}} \|_2^2 + |\beta_{n+1}|^2 + \dots + |\beta_m|^2$$

$\underline{x} \in \mathbb{R}^n$ " 0 " erreicht für $\underline{x}$ = die Lösung von $\tilde{\underline{\underline{R}}}\,\underline{x} = \tilde{\underline{\beta}}$

Bsp $\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 0 \\ 0 \end{bmatrix} \Longrightarrow$ Normalengleichung.

$\begin{bmatrix} 3 & 3 \\ 3 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 0 \end{bmatrix} \Rightarrow \begin{matrix} x_1 = 5 \\ x_2 = -3 \end{matrix}$

Mit QR-Zerlegung:

$$\underline{\underline{A}} = \underline{\underline{Q}}\,\underline{\underline{R}} = \begin{bmatrix} -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & ? \\ -\frac{1}{\sqrt{3}} & 0 & ? \\ -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & ? \end{bmatrix} \begin{bmatrix} -\frac{3}{\sqrt{3}} & -\frac{3}{\sqrt{3}} \\ 0 & -\frac{2}{\sqrt{2}} \\ 0 & 0 \end{bmatrix}$$

LGS

$$\begin{bmatrix} -\frac{3}{\sqrt{3}} & -\frac{3}{\sqrt{3}} \\ 0 & -\frac{2}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -\frac{6}{\sqrt{3}} \\ 6/\sqrt{2} \end{bmatrix} \Rightarrow \begin{matrix} x_1 = 5 \\ x_2 = -3 \end{matrix}$$

Fall 2 $r = \mathrm{Rang}(\underline{\underline{A}}) < n \Rightarrow$ SVD!

$$\underline{\underline{A}} = \left[\begin{array}{c|c} \underline{\underline{U}}_1 & \underline{\underline{U}}_2 \end{array}\right] \left[\begin{array}{c|c} \underline{\underline{\Sigma}}_r & 0 \\ \hline 0 & 0 \end{array}\right] \left[\begin{array}{c} \underline{\underline{V}}_1^T \\ \hline \underline{\underline{V}}_2^T \end{array}\right] \begin{matrix} r \\ n-r \end{matrix}$$

($\underline{\underline{U}}_1$: r Spalten, $\underline{\underline{U}}_2$: $m-r$ Spalten; $\underline{\underline{V}}_1^T$: r Zeilen, $\underline{\underline{V}}_2^T$: $n-r$ Zeilen)

$$\underline{\underline{\Sigma}}_r = \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_r \end{bmatrix} \quad \text{mit} \quad \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0.$$

$\underline{\underline{U}}$ orthogonal

$$\| \underline{\underline{A}}\underline{x} - \underline{b} \|_2 = \left\| \begin{bmatrix} \underline{\underline{\Sigma}}_r \underline{\underline{V}}_1^T \underline{x} \\ \hline 0 \end{bmatrix} - \begin{bmatrix} \underline{\underline{U}}_1^T \underline{b} \\ \hline \underline{\underline{U}}_2^T \underline{b} \end{bmatrix} \right\|_2$$

ist minimal für $\underline{x}$ = Lösung von

$$\underline{\underline{\Sigma}}_r \underline{\underline{V}}_1^T \underline{x} = \underline{\underline{U}}_1^T \underline{b}$$

($\underline{\underline{V}}_1$: r Spalten; $\underline{\underline{V}}_1^T$: r Zeilen; $\underline{\underline{V}}_1^T \underline{\underline{V}}_1 = I_r$)

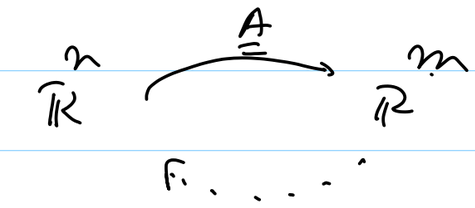
d.h. $\underline{x} = \underline{\underline{V}}_1 \underline{\underline{\Sigma}}_r^{-1} \underline{\underline{U}}_1^T \underline{b}$

und das Minimum ist $\| \underline{\underline{U}}_2^T \underline{b} \|_2$

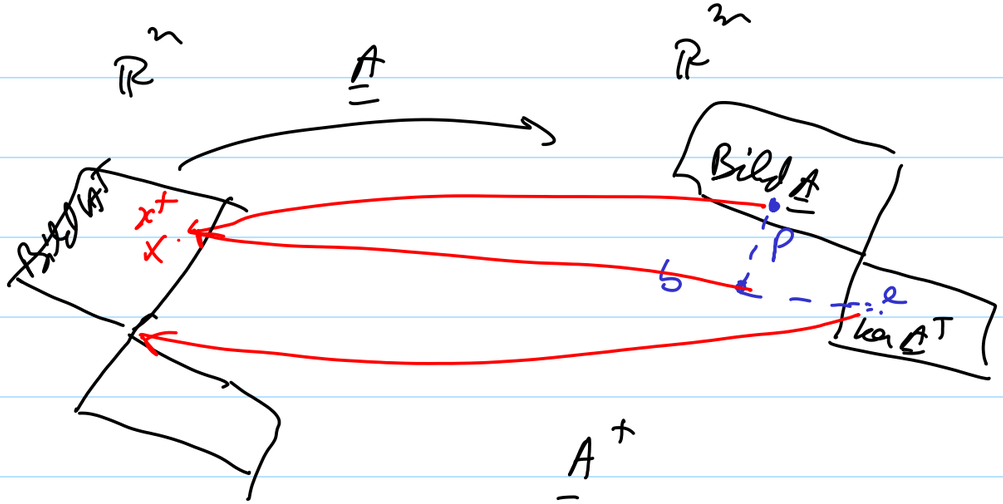
→ wird verwendet in standard codes Lsq($\underline{\underline{A}}, \underline{b}$)

Def Pseudoinverse einer Matrix (Moore-Penrose inverse)

$$\underline{\underline{A}}^+ := \underline{\underline{V}}\,\underline{\underline{\Sigma}}^+\,\underline{\underline{U}}^T \quad \text{mit} \quad \underline{\underline{\Sigma}}^+ = \begin{bmatrix} \sigma_1^{-1} & & & 0 \\ & \ddots & & \\ & & \sigma_r^{-1} & \\ 0 & & & 0 \end{bmatrix}$$



"die Ausgleichslösung" $\|Ax^+ - b\|_2^2$



§8.3. Lineare Ausgleichsrechnung mit linearen Nebenbedingungen.

Finde $x \in \mathbb{R}^n$ bei gegebenen $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$

mit $m \geq n = \text{Rang}(A)$

$$\begin{cases} \|Ax - b\|_2^2 = \text{minimal!} \\ Cx = d \end{cases}$$

mit $C \in \mathbb{R}^{p \times n}$ mit $p < n$

$\text{Rang } C = p$

$d \in \mathbb{R}^p$

Methode: Lagrange-Multiplicatoren $m \in \mathbb{R}^p$

$$L(x, m) = \frac{1}{2}\|Ax - b\|_2^2 + m^T(Cx - d)$$

$$\min_{x \in \mathbb{R}^n} \max_{m \in \mathbb{R}^p} L(x, m)$$

$$\begin{cases} \frac{\partial L}{\partial x}(\underline{x},\underline{m})=0 \\ \frac{\partial L}{\partial m}(\underline{x},\underline{m})=0 \end{cases} \qquad \begin{cases} \underline{\underline{A}}^T(\underline{\underline{A}}\underline{x}-b)+\underline{\underline{C}}^T\underline{m}=0 \\ 0+\underline{\underline{C}}\underline{x}-\underline{d}=0 \end{cases}$$

$$\begin{bmatrix} \underline{\underline{A}}^T\underline{\underline{A}} & \underline{\underline{C}}^T \\ \underline{\underline{C}} & \underline{\underline{0}} \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{m} \end{bmatrix} = \begin{bmatrix} \underline{\underline{A}}^T\underline{b} \\ \underline{d} \end{bmatrix}$$

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 $\underline{\underline{S}}^T\underline{\underline{S}} = -\underline{\underline{G}}\underline{\underline{G}}^T$

$$\begin{bmatrix} \underline{\underline{A}}^T\underline{\underline{A}} & \underline{\underline{C}}^T \\ \underline{\underline{C}} & \underline{\underline{0}} \end{bmatrix} = \begin{bmatrix} \underline{\underline{R}}^T & \underline{\underline{0}} \\ \underline{\underline{G}} & \underline{\underline{S}} \end{bmatrix} \begin{bmatrix} \underline{\underline{R}} & \underline{\underline{G}}^T \\ 0 & \underline{\underline{S}}^T \end{bmatrix}$$

Block 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 wir eventuel diese in den Algorithmen ausnutzen.

Methode 2 SVD besser für die Kondition, verliert die Struktur.

$$\underline{\underline{C}} = \underline{\underline{U}} \begin{bmatrix} \underline{\underline{\Sigma}} & \underline{\underline{0}} \end{bmatrix} \begin{bmatrix} \underline{\underline{V}}_1^H \\ \underline{\underline{V}}_2^H \end{bmatrix} \begin{matrix} p \\ n-p \end{matrix}$$

$\ker(\underline{\underline{C}}) = \mathrm{Bild}(\underline{\underline{V}}_2)$

Trick:

Definiere $\underline{x}_0 = \underline{\underline{V}}_1 \underline{\underline{\Sigma}}_1^{-1} \underline{\underline{U}}^H \underline{d}$

dann suche

$$\underline{x} = \underline{x}_0 + \underbrace{\underline{\underline{V}}_2 \underline{y}}_{\ker(\underline{\underline{C}}) = \mathrm{Bild}(\underline{\underline{V}}_2)} \quad \text{mit } \underline{y} \in \mathbb{R}^{n-p}$$

(und somit automatisch $\underline{\underline{C}}\underline{x} = \underline{d}$)

$$\|\underline{\underline{A}}\underline{x}-\underline{b}\|_2^2 = \|\underline{\underline{A}}\underline{x}_0 + \underline{\underline{A}}\underline{\underline{V}}_2\underline{y} - \underline{b}\|_2^2 =$$

$$= \|\underline{\underline{A}}\underline{\underline{V}}_2\underline{y} - (\underline{b}-\underline{\underline{A}}\underline{x}_0)\|_2^2$$

$\underline{\underline{A}}\underline{\underline{V}}_2\underline{y}$: min nach $\underline{y}$ — $(\underline{b}-\underline{\underline{A}}\underline{x}_0)$: fest

↑ standard (lin. Ausgleichsproblem.

§8.4. Totales lineares Ausgleichsproblem

total = Messfehler auch in den Messorten t_i

$\underline{\underline{A}}\underline{x} = \underline{b}$ mit Fehler in $\underline{b}$ auch in $\underline{\underline{A}}$

Ben $\underline{\underline{A}}\underline{x} = \underline{b}$ hat keine Lösung falls $\underline{b} \notin$ Bild$(\underline{\underline{A}})$

Ziel: finde $\hat{\underline{\underline{A}}}$, $\hat{\underline{x}}$, $\hat{\underline{b}}$ sodass $\hat{\underline{\underline{A}}}\hat{\underline{x}} = \hat{\underline{b}}$ (d.h. $\hat{\underline{b}} \in$ Bild$\hat{\underline{\underline{A}}}$)

$\hat{\underline{\underline{A}}}$ $m \times n$, $\hat{\underline{b}} \in \mathbb{R}^n$

"so nah wie möglich" an $\underline{\underline{A}}\underline{x} = \underline{b}$

$$\underline{\underline{C}} = [\underline{\underline{A}}\ \underline{b}]\ ;\quad \hat{\underline{\underline{C}}} = [\hat{\underline{\underline{A}}}\ \hat{\underline{b}}]$$

Finde $\hat{\underline{\underline{C}}}$ so dass $\|\underline{\underline{C}} - \hat{\underline{\underline{C}}}\|_F = \min$ Frobeniusnorm.

$$\|\underline{\underline{M}}\|_F^2 = \sum_{i,j} |m_{ij}|^2$$

mit $\hat{\underline{b}} \in$ Bild$(\hat{\underline{\underline{A}}})$.

Ben Rang $\underline{\underline{A}} = n \Rightarrow$ möchte auch Rang $(\hat{\underline{\underline{C}}}) = n$.

Lösung: Niedrigrangapproximation an $\underline{\underline{C}}$:

$$\underline{\underline{C}} = \underline{\underline{U}}\,\underline{\underline{\Sigma}}\,\underline{\underline{V}}^T = \sum_{j=1}^{n+1} \sigma_j\, \underline{u}_j \underline{v}_j^H$$

Definiere $\hat{\underline{\underline{C}}} = \sum_{j=1}^{n} \sigma_j \underline{u}_j \underline{v}_j^H$

↳ optimale Approx. an $\underline{\underline{C}}$ in der Menge der Matrizen von Rang n ist!

Ausserdem $\underline{v}_j$ sind Orthogonal $\Rightarrow \hat{\underline{\underline{C}}}\,\underline{v}_{n+1} = 0$

Falls $v_{n+1,n+1} \neq 0 \Rightarrow \hat{\underline{x}} = -\frac{1}{v_{n+1,n+1}}\,\underline{v}_{n+1}$

$\hat{\underline{b}} = \hat{\underline{\underline{A}}}\hat{\underline{x}}$

Beweis

$$\hat{\underline{A}}\,\hat{\underline{x}} = \hat{\underline{b}} \Leftrightarrow \hat{\underline{A}}\hat{\underline{x}} - \hat{\underline{b}} = 0 \Leftrightarrow \underbrace{[\hat{\underline{A}}\ \hat{\underline{b}}]}_{\hat{\underline{C}}} \begin{bmatrix} \hat{\underline{x}} \\ -1 \end{bmatrix} = 0$$

$$\Rightarrow \begin{bmatrix} \hat{\underline{x}} \\ -1 \end{bmatrix} \in \ker(\hat{\underline{C}}) \Rightarrow$$

$$\dim \ker(\hat{C}) = 1 \,, \quad \underline{v}_{n+1} \in \ker \hat{C}$$

$$\begin{bmatrix} \hat{\underline{x}} \\ -1 \end{bmatrix} = k\,\underline{v}_{n+1}$$

Definier $k = -\dfrac{1}{v_{n+1,n+1}} \Rightarrow \hat{\underline{x}} = -\dfrac{1}{v_{n+1,n+1}}\,\underline{v}_{n+1}$.

§8.5. Nichtlineare Ausgleichsrechnung

Modell $f(t, \underline{x}) = y$

↳ Parameter, aus Messungen zu bestimmen.

$f(t_i, \underline{x}) \approx y_i \,, \quad i = 1, 2, \ldots, m$

$$\underline{F}(\underline{x}) = \begin{bmatrix} f(t_1, \underline{x}) - y_1 \\ f(t_2, \underline{x}) - y_2 \\ \vdots \\ f(t_m, \underline{x}) - y_m \end{bmatrix}$$

$\underline{F} : \mathbb{R}^n \to \mathbb{R}^m$

Finde $\underline{x}^* \in \mathbb{R}^n$ sodass

$\|\underline{F}(\underline{x}^*)\|_2^2$ Minimal!

$\Phi(\underline{x}) = \frac{1}{2}\|\underline{F}(\underline{x})\|_2^2 \,, \quad \Phi : \mathbb{R}^n \to [0, \infty[$

zu minimieren!

Finde x^*: $D\Phi(\underline{x}^*) = 0$ ← Nullstellensuche!

→ mit Newton-Verfahren

↳ konvergiert lokal quadratisch.

$$D\Phi(x) = D\left(\tfrac{1}{2} F(x)^T F(x)\right) = DF(x)^T F(x)$$

$$D\Phi(x^*) = 0$$

$$DF(x^*)^T F(x^*) = 0$$

↑ dafür müssen wir Newton Verfahre anwenden.

In Newton brauchen wir die Ableitung davon.

$D(D\Phi(x)) = H\Phi(x)$ Hesse Matrix von Φ

$$H\Phi(x) = D\left(DF(x)^T F(x)\right) =$$

$$= DF(x)^T DF(x) + \sum_{j=1}^{m} F_j(x)\, D^2 F_j(x)$$

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 $x^{(k+1)} = x^{(k)} + \Delta$ mit

Newton-Korrektur Δ aus LGS

$$H\Phi(x^{(k)})\, \Delta = -DF(x^{(k)})^T F(x^{(k)})$$

Alternative: Gauss-Newton-Verfahren

Idee: linearisiere lokal in jeder Schritt einer Iteration ⇒ eine Folge von linearen Ausgleichsprobleme

linearisiere

$$\operatorname*{argmin}_{x \in \mathbb{R}^n} \|F(x)\|_2^2 \approx \operatorname*{argmin}_{x \in \mathbb{R}^n} \|F(x^{(0)}) + DF(x^{(0)})(x - x^{(0)})\|_2^2$$

$$= \operatorname*{argmin}_{x \in \mathbb{R}^n} \|Ax - b\|_2^2$$

mit $A = DF(x^{(0)})$ und $b = F(x^{(0)}) - DF(x^{(0)})\, x^0$

$\underline{x}^{(0)} \xrightarrow{\text{lin LSP}} \underline{x}^{(1)} \xrightarrow{\text{lin LSP}} \underline{x}^{(2)} \to \dots \to \underline{x}^{(k)}$

$\underline{x}^{(k+1)} = \underline{x}^{(k)} + \underline{\Delta}^{(k)}$, $\underline{\Delta}^{(k)} = \arg\min \| \underline{F}(\underline{x}^{(k)}) + \underline{\underline{D}}\underline{F}(\underline{x}^{(k)})\underline{\Delta} \|_2^2$

Nachteil: lineare Konvergenz.

Professionele Software: Levenberg-Marquadt

§9 Eigenwerte

§9.1. Motivation & Grundlagen

* kein Alg. die exakt die EW, EV berechnet
  ⇒ iterativ
* Software → eig($\underline{\underline{A}}$) kostet $O(N^3)$ für $N \times N$
  eigh($\underline{\underline{A}}$) kostet $O(N^2)$ für $\underline{\underline{A}}^H = \underline{\underline{A}}$
  ↳ QR-Algorithmus mit shift
* EV werden typischerweise nicht so gut approximiert!

§9.2 Potenzmethoden

$\underline{\underline{A}}$ $n \times n$ Matrix.

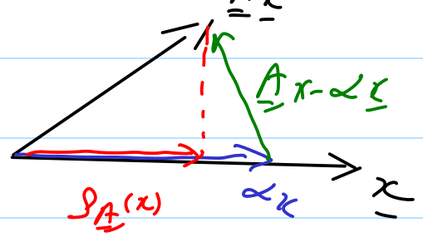
Def Rayleigh-Quotienten:

$$\rho_{\underline{\underline{A}}}(\underline{x}) = \frac{\underline{x}^H \underline{\underline{A}} \underline{x}}{\underline{x}^H \underline{x}}, \quad \underline{x} \neq \underline{0}$$

Bem

Falls $\underline{x}$ EV von $\underline{\underline{A}}$: $\underline{\underline{A}}\underline{x} = \lambda \underline{x}$

$$\rho_{\underline{\underline{A}}}(\underline{x}) = \frac{\underline{x}^H \lambda \underline{x}}{\underline{x}^H \underline{x}} = \lambda$$



Bem

$$\rho_{\underline{\underline{A}}}(\underline{x}) = \operatorname*{argmin}_{\alpha} \| \underline{\underline{A}}\underline{x} - \alpha \underline{x} \|_2$$

Ü: Beweis.

Bem

$$D\rho_{\underline{\underline{A}}}(\underline{x}) = \frac{(\underline{x}^T \underline{x})(\underline{\underline{A}}^T + \underline{\underline{A}})\underline{x} - 2(\underline{x}^T \underline{\underline{A}} \underline{x})\underline{x}}{(\underline{x}^T \underline{x})^2} = \frac{(\underline{\underline{A}}^T + \underline{\underline{A}}) - 2\rho_{\underline{\underline{A}}}(\underline{x})\underline{\underline{I}}}{(\underline{x}^T \underline{x})} \underline{x}$$

$\underline{x}$ EV von $\underline{\underline{A}}$ ⇒ $D\rho_{\underline{\underline{A}}}(\underline{x}) = \frac{(\underline{x}^T \underline{x}) 2\lambda \underline{x} - 2\lambda (\underline{x}^T \underline{x})\underline{x}}{(\underline{x}^T \underline{x})^2} = 0$

$\Rightarrow$ EV $\in$ ein stationärpunkt von $\rho_{\underline{\underline{A}}}(\cdot)$

Taylor für $\rho_{\underline{\underline{A}}}(\cdot)$ um ein EV:

$$\rho_{\underline{\underline{A}}}(\underline{x}) - \rho_{\underline{\underline{A}}}(EV) = O\left(\|\underline{x} - EV\|_2^2\right) \quad \text{für } \underline{x} \text{ nah am EV.}$$

Direkte Potenzmethode

Ziel: finde das betragsgrösste EW von $\underline{\underline{A}}$ und ein EV dazu

Annahme $\underline{\underline{A}}$ diagonalisierbar: $\underline{\underline{S}}^{-1}\underline{\underline{A}}\,\underline{\underline{S}} = \operatorname{diag}(\lambda_1, \dots, \lambda_n)$

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|$$

EV stehen in der Spalten von S, $\|\underline{s}_j\|_2 = 1$

$$\mathbb{R}^n \ni \underline{x} = \sum_{j=1}^{n} a_j \underline{s}_j \quad \text{mit} \quad a_1 \neq 0, \text{ sonst beliebig}$$

$$\underline{\underline{A}}\,\underline{x} = \sum_{j=1}^{n} a_j \underline{\underline{A}}\,\underline{s}_j = \sum_{j=1}^{n} a_j \lambda_j \underline{s}_j$$

$$\underline{\underline{A}}^2 \underline{x} = \sum_{j=1}^{n} a_j \lambda_j^2 \underline{s}_j$$

$$\dots$$

$$\underline{\underline{A}}^k \underline{x} = \sum_{j=1}^{n} a_j \lambda_j^k \underline{s}_j = \lambda_1^k \left(a_1 \underline{s}_1 + \sum_{j=2}^{n} \left(\frac{\lambda_j}{\lambda_1}\right)^k a_j \underline{s}_j\right)$$

$$\left|\frac{\lambda_j}{\lambda_1}\right| < 1 \Rightarrow \left|\frac{\lambda_j}{\lambda_1}\right|^k \ll 1$$

Bem Für grosses k zeigt $\underline{\underline{A}}^k \underline{x}$ in der Richtung von $\underline{s}_1$

$$\frac{\underline{\underline{A}}^k \underline{x}}{\|\underline{\underline{A}}^k \underline{x}\|_2} \longrightarrow \pm \underline{s}_1$$

Idee: notiere $\underline{x}_k = \underline{\underline{A}}^k \underline{x}$ und berechne $\rho_{\underline{\underline{A}}}(\underline{x}_k) =$

$$= \frac{x_k^H A x_k}{x_k^H x_k} = \frac{1}{x_k^H x_k}\left(x_k^H \sum_{j=1}^{n} a_j \lambda_j^{k+1} s_j \right) =$$

$$= \lambda_1 + O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)$$

Potenzmethode

Wähle x_0 zufällig, $\|x_0\| = 1$

für $k = 1, 2, \ldots$

$$w = A x_{k-1}$$

$$x_k = \frac{w}{\|w\|}$$

$$\lambda = x_k^H A x_k$$

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

↳ Grundlage für PageRank-Alg. von Google!

Beweis $A = A^H$ ONB von EV

$$x = \sum_{i=1}^{n} c_j v_j = U c \quad \text{mit } U \text{ unitär.}$$

$$\frac{x^H A x}{x^H x} = \frac{c^H U^H A U c}{c^H U^H U c} = \frac{c^H \Lambda c}{c^H c} =$$

$$= \frac{\lambda_1 |c_1|^2 + \lambda_2 |c_2|^2 + \ldots + \lambda_n |c_n|^2}{|c_1|^2 + \ldots + |c_n|^2} = \lambda_1 + \mathcal{O}\cdot\left|\frac{\lambda_2}{\lambda_1}\right|^{2k} + \ldots$$

Ziel Finde das kleinste EW:

Annahme $\underline{\underline{A}}$ invertierbar.

$$\underline{\underline{A}}\,\underline{x} = \lambda \underline{x} \Rightarrow \quad \underline{x} = \lambda \underline{\underline{A}}^{-1}\underline{x}$$

$\underline{\underline{A}}^{-1}|$

$$\frac{1}{\lambda} x = \underline{\underline{A}}^{-1}\underline{x}$$

$\Rightarrow$ betragskleinste EW: $\lambda_n \Rightarrow \quad \frac{1}{\lambda_n}\underline{x} = \underline{\underline{A}}^{-1}\underline{x}$

$\frac{1}{\lambda_n}$ ist der betragsgrösste EW von $\underline{\underline{A}}^{-1}$

$\Rightarrow$ "inverse Potenzmethode" = Potenzmethode für $\underline{\underline{A}}^{-1}$.

Bem Wir berechnen nicht $\underline{\underline{A}}^{-1}$ sondern nur ein mal eine LU-Zerlegung von $\underline{\underline{A}}$ (strukturerhaltend)

Dann Löse LGS mit Matrizen L, U um $\underline{\underline{A}}^{-1}\underline{x}$ zu implementieren.

Ziel Gegeben $\alpha \in \mathbb{C}$, finde EW nah an α

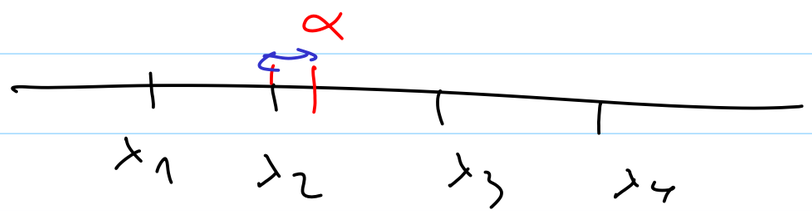
$|\alpha - \lambda| = \min\{|\alpha - \mu| \text{ mit } \mu \in EW \text{ von } \underline{\underline{A}}\}$

$$\underline{\underline{A}}\,\underline{x} = \lambda \underline{x} \Longleftrightarrow \underline{\underline{A}}\,\underline{x} - \alpha \underline{\underline{I}}\,\underline{x} = (\lambda - \alpha)\underline{x} \Leftrightarrow$$

$$(\underline{\underline{A}} - \alpha \underline{\underline{I}})\underline{x} = (\lambda - \alpha)\underline{x} \Leftrightarrow \frac{1}{\lambda - \alpha}\underline{x} = (\underline{\underline{A}} - \alpha \underline{\underline{I}})^{-1}\underline{x}$$

$\Rightarrow$ Potenzmethode für $(\underline{\underline{A}} - \alpha \underline{\underline{I}})^{-1} \Rightarrow \frac{1}{\lambda - \alpha} \Rightarrow \lambda$

"shifted inverse iteration"



Bem Die Potenzmethode ist schneller wenn $\alpha \approx \lambda_j$

$\Rightarrow$ Idee: wähle α adaptiv, z.B $\alpha = \rho_{\underline{\underline{A}}}(x^{(k-1)})$ im k-ten Schritt

$\Rightarrow$ beschleunigte Konvergenz

Rayleigh-Quotienten-Iteration.

Bem Wir brauchen immer einen guten Startvert.
z.B. für RQI einige Schritte von shifted inverse iteration.

$\Longrightarrow$ Konvergenzordnung 3!!!

$\underline{\underline{A}} = \underline{\underline{A}}^H$; $\underline{x} = \sum_{j=1}^{n} c_j \underline{u}_j = \underline{\underline{U}}\,\underline{c}$ mit $\underline{u}_1, \dots, \underline{u}_n \in V$

($\underline{x}$ beliebig.)

$$\rho_{\underline{\underline{A}}}(\underline{x}) = \frac{\underline{x}^H \underline{\underline{A}}\,\underline{x}}{\underline{x}^H \underline{x}} = \frac{\lambda_1 |c_1|^2 + \dots + \lambda_n |c_n|^2}{\|\underline{x}\|_2^2}$$

$(\lambda_1 \le \lambda_2 \le \dots \le \lambda_n)$

$$\lambda_1 \le \rho_{\underline{\underline{A}}}(\underline{x}) \le \lambda_n$$

$$\rho_{\underline{\underline{A}}}(\underline{x}) \in [\lambda_1, \lambda_n]$$

$\lambda_1 = \min_{\underline{x} \in \mathbb{C}^n} \rho_{\underline{\underline{A}}}(\underline{x})$ erreicht $\underline{c} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

$\lambda_n = \max_{\underline{x} \in \mathbb{C}^n} \rho_{\underline{\underline{A}}}(\underline{x})$ erreicht $\underline{c} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$

Bem $\underline{x} = \operatorname{span}\{\underline{u}_1\}^{\perp} \Rightarrow c_1 = 0$, $\underline{x} = c_2 \underline{u}_2 + \dots + c_n \underline{u}_n$

$\Downarrow$

$\rho_{\underline{\underline{A}}}(x) \ge \lambda_2$ erreicht für $\underline{c} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

$\lambda_2 = \min_{\underline{x} \in \mathbb{C}^n} \rho_{\underline{\underline{A}}}(\underline{x})$; $\lambda_{n-1} = \max_{\underline{x} \in \mathbb{C}^n} \rho_{\underline{\underline{A}}}(\underline{x})$

$\underline{x} \in \operatorname{span}\{\underline{u}_1\}^{\perp}$ $\underline{x} \in \operatorname{span}\{\underline{u}_n\}^{\perp}$

und so weiter

$\lambda_k = \min_{\underline{x} \in \operatorname{span}\{\underline{u}_1, \dots, \underline{u}_{k-1}\}^{\perp}} \rho_{\underline{\underline{A}}}(\underline{x}) = \min_{\underline{x} \in \{\underline{u}_k, \dots, \underline{u}_n\}^{\perp}} \rho_{\underline{\underline{A}}}(\underline{x}) =$

$= \max_{\underline{x} \in \operatorname{span}\{\underline{u}_{k+1}, \dots, \underline{u}_n\}^{\perp}} \rho_{\underline{\underline{A}}}(\underline{x}) = \max_{\underline{x} \in \operatorname{span}\{\underline{u}_1, \dots, \underline{u}_k\}} \rho_{\underline{\underline{A}}}(\underline{x})$

Theorem [Courant-Fischer]

$$\lambda_k = \min_{\dim U = k} \max_{x \in U} \frac{x^H A x}{x^H x} = \max_{\dim U = n-k+1} \min_{x \in U} \frac{x^H A x}{x^H x}$$

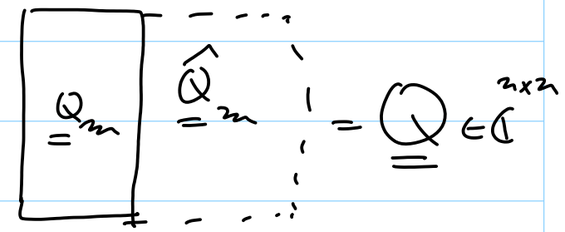
Theorem [Cauchy]

$$A = \begin{bmatrix} H & B^H \\ B & R \end{bmatrix} \quad \text{mit } H \in \mathbb{C}^{m \times m} \text{ mit Eigenwerten } \theta_1, \dots, \theta_m \quad (\text{mit } m < n)$$

Dann $\lambda_k \le \theta_k \le \lambda_{k+n-m}$

Konsequenz $Q_m \in \mathbb{C}^{n \times m}$ mit $Q_m^H Q_m = I$

erweitere Q_m auf ONB in $\mathbb{C}^n$



$$Q^H A Q = \begin{bmatrix} Q_m^H A Q_m & Q_m^H A \hat{Q}_m \\ \hat{Q}_m^H A Q_m & \hat{Q}_m^H A \hat{Q}_m^H \end{bmatrix}$$

hat dieselben EW wie A

Idee Für $m < n$, wähle Q_m so dass

Bild $Q_m \approx \operatorname{span}\{u_1, \dots, u_m\}$

$\to$ $\to$ EV von A

$\Rightarrow$ gute Approximation $\theta_k \approx \lambda_k$ für $k = 1, \dots, m$.

§ 9.3 Krylov-Verfahren

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)

für "kleine" Matrizen eig (QR-Alg.) gut

für "grosse" Matrizen eig zu langsam

ausserdem QR-Alg. zerstört Struktur

Krylov-Verfahren: grosse, dünnbesetzte Matrizen

Def Sei $\underline{0} \neq \underline{z} \in \mathbb{C}^n$, $\underline{A} \in \mathbb{C}^{n\times n}$

$$\mathcal{K}_l(\underline{A}, \underline{z}) := \operatorname{span}\{\underline{z}, \underline{A}\underline{z}, \underline{A}^2\underline{z}, \dots, \underline{A}^{l-1}\underline{z}\} =$$

$$= \{ p(\underline{A})\,\underline{z} \,;\, p = \text{Polynom vom Grad} \le l-1 \}$$

Krylov-Raum

Suche eine ONB in $\mathcal{K}_l(\underline{A}, \underline{z})$

$$\mathcal{K}_1 = \operatorname{span}\{\underline{z}\} \subset \operatorname{span}\{\underline{z}, \underline{A}\underline{z}\} = \mathcal{K}_2 \subset \mathcal{K}_3 \subset \dots \subset \mathcal{K}_l$$

Iterativ: gegeben $\underline{v}_1, \dots, \underline{v}_l$ ONB in $\mathcal{K}_l(\underline{A}, \underline{z})$

mit

$$\operatorname{span}\{\underline{v}_1, \dots, \underline{v}_j\} = \mathcal{K}_j(\underline{A}, \underline{z}) \text{ für } j = 1, 2, \dots, l$$

baue

$\underline{v}_1, \dots, \underline{v}_l, \underline{v}_{l+1}$ ONB in $\mathcal{K}_{l+1}(\underline{A}, \underline{z})$

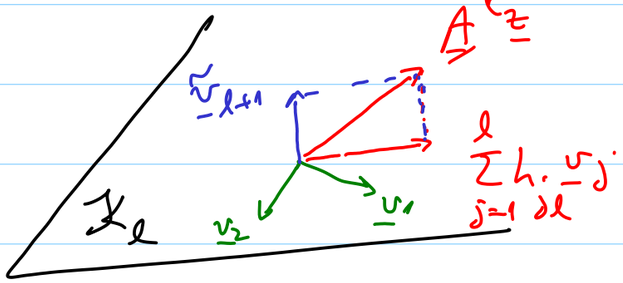
$$\mathcal{K}_{l+1}(\underline{A}, \underline{z}) = \operatorname{span}\{\underline{z}, \underline{A}\underline{z}, \dots, \underline{A}^{l-1}\underline{z}, \underline{A}^l\underline{z}\}$$

entweder $\underline{A}^l\underline{z} \in \mathcal{K}_l(\underline{A}, \underline{z})$, d.h.

$\underline{A}^l\underline{z}$ lin. abhängig von $\underline{z}, \underline{A}\underline{z}, \dots, \underline{A}^{l-1}\underline{z}$

oder $\underline{A}^l\underline{z} \notin \mathcal{K}_l(\underline{A}, \underline{z}) \Rightarrow \underline{A}^l\underline{z} \in \mathcal{K}_{l+1} \setminus \mathcal{K}_l$

dann $\underline{v}_{l+1}$ aus (modifiziertem) Gram-Schmidt



$$\tilde{v}_{l+1} = A^l z - \sum_{j=1}^{l} h_{jl} v_j$$

$$v_{l+1} = \frac{\tilde{v}_{l+1}}{\|\tilde{v}_{l+1}\|}$$

Dabei sind $h_{jl} = v_j^H A v_l$

(da $v_1, \dots, v_l$ ONB von K_l)

Algorithmus [Arnoldi Prozess]

z = beliebig

$v_1 = z / \|z\|$

für $l = 1, 2, \dots, k-1$: $\quad l=1 \quad l=2 \quad l=3$

$\tilde{v} = A v_l$

für $j = 1, 2 \dots, l$: $\quad h_{11} \quad h_{12} \quad h_{13}$

$h_{jl} = v_j^H \tilde{v}$ $\quad h_{22} \quad h_{23}$

mod. GS: $\tilde{v} = \tilde{v} - h_{jl} v_j$ $\quad h_{33}$

$h_{21} \quad h_{32} \quad h_{43}$

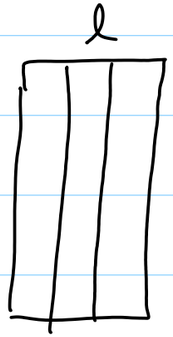
$h_{l+1,l} = \|\tilde{v}\|$

$v_{l+1} = \tilde{v} / h_{l+1,l}$

Bem Falls $h_{l+1,l} = 0 \Rightarrow$ Abbruch der Iteration

$$\underline{\underline{A}}\,\underline{v}_l \in \mathcal{K}_l(\underline{\underline{A}}, \underline{z})$$

$$\underline{\underline{V}}_l = \left[\, \underline{v}_1 \;\; \underline{v}_2 \;\cdots\; \underline{v}_l \,\right] \in \mathbb{C}^{n \times l}$$



$$\underline{\underline{\tilde{H}}}_l = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & h_{32} & h_{33} \\ & & h_{43} \\ 0 & 0 & \end{bmatrix} \in \mathbb{C}^{l+1,\,l}$$

$\underline{\underline{H}}_l \in \mathbb{C}^{l,l}$

$\underline{\underline{H}}_l$ obere Hessenbergmatrix

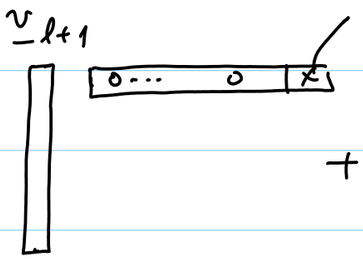
Aus Konstruktion:

$$\underline{\underline{A}}\,\underline{v}_k = h_{k+1,k}\,\underline{v}_{k+1} + \sum_{j=1}^{k} h_{jk}\,\underline{v}_j$$

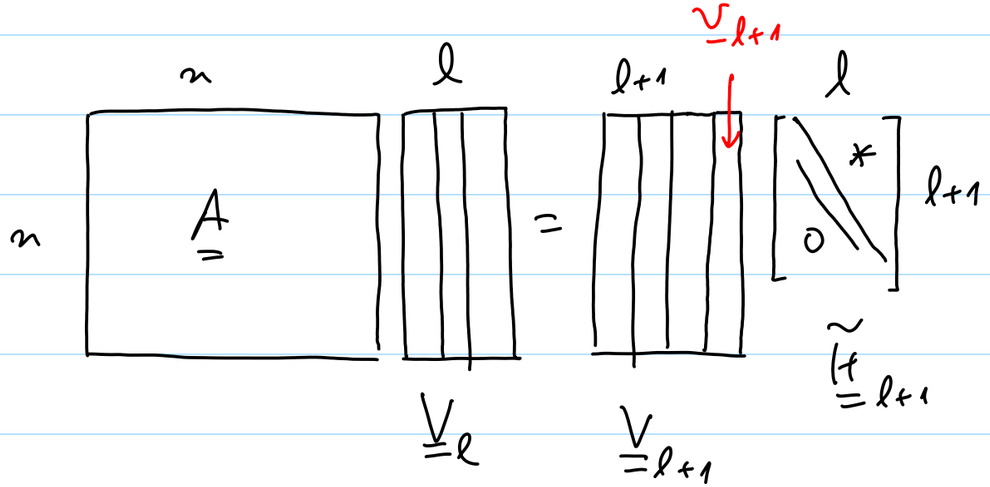
für $k = 1, 2, \ldots, l$

das heisst:

$$\underline{\underline{A}}\,\underline{\underline{V}}_l = \underline{\underline{V}}_{l+1}\,\underline{\underline{\tilde{H}}}_l =$$

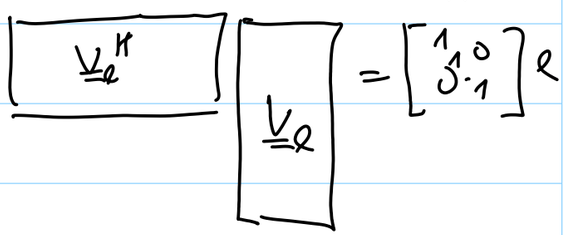


$$+\,\underline{\underline{V}}_l\,\underline{\underline{H}}_l$$



$$\underline{\underline{\tilde{H}}}_{l+1}$$

Bem 1) $\underline{\underline{V}}_l^H\,\underline{\underline{V}}_l = \underline{\underline{I}}_l$

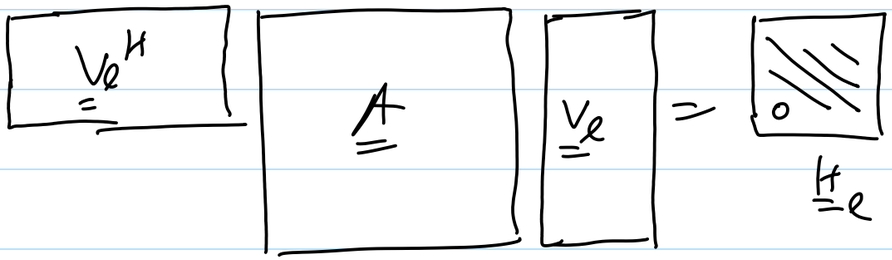


2)

$\underline{V}_\ell^H \cdot |$ $\Rightarrow$ $\underline{V}_\ell^H \underline{A} \underline{V}_\ell = \underline{V}_\ell^H \underline{v}_{\ell+1}$ [0 ... 0 x] $+ \underline{V}_\ell^H \underline{V}_\ell \underline{H}_\ell$

$= \underline{0} + \underline{I}_\ell \underline{H}_\ell = \underline{H}_\ell$

$\Rightarrow \underline{V}_\ell^H \underline{A} \underline{V}_\ell = \underline{H}_\ell$



3) falls $h_{\ell+1,\ell} = 0 \Rightarrow \mathcal{K}_{\ell+1} = \mathcal{K}_\ell$ und

$\underline{A} \underline{V}_\ell = \underline{V}_\ell \underline{H}_\ell$

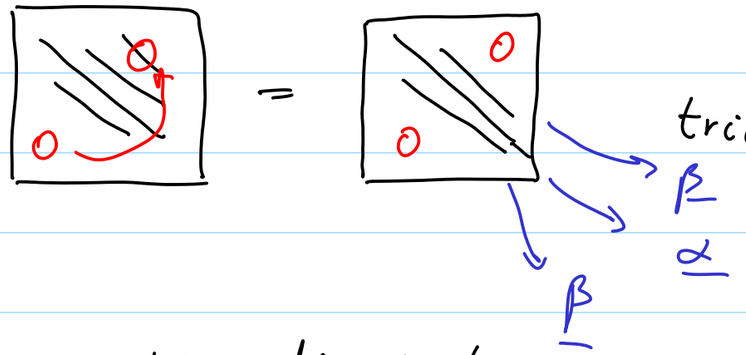
4) falls $\underline{A}^H = \underline{A}$ ($\underline{A}$ Hermite-symmetrisch):

$\underline{H}_\ell^H = (\underline{V}_\ell^H \underline{A} \underline{V}_\ell)^H = \underline{V}_\ell^H \underline{A}^H \underline{V}_\ell = \underline{V}_\ell^H \underline{A} \underline{V}_\ell = \underline{H}_\ell$

$\Rightarrow \underline{H}_\ell$ Hermite symmetrisch und obere Hessenberg

$\Downarrow$

tridiagonal



$\Rightarrow$ es reicht, die Vektoren α, β um $\underline{H}_\ell$ zu speichern.

und die innere Schleife im Arnoldi-Prozess hat die Länge 2

$$\tilde{\underline{v}}_{\ell+1} = \underline{A} \underline{v}_\ell - h_{\ell\ell} \underline{v}_\ell - h_{\ell-1,\ell} \underline{v}_{\ell-1}$$

Lanczos-Verfahren $O(nk)$

Arnoldi-Verfahren $O(nk^2)$

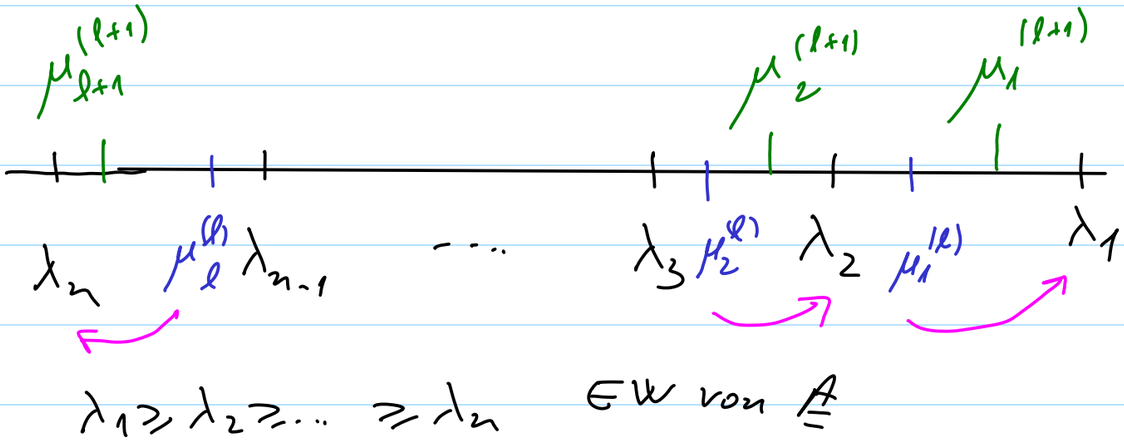
Allgemeinen Namen: Krylov-Raum-Verfahren

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

$$\mu_j^{(\ell)} \leq \mu_j^{(\ell+1)} \leq \lambda_j \,.$$



$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ $\in$ EW von $\underline{\underline{A}}$

$\mu_1^{(\ell)} \geq \mu_2^{(\ell)} \geq \ldots \geq \mu_\ell^{(\ell)}$ $\in$ EW von $\underline{\underline{H}}_\ell$

$\mu_1^{(\ell+1)} \geq \mu_2^{(\ell+1)} \geq \ldots \geq \mu_{\ell+1}^{(\ell+1)}$ $\in$ EW von $\underline{\underline{H}}_{\ell+1}$

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 $\underline{\underline{H}}_\ell$ ist auch EW von $\underline{\underline{A}}$

(2) falls $\underline{\underline{A}}$ regulär, dann gibt es $\underline{y} \in \mathbb{C}^\ell$ so dass $\underline{\underline{A}}\,\underline{x} = \underline{b}$ mit $\underline{x} = \underline{\underline{V}}_\ell \underline{y}$

skript: einfache Implementierung.

ARPACK → eigvals

§10 Lineare Anfangswertprobleme

1. Fall

$$\begin{cases} \dot{\underline{y}} = \underline{\underline{A}}\,\underline{y} \\ \underline{y}(0) = \underline{y}_0 \end{cases}$$

Falls $\underline{\underline{A}}$ diagonalisierbar

$$\underline{\underline{A}} = \underline{\underline{S}}^{-1}\,\underline{\underline{D}}\,\underline{\underline{S}}$$

Variablen wechsel $\hat{\underline{y}} = \underline{\underline{S}}^{-1}\underline{y}$ $\Rightarrow$ entkoppeln

$$\begin{cases} \dot{\hat{y}}_1 = \lambda_1 \hat{y}_1 \\ \dots \\ \dot{\hat{y}}_d = \lambda_d \hat{y}_d \end{cases} \Rightarrow \quad \hat{y}_i(t) = \left(\underline{\underline{S}}^{-1}\underline{y}_0\right)_i e^{\lambda_i t} \quad \text{für } t \in \mathbb{R}$$

$$\underline{y}(t) = \underline{\underline{S}} \begin{bmatrix} e^{\lambda_1 t} & & \\ & \ddots & \\ & & e^{\lambda_d t} \end{bmatrix} \underline{\underline{S}}^{-1}\underline{y}_0$$

ok für kleines d oder für exakt/analytisch diagonalisierbare $\underline{\underline{A}}$, sonst (z.B. $d \geq 5$) $\rightsquigarrow$ instabil.

Für $d = 5, 6, \dots, 20\,;\, 50\,,\, 100$

$$\underline{y}(t) = e^{\underline{\underline{A}}t}\,\underline{y}_0$$ Padé-Approximation

↳ expm$(\underline{\underline{A}}t)\,\underline{y}_0$

d gross, $\underline{\underline{A}}$ dünnbesetzt: Krylov-Verfahren.

Krylov: für $\underline{\underline{A}}$ gibt es $\underline{\underline{V}} \in \mathbb{C}^{d\times m}$ mit orthonormalen Spalten

$$\underline{\underline{V}}_m^H \underline{\underline{A}}\,\underline{\underline{V}}_m = \underline{\underline{H}}_m \quad m\times m \text{ mit } m \ll d$$

↳ obere Hessenberg Matrix.

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

$\underline{y}(t) \approx u_m(t) \in \mathcal{K}_m(\underline{\underline{A}}, \underline{y}_0) = \operatorname{span}\{\underline{y}_0, \underline{\underline{A}}\underline{y}_0, \dots, \underline{\underline{A}}^{m-1}\underline{y}_0\}$

$= \operatorname{span}\{\underline{v}_1, \dots, \underline{v}_m\}$ (Spalten von $\underline{\underline{V}}$)

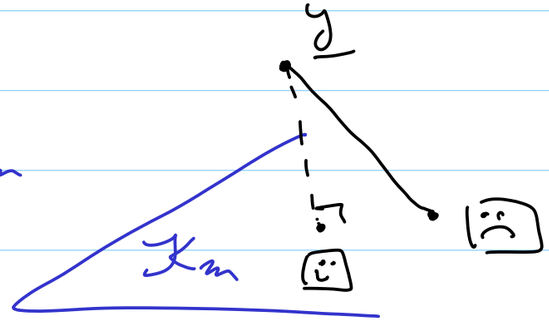
ONB in $\mathcal{K}_m$

$\dot{\underline{y}}(t) - \underline{\underline{A}}\,\underline{y}(t) = 0$

$\left(\dot{\underline{u}}_m(t) - \underline{\underline{A}}\,\underline{u}_m(t)\right) \perp \mathcal{K}_m$

$\Updownarrow$

$\langle \underline{w}, \dot{\underline{u}}_m(t) - \underline{\underline{A}}\,\underline{u}_m(t)\rangle = 0$ für alle $\underline{w} \in \mathcal{K}_m$.



Ersetze $\begin{cases} \dot{\underline{y}}(t) = \underline{\underline{A}}\,\underline{y}(t) \\ \underline{y}(0) = \underline{y}_0 \end{cases}$ durch $\begin{cases} \text{finde } \underline{u}_m(t) \in \mathcal{K}_m \text{ so dass} \\ \langle \underline{w}, \dot{\underline{u}}_m(t) - \underline{\underline{A}}\,\underline{u}_m(t)\rangle = 0 \\ \text{für alle } \underline{w} \in \mathcal{K}_m \end{cases}$

$\underline{u}_m(t) \in \mathcal{K}_m = \operatorname{span}\{\underline{v}_1, \dots, \underline{v}_m\} \Rightarrow$

$\underline{u}_m(t) = \sum_{k=1}^{m} c_k(t)\,\underline{v}_k \qquad \underline{c}(t) = \begin{bmatrix} c_1(t) \\ \vdots \\ c_m(t) \end{bmatrix} \in \mathbb{C}^m$

Einsetzen $\Rightarrow$

$\left\langle \underline{w}, \sum_{k=1}^{m} \dot{c}_k(t)\,\underline{v}_k - \sum_{k=1}^{m} c_k(t)\,\underline{\underline{A}}\,\underline{v}_k \right\rangle = 0$

für alle $\underline{w} \in \mathcal{K}_m(\underline{\underline{A}}, \underline{y}_0)$

Wähle $\underline{w} = \underline{v}_1 \Rightarrow$

$\left\langle \underline{v}_1, \sum_{k=1}^{m} \dot{c}_k(t)\,\underline{v}_k \right\rangle = \left\langle \underline{v}_1, \sum_{k=1}^{m} c_k(t)\,\underline{\underline{A}}\,\underline{v}_k \right\rangle$

$\sum_{k=1}^{m} \dot{c}_k(t) \langle \underline{v}_1, \underline{v}_k\rangle = \sum_{k=1}^{m} c_k(t) \langle \underline{v}_1, \underline{\underline{A}}\,\underline{v}_k\rangle$

$\underline{v}_1^H \underline{v}_k = 0$ falls $k \neq 1$, 1 für $k = 1$ $\qquad$ $\underline{v}_1^H \underline{\underline{A}}\,\underline{v}_k = (H_m)_{1k}$

$\Rightarrow \dot{c}_1(t) = \sum_{k=1}^{m} (H_m)_{1k}\, c_k(t)$

Für $\underline{w} = \underline{v}_2, \dots, \underline{v}_m$ 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 $\Downarrow$ kann mit Padé lösen!

$$\Rightarrow \underline{c}(t) = \operatorname{expm}(\underline{\underline{H}}_m t)\, \underline{c}(0)$$

$$\underline{u}_m(t) = \sum_{k=1}^{m} c_k(t)\, \underline{v}_k = \|\underline{y}_0\|\, \underline{\underline{V}}_m\, e^{\underline{\underline{H}}_m t} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

2 Fall $\dot{\underline{y}}(t) = \underline{\underline{A}}\, y(t) + \underline{g}(t)$ inhomogener Fall

Variation der Konstanten:

$$\underline{y}(t) = e^{\underline{\underline{A}}(t-t_0)}\, \underline{y}_0 + \int_{t_0}^{t} e^{\underline{\underline{A}}(t-\tau)}\, \underline{g}(\tau)\, d\tau .$$

3. Fall

$\dot{\underline{y}}(t) = \underline{\underline{A}}(t)\, \underline{y}(t)$ Magnus-Integratoren

Unter bestimmten Voraussetzungen

$$\underline{y}(t) = e^{\underline{\underline{\Omega}}(t,t_0)}\, \underline{y}_0$$ Magnus Entwicklung

$$\underline{\underline{\Omega}} = \sum_{k=1}^{\infty} \underline{\underline{\Omega}}_k$$

$[A,B] = AB - BA$

$$\begin{aligned} \Omega_1 &= \int_{t_0}^{t} \mathbf{A}(\tau_1) d\tau_1 , \\ \Omega_2 &= \frac{1}{2} \int_{t_0}^{t} \int_{t_0}^{\tau_1} [\mathbf{A}(\tau_1), \mathbf{A}(\tau_2)] d\tau_2 d\tau_1 , \\ \Omega_3 &= \frac{1}{12} \int_{t_0}^{t} \int_{t_0}^{\tau_1} \int_{t_0}^{\tau_2} [[\mathbf{A}(\tau_1), \mathbf{A}(\tau_2)], \mathbf{A}(\tau_3)] + [\mathbf{A}(\tau_1), [\mathbf{A}(\tau_2), \mathbf{A}(\tau_3)]] d\tau_3 d\tau_2 d\tau_1 . \end{aligned}$$

$$\left[\underline{\underline{A}}(\tau_1), \underline{\underline{A}}(\tau_2) \right] = \underline{\underline{A}}(\tau_1)\underline{\underline{A}}(\tau_2) - \underline{\underline{A}}(\tau_2)\underline{\underline{A}}(\tau_1)$$

Kommutator

Idee Statt $\dot{y} = \underline{\underline{A}}(t)\, y$ löse $\dot{\hat{y}} = \underline{\underline{\hat{A}}}(t)\, \hat{y}$

wobei $\underline{\underline{\hat{A}}}(t) = \sum_{i=1}^{s} l_i(t)\, \underline{\underline{A}}(t_n + c_i h)$

$l_i(t)$ = Lagrange-Polynom in $t_n + c_i h$

c_i = Quadraturknoten in $[0,1]$

$t_n + c_i h \in [t_n, t_n + h]$

$\underline{\underline{A}}(t)$ = Polynom vom Grad s in t auf $[t_n, t_n + h]$

$\underline{\underline{\hat{A}}}(t_n + c_i h) = \underline{\underline{A}}(t_n + c_i h)$ für $i = 1, 2, \dots, s$

Dann Magnus-Entwicklung für $\dot{\hat{y}} = \underline{\underline{\hat{A}}}(t)\, \hat{y}$

Vorteil: Integration nur für Polynome in t $\Rightarrow$ einfach exakt berechnen.

$\underline{\underline{\hat{A}}}(t)$ 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 $p \geq s$

$$y(t_n + h) - \hat{y}(t_n + h) = O(h^{p+1})$$

* Siehe Methoden & Beispiel im Skript!

§11. Exponentiele Integratoren

$$\begin{cases} \dot{\underline{y}} = \underline{f}(\underline{y}) & \text{autonom} \quad \text{mit } \underline{f} \text{ stetig differentierbar} \\ \underline{y}(0) = \underline{y}_0 \end{cases}$$

Idee der Linearisierung: $\underline{\underline{J}} = \underline{\underline{D}}\underline{f}(\underline{y}_0)$

$$\dot{\underline{y}} = \underline{\underline{J}}\underline{y} + \underline{f}(\underline{y}) - \underline{\underline{J}}\underline{y}$$

linear $\quad$ $\underline{g}(\underline{y})$

Variation der Konstanten:

$$\underline{y}(h) = e^{\underline{\underline{J}}h}\underline{y}_0 + \int_0^h e^{\underline{\underline{J}}(h-s)}\underline{g}(\underline{y}(s))\,ds$$

Ersetze $\underline{y}(s)$ durch $\underline{y}_0 \Rightarrow$ "Quadratur/Approximation"

$$\int_0^h e^{\underline{\underline{J}}(h-s)}\underline{g}(\underline{y}(s))\,ds \approx \int_0^h e^{\underline{\underline{J}}(h-s)}\underline{g}(\underline{y}_0)\,ds$$

$$= h\,\varphi(h\underline{\underline{J}})\,\underline{g}(\underline{y}_0)$$

$$\varphi(z) \overset{\text{def}}{=} \frac{e^z - 1}{z} = \sum_{n=1}^{\infty} \frac{1}{n!} z^{n-1} = \sum_{n=0}^{\infty} \frac{z^n}{(n+1)!}$$

für Matrizen: $\varphi(\underline{\underline{A}}) = \sum_{n=0}^{\infty} \frac{\underline{\underline{A}}^n}{(n+1)!} = (e^{\underline{\underline{A}}} - \underline{\underline{I}})\underline{\underline{A}}^{-1}$

Beweis

$$\int_0^h e^{\underline{\underline{J}}(h-s)}\underline{g}(\underline{y}_0)\,ds = \int_0^h \sum_{n=0}^{\infty} \frac{1}{n!}\left(\underline{\underline{J}}(h-s)\right)^n \underline{g}(\underline{y}_0)\,ds =$$

Umtauschen

$$= \sum_{n=0}^{\infty} \frac{1}{n!}\underline{\underline{J}}^n (-1)\frac{(h-s)^{n+1}}{n+1}\Big|_{s=0}^{s=h} \underline{g}(\underline{y}_0) = \quad \left(\int_0^h (h-s)^n ds\right)$$

$$\sum_{n=0}^{\infty} \frac{1}{n!} \underline{\underline{J}}^{n} \frac{h^{n+1}}{n+1} \underline{g}(\underline{y}_0) = h \sum_{n=0}^{\infty} \frac{1}{(n+1)!} \underline{\underline{J}}^{n} h^{n} \underline{g}(\underline{y}_0)$$

$$= h\, \underline{\underline{f}}(\underline{\underline{J}} h)\, \underline{g}(\underline{y}_0) .$$

Bem Definition von $\underline{g}(\underline{y}_0) = \underline{f}(\underline{y}_0) - \underline{\underline{J}}\,\underline{y}_0$

$$h\, \underline{\underline{f}}(\underline{\underline{J}} h)\, \underline{g}(\underline{y}_0) = h\, \underline{\underline{f}}(\underline{\underline{J}} h)\, \underline{f}(\underline{y}_0) - h\, \underline{\underline{f}}(\underline{\underline{J}} h) \cdot \underline{\underline{J}}\,\underline{y}_0$$

$$h\, \underline{\underline{f}}(\underline{\underline{J}} h)\, \underline{\underline{J}}\,\underline{y}_0 = h \sum_{n=0}^{\infty} \frac{1}{(n+1)!} \underline{\underline{J}}^{n} h^{n}\, \underline{\underline{J}}\,\underline{y}_0 = e^{\underline{\underline{J}} h}\, \underline{y}_0 - \underline{y}_0$$

$$h\, \underline{\underline{f}}(\underline{\underline{J}} h)\, \underline{g}(\underline{y}_0) = h\, \underline{\underline{f}}(\underline{\underline{J}} h)\, \underline{f}(\underline{y}_0) - e^{\underline{\underline{J}} h}\, \underline{y}_0 + \underline{y}_0$$

Somit bekommen wir

$$\underline{y}(h) \approx e^{\underline{\underline{J}} h}\, \underline{y}_0 + h\, \underline{\underline{f}}(\underline{\underline{J}} h)\, \underline{f}(\underline{y}_0) - e^{\underline{\underline{J}} h}\, \underline{y}_0 + \underline{y}_0$$

$$\Rightarrow \quad \underline{y}(h) \approx \underline{y}_0 + h\, \underline{\underline{f}}(\underline{\underline{J}} h)\, \underline{f}(\underline{y}_0)$$

exponentielles Eulerverfahren.

$$\underline{\underline{J}} = \underline{\underline{D}} \underline{f}(\underline{y}_0)$$

$$\underline{\underline{f}}(h\underline{\underline{J}}) = (e^{h\underline{\underline{J}}} - \underline{\underline{I}})(h\underline{\underline{J}})^{-1}$$

teuer für grosses d — $O(d^3)$

Bem Für grosses d kann man das Krylov-Verfahren für $e^{h\underline{\underline{J}}}$ verwenden!

$$f(\underline{\underline{A}})\,\underline{b} = \underline{\underline{V}}_m\, f(\underline{\underline{H}}_m) \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

aus Krylov-Verfahren mit $\mathcal{K}_m(\underline{\underline{A}}, \underline{b})$

Bem Stabilitätsfunktion $S(z) = e^z$

$\Rightarrow$ das ideale Stabilitätsgebiet!

da exakt für $\dot{\underline{y}} = \underline{\underline{A}}\,\underline{y} + \underline{g}$

↳ konstante

Verallgemeinerung: exponentielle RK-Verfahren.

$$\underline{\underline{J}} = \underline{\underline{D}} f(x)$$

semi-implizite Euler: $\underline{y}_1 = \underline{y}_0 + (\underline{\underline{I}} - h\underline{\underline{J}})^{-1} h\, \underline{f}(\underline{y}_0)$

exponentiele Euler: $\underline{y}_1 = \underline{y}_0 + \varphi(h\underline{\underline{J}})\, h\, \underline{f}(y_0)$

Idee: ersetze $\frac{1}{1-z}$ durch $f(z) = \frac{e^z - 1}{z}$ in Row $\Rightarrow$

$$\begin{cases} \underline{k}_i = f(\gamma h \underline{\underline{J}}) \left(f(\underline{u}_i) + h\underline{\underline{J}} \sum_{j=1}^{i-1} \gamma_{ij}\, \underline{k}_j \right) \\ \underline{u}_i = \underline{y}_0 + h \sum_{j=1}^{i-1} \alpha_{ij}\, \underline{k}_j \\ \underline{y}_1 = \underline{y}_0 + h \sum_{i=1}^{s} b_i\, \underline{u}_i \end{cases}$$

explizite RK: $f(z) = 1$ und $\underline{\underline{J}} = \underline{\underline{0}}$

Row : $f(z) = \frac{1}{1-z}$

exp. RK : $f(z) = \frac{e^z - 1}{z}$

ODEs	nicht steif	steif	ostillierend
Methode	expl. RK	impl. RK	Exponentielle RK
	ode45	ode23s	exp4
Stabilität	$h < \frac{1}{\lambda}$	alle h	alle h
Implement.	$f(y)$	$Df(y)$	$Df(y)$
		Lösen nicht.-lin. Gleichungssysteme	$\varphi(A)b$ schnell mit Krylov.

⊕ Erhaltungseigenschaften wichtig?

⊕ Autonom?

⊕ Ordnung der Dgl.?