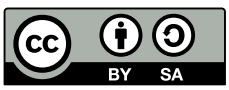
Numerik

Oliver Sander

13. Juli 2017

Getippt und gesetzt mit viel Hilfe von Johannes R. Stojanow



Inhaltsverzeichnis

1 Numerische Lösung nichtlinearer Gleichungen

1.1 Fixpunktiterationen im eindimensionalen Fall

Sei $f\colon \mathbb{R} \to \mathbb{R}$ eine Funktion. Wir interessieren uns für Lösungen x der Gleichung

$$f(x) = 0$$

Die Idee der Fixpunktiteration besteht darin, diese Gleichung äquivalent in eine *Fixpunktgleichung*

$$g(x) = x$$

umzuformen und mit Hilfe der Iterationsvorschrift

$$x_{k+1} = g\left(x_k\right) \qquad k = 0, 1, 2, \ldots$$

für einen gegebenen Startwert x_0 eine Folge $(x_0, x_1, \ldots)$ zu konstruieren.
Wunsch: Die Folge (x_k) konvergiert gegen einen *Fixpunkt* x^* mit $g\left(x^*\right) = x^*$, welcher auch Lösung der nichtlinearen Gleichung ist, d.h.

$$f\left(x^*\right) = 0$$

Die Existenz von Fixpunkten folgt z.B. aus dem Fixpunktsatz von Banach.

Satz 1.1. *Sei $I = [a, b] \subset \mathbb{R}$ ein Intervall und $g\colon I \to I$ eine kontrahierende Abbildung mit einer Lipschitz-Konstante $L < 1$. Dann folgt:*

(a) Es existiert genau ein Fixpunkt x^* von g, also $\exists! x^* \in I\colon g\left(x^*\right) = x^*$

(b) Für jeden Startwert $x_0 \in I$ konvergiert die Fixpunktiteration $x_{k+1} = g\left(x_k\right)$ gegen x^* mit $|x_{k+1} - x_k| \leq L|x_k - x_{k-1}|$ und

$$|x^* - x_k| \leq \frac{L^k}{1 - L}|x_1 - x_0|$$

Beweis. $\forall x_0 \in I\colon |x_{k+1} - x_k| = |g\left(x_k\right) - g\left(x_{k-1}\right)| \leq L|x_k - x_{k-1}|$. Induktiv: $|x_{k+1} - x_k| \leq L^k|x_1 - x_0|$. Wir wollen zeigen, dass (x_k) eine Cauchy-Folge ist und betrachten

$$\begin{aligned}
|x_{k+m} - x_k| &\leq |x_{k+m} - x_{k+m-1}| + \ldots + |x_{k+1} - x_k| \\
&\leq \underbrace{\left(L^{k+m-1} + L^{k+m-2} + \ldots + L^k\right)}_{=L^K(1+L+\ldots+L^{m-1})}|x_1 - x_0| \\
&\leq \frac{L^k}{1 - L}|x_1 - x_0|
\end{aligned}$$

Damit ist gezeigt, dass (x_k) eine Cauchy-Folge ist, welche in $\mathbb{R}$ gegen den Häufungspunkt

$$x^* = \lim_{k \longrightarrow \infty} x_k$$

konvergiert. Der Punkt x^* ist aber auch Fixpunkt von g, da

$$\begin{aligned} |x^* - g(x^*)| &= |x^* - x_{k+1} + x_{k+1} - g(x^*)| \\ &= |x^* - x_{k+1} + g(x_k) - g(x^*)| \\ &\leq |x^* - x_{k+1}| + |g(x_k) - g(x^*)| \\ &\leq |x^* - x_{k+1}| + L|x_k - x^*| \\ &\longrightarrow 0 \qquad k \longrightarrow \infty \end{aligned}$$

Hierbei wurde verwendet, dass

$$\forall L \in (0,1): \sum_{k=0}^{\infty} L^k = \frac{1}{1-L}$$

Damit haben wir (b) und die Existenz des Fixpunktes gezeigt. Seien zwei Fixpunkte x^*, y^* ausgezeichnet, dann

$$0 \leq |x^* - y^*| = |g(x^*) - g(y^*)| \leq L|x^* - y^*| < |x^* - y^*|$$

$L < 1 \implies |x^* - y^*| = 0$. Daher ist der Fixpunkt von g eindeutig bestimmt. □

Beispiel. Betrachte die nichtlineare Gleichung $f(x) = x^2 - \ln(x) - 2 = 0$ bzw. umgeformt $x^2 - 2 = \ln(x)$. Die Gleichung hat auf $\mathbb{R}$ zwei Lösungen x_1^* und x_2^*. Wie sieht eine geeignete Fixpunktiteration aus?

(a) $x = x^2 + x - \ln(x) - 2 =: g_1(x)$. Hinreichend für Lipschitz-Stetigkeit:

$$\max_{x \in \mathbb{R}} |g_1'(x)| = L < 1$$

Differenziere g_1 und erhalte

$$g_1'(x) = 2x + 1 - \frac{1}{x} \implies L \geq 1$$

Es gibt also keine Garantie für die Konvergenz gegen einen Fixpunkt.

(b) $\ln(x) = x^2 - 2 \iff x = e^{x^2-2} =: g_2(x)$ mit $g_2'(x) = 2xe^{x^2-2}$. $|g_2'(x)| < 1$ in einer Umgebung von $0 \implies$ Konvergenz gegen x_1^*.

(c) $x^2 = \ln(x) + 2 \iff x = \sqrt{\ln(x) + 2} =: g_3(x)$ mit

$$g_3'(x) = \frac{1}{2x\sqrt{\ln(x) + 2}}$$

$|g_3'(x)| < 1$ in $[1, 2] \implies$ Konvergenz gegen x_2^*.

Definition. *Eine gegen x^* konvergente Folge $(x_k) \subset \mathbb{R}^n$ besitzt die* Konvergenzordnung *$\alpha \geq 1$, wenn es eine Konstante $0 \leq c$ gibt, sodass*

$$\|x_{k+1} - x^*\| \leq c\|x_k - x^*\|^\alpha$$

Im Fall $\alpha = 1$ muss zusätzlich gelten, dass $c < 1$. Im Fall $\alpha = 1$ spricht man von linearer *Konvergenz, für $\alpha = 2$ von* quadratischer *Konvergenz. Weiter heißt eine Folge (x_k)* superlinear *konvergent, falls eine nicht-negative Nullfolge $(c_k) \subset [0, \infty)$ gibt, sodass*

$$\|x_{k+1} - x^*\| \leq c_k\|x_k - x^*\|$$

Alternative Darstellung für superlineare Konvergenz:

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0$$

Die Fixpunktiteration ist nur linear konvergent. Erstrebenswert wäre eine Iteration, die quadratisch konvergiert.
Kann in speziellen Fällen die Fixpunktiteration quadratisch konvergieren?

Satz 1.2. *Die Funktion $g\colon \mathbb{R} \to \mathbb{R}$ besitze in $x^* \in I$ einen Fixpunkt. Auf der Menge $U = [x^* - r, x^* + r] \subset I$ sei $g \in C^2(U)$ mit $g'(x^*) = 0$. Dann konvergiert die Fixpunktiteration für den Startwert x_0 mit $|x_0 - x^*| \leq \varrho = \frac{1}{\max_{x\in U} |g''(x)|}$ quadratisch.*

Beweis. Wir zeigen mittels vollständiger Induktion

$$|x_n - x^*| \leq \frac{1}{2^n}|x_0 - x^*| \tag{1}$$

Induktionsanfang: $n = 0$ klar
Induktionsschritt:

$$\begin{aligned}
|x_{n+1} - x^*| &= |g(x_n) - g(x^*)| \\
&\leq |g(x^*) + \underbrace{g'(x^*)}_{=0}(x_n - x^*) + \frac{g''(\xi)}{2}(x_n - x^*)^2 - g(x^*)| \\
&= |\frac{g''(\xi)}{2}(x_n - x^*)^2| \\
&\leq \frac{M}{2}|x_n - x^*| \qquad \text{mit } M := \max_{x\in U} |g''(x)|
\end{aligned} \tag{2}$$

Nun gilt: $|x_n - x^*| \leq |x_0 - x^*| \leq \varrho$, also

$$|x_{n+1} - x^*| \leq \frac{M}{2}\varrho\frac{1}{2^n}|x_0 - x^*| = \frac{1}{2^{n+1}}|x_0 - x^*|$$

. Also gilt (1) und es folgt Konvergenz. Aus (2) folgt Ordnung 2. □

1.2 Herleitung des Newton-Verfahrens in 1D

$f(x) = 0 \iff x = x - h(x)f(x) =: g(x)$ mit $g'(x) = 1 - h'(x)f(x) - h(x)f'(x)$. Die Funktion h muss noch bestimmt werden. Um den obigen Satz anwenden zu können, muss gelten:

$$g'(x^*) = 0 = 1 - h'(x^*) f(x^*) - h(x^*) f'(x^*) \implies \text{Idee} : h(x) = \frac{1}{f'(x)}$$

und somit $x = x - \frac{f(x)}{f'(x)}$. Wir erhalten die Fixpunktiteration (FPI):

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Diese FPI ist das Newton-Verfahren in 1D.

1.3 Newton-(Raphson-)Verfahren

Sei $f\colon \mathbb{R} \to \mathbb{R}$ stetig differenzierbar. *Gesucht:* $x^* \in \mathbb{R}\colon f(x^*) = 0$.
Wir wählen einen Startpunkt x_0. Für $k = 0, 1, 2, \ldots$ approximieren wir f durch eine Tangente p in x_k. Anstelle der Nullstelle x^* von f berechnen wir die Nullstelle der Tangente p und erhalten dadurch eine bessere Näherung x_{k+1}. Das Polynom p an x_k hat die Darstellung $p(x) = f(x_k) + f'(x_k)(x - x_k)$ Die Nullstelle davon ist

$$x^{k+1} = x^k - \frac{f\left(x^k\right)}{f'(x^k)}$$

unter der Voraussetzung, dass $\forall k = 0, 1, \ldots : f'(x_k) \neq 0$. [1]

Beispiel (Lokale Konvergenz). Sei

$$f(x) = \frac{x}{\sqrt{1 + c^3 x^2}} \qquad \text{mit} \qquad f(x^*) = 0 \iff x^* = 0$$
$$f'(x) = \ldots = \frac{1}{(1 + c^3 x^2)^{\frac{3}{2}}}$$
$$\frac{f(x)}{f'(x)} = x\left(1 + c^3 x^2\right) \implies g(x) = -c^3 x^3$$

Das Newton-Verfahren dazu ist

$$x^{k+1} = -c^3 (x_k)^3$$
$$\text{bzw. } x^k = (-1)^k c^{\frac{-3}{2}} \left(c^{\frac{3}{2}} x_0\right)^{3k}$$

Konvergenz ist nur dann garantiert, falls $c^{\frac{3}{2}} x_0 < 1$ gilt.

[1] Prinzip der Pfadverfolgung

Beispiel. Sei $f(x) = x^2 - 3$. Die Nullstellen sind $x^* = \pm\sqrt{3}$. $f'(x) = 2x \implies$ Newton-Verfahren ist durchführbar in der Nähe der Lösungen.
Mit $x_0 = 1$ ergibt sich folgende Tabelle der ersten 5 Iterationen:

0	1.00000000
1	2.00000000
2	1. 75000000
3	1. 7*3*214268
4	1. 7*320*5081
5	1. *73205081*

Was passiert bei einer mehrfachen Nullstelle x^*? - Betrachte dazu wieder eine Funktion $f\colon \mathbb{R} \to \mathbb{R}$ und f$f(x^*) = 0$. Sei $m \in \mathbb{N}_{\geq 2}$, x^* eine m-fache Nullstelle von f. Dann gilt

$$\forall \nu = 0, \ldots, m-1\colon f^{(\nu)}(x^*) = 0 \wedge f^{(m)}(x^*) \neq 0$$

Dann besitzen f und f' eine Darstellung der Form

$$\begin{aligned} f(x) &= (x - x^*)^m h(x) \\ f'(x) &= m(x - x^*)^{m-1} h(x) + (x - x^*)^m h'(x) \end{aligned}$$

mit einer differenzierbaren Funktion h, welche $h(x^*) \neq 0$ erfüllt.
Es folgt nun

$$\begin{aligned} g(x) = x - \frac{f(x)}{f'(x)} &= x - \frac{(x - x^*)^m h(x)}{m(x - x^*)^{m-1} h(x) + (x - x^*)^m h'(x)} \\ &= x - \frac{(x - x^*) h(x)}{m h(x) + (x - x^*) h'(x)} \\ \implies g'(x^*) &= 1 - \frac{1}{m} \end{aligned}$$

d.h. für $m > 1$, also mehrfache Nullstellen, ist das Verfahren zwar noch konvergent, aber nicht mehr quadratisch. (Denn nach dem Satz über FPI erhält man quadratische Konvergenz nur, wenn $g'(x^*) = 0$.)

1.4 Systeme von Gleichungen

Die Verallgemeinerung des Newton-Verfahrens auf Systeme von Gleichungen ist relativ einfach. Sei dazu $F\colon \mathbb{R}^n \to \mathbb{R}^n$ stetig differenzierbar (und besitze weitere Eigenschaften). Gesucht ist ein $x^* \in \mathbb{R}^n$ mit $F(x^*) = 0$. Sei $x^0 \in \mathbb{R}^n$ ein Startwert. Approximiere F durch Taylorentwicklung um x_0, also

$$F(x) \approx F\left(x^0\right) + F'\left(x^0\right)\left(x - x^0\right)$$

Falls $F'\left(x^0\right)$ invertierbar ist, dann setze

$$x^1 = x^0 - F'\left(x^0\right)^{-1} F\left(x^0\right)$$

Newton-Iteration: für $k = 0, 1, 2, 3, \ldots$

1) Berechne $\triangle x^k \in \mathbb{R}^n$, sodass $F'\left(x^k\right) \triangle x^k = -F\left(x^k\right)$ (Newton-Korrektur)

2) $x^{k+1} = x^k + \triangle x^k$

3) Falls $\left\|F\left(x^k\right)\right\|$ klein genug $\longrightarrow$ Abbruch

Wie im eindimensionalen Fall ist wieder x^{k+1} als Nullstelle der Tangente an F in x^k zu berechnen, also

$$x^{k+1} = x^k \underbrace{-F'\left(x^k\right)^{-1} F\left(x^k\right)}_{\text{Newton-Korrektur}}$$

Das Newton-Verfahren für nichtlineare Gleichungsysteme ist also auf eine Folge zu lösender linearer Gleichungsysteme zurückzuführen. Beobachtungen zum Konvergenzverhalten:

- konvergiert manchmal, manchmal auch nicht
- Wenn es konvergiert, dann konvergiert es *quadratisch* (d.h. schnell!)

$$\exists C > 0 \colon \left\|x^{k+1} - x^*\right\| \leq C \left\|x^k - x^*\right\|^2$$

Wann konvergiert das Verfahren gut?

- Falls F affin-linear ist, dann fertig in einem Schritt
- Vermutlich: Verfahren konvergiert, falls F „fast affin-linear“ ist

Was bedeutet nun „fast affin-linear“?

- F'' klein
- F' Lipschitz-stetig, also

$$\exists L \geq 0 \; \forall x, y \colon \left\|F'(x) - F'(y)\right\| \leq L\|x - y\|$$

Satz 1.3 (Fischer-Skript 5.2)**.** *Sei $D \subseteq \mathbb{R}^n$ offen und konvex, $F \colon D \to \mathbb{R}^n$ stetig differenzierbar mit invertierbarer Jacobi-Matrix an allen $x \in D$. Sei F' Lipschitz-stetig in D mit Konstante L. Sei $x^* \in D$ Nullstelle von F.*

a) Dann existiert eine offene Kugel $B_\varrho(x^*) := \{x \in \mathbb{R}^n \colon \|x - x^*\| < \varrho\} \subset D$, sodass das Newton-Verfahren für jede Startiterierte $x^0 \in B_\varrho(x^*)$ wohldefiniert ist.

b) Falls ϱ hinreichend klein ist konvergiert die Folge (x^k) quadratisch gegen x^*.

Korollar. Sei $F \in C^1$. Dann gilt $\forall x, y \colon$

$$F(y) = F(x) + F'(x)(y - x) + \int_0^1 (F'(x + s(y - x)) - F'(x))(y - x) \,\mathrm{d}\, s$$

Beweis. Taylor-Entwicklung mit Lagrange-Restglied

$$f(y) = \sum_{k=0}^{n} \frac{1}{k!} f^{(k)}(x)(y-x)^k + \frac{1}{n!} \int_0^1 (y+t)^n f^{(n+1)}(t) \,\mathrm{d}\, s$$

Spezialfall $n = 0$

$$f(y) = f(x) + \int_x^y f'(t) \,\mathrm{d}\, t = f(x) + \int_0^1 f'(x + s(y-x))(y-x) \,\mathrm{d}\, s$$

Für $F \colon \mathbb{R}^n \to \mathbb{R}^n$ gilt

$$F(y) = F(x) + \int_0^1 F'(x + s(y-x))(y-x) \,\mathrm{d}\, s$$

Addiere zur rechten Seite dieser Gleichung $F'(x)(y-x) - F'(x)(y-x) = 0$ □

Beweis. F' ist stetig in D

$$\implies \quad \exists \varrho_1 > 0 \; \exists M > 0 \colon B_{\varrho_1}(x^*) \subset D \wedge \forall x \in B_{\varrho_1}(x^*) \colon \|F'(x)^{-1}\| \leq M$$

Taylor-Formel mit Integralrestglied:

$$\forall x, y \in D \colon F(x) = F(y) + F'(y)(x-y) + \int_0^1 (F'(y + t(x-y)) - F'(y))(x-y) \,\mathrm{d}\, t$$

Für alle $x, y \in D$ gilt

$$\begin{aligned}
\|F(x) - F(y) - F'(y)(y-x)\| &= \left\| \int_0^1 \left[F'(y + t(x-y)) - F'(y) \right] (x-y) \,\mathrm{d}\, t \right\| \\
&\leq \int_0^1 \| \ldots \| \,\mathrm{d}\, t \\
&\leq \int_0^1 \|F'(y + t(x-y)) - F'(y)\| \,\mathrm{d}\, t \cdot \|x-y\| \\
&\leq \int_0^1 L \|(y + t(x-y)) - y\| \,\mathrm{d}\, t \cdot \|x-y\| \\
&= L \int_0^1 t \,\mathrm{d}\, t \cdot \|x-y\|^2 \\
&= \frac{1}{2} L \|x-y\|^2
\end{aligned}$$

Für beliebiges $x^k \in B_{\varrho_1}(x^*)$ gilt

$$\begin{aligned}
\left\| x^{k+1} - x^* \right\| &= \left\| x^k - F'\left(x^k\right)^{-1} F\left(x^k\right) - x^* \right\| \\
&= \left\| -F'\left(x^k\right)^{-1} \left[F\left(x^k\right) + F'\left(x^k\right)\left(x^k + x^*\right) \right] \right\| \\
&\leq M \left\| F\left(x^k\right) + F'\left(x^k\right)\left(x^* - x^k\right) \right\| \\
&= M \left\| F\left(x^*\right) - F\left(x^k\right) - F'\left(x^k\right)\left(x^* - x^k\right) \right\| \\
&\leq \frac{1}{2} ML \left\| x^* - x^k \right\|^2
\end{aligned}$$

Sei $\varrho \in (0, \varrho_1]$: $ML\varrho \leq 1$. Dann gilt für alle $x^k \in B_\varrho(x^*)$

$$\|x^{k+1} - x^*\| \leq \frac{1}{2} ML \left\|x^k - x^*\right\|^2 \leq \underbrace{\frac{1}{2} ML \underbrace{\left\|x^k - x^*\right\}}_{\varrho} \cdot \left\|x^k - x^*\right\}}_{\leq 1} \leq \frac{1}{2} \left\|x^k - x^*\right\|$$

Die Folge $\left(x^k\right)$ konvergiert gegen x^*, und zwar quadratisch. □

1.5 Affin-Invarianz

Es gibt viele Varianten des Satzes, dass das Newton-Verfahren lokal quadratisch konvergiert. P. Deuflhard hat das folgende Kriterium vorgeschlagen, dass solche Sätze erfüllen sollten.
Sei $A \in \mathbb{R}^{n \times n}$ invertierbar. Dann ist $F(x) = 0 \iff G(x) := AF(x) = 0$. Dann heißt das Problem $F(x) = 0$ affin-invariant.

Auch das Newton-Verfahren ist affin-invariant, denn $-\triangle x^k = F'\left(x^k\right)^{-1} F\left(x^k\right) = F'\left(x^k\right)^{-1} A^{-1} AF(x^k) = \left(AF'\left(x^k\right)\right)^{-1} \cdot AF(x^k) = G'\left(x^k\right)^{-1} \cdot G\left(x^k\right)$. Die vom Newton-Verfahren erzeugte Folge $(x^k)_{k \in \mathbb{N}}$ ist unabhängig von der Matrix A. Wir verlangen, dass auch die Konvergenzresultate unabhängig von A sein sollten. [2]

Satz 1.4 (D+H 4.10)**.** *Sei $D \subset \mathbb{R}^n$ offen und konvex, $C^1 \ni F\colon D \to \mathbb{R}^n$. Für alle $x \in D$ existiere $F'(x)^{-1}$. Für ein $\omega > 0$ gelte die Lipschitz-Bedingung*

$$\|F'(x)^{-1} \left(F'(x + sv) - F'(x)\right) v\| \leq s\omega \|v\|^2$$

für alle $s \in [0, 1], x \in D$ und $v \in \mathbb{R}^n$ mit $x + v \in D$. Es existiere eine Lösung $x^* \in D$ (des Problems $F(x^*) = 0$) und ein Startwert $x^0 \in D$ derart, dass

$$\varrho := \|x^* - x^0\| < \frac{2}{\omega} \qquad \text{und} \qquad B_\varrho(x^*) \subseteq D$$

Dann gilt:

i) Die durch die Newton-Iteration definiert Folge $\left(x^k\right)$ bleibt in der offenen Kugel $B_\varrho(x^*)$ und konvergiert gegen x^*.

ii) Konvergenz ist quadratisch, genauer

$$\forall k \in \mathbb{N}\colon \|x^{k+1} - x^*\| \leq \frac{\omega}{2} \|x^k - x^*\|^2$$

iii) Die Lösung von x^* ist eindeutig in $B_{\frac{2}{\omega}}(x^*)$.

[2] Das Streben nach Invarianz

Korollar. Unter den genannten Bedingungen gilt für alle $x, y \in D$:

$$\|F'(x)^{-1}\left(F(y) - F(x) - F'(x)(y-x)\right)\| \leq \frac{\omega}{2}\|y-x\|^2$$

Beweis. Taylor-Entwicklung mit Lagrange-Restglied:

$$f(y) = \sum_{k=0}^{n} \frac{1}{k!} f^{(k)}(x)(y-x)^k + \frac{1}{n!}\int_x^y (y-t)^n f^{n+1}(t)\,\mathrm{d}\,t$$

Spezialfall $n = 0$

$$\begin{aligned} f(y) &= f(x) + \int_x^y f'(t)\,\mathrm{d}\,t \\ &= f(x) + \int_0^1 f'(x + s(y-x))(y-x)\,\mathrm{d}\,s \end{aligned}$$

Für $F\colon \mathbb{R}^n \to \mathbb{R}^n$

$$F(y) = F(x) + \int_0^1 F'(x + s(y-x))(y-x)\,\mathrm{d}\,s$$

Subtrahiere $F'(x)(y-x)$ auf beiden Seiten

$$F(y) - F(x) - F'(x)(y-x) = \int_0^1 \left(F'(x + s(y-x)) - F'(x)\right)(y-x)\,\mathrm{d}\,s$$

Damit folgt

$$\begin{aligned} \|F'(x)^{-1}\left(F(y) - F(x) - F'(x)(y-x)\right)\| &= \left\| \int_0^1 F'(x)^{-1} F'(x + s\underbrace{(y-x)}_{v}) - F'(x)\underbrace{(y-x)}_{v}\,\mathrm{d}\,s \right\| \\ &\leq \int_0^1 s\omega\|y-x\|^2\,\mathrm{d}\,s \qquad \text{nach Voraussetzung} \\ &= \frac{\omega}{2}\|y-x\|^2 \end{aligned}$$

□

Beweis des Satzes.

$$\begin{aligned} x^{k+1} - x^* &= x^k - F'\left(x^k\right)^{-1} F\left(x^k\right) - x^* \\ &= x^k - x^* - F'\left(x^k\right)^{-1}\left(F\left(x^k\right) - \underbrace{F\left(x^*\right)}_{=0}\right) \\ &= F'\left(x^k\right)^{-1} F'\left(x^k\right)\left(x^k - x^*\right) - F'\left(x^k\right)^{-1}\left(F\left(x^k\right) - F\left(x^*\right)\right) \\ &= F'\left(x^k\right)^{-1}\left(F\left(x^*\right) - F\left(x^k\right) - F'\left(x^k\right)\left(x^* - x^k\right)\right) \end{aligned}$$

Mit dem Hilfssatz folgt daraus

$$\left\|x^{k+1} - x^*\right\| \leq \frac{\omega}{2} \left\|x^k - x^*\right\|^2$$

Das heißt, falls (x^k) konvergiert, so konvergiert es quadratisch.
Nach Wahl von ϱ gilt

$$\left\|x^k - x^*\right\| \leq \varrho \implies \left\|x^{k+1} - x^*\right\| \leq \underbrace{\frac{\omega}{2} \left\|x^k - x^*\right\|}_{\leq \varrho \frac{\omega}{2} < 1} \cdot \left\|x^k - x^*\right\|$$

Da $\|x^0 - x^*\| = \varrho$ gilt $\forall k > 0$: $\left\|x^k - x^*\right\| < \varrho$ und $\left(x^k\right)$ konvergiert gegen x^*. □

Beweis von (3). Sei x^{**} eine weitere Lösung in $B_{\frac{2}{\omega}}(x^*)$.

$$\begin{aligned}
\|x^{**} - x^*\| &= \left\|F'(x^*)^{-1} F'(x^*)(x^{**} - x^*)\right\| \\
&= \left\|F'(x^*)^{-1} \left(F(x^{**}) - F(x^*) - F'(x)\right)(x^{**} - x^*)\right\| \\
&\leq \frac{\omega}{2} \|x^{**} - x^*\| \cdot \|x^{**} - x^*\|
\end{aligned}$$

Es gilt

$$x^{**} \in B_{\frac{2}{\omega}}(x^*) \implies \frac{\omega}{2} \|x^{**} - x^*\| < 1 \implies \|x^{**} - x^*\| = 0$$

□

1.6 Konvergenzkriterien

Die Voraussetzungen des vorherigen Satzes können nicht algorithmisch geprüft werden. Trotzdem möchte man gern während der Durchführung der Newton-Methode wissen, ob das Verfahren konvergiert.

1.6.1 Monotonietests

Betrachte das Residuum $F\left(x^k\right)$. Das Lösen von $F(x) = 0$ ist äquivalent zum Minimieren von $\|F(x)\|$ (oder $\|F(x)\|$). Wir vermuten/hoffen: falls $\left(x^k\right)$ konvergiert, dann ist $\left\|F\left(x^k\right)\right\|$ eine monoton fallende Folge.

a) *Monotonietest:* Für ein $\theta < 1$ prüfe nach jedem Schritt, ob

$$\left\|F\left(x^{k+1}\right)\right\| \leq \theta \left\|F\left(x^k\right)\right\|$$

Dieses Verfahren ist nicht affin-invariant. Es folgt ein Gegenbeispiel.

Beispiel. Ausgangssituation

$$F\left(x^{k+1}\right) = \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix},\ F\left(x^{k}\right) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \implies \left\| \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} \right\| \leq \left\| \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\|$$

Sei ε klein. Wähle nun $A \in \mathbb{R}^{2\times 2}$ wie folgt

$$\begin{aligned} A &= \begin{pmatrix} 1 & 0 \\ 0 & \varepsilon \end{pmatrix} \\ \text{Wunsch :} \quad \left\| AF\left(x^{k+1}\right) \right\| &\leq \left\| AF\left(x^{k}\right) \right\| \\ \iff \left\| \begin{pmatrix} 1 & 0 \\ 0 & \varepsilon \end{pmatrix} \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} \right\| &\leq \left\| \begin{pmatrix} 1 & 0 \\ 0 & \varepsilon \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\ \iff \left\| \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} \right\| &\leq \left\| \begin{pmatrix} 0 \\ \varepsilon \end{pmatrix} \right\| \end{aligned}$$

Nicht erfüllt, obwohl Verfahren nicht verändert.

b) *Natürlicher Monotonietest:*

$$\left\| F'\left(x^{k}\right)^{-1} F\left(x^{k+1}\right) \right\| \leq \theta \left\| F'\left(x^{k}\right)^{-1} F\left(x^{k}\right) \right\|$$

Wie berechnet man diese Terme? - Rechte Seite

$$F'\left(x^{k}\right)^{-1} F\left(x^{k}\right) = \triangle x^{k}$$

Lösung des linearen Gleichungsystems ist die Newton-Korrektur, die wir ohnehin berechnen müssen. Linke Seite

$$F'\left(x^{k}\right)^{-1} F\left(x^{k+1}\right) = \overline{\triangle x^{k+1}}$$

ist die Lösung eines weiteren linearen Gleichungssystems, was teuer werden kann. ABER: es liegt die gleiche Matrix $F'\left(x^{k}\right)$ vor, wovon die LR-Zerlegung bereits bekannt ist. Es sind nur Vorwärts- und Rückwärtssubstitution nötig, der Aufwand ist damit $\mathcal{O}\left(n^2\right)$.

1.7 Newton-Verfahren mit Dämpfung

Wir verfolgen nun das Ziel der Globalisierung des Newton-Verfahrens. Kann man das Verfahren so erweitern, dass es für alle (oder zumindest mehr) Startwerte konvergiert? Dabei möchte man die schnelle quadratische Konvergenz behalten.

Idee. *Wir messen die „Güte“ einer Approximation x^k von x^* durch eine skalare Funktion*

$$\phi(x) := \frac{1}{2}\|F(x)\|_2^2$$

Wie wirkt das Newton-Verfahrens auf ϕ?

Lemma 1.1. *Die Newton-Korrektur $\triangle x = -\left(F'(x)\right)^{-1} F(x)$ ist nur eine Abstiegsrichtung für ϕ, d.h.*

$$\phi\left(x^k + t^k \triangle x^k\right) < \phi\left(x^k\right)$$

für $t^k > 0$ klein genug.

Beweis.

$$\begin{aligned}
\phi'(x) &= F(x)^T F'(x) \\
\phi'(x)\triangle x &= F^T F'(x)\triangle x = -F^T(x)F'(x)F'(x)^{-1}F(x) \\
&= -F^T(x)F(x) = -2\phi(x) < 0
\end{aligned}$$

falls x nicht Lösung von $F(x) = 0$ ist.

$$\begin{aligned}
\phi(x + t\triangle x) &= \phi(x) + \phi'(x)\cdot t\triangle x + o(t) \qquad \text{(Taylor)} \\
&= \phi(x) - 2t\phi(x) + o(t) \\
&= (1 - 2t)\phi(x) + o(t)
\end{aligned}$$

□

Durch geeignete Wahl der t_k entsteht eine Folge (x^k) mit $\varphi(x^{k+1}) < \varphi(x^k)$.

Idee. *Wähle anstelle der Korrektur $\triangle x^k$ die Korrektur $t_k \triangle x^k$ mit $t \in (0,1)$, sodass „hinreichender Abstieg“ von ϕ erzeugt wird. Was heißt „hinreichender Abstieg“?*

- *Angenommen, $\phi\left(x^k\right)$ sei streng monoton fallend.*

$\rightarrow$ *Dann konvergiert diese Folge (da ϕ von unten beschränkt)*

$\rightarrow$ *Aber sie konvergiert nicht notwendigerweise gegen 0.*

$\rightarrow$

Die Armijo-Schrittweitenregel (nach Larry Armijo): Wähle t_k als das größte Element aus $\left\{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots, \frac{1}{2^n}, \ldots\right\}$, für welches

$$\phi\left(x^k + t_k \triangle x^k\right) \leq (1 - qt)\, \phi\left(x^k\right)$$

für $q \in (0,1)$ fix, d.h.: Der Abstieg ist mindestens linear in t.

Satz 1.5 (Fischer-Skript 5.3)**.** *Es sei $C^1 \ni F\colon \mathbb{R}^n \to \mathbb{R}^n$. Sei F' lokal Lipschitz-stetig und $F'(x)$ regulär für alle $x \in W\left(x^0\right) := \{x \in \mathbb{R}^n \colon \phi(x) \leq \phi\left(x_0\right)\}$. Dann ist das Newton-Verfahren mit der Armijo-Dämpfung wohldefiniert. Falls die Folge $\left(x^k\right)$ eine Teilfolge besitzt, die gegen ein $\tilde{x}$ konvergiert, so gilt $F(\tilde{x}) = 0$.*

Beweis. Sei $x^k \in W\left(x^0\right)$. Nach Voraussetzung ist $F'\left(x^k\right)$ regulär, daher ist $\triangle x^k$ wohldefiniert und nach Konstruktion ist auch t_k wohldefiniert und

$$\phi\left(x^{k+1}\right) < \phi\left(x^k\right) \implies x^{k+1} \in W\left(x^0\right)$$

(1) Obere Schranke für $\left\|\triangle x^k\right\|$
   - F und F' sind stetig $\in \mathbb{R}^n$.
   - $\tilde{x} \in W\left(x_0\right) \implies F'\left(x_0\right)$ regulär
   - Wähle $\varrho > 0$ so klein, dass $B_\varrho(\tilde{x}) \subset W\left(x_0\right)$
   - Dann ist $x \mapsto \|\triangle x\| = \|F'(x)^{-1} F(x)\|$ stetig in $B_\varrho(\tilde{x})$

   $\implies \exists c > 0 \forall x \in B_\varrho(\tilde{x})\colon \|\triangle x\| \leq c$

(2) Da F' Lipschitz-stetig ist, so ist auch $\varphi' = F^T F'$ Lipschitz-stetig

$$\exists L > 0\ \forall x, y \in B_\varrho(\tilde{x})\colon \|\varphi'(x) - \varphi'(y)\| \leq \|x - y\|$$

(3) Nach Voraussetzung existiert Teilfolge $\left(x^{k_i}\right)$ gegen $\tilde{x}$. Bezeichne diese Teilfolge wieder als x^k.
   - $x^k \longrightarrow \tilde{x} \implies \exists k_0 \in \mathbb{N} \forall k > k_0\colon x^k \in B_\varrho(\tilde{x})$

– Da x^k gegen den Kugelmittelpunkt konvergiert, bildet sich ein endlicher Abstand zum Kugelrand, also

$$\exists \overline{t} > 0 \exists k_0 \in \mathbb{N} \forall k \geq k_0 \forall t \in [0, \overline{t}] \colon x^k + t \triangle x^k \in B_\varrho(\tilde{x})$$

- Taylor-Formel mit Integral-Restglied.

$$\phi(y) = \phi(x) + \phi'(x)(y-x) + \int_0^1 \left(\phi'(x + s(y-x)) - \phi'(x)\right)(y-x)\,\mathrm{d}\,s$$

Für $y = x + t\triangle x$:

$$\phi(x + t\triangle x) = \phi(x) + \phi'(t\triangle x) + \int_0^1 \left(\phi'(x + st\triangle x) - \phi'(x)\right) t\triangle x\,\mathrm{d}\,s$$

- Wegen $\phi'(x)\triangle x = -2\phi(x)$ folgt

$$\begin{aligned}
\phi(x + t\triangle x) &= (1-2t)\phi(x) + \int_0^1 \ldots \mathrm{d}\,s \\
&\leq (1-2t)\phi(x) + \left|\int_0^1 \ldots \mathrm{d}\,s\right| \\
&\leq (1-2t)\phi(x) + \int_0^1 \left\|\phi'(x + st\triangle x) - \phi'(x)\right\| \mathrm{d}\,s \cdot t \underbrace{\|\triangle x\|}_{\leq c} \\
&\leq (1-2t)\phi(x) + \max_s \left\|\phi'(x + st\triangle x) - \phi'(x)\right\| \cdot t \cdot c \\
&\leq (1-2t)\phi(x) + \max_s \|x + st\triangle x - x\| \cdot t \cdot c \\
&\leq (1-2t)\phi(x) + L \cdot t \,\|\triangle x\| \cdot t \cdot c \\
&\leq (1-2t)\phi(x) + Lt^2c^2
\end{aligned}$$

Wir wollen zeigen, dass die t_k von 0 weg beschränkt sind. Armijo: Wähle t als größtes t, sodass

$$\phi\left(x^k + t_k\triangle x^k\right) \leq (1 - qt_k)\left(\phi\left(x^k\right)\right) \underbrace{\leq (1-2t)\phi(x) + Lt^2c^2}_{\text{Wunder und Magie}}$$

Mit der obigen „magischen“ Forderung wird das t_k höchstens kleiner, aber selbst dieses t_k ist von 0 weg beschränkt. Nach Auflösen nach t_k folgt

$$\begin{aligned}
-2t_k\phi\left(x^k\right) + Lt_k^2c^2 &\geq -qt_k\phi\left(x^k\right) \\
Lt_k^2c^2 &\geq (2-q)t_k\phi\left(x^k\right) \\
t_k &\geq \frac{(2-q)\phi\left(x^k\right)}{Lc^2}
\end{aligned}$$

Das heißt noch nicht, dass t_k von 0 weg beschränkt ist, da $\phi\left(x^k\right) \longrightarrow 0$ sein könnte. Angenommen $\phi(\tilde{x}) > 0$.

$$\phi\left(x^k\right) \geq \frac{1}{2}\phi\left(\tilde{x}\right)$$
$$\implies t_k \geq \hat{t} := \frac{(2-q)\phi(\tilde{x})}{2Lc^2} > 0$$

Aber $\left(\phi\left(x^k\right)\right)$ ist monoton fallend

$$\phi(x^{k+1}) = \phi(x^k + t_k \triangle x^k) < (1 - qt_k)\phi(x^k) \leq \underbrace{(1-q\hat{t})}_{\longrightarrow 0}\phi(x^k)$$

Diese Ungleichung gilt für fast alle Indizes k. Daraus folgt $\lim \phi\left(x^k\right) = 0$ und Stetigkeit von ϕ liefert

$$\phi\left(\lim x^k\right) = \phi(\tilde{x}) = 0$$

□

1.8 Newton-Verfahren mit Armijo-Dämpfung

Sei $x_0 \in \mathbb{R}^n$. Für $k = 1, 2, \ldots$.

- Berechne Newton-Korrektur $\triangle x_k$ als Lösung von $F'\left(x^k\right) \triangle x^k = -F\left(x^k\right)$
- Wähle Schrittweite t^k als größtes Element aus

  $$\left\{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots\right\}$$

  so dass (mit $\phi(x) := \frac{1}{2}\|F(x)\|^2$)

  $$\phi\left(x^k + t^k \triangle x^k\right) \leq \left(1 - qt^k\right)\phi\left(x^k\right) \qquad q \in (0,1)$$

- ?

Satz 1.6 (Fischer 5.3)**.** *Sei $F\colon \mathbb{R}^n \to \mathbb{R}^n$ stetig differenzierbar. Weiter sei F' lokal Lipschitz-stetig mit Konstante L_0 und für alle $x \in W(x_0) := \{x \in \mathbb{R}^n \colon \phi(x) < \phi(x_0)\}$ sei $F'(x_0)$ invertierbar. Dann gibt es ein $k_0 \in \mathbb{N}$, sodass $t_k = 1$ für alle $k \geq k_0$. Die Dämpfung schaltet sich also irgendwann automatisch ab.*

Korollar. Das gedämpfte Verfahren konvergiert lokal quadratisch.

Beweis. Für k groß genug degeneriert das Verfahren zum normalen Newton-Verfahren. Da das Verfahren konvergiert, kommt irgendwann eine Iterierte x_k, die so nah an x^* liegt, sodass der lokale Konvergenzsatz greift. Aus der Konvergenz einer Teilfolge folgt die Konvergenz der ganzen Folge. □

Beweis von Satz 5.3. Man verwendet wieder die Taylor-Formel.

- Sei ϱ so klein, dass $B_\varrho(x^*) \subset W(x_0)$.
- Dann ist F' regulär für alle $x \in B_\varrho(x^*)$ und es existiert ein $M > 0$, sodass $\|F'(x)^{-1}\| \leq M$ für alle $x \in B_\varrho(x^*)$.
- Da x^k gegen x^* konvergiert gibt es ein $N \in \mathbb{N}$, sodass $x^k \in B_\varrho(x^*)$ für alle $k \geq N$.
- Für solche k gilt für jedes $s \in [0,1]$

$$\begin{aligned}\left\|F'\left(x^k + s\triangle x^k\right) - F'\left(x^k\right)\right\| &\leq L_0 \left\|s\triangle x^k\right\| \leq L_0 \left\|\triangle x^k\right\| \\ &= L_0 \left\|F'\left(x^k\right)^{-1} F\left(x^k\right)\right\| \leq L_0 M \left\|F\left(x^k\right)\right\|\end{aligned}$$

- Taylor-Formel

$$\begin{aligned}\left\|F\left(x^k + \triangle x^k\right)\right\| &= \left\|\underbrace{F\left(x^k\right) + F'\left(x^k\right)\triangle x^k}_{=0} + \int_0^1 \left[F'\left(x^k + s\triangle x^k\right) - F'\left(x^k\right)\right]\triangle x^k \,\mathrm{d}\, s\right\| \\ &\leq \underbrace{\max_{s\in[0,1]} \left\|F'\left(x^k + s\triangle x^k\right) - F'\left(x^k\right)\right\|}_{\leq L_0 M \left\|F\left(x^k\right)\right\|} \cdot \underbrace{\left\|\triangle x^k\right\|}_{\leq M} \left\|F\left(x^k\right)\right\| \\ &\leq L_0 M^2 \|F\left(x^k\right)\|^2\end{aligned}$$

- Wähle $\varrho > 0$ so klein, dass

$$\|F(x)\| \leq L_0^{-1} M^{-2} \sqrt{1-q} \qquad \forall x \in B_\varrho(x^*)$$

- Dann ist $\forall k$ groß genug

$$\left\|F\left(x^k + \triangle x^k\right)\right\| \leq \sqrt{1-q}\left\|F\left(x^k\right)\right\|$$

  also

$$\begin{aligned}\varphi\left(x^k + \triangle x^k\right) &= \frac{1}{2}\left\|F\left(x^k + \triangle x^k\right)\right\|^2 \leq \frac{1}{2}(1-q)\left\|F\left(x^k\right)\right\|^2 \\ &= (1-q)\varphi\left(x^k\right)\end{aligned}$$

- Das Armijo-Kriterium ist mit $t = 1$ erfüllt.

□

2 Nichtlineare Ausgleichsprobleme

Bisher haben wir Vektoren $x^* \in \mathbb{R}^n$ gesucht, sodass

$$F(x^*) = 0 \qquad \text{bzw.} \qquad \|F(x^*)\| = 0$$

Wir verallgemeinern nun diese Situation.

- *Gegeben:* m Messwerte $b_1, \ldots, b_m \in \mathbb{R}$ zu Zeitpunkten $t_1, \ldots, t_m$.
- *Gesucht:* Funktion $\phi\colon t \mapsto b$, die die Daten „möglichst gut approximiert".

Wir kennen schon:

1) Polynominterpolation: $\exists!$ Polynom p mit $\deg p = m - 1$, welches die Daten interpoliert. Dies funktioniert aber schlecht, wenn m groß. Jedoch interessiert uns gerade der Fall, wenn m groß ist.
2) Spline-Interpolation: Ja, *aber:* häufig vermuten wir schon eine Gesetzmäßigkeit und wollen eigentlich nur ein paar Parameter bestimmen.

Beispiel (Normalverteilung).

$$\phi(t; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(t-\sigma)^2}{2\varrho^2}\right)$$

mit den Parametern μ, dem Erwartungswert und σ, der Standardabweichung.
Das Problem ist hier: Finde $\mu, \sigma \in \mathbb{R}$, sodass $\forall i = 1, \ldots, m\colon \varphi(t_i; \mu, \sigma) = b_i$
Abstrakt: Wir suchen eine Modellfunktion mit n reellen Parametern [1] $x_1, \ldots, x_n$

$$\phi(t; x_1, x_2, \ldots, x_n)$$

Falls die Messwerte die Gesetzmäßigkeit exakt erfüllen, dann gibt es Parameter $x_1, \ldots, x_n$ so dass

$$b_i = \phi(t_i; x_1, \ldots, x_n) \qquad \forall i = 1, \ldots, m$$

Modellfunktionen sind immer nur Approximationen, also nie richtig oder falsch.

Beispiel. Newtonsche Mechanik macht Fehler (relativistische Korrekturen), Einsteinsche Mechanik dagegen weniger.

In der Praxis ist es wichtig, Rücksicht sowohl auf Messfehler aber auch auf Modellfehler zu nehmen. Deswegen kann man nur

$$b_i \approx \phi(t_0; x_1, \ldots, x_n) \qquad \forall i = 1, \ldots, m \qquad \text{erwarten.}$$

Betrachte die Differenzen

$$\triangle_i := b_i - \varphi(t_i; x_1, \ldots, x_n) \qquad \forall i = 1, \ldots, m$$

[1] Parametersatz

2.1 Prinzip der kleinsten Quadrate

Es ist sinnvoll, die $x_1, \ldots, x_n \in \mathbb{R}$ so zu wählen, dass das Fehlerfunktional

$$\triangle^2 := \sum_{i=1}^{m} \triangle_i^2 = \sum_{i=1}^{m} b_i - \phi(t_i; x_1, \ldots, x_n)$$

minimal wird. Es gibt Alternativen:

- Wähle den Parametersatz so, dass $\sum_i |\triangle_i|$ minimal wird.
- Wähle den Parametersatz so, dass $\max_i |\triangle_i|$ minimal wird.

Letztere werden nicht oft verwendet, z.B. aufgrund fehlender DIfferenzierbarkeit. Dabei sind diese Normen äquivalent. Es gelten Ungleichungen, wie

$$\exists C > 0 \colon \sum_i \triangle_i^2 < C \max_i |\triangle_i|$$

Beispiel (Spezielles Beispiel). ϕ ist linear in $x_1, \ldots, x_n$, also

$$\phi(t; x_1, \ldots, x_n) = a_1(t)x_1 + a_2(t)x_2 + a_3(t)x_3 + \ldots + a_n(t)x_n$$

Dann ist

$$\triangle^2 = \sum_{i=1}^{m} [b_i - (a_1(t_i)x_1 + \ldots + a_n(t_i)x_n)]^2 = \|b - Ax\|_2^2 =: \|F(x)\|_2^2$$

mit $A \in \mathbb{R}^{m \times n},\ A_{ij} = a_j(t_i)$, $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$ und $F \colon \mathbb{R}^n \to \mathbb{R}^m,\ x \mapsto b - Ax$. Es liegt also ein lineares Ausgleichproblem vor, was wir schon kennen.

Nun zu *nichtlinearen* Ausgleichsproblemen. Folgende Situation: $(m, n \in \mathbb{N}, m > n)$

- Seien $b_1, \ldots, b_m \in \mathbb{R}$ Messdaten zu Zeitpunkten $t_1, \ldots, t_m$ und $F \colon \mathbb{R}^n \to \mathbb{R}^m$ zweimal stetig differenzierbar.
- Gesucht ist eine Modellfunktion $\phi(\cdot, x_1, \ldots, x_n)$ mit einem reellen Parametersatz $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$, sodass das Residuum

  $$\|F(x)\|^2 = \sum_{i=1}^{m} (b_i - \phi(t_i, x_1, \ldots, x_n))^2$$

  (lokal) minimal wird. Das nichtlineare Ausgleichsproblem wird so zu einem Minimierungsproblem

  $$g(x) := \frac{1}{2} \|F(x)\|_2^2 \longrightarrow \min$$

  mit lokalem Minimierer $x^* \in \mathbb{R}^n$.
- Hinreichende Kriterien sind

  $$g'(x^*) = 0, \qquad g''(x^*) \text{ positiv definit}$$

Idee. *Benutze ein Newton-Verfahren für die Gleichung*

$$0 = g'(x) = F'(x)^T F(x) =: G(x), \qquad G\colon \mathbb{R}^n \to \mathbb{R}^n$$

Eine Newton-Korrektur $\triangle x^k$ für diese Gleichung löst:

$$G'\left(x^k\right) \triangle x^k = -G\left(x^k\right) \qquad \forall k = 0, 1, 2, \ldots$$

Ausrechnen

$$G'(x) = F'(x)^T F'(x) + \underbrace{F''(x)^T}_{\text{3D}-\text{Matrix}} F(x)$$

Nach Annahme positiv definit, also invertierbar.

Wir haben ein Problem: F'' wird benötigt.

→ Ausrechnen davon kann beschwerlich sein.

→ Ausrechnen davon kann teuer sein, denn F'' hat n^3 Einträge

→ Können wir den 2. Summanden in G' einfach weglassen?

Definition (kompatibel)**.** *Ein Ausgleichsystem heißt kompatibel, falls es ein $x^* \in \mathbb{R}^n$ gibt dass $F(x^*) = 0$. In diesem Fall gilt zumindest in x^**

$$G'\left(x^*\right) = F'\left(x^*\right)^T F'\left(x^*\right)$$

Kompatibilität heißt: Es existiert ein Parametersatz $x \in \mathbb{R}$, sodass alle Messwerte $b_1, \ldots, b_m$ exakt prognostiziert werden.

→ *Dies kommt in der Praxis quasi nie vor.*

- Aber*: Wir erwarten/glauben, dass die Modellfunktion ϕ „gut“ ist, d.h., dass es Punkte/offene Mengen gibt, bei denen $\|F(x)\|^2$ zumindest „klein“ wird.*

→ *Diese Probleme nennt man dann „fast kompatible Probleme“.*

Wir lassen also den zweiten Summanden in G' weg und hoffen auf das Beste.

2.2 Gauß-Newton-Verfahren

Kein echtes Newton-Verfahren. Deswegen bekommt man nicht alle schönen Eigenschaften eines Newton-Verfahrens. Iterationsvorschrift mit modifizierten G' lautet:

$$F'\left(x^k\right)^T F'\left(x^k\right) \triangle x^k = -F'\left(x^k\right)^T F\left(x^k\right) \tag{2.1}$$

Bemerkung. Sei $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. Gesucht $x \in \mathbb{R}^n$ so dass $\|b - Ax\|^2$ minimal wird. Normalengleichung: Die Lösung x^* davon löst

$$A^T A x = A^T b \qquad \text{(Normalengleichung)}$$

$\Longrightarrow$ (2.1) ist Normalengleichung des *linearen* Ausgleichsproblems

$$\left\| F\left(x^k\right) + F'\left(x^k\right) \triangle x^k \right\| \longrightarrow \min$$

Wir können ein nichtlineares Ausgleichsproblem lösen, indem wir eine Folge von linearen Problemen lösen. Betrachte hierzu das lineare Ausgleichsproblem:

$$\|b - Ax\| \longrightarrow \min$$

mit $A \in \mathbb{R}^{m \times n}$, $m \geq n$. Es gilt

- Falls A invertierbar ist, dann gilt $x = A^{-1}b$
- Falls A nicht invertierbar ist, dann löst x die Normalengleichung

$$A^T Ax = A^T b$$

- Die Matrix A habe Rang n. Dann ist $A^T A$ invertierbar und es folgt

$$x = \underbrace{(A^T A)^{-1} A^T}_{\text{Pseudoinverse } A^+} b$$

Definition (Moore-Penrose-Pseudoinverse)**.** *Die Moore-Penrose Pseudoinverse von A ist* $A^+ := (A^T A)^{-1} A^T$.

Satz 2.1. *Die Moore-Penrose-Pseudoinverse* $A^+ \in \mathbb{R}^{n \times m}$ *einer Matrix* $A \in \mathbb{R}^{m \times n}$ *besitzt folgende Eigenschaften*

1) $AA^+A = A$

2) $A^+AA^+ = A^+$

3) A^+A *und* AA^+ *sind symmetrisch.*

4) $(A^+)^+ = A$

5) $(A^T)^+ = (A^+)^T$

6) $\forall \lambda \in \mathbb{R} \setminus \{0\}\colon\ (\lambda A)^+ = \lambda^{-1} A^+$

Falls $A \in \mathbb{R}^{m \times n}$ *vollen Rang hat, dann gilt*

$$A^+A = I_n \in \mathbb{R}^{n \times n}$$

und

$$AA^+ = I_m \in \mathbb{R}^{m \times m}$$

$\Longrightarrow$ (2.1) ist Normalengleichung des *linearen* Ausgleichsproblems

$$\left\| F\left(x^k\right) + F'\left(x^k\right) \triangle x^k \right\| \implies \triangle x^k = -F'\left(x^k\right)^+ F\left(x^k\right)$$

Satz 2.2 (Deuflhard & Hohmann 4.15)**.** *Sei $D \subset \mathbb{R}^m$ offen und konvex, $F\colon D \to \mathbb{R}^m$, $m \geq n$, stetig differenzierbar und $F'(x)$ habe vollen Rang $\forall x \in D$. Es existiere eine Lösung x^* des dazugehörigen Ausgleichsproblems. F' sei affin-invariant Lipschitz-stetig:*

$$\exists \omega > 0\ \forall s \in [0,1]\colon \left\| F'(x)^+ (F'(x+sv) - F'(x)) \right\| \leq s\omega \|v\|^2 \qquad \forall v \in \mathbb{R}^n\colon x + v \in D$$

Es gebe eine Konstante $\kappa_* \in [0,1)$ so, dass

$$\forall x \in D\colon \left\| F'(x)^+ F(x^*) \right\| \leq \kappa_* \|x - x^*\|$$

Diese Bedingung fordert, dass das Problem „fast kompatibel “, also $\|F(x^*)\|$ klein ist. Sei weiterhin der Startwert $x^0 \in D$ so, dass

$$\left\| x^0 - x^* \right\| < \frac{2}{\omega}(1 - \kappa_*) := \mathcal{G}$$

1) Dann konvergiert das Gauß-Newton-Verfahren gegen x^*

2) Die Konvergenzgeschwindigkeit ist

$$\left\| x^{k+1} - x^* \right\| \geq \frac{\omega}{2} \left\| x^k - x^* \right\|^2 + \kappa_* \left\| x^k - x^* \right\| \tag{2.2}$$

Beweis. Der Beweis ist dem Beweis zum Newton-Verfahren sehr ähnlich. Für alle $x, y \in D$ gilt:

$$\begin{aligned}
\left\| F'(x)^+ \left[F(y) - F(x) - F'(y-x) \right] \right\| &\leq \left\| F'(x)^+ \int_0^1 \left[F'(x + s(y-x)) - F'(x) \right] (y-x)\, \mathrm{d}\, s \right\| \\
&\leq \int_0^1 \left\| F'(x)^+ \left[F'(x + s(y-x)) - F'(x) \right] (y-x) \right\| \mathrm{d}\, s \\
&\leq \int_0^1 s\omega \|y - x\|_2^2 \, \mathrm{d}\, s \\
&= \frac{\omega}{2} \|y - x\|_2^2
\end{aligned}$$

Beachte: $F'(x)^+ F'(x) = I_n\ \forall x \in D$, da Moore-Penrose-Pseudoinverse.

$$\begin{aligned}
x^{k+1} - x^* &= \left(x^k - x^* \right) - F'\left(x^k \right)^+ F\left(x^k \right) \\
&= \underbrace{F'\left(x^k \right)^+ F'\left(x^k \right)}_{=I} \left(x^k - x^* \right) - F'\left(x^k \right)^+ F\left(x^k \right) + \underbrace{F'\left(x^k \right)^+ F(x^*) - F'\left(x^k \right)^+ F(x^*)}_{=0} \\
&= \underbrace{F'\left(x^k \right)^+ \left[F(x^*) - F\left(x^k \right) - F'\left(x^k \right) \left(x^* - x^k \right) \right]}_{\leq \frac{\omega}{2} \left\| x^k - x^* \right\|^2} - \underbrace{F'\left(x^k \right)^+ F(x^*)}_{\leq \kappa_* \left\| x^k - x^* \right\|} \\
&\implies \|x^{k+1} - x^*\| \leq \left(\frac{\omega}{2} \left\| x^k - x^* \right\| + \kappa_* \right) \left\| x^k - x^* \right\| \\
&\implies \text{Konvergenz, falls } \frac{\omega}{2} \left\| x^k - x^* \right\| + \kappa_* < c < 1 \qquad \forall k
\end{aligned}$$

Nach Voraussetzung:

$$\left\|x^0 - x^*\right\| < \frac{2}{\omega}(1-\kappa_*) \iff \underbrace{\frac{\omega}{2}\left\|x^0 - x^*\right\| + \kappa_*}_{:=c} < 1$$

Nach Induktion ist damit

$$\begin{aligned} &\forall k \in \mathbb{N}\colon \left\|x^{k+1} - x^*\right\| < \left\|x^k - x^*\right\| \\ \implies &\forall k \in \mathbb{N}\colon \frac{\omega}{2}\left\|x^k - x^*\right\| + \kappa_* < c \end{aligned}$$

Das Verfahren konvergiert. □

Das Verfahren konvergiert nur dann lokal quadratisch, falls κ_*, falls also das Problem kompatibel ist. Dasist der Preis dafür, dass wir F'' weggelassen haben.

3 Optimierung

Wir verallgemeinern unser Problem noch weiter. Sei nun $f\colon \mathbb{R}^n \to \mathbb{R}$ gegeben. Unsere Aufgabe: Finden eines (lokalen) Minimierers von f.

Beispiel. $x_1, \dots, x_n$ bezeichnen Designparameter eines Rennautos (z.B. Hubraum, Reifengröße, Gewicht, etc.). $f\colon \mathbb{R}^n \to \mathbb{R}$ (Höchst-)Geschwindigkeit des Autos. Finde Minimierer von $-f$.

Beispiel (Festkörpermechanik). Deformierbares Objekt $\Omega \subset \mathbb{R}^3$. Deformation: $\Phi\colon \Omega \to \mathbb{R}^3$. Hyperelastizität: stabile Zustände Φ minimieren eine Energie (Funktion)

$$\mathcal{J}\colon C^1(\Omega, \mathbb{R}^3) \to \mathbb{R} \qquad \Phi \mapsto \int_\Omega W(\Phi(x), \nabla\Phi(x)) \,\mathrm{d}\, x$$

Diskretisierung: Fülle Ω mit Dreiecken bzw. Tetraedern. Betrachte nur noch Position der Eckpunkte der Dreiecke (Knoten). Das sind n viele. Aus $\mathcal{J}\colon C^1(\Omega, \mathbb{R}^3) \to \mathbb{R}$ wird $f\colon \mathbb{R}^{3n} \to \mathbb{R}$ Stichwort: Finite Elemente [1]

Sei zunächst f quadratisch, d.h. mit symmetrischer Matrix $A \in \mathbb{R}^{n\times n}$

$$f(x) = \frac{1}{2} x^T A x - b^T x + c$$

- Falls A positiv definit ist, dann existiert genau ein Minimierer x^*.
- Dieser löst
$$\nabla f\,(x^*) = 0 = Ax - b \iff Ax = b$$
- Problem zurückgeführt auf lineares Gleichungsystem → schon bekannt

Sei ab jetzt f *nicht* quadratisch. Alle bekannten Verfahren sind iterativ.
Allgemeiner Ansatz: Sei $x^0 \in \mathbb{R}^n$. Für $k = 1, 2, \dots$

- Wähle eine Richtung p_k
- Wähle eine Schrittweite $t_k \in \mathbb{R}, t_k > 0$
- Setze $x^{k+1} = x^k + t_k p_k$

Hoffnung:

1) (x_k) konvergiert gegen den Minimierer von f für möglichst viele Startwerte.
2) Die Konvergenz ist schnell.

[1] Unsere Welt minimiert die Funktion über allen möglichen Welten.

Hängt ab von:

a) geschickter Wahl von p_k

b) geschickter Wahl von t_k

In den allermeisten Fällen will man *Abstiegsverfahren*, d.h. es soll gelten

$$\forall k \in \mathbb{N}\colon f\left(x^{k+1}\right) \leq f\left(x^k\right)$$

3.1 Schrittweiten

Seien $x^k \in \mathbb{R}^n$ und eine Abstiegsrichtung p_k gegeben. Wie sollte man ein „gutes“ t_k wählen? Es entsteht für uns ein Dilemma

- t_k muss sorgfältig gewählt werden, um möglichst viel Energiereduktion zu erhalten.
- Die Wahl von t_k selbst darf nicht zu aufwändig sein.

Idealerweise: Wähle t_k als globalen Minimierer von

$$\Theta\colon \mathbb{R} \to \mathbb{R}, t \mapsto f(x^k + tp_k)$$

Diese exakte Liniensuche ist aber i.A. viel zu teuer. Stattdessen: inexakte Liniensuche

- Gegeben sei eine Folge von möglichen Schrittweiten t_k.
- Wähle die erste Schrittweite, die einer gewissen Bedingung genügt.
- Hoffnung: mit deutlich weniger Aufwand eine Schrittweise zu finden, die fast genauso gut ist.

Diese Methode ist der Dämpfungsstrategie im gedämpften Newton-Verfahren sehr ähnlich. Jedoch wusste man dort, dass $t_k \in (0, 1]$ sein muss und dass $t_k = 1$ gewisse Vorteile biete. Dieses Wissen hat man hier nicht. Die einfachste Bedingung an die Schrittweite ist

- Wähle t_k so, dass $f\left(x^k + t_k p_k\right) < f\left(x^k\right) \forall k \in \mathbb{N}$

Das reicht nicht: betrachet beispielsweise die Funktion $f(x) = x^2 - 1$. Wir brauchen *hinreichenden Abstieg.*

a) Armijo-Regel: Fordere Reduktion, die linear ist in der Schrittweite t_k und der Richtungsableitung

$$\frac{df}{dp} = \frac{d}{dt} f\left(x^k + tp_k\right)\Big|_{t=0} = \nabla f(x)^T p_k = \left\langle p_k, \nabla f\left(x^k\right)\right\rangle$$

Das bedeutet

$$f\left(x^k + tp_k\right) \leq f\left(x^k\right) + c_1 t \nabla f\left(x^k\right)^T p_k, \qquad c_1 \in (0, 1) \tag{I}$$

Achtung: wird nur die Armijo-Regel verwendet, dann können die Schrittweiten unnötig klein werden. Diese Bedingung allein reicht nicht.

Bemerkung. Falls man eine maximale Schrittweite weiß/schätzen kann, so kann man Backtracking verwenden

- Wähle $t_0 = t_{\max}$.
- Für alle $l = 0, 1, 2, \ldots$.
- Falls $f\left(x^k + t_l p_k\right) \leq f\left(x^k\right) + c t_l \nabla f\left(x^k\right)^T p_k \longrightarrow$ STOP.
- Setze $t_{l+1} = \frac{1}{2} t_l$.

Um das zu vermeiden fordern wir

b) Krümmungsbedingung: Falls $\Theta'(t)$ stark negativ ist, bekomme ich mehr Abstieg, wenn ich t vergrößere. Falls $\Theta'(t)$ positiv oder nur wenig negativ ist, dann lohnt es sich nicht/kaum, t zu vergrößern. Forderung:

$$\Theta'(t) \geq \underbrace{c_2 \Theta'(0)}_{<0}$$

oder auch

$$\nabla f\left(x^k + t p_k\right) \geq c_2 \nabla f\left(x^k\right) p_k, \qquad c_2 \in (c_1, 1) \tag{II}$$

- (I) & (II) zusammen ergeben die Wolfe-Bedingungen.

Satz 3.1 (Nocedal & Wright, Lemma 3.1)**.** *Sei $f\colon \mathbb{R}^n \to \mathbb{R}$ stetig differenzierbar und von unten bschränkt, p_k Abstiegsrichtung in x^k. Für alle $c_1, c_2 \in \mathbb{R}$ mit $0 < c_1 < c_2 < 1$ existieren zulässige Schrittweiten im Sinne der Wolfe-Bedingung.*

Beweis. $\phi(t) = f\left(x^k + t p_k\right)$ ist von unten beschränkt.

- Da p_k Abtiegsrichtung $\implies \nabla f\left(x^k\right)^T p_k < 0$.
- Deshalb ist $\ell(t) = f\left(x^k\right) + t c_1 \nabla f\left(x^k\right)^T p_k$ für $t > 0$ nicht von unten beschränkt.
- f ist stetig: $\exists$ ein kleinstes $t' > 0$ mit

$$f\left(x^k + t' p_k\right) = f\left(x^k\right) + t' c_1 \nabla f\left(x^k\right)^T p_k$$

- Also gilt die Armijo-Bedingung (I) für alle $t < t'$.
- Mittelwertsatz: $\exists t'' \in (0, t')$ so dass

$$\underbrace{f\left(x^k + t' p_k\right) - f\left(x^k\right)}_{= t' c_1 \nabla f\left(x^k\right)^T p_k} = t' \nabla f\left(x^k + t'' p_k\right)^T p_k$$

Teile durch t':

$$\nabla f(x^k + t'' p_k)^T p_k = \underbrace{c_1}_{<c_2} \underbrace{\nabla f(x^k)^T p_k}_{<0} \geq c_2 \nabla f(x^k)^T p_k$$

- t'' erfüllt auch Bedingung (II).

□

Liniensuchverfahren entsprechen i.A. dem folgenden Algorithmus: Sei $x^0 \in \mathbb{R}^n$ gegeben. Für $k = 0, 1, 2, \ldots$

- Wähle Richtung $p_k \in \mathbb{R}^n$
- Wähle Schrittweite $t_k \in (0, \infty)$
- Setze $x^{k+1} = x^k + t_k p_k$ [2]

Sinnvolle Schrittweiten t_k erfüllen z.B.

$$f(x_k + t_k p_k) \leq f\left(x^k\right) + c_1 t_k \nabla f\left(x^k\right)^T p_k \qquad \text{(1 Armijo-Bedingung)}$$

$$\triangledown f\left(x^k + t_k p_k\right)^T p_k \geq c_2 \nabla f\left(x^k\right)^T p_k, \qquad c_2 > c_1 \qquad \text{(2 Krümmungsbedingung)}$$

3.2 Suchrichtungen

Wie wählt man die Suchrichtungen p_k? Eine scheinbar vernünftige Idee: Wähle p_k als Richtung des steilsten Abstiegs von f in x^k (engl.: steepest descent). Richtungsableitung

$$\frac{df}{dp} = \frac{d}{dx} f(x + \alpha p)|_{\alpha=0}$$

Definition. *Die Richtung des steilsten Abstiegs von f in x ist der Minimierer von $\frac{df}{dp}$ bezüglich p unter der Nebenbedingung $\|p\| = 1$.*

Lemma 3.1. *Der Minimierer ist*

$$p^* = \frac{-\nabla f(x)}{\|\nabla f(x)\|}$$

Beweis. Sei θ der WInkel zwischen p und $\nabla f(x)$.

- Dann ist
$$\frac{df}{dp} = p^T \nabla f(x) = \|p\| \|\nabla f(x)\| \cos\theta = \|\nabla f(x)\| \cos\theta$$
- Minimal, wenn $\cos\theta = -1 \implies \theta = \pi$
$$p = \frac{-\nabla f}{\|\nabla f\|}$$

□

[2] Schwarze Magie

Das Gradientenverfahren wirkt vernünftig, kann aber sehr langsam sein.

Beispiel. Sei

$$f\colon \mathbb{R}^2 \to \mathbb{R}, x \mapsto x^T \underbrace{\begin{pmatrix} \varepsilon & 0 \\ 0 & 1 \end{pmatrix}}_{=:A} x$$

mit kleinem ε. Die Optimale Schrittweite ist

$$t = \frac{\nabla f\left(x^k\right)^T \nabla f\left(x^k\right)}{\nabla f\left(x^k\right)^T A \nabla f\left(x^k\right)}$$

Bemerkung. Das Problem ist schlecht konditioniert.

Viele Algorithmen verwenden deshalb andere Suchrichtungen. Man will aber häufig, dass die Richtungen p_k wenigstens ähnlich dem steilsten Abstieg sind (engl.: gradient-related), die also nur ungefähr in Richtung $-\nabla f\left(x^k\right)$ zeigen. Definiere θ_k als Winkel zwischen p_k und $-\nabla f\left(x^k\right)$.

$$\cos\theta_k = \frac{-\nabla f\left(x^k\right)^T p_k}{\|\nabla f\left(x^k\right)\| \cdot \|p_k\|}$$

Satz 3.2 (Nocedal & Wright, 3.2)**.** *Gegeben sei ein Verfahren*

$$x^{k+1} = x^k + t_k p_k$$

wobei p_k immer Absiegsrichtung (nicht unbedingt maximale Abstiegsrichtung) ist und t_k immer die Wolfe-Bedingung erfüllt. Sei $C^1 \ni f\colon \mathbb{R}^n \to \mathbb{R}$ von unten beschränkt. Der Gradient ∇f sei Lipschitz-stetig mit Konstante L. Dann folgt

$$\sum_{k=0}^{\infty} \cos^2\theta_k \left\|\nabla f\left(x^k\right)\right\|^2 < \infty$$

Konsequenzen:

- Es gilt
$$\lim_{k\to\infty} \cos^2\theta_k \left\|\nabla f\left(x^k\right)\right\|^2 = 0$$
- Falls p_k so gewählt ist, dass θ_k von 90° weg beschränkt ist, dann
$$\forall k \exists \delta > 0\colon \cos\theta_k \le \delta > 0 \implies \lim \|\nabla f\left(x^k\right)\| = 0$$
$\implies$ Das Verfahren konvergiert gegen einen stationären Punkt, wenn die Suchrichtungen nicht „zu senkrecht“ auf $-\nabla f\left(x^k\right)$ stehen. Insbesondere „konvergiert“ das Gradientenverfahren gegen einen stationären Punkt, wenn die Schrittweite immer die Wolfe-Bedingung erfüllt.

- *Beachte*: Sattelpunkte/Maxima sind instabil! - Sei x^* ein Sattelpunkt und $\left(x^k\right)$ mit $x^k \longrightarrow x^*$ erzeugt durch ein Liniensuchverfahren. Dann ist $f\left(x^k\right) \geq f\left(x^*\right)\ \forall k$.
- Konvergenz gegen stationäre Punkte, *nicht* gegen Minimierer!
- Mehr ist mit den oben genannten Annahmen nicht zu erreichen.
- Konvergenz gegen Minimierer nur mit zusätzlichen Annahmen an die p_k
- Das ist natürlich teuer.

Beachte: Stationäre Punkte (außer Minimierer) sind instabil!

- Sei x^* ein Sattelpunkt und $\left(x^k\right)$ mit $x^k \longrightarrow x^*$ durch das Liniensuchverfahren erzeugt.
- Dann ist $\forall k\colon f\left(x^k\right) \geq f\left(x^*\right)$.
- Tatsächlich aber treten Rundungsfehler auf: Wenn x^k schon sehr nah an x^* ist, kann eventuell gelten
$$f\left(x^{k+1}\right) \geq f\left(x^*\right)$$
aber
$$f\left(\approx x^{k+1}\right) < f\left(x^*\right)$$
- Danach kann die Folge nicht mehr gegen x^* konvergieren.

Beweis von Satz 3.2. Die Wolfe-Bedingungen sind:

a) $f\left(x^{k+1}\right) \leq f\left(x^k\right) + c_1 t_k \nabla f\left(x^k\right)^T p_k$ (Armijo)

b) $\nabla f\left(x^{k+1}\right) p_k \geq c_2 \nabla f\left(x^k\right)^T p_k$ (Krümmung)

Subtrahiere $\nabla f\left(x^k\right)^T p_k$ von b)

$$\left(\nabla f\left(x^{k+1}\right) - \nabla f\left(x^k\right)\right)^T p_k \geq (c_2 - 1) \nabla f\left(x^k\right)^T p_k$$

∇f ist Lipschitz-stetig, deshalb gilt

$$\begin{aligned}\left|\left(\nabla f\left(x^{k+1}\right) - \nabla f\left(x^k\right)\right)^T p_k\right| &\leq \left\|\left(\nabla f\left(x^{k+1}\right) - \nabla f\left(x^k\right)\right)\right\| \cdot \|p_k\| \\ &\leq L \overbrace{\left\|x^{k+1} - x^k\right\|}^{t_k p_k} \cdot \|p_k\| = L t_k \|p_k\|^2\end{aligned}$$

Zusammen

$$t_k L \left\|p_k\right\|^2 \geq \left(\nabla f\left(x^{k+1}\right) - \nabla f\left(x^k\right)\right)^T p_k \geq (c_2 - 1)\nabla f\left(x^k\right)^T p_k$$

$$\implies t_k \geq \frac{c_2 - 1}{L} \cdot \frac{\nabla f\left(x^k\right)^T p_k}{\left\|p_k\right\|^2}$$

Einsetzen in Armijo-Bedingung

$$\begin{aligned} f\left(x^{k+1}\right) &\leq f\left(x^k\right) + c_1 t_k \underbrace{f\left(x^k\right)^T p_k}_{\leq 0} \\ &\leq f\left(x^k\right) + c_1 \frac{c_2 - 1}{L} \frac{\left(\nabla f\left(x^k\right)^T p_k\right)^2}{\left\|p_k\right\|^2} \\ &= f\left(x^k\right) - \underbrace{c_1 \frac{1 - c_2}{L}}_{=:c} \cdot \underbrace{\frac{\left(\nabla f\left(x^k\right)^T p_k\right)^2}{\left\|p_k\right\|^2 \cdot \left\|\nabla f\left(x^k\right)\right\|^2}}_{=\cos^2\theta_k} \cdot \left\|\nabla f\left(x^k\right)\right\|^2 \\ &= f\left(x^k\right) - c \cdot \cos^2\theta_k \cdot \left\|\nabla f\left(x^k\right)\right\|^2 \end{aligned}$$

Rekursives Einsetzen

$$f\left(x^{k+1}\right) \leq f\left(x^0\right) - c \sum_{j=0}^{k} \cos^2\theta_j \left\|\nabla f\left(x^j\right)\right\|^2$$

beziehungsweise

$$\sum_{j=0}^{k} \cos^2\theta_j \left\|\nabla f\left(x^j\right)\right\|^2 \leq \frac{1}{c}\left(f\left(x^0\right) - f\left(x^{k+1}\right)\right)$$

Rechte Seite ist nach oben beschränkt, da f nach unten beschränkt ist. Partialsummen sind also beschränkt, außerdem monoton steigend

$$\implies \lim_{k\to\infty} \sum_{j=0}^{k} \cos^2\theta_j \left\|\nabla f\left(x^j\right)\right\|^2 < \infty$$

□

3.3 Das Gradientenverfahren

Gegeben $x^0 \in \mathbb{R}^n$. Für $k = 0, 1, 2, \ldots$

$$x^{k+1} = x^k - t_k \nabla f\left(x^k\right)$$

Verfahren konvergiert global, falls die t_k die Wolfe-Bedingungen erfüllen. *Aber*: Konvergenz teilweise langsam.

Satz 3.3 (Nocedal & Wright 3.3)**.** *Sei f quadratisch, also*

$$f(x) = \frac{1}{2}x^T Ax - b^T x$$

mit symmetrischer, positiv definiter Matrix A.

- *Exakte Liniensuche*
$$t_k = \frac{\nabla f\left(x^k\right)^T \nabla f\left(x^k\right)}{\nabla f\left(x^k\right)^T A \nabla f\left(x^k\right)}$$
- *Energie-Norm (auch gewichtete Norm) $\|x\|_A^2 := x^T Ax$*

Dann gilt für den $k+1$-ten Fehler

$$\left\|x^{k-1} - x^*\right\|_A \leq \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \left\|x^k - x^*\right\|_A$$

mit $0 < \lambda_1 \leq \ldots \leq \lambda_n$ den Eigenwerten von A.

- *Konvergenz umso besser, je näher die Eigenwerte beieinander liegen*

Beachte*: $\kappa(A)$ Kondition von A*

$$\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} = \frac{\frac{\lambda_n}{\lambda_1} - 1}{\frac{\lambda_n}{\lambda_1} + 1} = \frac{\kappa(A) - 1}{\kappa(A) + 1} \longrightarrow \begin{cases} 1 & , k \longrightarrow \infty \\ 0 & , k \longrightarrow 1 \end{cases}$$

Für nichtquadratische f gilt folgendes Resultat.

Satz 3.4 (Nocedal & Wright 3.4)**.** *Sei $f\colon \mathbb{R}^n \to \mathbb{R}$ zweimal stetig differenzierbar. Angenommen, die Iterierten x^k des Gradientenverfahrens konvergieren zu einem x^*, wo $\nabla^2 f(x^*)$ positiv definit ist. Sei außerdem*

$$r \in \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}, 1\right)$$

wobei $\lambda_1 \leq \ldots \leq \lambda_n$ die Eigenwerte von $\nabla^2 f(x^*)$. Dann gilt

$$f\left(x^{k+1}\right) - f(x^*) \leq r^2 \left(f\left(x^k\right) - f(x^*)\right)$$

für alle k groß genug.

3.4 Das Newton-Verfahren

Sei $x^k \in \mathbb{R}^n$. Approximiere f um x^k durch ein quadratisches Modell

$$m_k\left(x^k + p\right) = f\left(x^k\right) + \nabla f\left(x^k\right)^T p + \frac{1}{2} p^T \nabla^2 f\left(x^k\right) p$$

- Falls $\nabla^2 f\left(x^k\right)$ positiv definit ist, hat m_k einen eindeutigen Minimierer

$$p_k = -\nabla^2 f\left(x^k\right)^{-1} \nabla f\left(x^k\right)$$

Im Prinzip liegt das bekannte Newton-Verfahren für die Optimalitätsbedingung $F(x) = \nabla f(x) = 0$ vor. Aber Abstiegsrichtungen erhält man nur, falls $\nabla F(x) = \nabla^2 f(x)$ positiv definit ist.
Wir wissen also:

- Das ungedämpfte Newton-Verfahren konvergiert lokal quadratisch, also viel schneller als das Gradientenverfahren.
- Insbesondere gibt es eine besondere Schrittweite: zumindest in der Nähe einer Läsung ist $t_k = 1$ eine gute Wahl.
- Weiterer Vorteil bei Optimierungsproblemen: Anders als ∇F für beliebige Vektorfunktionen ist $\nabla^2 F$ auf jeden Fall symmetrisch.

Falls $\nabla^2 f\left(x^k\right)$ nicht positiv definit ist, dann ... [3]

- Ersetze $\nabla^2 f\left(x^k\right)$ durch eine ähnliche Matrix, die positiv definit ist
- Addiere Vielfaches der Identität (Einheitsmatrix)
- Modifizierte Cholesky-Zerlegung
- ...

3.5 Konvergenzeigenschaften

Ein allgemeines Liniensuchverfahren konvergiert, falls ρ existiert, falls

$$\forall k\colon \cos\theta_k := \frac{-\nabla f\left(x^k\right)^T p_k}{\|\nabla f\left(x^k\right)\| \cdot \|p_k\|} > \rho$$

Lemma 3.2 (Übung)**.** *Sei $M \in \mathbb{R}$ eine obere Schranke der Kondition von $\nabla^2 f$, also*

$$\forall x^k\colon \kappa(\nabla^2 f) = \left\|\nabla^2 f\right\| \cdot \left\|\nabla^2 f^{-1}\right\| \leq M$$

Dann gilt $\cos\theta_k \geq \frac{1}{M}$.

Das Verfahren konvergiert , falls die Folge der $\nabla^2 f\left(x^k\right)$ beschränkte Kondition hat.

[3] müssen wir tricksen

Satz 3.5 (Nocedal & Wright, Satz 3.5)**.** *Sei $f\colon \mathbb{R}^n \to \mathbb{R}$ zweimal stetig differenzierbar und $x^* \in \mathbb{R}^n$ mit $\nabla f(x^*) = 0$ und $\nabla^2 f(x^*)$ positiv definit. Sei $\nabla^2 f$ Lipschitz-stetig in einer Umgebung von x^*, $\left(x^k\right)$ die Newton-Folge $x^{k+1} = x^k - \nabla^2 f\left(x^k\right)^{-1} \nabla f\left(x^k\right)$. Falls x^0 hinreichend nah an x^* liegt*

a) konvergiert die Folge gegen x^*.

b) Die Konvergenz ist lokal quadratisch.

c) Die Folge $\left(\left\|\nabla f\left(x^k\right)\right\|\right)$ konvergiert quadratisch gegen 0.

Beachte: für alle $k = 1$ hinreichend nah an x^* erfüllt die Schrittweite $t_k = 1$ die Wolfe-Bedingung.

3.6 Quasi-Newton-Verfahren

Nachteil vom Newton-Verfahren: die Auswertung von $\nabla^2 f$ kann schwierig/teuer sein. Quasi-Newton-Verfahren:

- Ersetze $\nabla^2 f\left(x^k\right)$ durch Approximation $B_k \in \mathbb{R}^{n\times n}$.
- Suchrichtung $p_k = -B_k^{-1} \nabla f\left(x^k\right)$.
- Konstruktion der B_k:

Idee: Die Folge der Gradienten $\left(\nabla f\left(x^k\right)\right)_k$ enthält Information über die zweiten Ableitungen von f.

$$f'\left(x^{k+1}\right) > f'\left(x^k\right) \implies f''\left(x^{k+1}\right) > 0 \quad ?$$

Formaler: Taylor-Entwicklung

$$\nabla f(x+p) = \nabla f(x) + \nabla^2 f(x) \cdot p + \underbrace{\int_0^1 \left[\nabla^2 f(x+sp) - \nabla^2 f(x)\right] p \,\mathrm{d}\, s}_{=:R(p)}$$

∇f ist stetig, deshalb gilt

$$\|R(p)\| \in o\left(\|p\|\right) \iff \lim_{\|p\|\to 0} \frac{\|R(p)\|}{\|p\|} = 0$$

Also folgt

$$\nabla f\left(x^{k+1}\right) = \nabla f\left(x^k\right) + \nabla^2 f\left(x^k\right) \cdot \left(x^{k+1} - x^k\right) + o\left(\left\|x^{k+1} - x^k\right\|\right)$$

Seien x^{k+1}, x^k in einer Umgebung von x^*, in der $\nabla^2 f$ „hinreichend“ positiv definit ist. Dann ist

$$\nabla^2 f(x^k) \underbrace{\left(x^{k+1} - x^k\right)}_{=:s^k} \approx \underbrace{\nabla f(x^{k+1}) - \nabla f(x^k)}_{=:y^k} \qquad \text{(Sekantengleichung)}$$

Idee des Quasi-Newton-Verfahrens: Konstruiere B_{k+1} so, dass diese Bedingung

$$B_{k+1}s^k = y^k$$

erfüllt ist. Die Sekantengleichung bestimmt B_{k+1} allerdings nur falls $n = 1$. Wir benötigen also zusätzliche Bedingungen. Weitere Wünsche:

- Symmetrie
- $B_{k+1} - B_k$ habe niedrigen Rang (spart viel Speicher, falls $k \leq n$)

Es existieren viele Varianten. Die wohl wichtigste Variante ist die *BFGS-Formel* (nach Broyden, Fletcher, Goldfarb, Shannon)

$$B_{k+1} = B_k - \frac{B_k s_k \cdot s_k^T B_k^T}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$$

Nützlich:

- $B_{k+1} - B_k$ hat Rang 2
- Alle B_k sind symmetrisch.
- Alle B_k erfüllen die Sekantengleichung.
- Alle B_k sind positiv definit, falls auch B_0 positiv definit ist.

Für Quasi-Newton-Methoden brauchen wir eine neue Art von Konvergenzgeschwindigkeit.

Definition. *Eine Folge* $\left(x^k\right)$ *konvergiert* superlinear *gegen* x^**, falls*

$$\lim_{k\to\infty} \frac{\left\|x^{k+1} - x^k\right\|}{\|x^k - x^*\|} = 0$$

Satz 3.6 (Nocedal & Wright, Satz 3.6)**.** *Sei* $f \in C^2\left(\mathbb{R}^n, \mathbb{R}\right)$*. Betrachte die Iteration*

$$x^{k+1} = x^k + t_k p_k$$

wobei:

a) p_k *ist Abstiegsrichtung*

b) t_k *erfülle die Wolfe-Bedingungen mit* $c \leq \frac{1}{2}$

Die Folge $\left(x^k\right)$ *konvergiere gegen ein* x^* *mit* $\nabla f\left(x^*\right) = 0$ *und* $\nabla^2 f\left(x^*\right)$ *positiv definit. Falls*

$$\lim_{k\to\infty} \frac{\left\|\nabla f\left(x^k\right) + \nabla^2 f\left(x^k\right) p_k\right\|}{\|p_k\|} = 0$$

Dann gilt:

i) Die Schrittweite $t_k = 1$ erfüllt die Wolfe-Bedingungen für alle k groß genug.

ii) Falls $t_k = 1$ gewählt wird für alle k groß genug, dann konvergiert $\left(x^k\right)$ superlinear.

Was heißt das für Quasi-Newton-Verfahren? Sei $p_k = -B_k^{-1}\nabla f\left(x^k\right)$. Dann ist die zentrale Bedingung aus Satz 3.6

$$\lim \frac{\left\|\nabla f\left(x^k\right) + \nabla^2 f\left(x^k\right) p_k\right\|}{\|p_k\|} = \lim \frac{\left\|\left(B_k - \nabla^2 f\left(x^k\right)\right) p_k\right\|}{\|p_k\|}$$

Beachte hierbei, dass

$$\nabla f\left(x^k\right) = B_k B_k^{-1} \nabla f\left(x^k\right) = -B_k p_k$$

Das sind gute Nachrichten!

→ Es heißt nämlich NICHT, dass die B_k immer bessere Approximationen von $\nabla^2 f\left(x^k\right)$ werden müssen.

→ Sie müssen $\nabla^2 f\left(x^k\right)$ nur *entlang der Suchrichtungen* immer besser approximieren.

3.7 Trust-Region-Verfahren

Bisher haben wir Liniensuchmethoden behandelt

- Suche *erst* eine Richtung $p_k \in \mathbb{R}^n$
- Suche dann eine Schrittweite $t_k \in \mathbb{R}$
- Setze $x^{k+1} = x^k + t_k p_k$

Trust-Region-Verfahren wählen p_k und t_k zusammen.

- Sei x^k die aktuelle Iterierte
- Approximiere f um x^k durch ein quadratisches Modell

  $$m_k(p) = f\left(x^k\right) + g^T p + \frac{1}{2} p^T B_k p$$

  mit $g_k = \nabla f\left(x^k\right)$, $B_k \in \mathbb{R}^{n\times n}$ ist Approximation von $\nabla^2 f\left(x^k\right)$, zum Beispiel $\nabla^2 f\left(x^k\right)$ selbst.

Idee: Vertraue m_k nur in einer Kugel um x^k (der Trust-Region) mit Radius $\triangle_k$.
Wähle als Korrektur-Schritt

$$p_k = \operatorname{argmin}_{\|p\| \le \triangle_k} m_k(p)$$

Wir erhalten den Vorteil, dass p_k immer definiert ist, selbst wenn B_k nicht positiv definit ist, dafür aber den Nachteil, dass in jedem Schritt ein Minimierungsproblem *mit Nebenbedingung* zu lösen ist. *Wie wählt man $\triangle_k$?* Grundlage: Wie gut hat m_k den Energieverlust $x^k \to x^k + p_k$ prognostiziert? Definiere:

$$\rho_k = \frac{f\left(x^k\right) - f\left(x^k + p_k\right)}{m_k(0) - m_k\left(p_k\right)}$$

Wähle zwei Konstanten $0 < \eta_1 < \eta_2 < 1$, z.B. $\eta_1 = 0,1, \eta_2 = 0,9$. Für $k = 0, 1, 2, \ldots$

- Setze $p_k = \operatorname{argmin}_{\|p\| < \triangle_k} m_k(p)$
- Berechne ϱ_k
- Fall 1: $\varrho_k < \eta_1$
  1) $x^{k+1} = x^k$
  2) $\triangle_{k+1} = \frac{1}{2}\triangle_k$
- Fall 2: $\eta_1 \leq \varrho_k \leq \eta_2$
  1) $x^{k+1} = x^k + p_k$
- Fall 3: $\varrho_k > \eta_2$
  1) $x^{k+1} = x^k + p_k$
  2) $\triangle_{k+1} = 2\triangle_k$

3.8 Globale Konvergenz

Definition (Cauchy-Punkt)**.** *Der Cauchy-Punkt p_k^c ist der Minimierer von m_k innerhalb der Trust-Region in Richtung des negativen Gradienten.*

$$p_k^c = -\frac{g_k}{\|g_k\|} \cdot \tau_k$$

wobei

$$\tau_k = \begin{cases} \triangle_k & , \text{falls } g_k^T B_k g_k < 0 \\ \min\left\{ \triangle_k, \underbrace{\frac{\|g_k\|^3}{g_k^T B_k g_k}}_{\text{Billig}} \right\} & , \text{sonst} \end{cases}$$

Der Cauchy-Punkt erzeugt Energieabstieg *im Modell* ähnlich wie der von der Armijo-Bedingung gefordert.

Lemma 3.3 (Nocedal & Wright, 4.3)**.** *Für den Cauchy-Punkt p_k^c gilt*

$$m(0) - m\left(p_k^c\right) \geq \frac{1}{2}\|g_k\| \cdot \min\left\{ \triangle_k, \frac{\|g_k\|}{\|B_k\|} \right\} \qquad (*)$$

Satz 3.7. *Falls [technische Bedingungen], und p_k so gewählt wird, dass (*) für alle $k \in \mathbb{N}$ erfüllt ist, dann folgt*

$$\lim_{k\to\infty} \|g_k\| = 0$$

3.9 Das Hundebein-Verfahren

[dogleg method] Sei B_k positiv definit. Der Minimierer von m_k ohne Nebenbedingung ist

$$p^B = -B_k^{-1} g_k$$

Falls p^B zulässig ist, also $\left\|p^B\right\| \leq \triangle$, dann ist p^B auch Lösung des quadratischen Minimierungsproblems *mit Nebenbedingungen*. Sei $p^*(\triangle)$ der Minimierer von m_k in der Trust-Region als Funktion des Radius. Sei $\triangle$ klein im Verhältnis zu $\left\|p^B\right\|$. Dann ist der quadratische Term in $m(p) = f\left(x^k\right) + g_k^T p + \frac{1}{2} p^T B_k p$ eher irrelevant. Dann ist der Minimierer von m_k ungefähr der Cauchy-Punkt

$$p^*(\triangle) \approx -\triangle \frac{g}{\|g\|}$$

dogleg-Methode: Wähle p_k als Minimierer von m_k auf dem gelben Pfad unter der Nebenbedingung $\|p_k\| \leq \triangle_k$.

Satz 3.8. *Der Vektor p^* ist Minimierer von*

$$\min_{\|p\|\leq\triangle_k} m(p) = f\left(x^k\right) + g^T p + \frac{1}{2} p^T B_k p$$

$\iff$ *$\|p^*\| \leq \triangle_k$, und es eine Zahl $\lambda \geq 0$ gibt so dass*

$$\begin{aligned} (B + \lambda I) p^* &= -g \\ \lambda(\triangle - \|p^*\|) &= 0 \end{aligned}$$

und $(B + \lambda I)$ ist positiv semidefinit.

Berechnungsmethode von Minimierern

Algorithmus. *Für λ groß genug definiere*

$$p(\lambda) = -(B + \lambda I)^{-1} g$$

Falls $\|p^*\|$ auf dem Rand der Trust-Region liegt, dann verwende das Newton-Verfahren zu Lösen von

$$\|p(\lambda)\| - \triangle_k = 0$$

4 Iterative Lösungsverfahren für große, dünnbesetzte Gleichungssysteme

In manchen Anwendungen stößt man auf Matrizen, die sehr groß sind, aber fast ausschließlich Nullen enthalten.

Solche Matrizen nennt man *dünnbesetzt* oder *dünn* (engl. *sparse*).

4.1 Motivation: Das Poisson-Problem

Sei Ω eine offene, beschränkte Menge in $\mathbb{R}^2$, und $f : \Omega \to \mathbb{R}$ eine gegebene Funktion. Gesucht wird eine Funktion $u : \Omega \to \mathbb{R}$ für die

$$\begin{aligned} -\Delta u := -\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} &= f \qquad \text{auf } \Omega, \\ u &= 0 \qquad \text{auf dem Rand von } \Omega. \end{aligned}$$

Solch eine Funktion u beschreibt z. B.

- Temperaturverteilung bei gegebener Wärmezufuhr f,
- Elektrostatisches Potential bei gegebener Ladungsdichte f,
- Flüssigkeitsdruck in einem porösen Medium.

Wie findet man so ein u?
Eine Möglichkeit: Finite Differenzen

- Sei $g : \mathbb{R} \to \mathbb{R}$ hinreichend oft stetig differenzierbar.
- Taylorentwicklung um ein $x \in \mathbb{R}$:

  $$g(x+h) = g(x) + g'(x)h + \text{Rest},$$

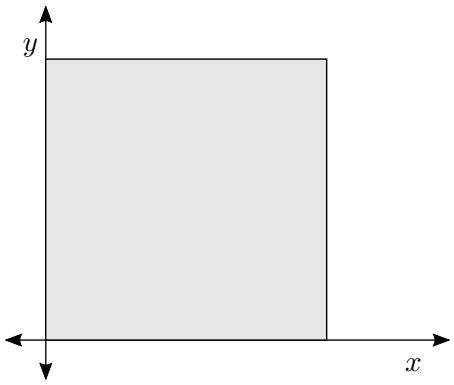
  also

  $$g'(x) \approx \frac{g(x+h) - g(x)}{h}$$

- Ähnlich erhält man

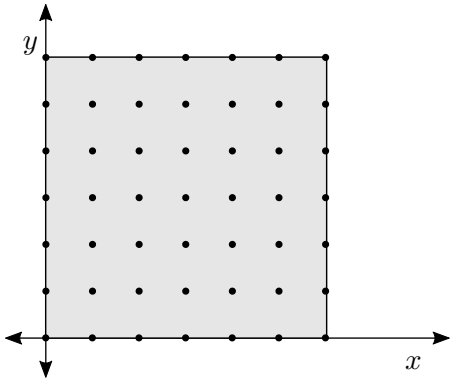
  $$g''(x) \approx \frac{g(x+h) - 2g(x) + g(x-h)}{h^2}.$$

Das machen wir jetzt für die zweiten partiellen Ableitungen.
Sei Ω der Einfachheit halber das Einheitsquadrat.



Wähle ein $N \in \mathbb{N}$, definiere die Gitterweite $h := \frac{1}{N}$, und das Gitter

$$(x_i, y_j) := (ih, jh), \qquad 0 \le i, j \le N.$$



Betrachte den Laplace-Operator an einem inneren Gitterpunkt (x_i, y_j):

$$\begin{aligned}
\Delta u(x_i, y_j) &= \frac{\partial^2 u(x_i, y_j)}{\partial x^2} + \frac{\partial^2 u(x_i, y_j)}{\partial y^2} \\
&\approx \frac{u(x_i + h, y_j) - 2u(x_i, y_j) + u(x_i - h, y_j)}{h^2} + \frac{u(x_i, y_j + h) - 2u(x_i, y_j) + u(x_i, y_j - h)}{h^2} \\
&= \frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{h^2} + \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1})}{h^2}.
\end{aligned}$$

Nummeriere die Gitterknoten von links unten nach rechts oben durch.
Seien u_i, f_i die Werte der Funktionen u bzw. f am i-ten Gitterknoten.

Man erhält das lineare Gleichungssystem

$$\frac{1}{h^2}\begin{pmatrix} 4 & -1 & 0 & \dots & 0 & -1 & 0 & \dots \\ -1 & 4 & -1 & 0 & \dots & 0 & -1 & \dots \\ 0 & -1 & 4 & -1 & \dots & \dots & 0 & \dots \\ & 0 & -1 & 4 & -1 & & & \\ & & & & \ddots & & & \\ \vdots & 0 & & \dots & -1 & 4 & -1 & 0 \\ 0 & -1 & 0 & \dots & 0 & -1 & 4 & -1 \\ \dots & 0 & -1 & 0 & \dots & 0 & -1 & 4 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}.$$

4.1.1 Eigenschaften der Matrizen

Größe

Die Matrizen aus dem obigen Beispiel können sehr groß werden.

- Für jeden Gitterknoten eine Gleichung
- Für d-dimensionale Gebiete hat man etwa $n \approx \mathrm{Vol}(\Omega) \cdot h^{-d}$ Knoten
- Je feiner das Gitter, desto präziser die Lösung, desto größer aber auch die Matrix.
- Mein Laptop: ca. 8 GB RAM; eine Zahl in doppelter Genauigkeit braucht 8 Byte. Also hat man Platz für 1 Milliarde Zahlen.
- Aktuelle Hochleistungsrechner: $n \approx 10^{11}$.

Wie löst man diese Gleichungssysteme? Direkte Verfahren wie Gauß-Elimination oder Cholesky-Zerlegung sind i.A. zu teuer.
Erinnerung: Gauß-Elimination braucht $O(n^3)$ Rechenoperationen!

Dünnbesetztheit

- Sei i ein innerer Knoten im Gitter für das Poisson-Problem.
- Die Gleichung für u_i ist

  $$4u_i - u_{i-1} - u_{i+1} - u_{i-(N+1)} - u_{i+N+1} = f_i.$$

  Die i-te Zeile von A enthält also nur 5 Einträge, der Rest ist Null.
- Allgemein: Die Anzahl der Nicht-Null-Einträge pro Zeile ist durch eine kleine Konstante beschränkt.
- Die Matrix enthält also nur $O(n)$ Einträge

- Wendet man das Gauß-Verfahren auf solch eine Matrix an, so entstehen bei den Zwischenschritten in der Matrix eine beträchtliche Anzahl von zusätzlichen Einträgen („fill-in“).
- Das Gauß-Verfahren ist deshalb nicht nur zu langsam, es braucht auch zu viel Speicher.

Konsequenz: Wir brauchen Algorithmen und Datenstrukturen, die die Dünnbesetztheit ausnutzen.

Beispiel: Matrix–Vektor-Multiplikation $v = Aw$.

1. Naiv:
```
for alle Zeilen i do
    v_i = 0
    for alle Spalten j do
        v_i = v_i + A_ij w_j
    end
end
```
Das braucht $O(n^2)$ Operationen.

2. Pseudo-schlau:
```
for alle Zeilen i do
    v_i = 0
    for alle Spalten j do
        if A_ij ≠ 0 then
            v_i = v_i + A_ij w_j
        end
    end
end
```
Das braucht ebenfalls $O(n^2)$ Operationen!

3. Wirklich schlau:
```
for alle Zeilen i do
    v_i = 0
    for alle Spalten j in denen A_ij ≠ 0 do
        v_i = v_i + A_ij w_j
    end
end
```
Das braucht nur $O(\#\text{Nichtnulleinträge})$ Operationen.

Besondere Forschungsrichtung: direkte Sparse-Verfahren (Das machen wir in Kapitel 5.)

4.2 Lineare iterative Verfahren

Sei $A \in \mathbb{R}^{n\times n}$ nichtsingulär, groß und dünnbesetzt und $b \in \mathbb{R}^n$. Finde $x \in \mathbb{R}^n$ so, dass

$$Ax = b$$

(x^* sei von nun an die Lösung).

Idee der iterativen Verfahren:

1. Wähle eine Startiterierte $x^0 \in \mathbb{R}^n$.
2. Berechne daraus eine Iterierte $x^1 \in \mathbb{R}^n$. x^1 ist zwar nicht die Lösung, aber hoffentlich „näher dran“ als x^0.
3. Wiederhole 2) so lange, bis ausreichend Genauigkeit erreicht ist.

Man erhält eine Folge $x^0, x^1, x^2, \ldots$, die (hoffentlich) gegen x^* konvergiert.

Es gibt sehr viele Ansätze für 2). Ein paar werden wir jetzt betrachten.

Folgende Idee führt auf eine ganze Klasse von Verfahren: Das *Residuum* $b - Ax$ ist eine Art Fehler. Dann ist

$$x^{k+1} := x^k + b - Ax^k$$

vielleicht näher an x^* als x^k.

Formaler: Schreibe $Ax = b$ als Fixpunktgleichung

$$x = x + C(b - Ax)$$

mit $C \in \mathbb{R}^{n\times n}$ nichtsingulär. Setze

$$\Phi\colon \mathbb{R}^n \to \mathbb{R}^n, \qquad x \mapsto x + C(b - Ax).$$

Die Lösung x^* ist Fixpunkt von Φ.

Dafür machen wir jetzt eine Fixpunktiteration:

$$x^{k+1} := \Phi(x^k) = x^k + C(b - Ax^k) = (I - CA)x^k + Cb, \qquad k = 0, 1, 2, \ldots$$

4.2.1 Konvergenz

Unter welchen Umständen konvergiert dieses Verfahren?

Der Fehler im k-ten Iterationsschritt ist $e^k = x^k - x^*$. Es gilt

$$e^{k+1} = x^{k+1} - x^* = \Phi(x^k) - \Phi(x^*) = \underbrace{(I - CA)}_{\text{Iterationsmatrix}} e^k.$$

Also gilt

$$e^k = (I - CA)^k e^0 \qquad \forall k = 0, 1, 2, \ldots \tag{4.1}$$

Definition. *Die Matrix $I - CA$ heißt* Iterationsmatrix *der Methode.*

Die Fehlerfortpflanzung (4.1) ist linear, deshalb werden solche Verfahren lineare Verfahren genannt.

Für die Konvergenz gilt der folgende wichtige Satz.

Satz 4.1. *Sei $\rho(I - CA)$ der Spektralradius von $I - CA$. Das Verfahren konvergiert für jeden Startwert $x^0 \in \mathbb{R}$ gegen die Lösung von $Ax = b$ genau dann, wenn $\rho(I - CA) < 1$.*

Beweis. Wir beweisen nur den einfachen, aber wichtigen Fall, dass $I - CA$ symmetrisch positiv-definit (s.p.d.) ist.
Erstens: Aus $\rho < 1$ folgt Konvergenz.

- $I - CA$ ist symmetrisch und positiv definit, also diagonalisierbar. Das heißt es existiert eine nichtsinguläre Matrix T so dass

$$T^{-1}(I - CA)T = \begin{pmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{pmatrix} =: D,$$

  wobei $\lambda_1, \lambda_2, \ldots, \lambda_n$ die Eigenwerte von $I - CA$ sind.

- $e^k = (I - CA)^k e^0 = \left(TDT^{-1}\right)^k e^0 = TD^kT^{-1}e^0$.

- Schätze e^k in einer Norm ab, z.B. der $\|\cdot\|_2$-Norm

$$\begin{aligned} \|e^k\|_2 &= \|TD^kT^{-1}e^0\|_2 \\ &\leq \|T\|_2 \cdot \|D^k\|_2 \cdot \|T^{-1}e^0\|_2 \\ &\leq \|T\|_2 \cdot \|T^{-1}e^0\|_2 \cdot \underbrace{\max_{i=1,\ldots,n} |\lambda_i|^k}_{=\rho(I-CA)}. \end{aligned}$$

- Dieser Term geht gegen 0, wenn $\rho(I - CA) = \max_i |\lambda_i| < 1$ ist.

Zeige jetzt: Aus Konvergenz folgt $\rho < 1$

- Angenommen $|\lambda_j| \geq 1$ für ein j, und $|\lambda_j| = \rho(I - CA)$.

- Sei v ein zu λ_j gehörender Eigenvektor.

- Wähle als Startwert $x^0 = x^* + v$, also $e^0 = v$.

- Dann folgt

$$\|e^k\|_2 = \|(I - CA)^k e_0\|_2 = \|(I - CA)^k v\|_2 = |\lambda_j|^k \|v\|_2 \geq \|e^0\|_2$$

  für alle k.

$\Longrightarrow$ Das Verfahren konvergiert nicht.

In allen endlich-dimensionalen Vektorräumen sind alle Normen äquivalent. Deshalb ändert sich das Resultat auch nicht, wenn man eine andere Norm betrachtet. □

Der Spektralradius einer Matrix ist nur mit Mühe auszurechnen. Allerdings gilt $\rho(B) \leq \|B\|$ für jede submultiplikative Matrixnorm.

[Denn: Sei v ein Eigenvektor von B zum Eigenwert λ. Dann ist

$$|\lambda|\|v\| = \|\lambda v\| = \|Bv\| \leq \|B\|\|v\|.$$

Deshalb gilt $|\lambda| \leq \|B\|$ für alle Eigenwerte λ von B.]

Deshalb:

Korollar. Für jede Vektornorm $\|\cdot\|$ mit dazugehöriger Operatornorm gilt

$$\forall k = 0, 1, 2, \ldots : \|x^k - x^*\| \leq \|I - CA\|^k \cdot \|x^0 - x^*\|.$$

Das Verfahren konvergiert genau dann, wenn $\|I - CA\| < 1$ für eine beliebige Norm gilt.

4.2.2 Konvergenzgeschwindigkeit

Man möchte gerne wissen, *wie schnell* die Folge $x^0, x^1, x^2, \ldots$ gegen x^* konvergiert. Fehlerreduktionsrate im k-ten Schritt: $\frac{\|e^k\|}{\|e^{k-1}\|}$ und gemittelt über die ersten k Schritte

$$\sqrt[k]{\frac{\|e^k\|}{\|e^0\|}} =: \rho_k$$

Auch die Größe ρ_k hängt mit $\rho(I - CA)$ zusammen!

Beweis.

- Sei $I - CA$ wieder diagonalisierbar
- Eigenvektorbasis: $v_1, v_2, \ldots, v_n$
- Nummerierung nach absteigenden Eigenwerten

$$|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|$$

- Stelle den Anfangsfehler e^0 in der Eigenvektorbasis dar:

$$e^0 = \sum_{i=0}^{n} c_i v_i$$

- Sei o.B.d.A. $c_1 \neq 0$

- Es gilt

$$\begin{aligned} e^k = (I - CA)^k \sum_{i=1}^{n} c_i v_i &= \sum_{i=1}^{n} c_i \lambda_i^k v_i \\ &= c_1 \lambda_i^k v_1 + \sum_{i=2}^{n} c_i \lambda_i^k v_i \\ &= \lambda_1^k \left(c_1 v_1 + \underbrace{\sum_{i=2}^{n} c_i \left(\frac{\lambda_i}{\lambda_1} \right)^k v_i}_{=:r^k} \right) \\ &= \lambda_1^k (c_1 v_1 + r^k). \end{aligned}$$

- Es gilt $c_1 \neq 0$ und $\left|\frac{\lambda_i}{\lambda_1}\right| \leq 1 \implies \exists$ Konstanten $c_{\min}, c_{\max}$ (unabhängig von k), sodass

$$0 < c_{\min} < \left\| c_1 v_1 + r^k \right\| \leq c_{\max}$$

- Deshalb:

$$\rho_k = \sqrt[k]{\frac{\|e^k\|}{\|e^0\|}} = \frac{|\lambda_1| \|c_1 v_1 + r^k\|^{\frac{1}{k}}}{\|e^0\|^{\frac{1}{k}}} \xrightarrow{k \longrightarrow \infty} |\lambda_1| = \rho\left(I - CA\right) \qquad \square$$

Wie viele Schritte braucht man, um den Fehler um den Faktor $\frac{1}{e} \approx \frac{1}{2{,}718\ldots}$ zu reduzieren?

$$(\rho_k)^k := \frac{\|e^k\|}{\|e^0\|} \approx \frac{1}{e} \iff k \approx \frac{1}{-\ln \rho_k}$$

- Interpretiere $-\ln \rho^k$ als Konvergenzgeschwindigkeit.
- Asymptotische Konvergenzgeschwindigkeit:

$$-\ln(\rho(I - CA))$$

- Für große k ist $\rho(I - CA)$ in etwa die gemittelte Fehlerreduktionsrate.

4.2.3 Die Wahl von C

Wie soll man die Matrix C wählen?
Ziel: $\rho(I - CA)$ soll möglichst klein sein.

- Ideal wäre $C = A^{-1} \implies \rho(I - CA) = 0$. Das Verfahren konvergiert dann in einem Schritt

$$x^1 = x^0 + A^{-1}(b - Ax^0) = A^{-1}b = x^*.$$

- Die Durchführung dieses Schrittes wäre aber sehr teuer. Es muss das LGS
$$A(x^1 - x^0) = b - Ax^0$$
gelöst werden.

Wir haben somit nichts gewonnen. Es ergibt sich folgendes Dilemma:

1. C soll A^{-1} möglichst gut approximieren
2. Die Operation $y \mapsto Cy$ soll möglichst billig sein

Beachte: Die Matrix C wird nie explizit ausgerechnet!

4.2.4 Das Jacobi-Verfahren

Wir nehmen im Folgenden an, dass $a_{ii} \neq 0$ für alle $i = 1, \ldots, n$.

Betrachte das lineare Gleichungssystem:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1m}x_m &= b_1 \\ a_{12}x_1 + a_{22}x_2 + \ldots + a_{2m}x_m &= b_2 \\ &\vdots \end{aligned}$$

Idee: Löse i-te Zeile nach x_i auf für $i = 1, \ldots, n$.

$$\begin{aligned} x_1 &= \frac{1}{a_{11}} (b_1 - a_{12}x_2 - \ldots - a_{1n}x_n) \\ x_2 &= \frac{1}{a_{22}} (b_2 - a_{21}x_1 - \ldots - a_{2n}x_n) \\ &\vdots \end{aligned}$$

Mache daraus ein iteratives Verfahren:

$$\begin{aligned} x_1^{k+1} &= \frac{1}{a_{11}} \left(b_1 - a_{12}x_2^k - \ldots - a_{1n}x_n^k\right) \\ x_2^{k+1} &= \frac{1}{a_{22}} \left(b_2 - a_{21}x_1^k - \ldots - a_{2n}x_n^k\right) \\ &\vdots \end{aligned}$$

Für $i = 1, \ldots, n$

$$x_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - a_{i1}x_1^k - a_{i2}x_2^k - \ldots - a_{i,i-1}x_{i-1}^k - a_{i,i+1}x_{i+1}^k - \ldots - a_{in}x_n^k\right).$$

Oder, kompakter

$$\forall i \in \{1, 2, \ldots, n\}: \qquad x_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij}x_i^k\right)$$

Beachte:

- Die Rechnungen für die verschiedenen i sind voneinander unabhängig $\implies$ leicht zu parallelisieren.
- In der Praxis gilt die Summe natürlich nur über die Einträge der i-ten Zeile von A, die $\neq 0$ sind.

Darstellung als lineares Verfahren

Sei $D := \operatorname{diag}(a_{11}, \ldots, a_{nn}) \in \mathbb{R}^{n \times n}$. Die Iterationsvorschrift lässt sich schreiben als

$$\begin{aligned} x^{k+1} &= D^{-1}(b - (A - D)x^k) \\ &= D^{-1}b - D^{-1}(A - D)x^k \\ &= D^{-1}b - D^{-1}Ax^k + x^k \\ &= x^k + D^{-1}(b - Ax^k) \end{aligned}$$

$\implies$ lineares Verfahren mit $C = D^{-1}$.

Alternative Formulierung

- Sei $-L$ die Matrix aller Einträge von A unterhalb der Diagonalen
- Sei $-U$ die Matrix aller Einträge von A oberhalb der Diagonalen
- Also $A = D - L - U$
- Jacobi-Iteration:
$$Dx^{k+1} = (L + U)x^k + b$$

Konvergenz

Wir wenden das bekannte Konvergenzkriterium an:

Satz 4.2. *Das Jacobi Verfahren konvergiert genau dann, wenn* $\rho(I - D^{-1}A) < 1$.

Leider gilt diese Bedingung nicht immer.

Beispiel. Folgende Situation:

$$A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \implies I - \underbrace{D^{-1}}_{=I} A = \begin{pmatrix} 0 & -2 \\ -2 & 0 \end{pmatrix}$$

$$\lambda_{1,2} = \pm 2 \text{ sind Eigenwerte}$$

$$v_{1,2} = \begin{pmatrix} \pm 1 \\ 1 \end{pmatrix} \text{ sind Eigenvektoren}$$

$\implies \rho(I + D^{-1}A) = 2 > 1$. Das Verfahren konvergiert also *nicht*.

Es gibt schwächere Kriterien, die aber einfacher zu handhaben sind.

Definition. *$A \in \mathbb{R}^{n\times n}$ heißt* irreduzibel, *falls es keine Permutationen der Zeilen und Spalten gibt, so dass A die Form*

$$\begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{pmatrix}$$

bekommt, wobei $\tilde{A}_{11} \in \mathbb{R}^{k\times k}, 1 \le k < n$ (quadratisch) ist.

A heißt diagonaldominant, *falls $|a_{ii}| \ge \sum_{j\neq i}|a_{ij}|$ für $i = 1, \dots, n$ mit strikter Ungleichheit für mindestens ein i.*

Satz 4.3. *Das Jacobi-Verfahren konvergiert, falls mindestens eine der folgenden Bedingungen gilt:*

- *A ist symmetrisch positiv definit, und $2D - A$ ist auch symmetrisch positiv definit.*
- *A ist irreduzibel und diagonaldominant.*

Anwendung auf das Poissonproblem

Quelle angeben!

Sei A die Matrix des Poisson-Problems. Gleichungssystem:

$$\frac{1}{h^2}\left(4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1}\right) = f_i.$$

Jacobi-Verfahren: $D = 4h^{-2}I$.

Wir versuchen jetzt, den Spektralradius von $I - CA = I - D^{-1}A$ abzuschätzen.

Dazu benutzen wir den *Rayleigh-Quotienten*

$$R(A, x) := \frac{x^T Ax}{\|x\|^2}.$$

Für symmetrische A gilt $\lambda_{\min}(A) \le R(A, x) \le \lambda_{\max}(A)$, und diese Schranken werden angenommen, wenn x entsprechende Eigenvektoren sind.

Damit

$$\begin{aligned}
\rho(I - D^{-1}A) &= \sup_{x\neq 0}\left|\frac{x^T\left(I - D^{-1}A\right)x}{\|x\|^2}\right| \\
&= \sup_{x\neq 0}\left|1 - \frac{x^T D^{-1}Ax}{\|x\|^2}\right| \\
&= \sup_{x\neq 0}\left|1 - \frac{1}{4}h^2\frac{x^T Ax}{\|x\|^2}\right|
\end{aligned}$$

Daraus folgt dass

$$\rho(I - D^{-1}A) = \sup\left\{\left|1 - \frac{1}{4}h^2\lambda\right| : \lambda \text{ Eigenwert von A}\right\}$$

Für die spezielle Matrix können wir die Eigenwerte ausrechnen. Diese sind alle nichtnegativ.

Der größte Eigenwert von A ist: Quelle angeben!

$$\lambda = \frac{8}{h^2} \sin^2\left(\frac{1}{2}\pi h\right)$$

Es folgt, dass

$$\rho(I - D^{-1}A) = 1 - 2\sin^2\left(\frac{1}{2}\pi h\right) = \cos(\pi h)$$

Taylor-Entwicklung:

$$\cos(\pi h) = 1 - \frac{1}{2}\pi^2 h^2 + \dots$$

Also ist

$$\frac{\|e^{k+1}\|}{\|e^k\|} \approx \rho(I - D^{-1}A) \approx 1 - \frac{1}{2}\pi^2 h^2$$

Dieser Ausdruck geht „quadratisch“ (also ziemlich schnell) gegen 1 wenn $h \to 0$.

Wir wollen ausrechnen, wie viele Iterationen man ungefähr braucht, um den Anfangsfehler um einen Faktor R zu reduzieren. D.h., welches k soll man wählen, um ungefähr

$$\frac{\|e^k\|}{\|e^0\|} \leq \frac{1}{R}$$

zu erhalten?

Da $\frac{\|e^k\|}{\|e^0\|} \approx \rho^k$ erhält man

$$k = \log_\rho \frac{1}{R} = \frac{-\ln R}{\ln \rho\,(I - D^{-1}A)} \approx \frac{-\ln R}{\ln\left(1 - \frac{1}{2}\pi^2 h^2\right)} \approx \frac{2}{\pi^2 h^2} \ln R,$$

da $\ln x \approx (x-1) - \frac{1}{2}(x-1)^2 + \dots$ ist.

Für ein doppelt so feines Gitter braucht man viermal so viele Iterationen (und diese sind natürlich auch noch teurer.).

Also ist dies kein so gutes Verfahren.

4.2.5 Das Gauß-Seidel-Verfahren

- Carl-Friedrich Gauß
- Philipp Ludwig von Seidel, 1821–1896, Mathematiker, Optiker, Astronom

Betrachte noch einmal die Jacobi-Rechenvorschrift:

$$x_i^{k+1} = \frac{1}{a_{ii}}\left(b_i - a_{i1}x_1^k - a_{i2}x_2^k - \dots - a_{i,i-1}x_{i-1}^k - a_{i,i+1}x_{i+1}^k - \dots - a_{in}x_n^k\right).$$

Eigentlich haben wir für $x_1, \dots, x_{i-1}$ schon bessere Werte als $x_1^k, \dots, x_{i-1}^k$, nämlich $x_1^{k+1}, \dots, x_{i-1}^{k+1}$.

Gauß-Seidel-Verfahren:

$$\begin{aligned} x_i^{k+1} &= \frac{1}{a_{ii}} \left(b_i - a_{i1} x_1^{k+1} - \ldots - a_{i,i-1} x_{i-1}^{k+1} - a_{i,i+1} x_{i+1}^{k} - \ldots - a_{in} x_n^k \right) \\ &= \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij} x_j^k \right). \end{aligned}$$

x_i^{k+1} hängt von x_{i-1}^{k+1} ab $\implies$ keine Parallelisierung möglich.

Gauß-Seidel als lineares Verfahren

Seien $D, -L, -U$ Diagonalteil, linker, rechter Dreiecksteil von A.

$$x^{k+1} = D^{-1}(b + Lx^{k+1} + Ux^k)$$

Wir ziehen D und L auf die linke Seite

$$(D - L)x^{k+1} = Ux^k + b,$$

und lösen nach x^{k+1} auf

$$\begin{aligned} x^{k+1} &= (D-L)^{-1}Ux^k + (D-L)^{-1}b \\ &= x^k + \left[(D-L)^{-1}U - I\right] + (D-L)^{-1}b \\ &= x^k + (D-L)^{-1} \underbrace{\left[U - D + L\right]}_{=-A} x^k + (D-L)^{-1}b \\ &= x^k + (D-L)^{-1}(b - Ax^k). \end{aligned}$$

$\implies$ Lineares Verfahren mit $C = (D-L)^{-1}$.

Konvergenz

Wir erwarten bessere Konvergenzeigenschaften als für das Jacobi-Verfahren.

Und in der Tat:

Satz 4.4. *Das Gauß-Seidel-Verfahren konvergiert, wenn*

- *A symmetrisch und positiv definit ist und/oder*
- *A irreduzibel und diagonal dominant ist.*

Konvergenzgeschwindigkeit

Beispiel. Sei A die Matrix des Poisson-Problems für ein quadratisches Gebiet.

- Es gilt $\rho(I - (D-L)^{-1}A) = (\rho(I - D^{-1}A))^2$
  Quelle angeben! Steht angeblich irgendwo bei Hackbusch.

- Deshalb
$$\rho(I - (D - L)^{-1}A) = \cos^2(\pi h)$$
- Taylor-Entwicklung für $\cos^2$
$$\cos^2 \pi h \approx \left(1 - \frac{1}{2}\pi^2 h^2 + \dots\right)^2 = 1 - 2\frac{1}{2}\pi^2 h^2 + \frac{1}{4}\pi^4 h^4 + \dots \approx 1 - \pi^2 h^2$$
- Fehlerreduktion
$$\frac{\|e^{k+1}\|}{\|e^k\|} \approx 1 - \pi^2 h^2$$
- Um den Startfehler um den Faktor R zu reduzieren, braucht man etwa
$$\frac{-\ln R}{\ln \rho(I - (D - L)^{-1}A)} \approx \frac{-\ln R}{\ln(1 - \pi^2 h^2)} \approx \frac{\ln R}{\pi^2 h^2}$$
Iterationen.

Faustregel: Das Gauß-Seidel-Verfahren braucht nur etwa halb so viele Iterationen wie das Jacobi-Verfahren.

4.2.6 Abbruchkriterien

Wie viele Iterationen soll man machen?

- Schätzungen wie „$\frac{1}{\pi^2 h^2} \ln R$“ sind nur für wenige Spezialfälle bekannt.

Idealerweise iteriert man so lange, bis $\|e^k\| < K$ mit K vorgegeben.

Wie sollte man $\|e^k\|$ ausrechnen/abschätzen?

Residuum soll den Index unten bekommen, das spart später Klammern.

Beliebter Ansatz: Betrachte das Residuum $r^k := b - Ax^k$. Es gilt
$$\|e^k\| = \|x^* - x^k\| = \|A^{-1}(b - Ax^k)\| = \|A^{-1}r^k\| \le \|A^{-1}\| \cdot \|r^k\|.$$
Das Residuum schätzt den Fehler von oben ab, *wenn* $\|A^{-1}\|$ *bekannt ist.*

Abbruchbedingung $\|r^k\| < K$ ist nicht sinnvoll!

Stattdessen: Breche ab, sobald
$$\frac{\|r^k\|}{\|r^0\|} < K$$
Die Idee dahinter ist
$$\frac{\|e^k\|}{\|e^0\|} = \frac{\|A^{-1}r^k\|}{\|A^{-1}r^0\|} \approx \frac{\|r^k\|}{\|r^0\|}$$
Das ist aber nicht wirklich mathematisch zu rechtfertigen.

Ausgefeilterer Ansatz: Um $\|e^k\|$ abzuschätzen:

- Berechne m weitere Iterationen,
- Schätze e^k durch $e^m - e^k$ ab.

4.3 Das Gradientenverfahren

(Auch bekannt als: Verfahren des steilsten Abstiegs, steepest descent, etc.)

Die folgenden Verfahren sind *nichtlinear*. Das heißt, dass die Fehlerfortpflanzung von einem Schritt zum nächsten nicht linear ist.

Der Inhalt dieses Kapitels ist weitestgehend dem Artikel von Shewchuk, *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain* [6] entnommen.

Ab jetzt sei A immer symmetrisch und positiv definit. Betrachte die Funktion

$$f\colon \mathbb{R}^n \to \mathbb{R}, \qquad f(x) = \frac{1}{2}x^T Ax - bx.$$

Satz 4.5. *Die Lösung x^* von $Ax = b$ ist eindeutiger Minimierer von f.*

Beweis. 1. x^* ist stationärer Punkt von f, denn

$$f'(x) = \frac{1}{2}A^T x + \frac{1}{2}Ax - b = Ax - b \implies f'(x^*) = 0.$$

2. x^* ist Minimierer, denn für $p \neq x^*$ ergibt etwas Rechnen

$$f(p) = f(x^*) + \frac{1}{2}\underbrace{(p-x^*)^T A\,(p-x^*)}_{>0} > f(x^*),$$

da A positiv definit ist. □

Die Umformulierung des Gleichungssystems $Ax = b$ in ein Minimierungsproblem ermöglicht ein neue Sichtweise des Problems. Statt Lösungen eines Gleichungssystems können wir jetzt nach Minimierern einer Energie suchen.

4.3.1 Idee des Gradientenverfahrens

Idee: Ausgehend von x^k, mache einen Schritt in Richtung des steilsten Abstieges.

Diese Richtung ist $-f'(x^k) = b - Ax^k = r^k$ (das Residuum).
Ein Schritt ist

$$x^{k+1} = x^k + \alpha^k r^k, \qquad \alpha^k \in \mathbb{R} \text{ die Schrittlänge.} \tag{4.2}$$

Wie lang soll der Schritt sein?
Liniensuche: Minimiere f entlang der Suchrichtung.
Also

$$0 = \frac{d}{d\alpha}f(x^{k+1}) = f'(x^{k+1})^T \cdot \frac{dx^{k+1}}{d\alpha} = f'(x^{k+1})^T r^k.$$

Es ist aber $f'(x^{k+1}) = -r^{k+1}$, und deshalb

$$\begin{aligned} 0 = (r^{k+1})^T r^k &= (b - Ax^{k+1})^T r^k \\ &= \left(b - A(x^k + \alpha^k r^k)\right)^T r^k = \underbrace{(b - Ax^k)^T}_{=(r^k)^T} r^k - \alpha^k (Ar^k)^T r^k \end{aligned}$$

$$\implies \alpha^k = \frac{(r^k)^T r^k}{(r^k)^T A r^k}$$

Die Berechnung von α^k besteht also (hauptsächlich) aus einer Matrix-Vektor-Multiplikation.

Jeder Schritt ist orthogonal zu seinem Vorgänger.

4.3.2 Konvergenzanalyse

Zuerst ein einfacher Fall: Sei e^k Eigenvektor von A.

Dann ist r^k parallel zum Fehler e^k, denn

$$r^k = b - Ax^k = Ax^* - Ax^k = -Ae^k = -\lambda e^k$$

mit λ Eigenwert von A zu e^k.

Dann gilt:

$$e^{k+1} = e^k + \frac{(r^k)^T r^k}{(r^k)^T A r^k} r^k = e^k + \frac{(r^k)^T r^k}{(r^k)\underbrace{A(-\lambda e^k)}_{=\lambda r^k}} \cdot (-\lambda e^k) = 0.$$

Das Verfahren konvergiert in einem Schritt.

Anschauung: x^k liegt auf einer Achse des Ellipsoids.

BILD!

Allgemeiner: e^k ist Linearkombination von Eigenvektoren.

Sei $\{v_j\}$ Orthonormalbasis von Eigenvektoren

$$e^k = \sum_{j=1}^{n} \xi_j v_j.$$

(Zur Einfachheit lassen wir das k weg.)

Wir erhalten:

$$\begin{aligned} r^T r = (-Ae)^T(-Ae) &= \left(A \sum_i \xi_i v_i\right)^T \left(A \sum_j \xi_j v_j\right) \\ &= \left(\sum_i \xi_i \lambda_i v_i\right)^T \left(\sum_j \xi_j \lambda_j v_j\right) = \sum_j \xi_j^2 \lambda_j^2. \end{aligned}$$

Ebenso:

$$r^T A r = \sum_j \xi_j^2 \lambda_j^3$$

Der nächste Fehler ist damit

$$e^{k+1} = e^k + \frac{(r^k)^T r^k}{(r^k)^T A r^k} r^k = e^k + \frac{\sum_j \xi_j^2 \lambda_j^2}{\sum_j \xi_j^2 \lambda_j^3} r^k.$$

Beachte: Falls alle λ_j gleich sind, so sind wir wieder in einem Schritt fertig, da dann $r^k = -\lambda e^k$.

Anschauung dazu: Das Funktional ist dann kugelsymmetrisch.

Der folgende Satz beschreibt den allgemeinen Fall. Den Beweis findet man bei Shewchuk [6].

Satz 4.6 ([6, Kapitel 6])**.** *Sei κ die Kondition der Matrix A. Dann gilt nach k Schritten des Gradientenverfahrens*

$$\|e^k\|_A \leq \Big(\frac{\kappa - 1}{\kappa + 1}\Big)^k \|e^0\|_A,$$

wobei $\|\cdot\|_A$ die Energienorm ist.

4.4 Das Verfahren der konjugierten Gradienten (CG)

(Ursprünglich vorgestellt von Hestenes und Stiefel im Jahr 1952 [3])

Wir halten fest:

- Das Gradientenverfahren minimiert häufig mehrfach in ähnliche Richtungen.

Besser wäre doch:

1. n orthogonale Suchrichtungen $d_1, \ldots, d_n$
2. Bei jedem Schritt könnten wir die richtige Schrittweite α^k bestimmen.

Damit hätten wir die exakte Lösung nach n Schritten.

Problem: 2) heißt gerade: e^{k+1} muss senkrecht auf der Suchrichtung d_k stehen.

Bestimme α:

$$0 = d_k^T e^{k+1} = d_k^T (e^k + \alpha^k d_k) \iff \alpha^k = -\frac{d_k^T e^k}{d_k^T d_k}.$$

Geht nicht, denn e^k ist unbekannt!

Stattdessen: *Gute Idee Nr. 1:* Wähle stattdessen A-orthogonale Suchrichtungen (auch *konjugierte* Suchrichtungen).

Bestimme α^k so, dass d_k und e^{k+1} A-orthogonal sind:

$$\begin{aligned} d_k^T A e^{k+1} &= d_k^T A(e^k + \alpha^k d_k) = 0 \\ \implies \alpha^k &= -\frac{d_k^T A e^k}{d_k^T A d_k} = \frac{d_k^T r^k}{d_k^T A d_k}. \end{aligned}$$

Dabei haben wir benutzt dass

$$r^k = b - Ax^k = A(A^{-1}b - x^k) = A(x^* - x^k) = -Ae^k.$$

Lemma 4.1. *Auch mit A-orthogonalen Suchrichtungen ist man nach n Schritten fertig.*

Beweis. Schreibe Anfangsfehler e^0 als Linearkombination der Suchrichtungen

$$e^0 = \sum_{j=1}^{n} \delta_j d_j. \tag{4.3}$$

Gesucht ist eine Formel für δ_k. Multipliziere (4.3) mit $d_k^T A$. Man erhält

$$\begin{aligned} d_k^T A e^0 &= \sum_{j=1}^{n} \delta_j d_k^T A d_j = \delta_k d_k^T A d_k \\ \implies \delta_k &= \frac{d_k^T A e^0}{d_k^T A d_k} = \frac{d_k^T A (e^0 + \sum_{i=1}^{k-1} \alpha^i d_i)}{d_k^T A d_k} \\ &= \frac{d_k^T A e^k}{d_k^T A d_k} = -\alpha^k. \end{aligned}$$

Bei jedem Schritt wird genau ein Summand aus der Fehlerdarstellung (4.3) entfernt.
$\implies$ fertig nach n Schritten, da dann $e^n = 0$. □

4.4.1 Das Gram-Schmidt-Verfahren

Wie erzeugt man A-orthogonale Richtungen?

Wir erklären jetzt, wie man n A-orthogonale Suchrichtungen konstruieren kann. Im CG-Verfahren passiert das synchron zum eigentlichen Suchen des Minimierers. Es wird *nicht* zuerst die Menge der Suchrichtungen konstruiert, und dann erst eine nach der anderen zur Suche genommen.

- Seien $u_1, \ldots, u_n$ linear unabhängige Vektoren.
- Setze:

$$d_i = u_i + \sum_{j=1}^{i-1} \beta_{ij} d_j, \qquad \beta_{ij} = -\frac{u_i^T A d_j}{d_j^T A d_j}. \tag{4.4}$$

Funktioniert, aber:

1. Man muss sich alle d_j merken $\longrightarrow \mathcal{O}(n^2)$ Speicherverbrauch
2. Benötigt $\mathcal{O}(n^3)$ Rechenoperationen. Das ist in etwa so viel wie wie das Gauß–Seidel-Verfahren, also zu viel.

4.4.2 Das Verfahren der konjugierten Gradienten

(Eigentlich ein schlechter Name: Es kommen keine konjugierten Gradienten vor.)

Gute Idee Nr. 2: Wähle $u_i = r^i$ für $i = 1, \ldots, n$.

- Warum ist das eine gute Idee?
- Geht das überhaupt?
- Bilden die r^i eine linear unabhängige Menge?

Bemerkung: Wie kann das gehen? Zum Berechnen der Residuen r^i brauchen wir die Suchrichtungen, und jetzt sollen wir umgekehrt die Residuen zur Berechnung der Suchrichtungen brauchen? Der Trick: Wir berechnen beide abwechselnd. Aus der Startiterierten x^0 folgt das erste Residuum. Damit kann die erste Suchrichtung berechnet werden. Damit berechnen wir x^1 und damit das zweite Residuum. Damit dann die nächste Richung usw.

Lemma 4.2. *$d_l^T r_i = 0$ für alle $l < i$.*

Beweis. Es gilt

$$e^i = e^0 + \sum_{j=0}^{i-1} \alpha^j d_j = \sum_{j=0}^{n} \delta_j d_j - \sum_{j=0}^{i-1} \delta_j d_j = \sum_{j=i}^{n} \delta_j d_j.$$

Multipliziere beide Seiten mit $-d_l^T A$:

$$-d_l^T A e^i = -\sum_{j=i}^{n} \delta_j d_l^T A d_j.$$

Die linke Seite ist $d_l^T r^i$.

Die rechte Seite ist 0, da die d_i A-orthogonal sind. □

Nicht nur ist r^i orthogonal zu allen Suchrichtungen d_l mit $l < i$, es ist auch senkrecht auf allen Residuen r^l mit $l < i$! Deshalb bilden die r^k eine linear unabhängige Menge; das Gram-Schmidt-Verfahren kann also angewandt werden.

Lemma 4.3. *r^i ist orthogonal zu r^l falls $l < i$.*

Beweis. Gram-Schmidt-Formel

$$d_l = r^l + \sum_{k=0}^{l-1} \beta_{lk} d_k.$$

Multipliziere von rechts mit r^i:

$$\underbrace{d_l^T r^i}_{=0} = (r^l)^T r^i + \sum_{k=0}^{l-1} \beta_{lk} \underbrace{d_k^T r^i}_{=0}.$$

Es folgt

$$(r^l)^T r^i = 0.$$

Die Residuen $r^1, \ldots, r^n$ sind linear unabhängig. □

Es passiert etwas magisches!

Lemma 4.4. *Fast alle β_{ij} verschwinden! Gram-Schmidt wird billig.*

Beweis.

- $r^{j+1} = -Ae^{j+1} = -A(e^j + \alpha^j d_j) = r^j - \alpha^j A d_j$
- Multipliziere von links mit r_i^T

$$\alpha^j (r_i^T A d_j) = \underbrace{r_i^T r_j}_{=0 \Leftarrow i \neq j} - \underbrace{r_i^T r_{j+1}}_{=0 \Leftarrow i \neq j+1}$$

  Wegen Lemma 4.3 ist der Term auf der rechten Seite fast immer Null! Genauer:

$$r_i^T A d_j = \begin{cases} \frac{1}{\alpha^i} (r^i)^T r^i & \text{falls } i = j, \\ -\frac{1}{\alpha^{i-1}} (r^i)^T r^i & \text{falls } i = j+1, \\ 0 & \text{sonst.} \end{cases}$$

- Gram-Schmidt-Koeffizienten:

$$\beta_{ij} = -\frac{(r^i)^T A d_j}{d_j^T A d_j} = \begin{cases} \frac{1}{\alpha^{i-1}} \frac{(r^i)^T r^i}{d_j^T A d_j} & \text{falls } i = j+1, \\ 0 & \text{sonst.} \end{cases}$$

  Der Fall $i = j$ tritt im Gram-Schmidt-Verfahren nicht auf! □

Formel (4.4) reduziert sich:

$$\text{Aus} \qquad d_i = u_i + \sum_{j=1}^{i-1} \beta_{ij} d_j \qquad \text{wird} \qquad d_i = u_i + \beta_{i,i-1} d_{i-1}. \tag{4.5}$$

- Fast alle β_{ij} sind Null!
- Deshalb: Einfachere Notation: Schreibe $\beta_{(i)}$ statt $\beta_{i,i-1}$.
- Es werden nicht mehr alle d_i benötigt, um die nächste Richtung auszurechnen.
- Speicheraufwand und Rechenzeit geht von $\mathcal{O}(n^2)$ nach $\mathcal{O}(n \cdot$ Anzahl der Einträge von $A)$.

Wir können die Darstellung von $\beta_{(i)}$ noch weiter vereinfachen.

- Setze zunächst $\alpha^i = \frac{d_i^T r^i}{d_i^T A d_i}$ in die Definition von $\beta_{i,i-1}$ ein. Man erhält

$$\beta_{(i)} = \frac{(r^i)^T r^i}{d_{i-1}^T r_{i-1}}$$

- Aus (4.5) folgt

$$d_{i-1}^T r_{i-1} = u_{i-1}^T r_{i-1} + \beta_{i-1,i-2} \underbrace{d_{i-2}^T r_{i-1}}_{=0} = u_{i-1}^T r_{i-1} = r_{i-1}^T r_{i-1}.$$

- Damit erhält man

$$\beta_{(i)} = \frac{(r^i)^T r^i}{r_{i-1}^T r_{i-1}}.$$

4.4.3 Das komplette Verfahren

- Berechne $d_0 = r_0 = b - Ax^0$.
- Für $k = 1, 2, 3, \ldots$

$$\begin{aligned}
\alpha^k &= \frac{r_k^T r_k}{d_k^T A d_k} \\
x^{k+1} &= x^k + \alpha^k d^k \\
r^{k+1} &= b - Ax^{k+1} = r^k - \alpha^k A d_k \\
\beta_{(k+1)} &= \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k} \\
d_{k+1} &= r_{k+1} + \beta_{(k+1)} d_k
\end{aligned}$$

Direkter Löser mit Komplexität $\mathcal{O}(n \cdot$ Anzahl der Einträge von $A)$. Einige Fakten zu CG

- Zuerst vorgeschlagen von Magnus Hestenes und Eduard Stiefel im Jahr 1952
- Toll: ein direkter Löser für dünnbesetzte lineare GS mit Komplexität $\mathcal{O}(n \cdot$ Anzahl der Einträge von $A)$.
- Funktioniert in der Praxis aber schlecht: Rundungsfehler zerstören A-Orthogonalität der Richtungen, man hat nach n Iterationen also *nicht* die Lösung
- Geriet zwischenzeitlich in Vergessenheit
- Erlebte Revival als iterative Methode

4.4.4 Interpretation als Krylov-Verfahren

CG hat weitere interessante Eigenschaften.

Die Folge der Suchrichtungen $d_0, d_1, \ldots$ definiert eine Folge von Räumen

$$\begin{aligned}
\mathcal{D}_0 &= \operatorname{span}\{d_0\} \\
\mathcal{D}_1 &= \operatorname{span}\{d_0, d_1\} \\
\mathcal{D}_2 &= \operatorname{span}\{d_0, d_1, d_2\} \\
&\vdots
\end{aligned}$$

Satz 4.7. *Das CG-Verfahren wählt x_k so aus dem Raum $e_0 + \mathcal{D}_k$, dass $\|e_k\|_A$ minimal ist.*

[Dies ist auch eine alternative Motivation des Verfahrens.]

Alternative Charakterisierung der $\mathcal{D}_k$

$$\begin{aligned}
\mathcal{D}_k &= \operatorname{span}\left\{d_0, Ad_0, A^2 d_0, \ldots, A^k d_0\right\} \\
&= \operatorname{span}\left\{r_0, Ar_0, A^2 r_0, \ldots, A^k r_0\right\}.
\end{aligned}$$

- Solche Räume heißen *Krylov-Räume.*[1]
- CG heißt deshalb auch ein „Krylov-Verfahren".
- Es gibt noch weitere Krylov-Raum-basierte Verfahren, z.B. BiCGStab, MinRes, GMRes.

4.4.5 Konvergenz des CG-Verfahren als iterativem Verfahren

Satz 4.8. *Für alle $k = 1, \dots, n$ hat der Fehler e^k die Darstellung*

$$e^k = \left(I + \sum_{j=1}^{k} \psi_j A^j \right) e^0,$$

wobei die Koeffizienten $\psi_1, \dots, \psi_k \in \mathbb{R}$ von α^i und $\beta_{(i)}$ für $i = 1, \dots, k$ abhängen.

Hauptidee:

- CG minimiert $\|e_k\|_A$
- Der Ausdruck in der Klammer ist ein Polynom in A, also
$$e^k = P_k(A) e^0$$
- Interpretation von CG:
  1. CG wählt die Koeffizienten $\alpha_i, \beta_{(i)}$
  2. CG konstruiert das Polynom $P_k(A)$

Wie wirkt $P_k(A)$ auf e^0?

- Schreibe e^0 in orthonormaler Eigenvektor-Basis
$$e^0 = \sum_{j=i}^{n} \xi_j v_j$$
- Daraus folgt
$$\begin{aligned} e^k &= \left(I + \sum_{i=1}^{k} \psi_i A^i \right) \sum_{j=1}^{n} \xi_j v_j \\ &= \sum_{j=1}^{n} \xi_j \left(I + \sum_{i=1}^{k} \psi_i A^i \right) v_j \\ &= \sum_{j=1}^{n} \xi_j \left(1 + \sum_{i=1}^{k} \psi_i \lambda_j^i \right) v_j \\ &= \sum_{j=1}^{n} \xi_j P_k(\lambda_j) v_j \end{aligned}$$

[1] Nach Alexei Nikolajew Krylow, 1863-1945

- Multiplikation von links mit A

$$Ae_k = \sum_{j=1}^{n} \xi_j P_k(\lambda_j)\lambda_j v_j$$

$$\|e_k\|_A^2 = e_k^T(Ae_k) = \left(\sum_{i=1}^{n} \xi_i P_k(\lambda_i) v_i\right)\left(\sum_{j=1}^{n} \xi_j P_k(\lambda_j)\lambda_j v_j\right) = \sum_{j=1}^{n} \xi_j^2 \left(P_k\left(\lambda_j\right)\right)^2 \lambda_j$$

- Das CG-Verfahren minimiert also

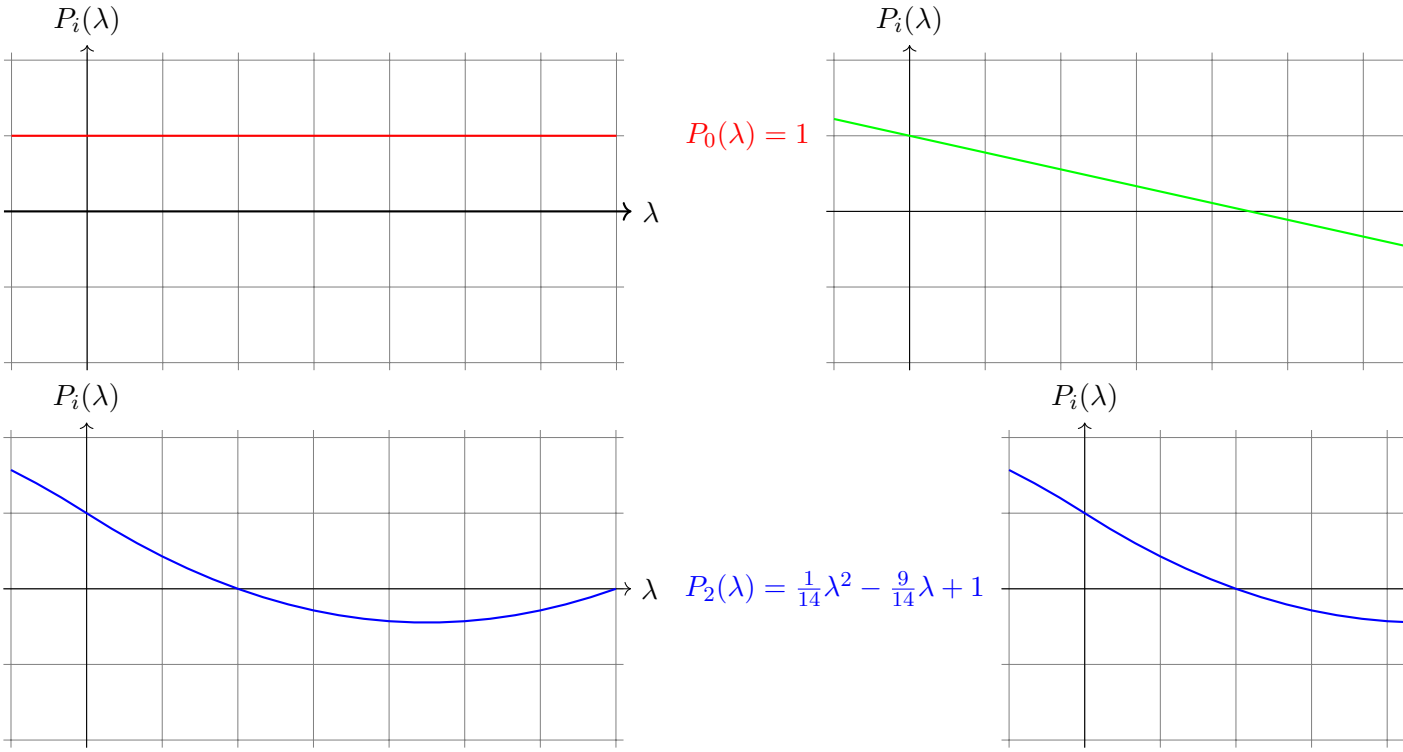
$$\begin{aligned}\|e_k\|_A^2 &= \min_{\tilde{P}_k, \tilde{P}_k(0)=1} \sum_{j=1}^{n} \xi_j^2 \tilde{P}_k(\lambda_j)^2 \lambda_j \\ &\leq \min_{\tilde{P}_k, \tilde{P}_k(0)=1} \max_{\lambda \in \sigma(A)} (\tilde{P}_k(\lambda))^2 \underbrace{\sum_{j=1}^{n} \xi_j^2 \lambda_j}_{\|e^0\|_A^2}\end{aligned}$$

  Dabei ist $\sigma(A)$ das *Spektrum* von A, d.h. die Menge aller Eigenwerte.

Die Aufgabe lautet also:

- Finde Polynom P_k mit $P_k(0) = 1$, dessen Werte für Eigenwerte λ möglichst klein sind.

Beispiel. $A \in \mathbb{R}^{2\times 2}$ mit $\sigma(A) = \{2, 7\}$, also $\lambda_1 = 2, \lambda_2 = 7$.



Male die Eigenwerte als kleine schwarze Kügelchen ein.

Schnelle Konvergenz, wenn

- es ein Polynom niedrigen Grades gibt, dass bei allen Eigenwerten von A niedrige Werte annimmt.
- Eigenwerte von A in Haufen auftreten
- viele Eigenwerte mehrfach auftreten

Schlimmst-möglicher Fall:

- Eigenwerte sind gleichverteilt in $[\lambda_{\min}, \lambda_{\max}]$.
- Wenig doppelte Eigenwerte.

Allgemein ist zuwenig über die Eigenwerte von A bekannt.

Ansatz: Anstelle das Maximum von P_k über alle Eigenwerte von A zu minimieren, minimieren wir das Maximum von P_k auf ganz $[\lambda_{\min}, \lambda_{\max}]$.

$$\|e^k\|_A^2 \leq \min_{\tilde{P}_k, \tilde{P}_k(0)=1} \max_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} (\tilde{P}_k(\lambda))^2 \|e^0\|_A^2$$

Definition. *Das Tschebyshew-Polynom vom Grad $i \in \mathbb{N}$ ist*

$$T_i(s) = \frac{1}{2}\Big[(s + \sqrt{s^2+1})^i + (s - \sqrt{s^2-1})^i\Big].$$

Satz 4.9. *Es gilt*

$$|T_i(s)| \leq 1 \qquad \text{für alle } s \in [-1, 1],$$

und T_i ist „maximal außerhalb von $[-1, 1]$“ unter allen Polynomen mit dieser Eigenschaft.

Umskalieren:

Lemma 4.5. *Das Polynom*

$$\tilde{T}_i(\lambda) = \frac{T_i\left(\frac{\lambda_{\max}+\lambda_{\min}-2\lambda}{\lambda_{\max}-\lambda_{\min}}\right)}{T_i\left(\frac{\lambda_{\max}+\lambda_{\min}}{\lambda_{\max}-\lambda_{\min}}\right)}$$

oszilliert auf $[\lambda_{\min}, \lambda_{\max}]$ zwischen

$$\pm T_i\left(\frac{\lambda_{\max}+\lambda_{\min}}{\lambda_{\max}-\lambda_{\min}}\right)^{-1},$$

und erfüllt $\tilde{T}_i(0) = 1$.

Damit können wir den Fehler abschätzen:

$$\begin{aligned}\|e^k\|_A &\leq \min_{\tilde{P}_k, \tilde{P}_k(0)=1} \max_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} |P_k(\lambda)| \cdot \|e^0\|_A \\ &\leq \max_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} |\tilde{T}_k(\lambda)| \cdot \|e^0\|_A \\ &\leq T_i \left(\frac{\lambda_{\max} + \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}} \right)^{-1} \cdot \|e^0\|_A\end{aligned}$$

Da A s.p.d. ist gilt $\kappa = \kappa(A) = \frac{|\lambda_{\max}|}{|\lambda_{\min}|} = \frac{\lambda_{\max}}{\lambda_{\min}}$, und deshalb

$$\begin{aligned}\|e^k\|_A &= T_i \left(\frac{\kappa + 1}{\kappa - 1} \right)^{-1} \|e^0\|_A \\ &= 2 \left[\left(\frac{\sqrt{\kappa} + 1}{\sqrt{\kappa} - 1} \right)^k + \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \right]^{-1} \cdot \|e^0\|_A\end{aligned}$$

Der zweite Summand geht gegen 0 für $k \to \infty$. Deshalb

$$\|e^k\|_A \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \cdot \|e^0\|_A$$

- Vergleiche mit Gradientenverfahren. Dort:

$$\|e^k\|_A \leq \left(\frac{\kappa - 1}{\kappa + 1} \right)^k \|e^0\|_A$$

- Das ist langsamer als für das CG-Verfahren.
- Diese Abschätzung für CG ist aber sehr schwach. In vielen Fällen konvergiert das Verfahren deutlich besser!

4.5 Vorkonditionierung

Die Konvergenzrate von CG (und diversen anderen Verfahren) hängt von der Kondition von A ab.

4.5.1 Idee der Vorkonditionierung

Wähle eine Matrix W, die A „gut approximiert“, und betrachte das Gleichungssystem

$$W^{-1}Ax = W^{-1}b.$$

- Gleiche Lösung wie $Ax = b$.
- Bei geschickt gewähltem W ist $\kappa\left(W^{-1}A\right) \ll \kappa(A)$

Löse $W^{-1}Ax = W^{-1}b$ mit dem CG-Verfahren.

Problem: Das CG-Verfahren kann nur verwendet werden, wenn $W^{-1}A$ s.p.d. ist, aber aus W und A s.p.d. folgt *nicht*, dass $W^{-1}A$ s.p.d. ist.

Trick: Wähle W s.p.d.
Dann existiert die Cholesky-Zerlegung

$$W = L_1 L_1^T = LDL^T$$

mit

- D Diagonalmatrix
- L untere Dreiecksmatrix mit Einsen auf der Diagonalen
- $L_1 = LD^{\frac{1}{2}}$ untere Dreiecksmatrix

Statt $Ax = b$ betrachte

$$\underbrace{L_1^{-1}AL_1^{-T}}_{=:\tilde{A}} \underbrace{L_1^T x}_{=:\tilde{x}} = \underbrace{L_1^{-1}b}_{=:\tilde{b}}$$

also

$$\tilde{A}\tilde{x} = \tilde{b}.$$

Die neue Matrix $\tilde{A}$ ist s.p.d., es gilt sogar:

Satz 4.10. *$W^{-1}A$ und $\tilde{A} = L_1^{-1}AL_1^{-T}$ haben die gleichen Eigenwerte.*

Beweis. Sei v Eigenvektor von $W^{-1}A$, dann ist $L_1^T v$ Eigenvektor von $\tilde{A}$ zum selben Eigenwert. □

Da $\tilde{A}$ s.p.d. ist können wir CG verwendet werden:

$$\tilde{d}_0 = \tilde{r}^0 = \tilde{b} - \tilde{A}\tilde{x}^0 = L_1^{-1}b - L_1^{-1}AL_1^{-T}\tilde{x}^0 = L_1^{-1}(b - Ax^0)$$

Für $k = 1, 2, 3, \ldots$

$$\begin{aligned}
\alpha^k &= \frac{\tilde{r}_k^T \tilde{r}_k}{\tilde{d}_k L_1^{-1}AL_1^{-T}\tilde{d}_k} \\
\tilde{x}^{k+1} &= \tilde{x}^k + \alpha^k \tilde{d}_k \\
\tilde{r}_{k+1} &= \tilde{r}_k - \alpha^k L_1^{-1}AL_1^{-T}\tilde{d}_k \\
\beta_{(k+1)} &= \frac{\tilde{r}_{k+1}^T \tilde{r}_{k+1}}{\tilde{r}_k^T \tilde{r}_k} \\
\tilde{d}_{k+1} &= \tilde{r}_{k+1} + \beta_{(k+1)}\tilde{d}_k
\end{aligned}$$

Zu teuer: L_1 muss bekannt sein!

Stattdessen: Setze $d_k = L_1^{-T}\tilde{d}_k$.

Das umgeformte Verfahren ist:

$$d_0 = L_1^{-T}\tilde{d}_0 = L_1^{-T}L_1^{-1}(b - Ax^0) = W^{-1}r^0$$

Für $k = 1, 2, 3, \ldots$

$$\begin{aligned}
\alpha^k &= \frac{r_k^T W^{-1} r_k}{d_k^T A d_k} \\
x^{k+1} &= x^k + \alpha^k d^k \\
r_{k+1} &= r_k - \alpha^k A d_k \\
\beta_{(k+1)} &= \frac{r_{k+1}^T W^{-1} r_{k+1}}{r_k^T W^{-1} r_k} \\
d_{k+1} &= W^{-1} r_{k+1} + \beta_{(k+1)} d_k
\end{aligned}$$

Mit einem Wort: Wir nehmen ersetzen im normalen CG-Verfahren einfach überall r durch $W^{-1}r$.

Dieses Verfahren heißt *Vorkonditioniertes CG-Verfahren*. Es funktioniert gut, wenn

1. $\kappa\left(W^{-1}A\right)$ klein ist
2. Die Lösung von $Wz = r^k$ billig berechnet werden kann.

4.5.2 Unvollständige Cholesky-Zerlegung (ICH,ILU,...)

Für die Wahlt von W sind extrem viele verschiedene Möglichkeiten vorgeschlagen worden.

Wir zeigen zwei wichtige Ansätze.

Idee: Die beste Kondition bekäme man natürlich für $W = A$.
Zur Lösung von $Wz = Az = r^k$ Cholesky-Zerlegung $A = LDL^T$ berechnen
(viele GS mit fester Matrix).

- Selbst für dünnbesetzte A ist L aber vollbesetzt
  $\Longrightarrow$ Lösen mit Cholesky-Zerlegung ist zu teuer und braucht zu viel Speicher.

Definition. *Sei $A \in \mathbb{R}^{n\times n}$ symmetrisch und positiv definit. Eine unvollständige Cholesky-Zerlegung von A ist $A \approx \tilde{L}\tilde{D}\tilde{L}^T$, wobei*

- *$\tilde{L}$: normierte untere Dreiecksmatrix*
- *$\tilde{D}$: Diagonalmatrix*
- *$\tilde{L}_{ij} = 0$ falls $A_{ij} = 0$*

Vorkonditionierer: $W = \tilde{L}\tilde{D}\tilde{L}^T$

- häufig: $\kappa\left(W^{-1}A\right) \ll \kappa(A)$
- $\tilde{L}$ ist dünnbesetzt: $\Longrightarrow$ $\tilde{L}\tilde{D}\tilde{L}^T z^k = r^k$ ist billig zu lösen.

Konstruktion der unvollständigen Cholesky-Zerlegung

Statt $\tilde{L}\tilde{D}\tilde{L}^T$ berechnen wir $\tilde{L}\tilde{R}$.

Dann ist nämlich $\tilde{R} = \tilde{D}\tilde{L}^T$ mit $\tilde{D} = \operatorname{diag}\tilde{R}$.

Sei $A = LR$ mit L, R Dreiecksmatrizen, L normiert.
Dann ist

$$A_{ik} = \sum_{j=1}^{n} L_{ij}R_{jk} = \sum_{j=1}^{i} L_{ij}R_{jk} = \underbrace{L_{ii}}_{=1} R_{ik} + \sum_{j=1}^{i-1} L_{ij}R_{jk}$$

Damit kann man den Eintrag R_{ik} berechnen:

$$R_{ik} = A_{ik} - \sum_{j=1}^{i-1} L_{ij}R_{jk} \qquad \text{da } L_{ii} = 1, \quad 1 \le i < k \le n$$

Ähnlich:

$$L_{ik} = R_{ii}^{-1}\Big(A_{ik} - \sum_{j=1}^{k-1} L_{ij}R_{jk}\Big) \qquad 1 \le k < i \le n$$

Mit diesen Formeln kann man eine LR-Zerlegung berechnen.

input : $A \in \mathbb{R}^{n\times n}$
Setze $L = I \in \mathbb{R}^{n\times n}, R = 0 \in \mathbb{R}^{n\times n}$
for $i = 1, 2, \ldots, n$ **do**
  for $k = 1, \ldots, i-1$ **do**
    $L_{ik} = R_{kk}^{-1}\Big(A_{ik} - \sum_{j=1}^{k-1} L_{ij}R_{jk}\Big)$
  end
  for $k = i, \ldots, n$ **do**
    $R_{ik} = A_{ik} - \sum_{j=1}^{i-1} L_{ij}R_{jk}$
  end
end

Um stattdessen eine *unvollständige* LR-Zerlegung zu berechnen lassen wir einfach alle Einträge i, j aus, für die $A_{ij} = 0$ gilt.

input : $A \in \mathbb{R}^{n\times n}$
Setze $\tilde{L} = I \in \mathbb{R}^{n\times n}, \tilde{R} = 0 \in \mathbb{R}^{n\times n}$
for $i = 1, 2, \ldots, n$ **do**
  foreach $k = 1, \ldots, i-1$ *with* $A_{ik} \neq 0$ **do**
    $\tilde{L}_{ik} = \tilde{R}_{kk}^{-1}\Big(A_{ik} - \sum_{j=1}^{k-1} \tilde{L}_{ij}\tilde{R}_{jk}\Big)$ Summe nur über das Muster
  end
  foreach $k = i, \ldots, n$ *with* $A_{ik} \neq 0$ **do**
    $\tilde{R}_{ik} = A_{ik} - \sum_{j=1}^{i-1} \tilde{L}_{ij}\tilde{R}_{jk}$ Summe nur über das Muster
  end
end

Beispiel. Poisson-Problem. CG vs. CG mit ILR-Vorkonditionierer

h	$\frac{1}{40}$	$\frac{1}{80}$	$\frac{1}{160}$	$\frac{1}{320}$
CG	65	130	262	525
ILR-CG	20	40	79	157

Tabelle: Anzahl der Iterationen um das Residuum um den Faktor 1000 zu reduzieren.

Es sind viele Varianten möglich!

4.5.3 Lineare Verfahren als Vorkonditionierer

Erinnerung: Lineares Verfahren

$$x^{k+1} = x^k + C(b - Ax^k)$$

Sei

1. A symmetrisch und positiv definit
2. C so, dass $\rho(I - CA) < 1$, d.h. das Verfahren konvergiert.

Wir hatten C als Approximation von A^{-1} interpretiert.

Idee: Wähle $W^{-1} = C$ als Vorkonditionierer für CG.

Praktische Umsetzung: Für CG müssen Ausdrücke der Form $z = W^{-1}r^k = Cr^k$ berechnet werden.

Problem: C ist i. A. nicht explizit gegeben.

Lösung: Betrachte das lineare Gleichungssystem $Az = r^k$

- Setze $z^0 = 0 \in \mathbb{R}^n$

- Ein Schritt des linearen Verfahrens:
$$z^1 = z^0 + C(r^k - Az^0) = Cr^k$$

Viele lineare Verfahren können als Vorkonditionierer verwendet werden!

- Z. B.: Der Jacobi-Vorkonditionierer: $C = \operatorname{diag}(A)^{-1}$
- Viele lineare Verfahren werden überhaupt nur betrachtet, um als Vorkonditionierer zu dienen (z. B. Mehrgitterverfahren, Gebietszerlegungsverfahren).

Ein Problem noch: Wie steht es z. B. mit Gauß–Seidel
$$C = (D - L)^{-1} \qquad ?$$

- Funktioniert nicht, denn $W = C^{-1} = D - L$ ist nicht symmetrisch.

Stattdessen: Symmetrischer Gauß–Seidel

- Abwechselnd
  - Vorwärtsiteration:
$$x^{k+\frac{1}{2}} = x^k + (D - L)^{-1}(b - Ax^k)$$
  - Rückwärtsiteration:
$$x^{k+1} = x^{k+\frac{1}{2}} + (D - R)^{-1}(b - Ax^{k+\frac{1}{2}})$$
- Einsetzen:
$$x^{k+1} = x^k + \underbrace{\left[(D - L)^{-1} + (D - R)^{-1} - (D - R)^{-1}A(D - L)^{-1}\right]}_{=C}(b - Ax^k)$$
  Dieses C will man sicher nicht explizit ausrechnen.
- Aber: $C^{-1} = W$ ist s.p.d.!

Auch im linearen Verfahren selbst kann man C als Vorkonditionierer interpretieren.

- Lineares Gleichungssystem $Ax = b$
- Richardson-Verfahren
$$x^{k+1} = x^k + (b - Ax^k)$$
- Vorkonditionierung: Wähle ein W als Approximation von A. Betrachte
$$W^{-1}Ax = W^{-1}b$$
  Statt W^{-1} schreibe C:
$$CAx = Cb$$
- Richardson-Iteration dafür:
$$x^{k+1} = x^k + (Cb - CAx^k) = x^k + C(b - Ax^k).$$

5 Direkte Lösungsverfahren für dünnbesetzte Gleichungssysteme

Problem: Sei $A \in R^{n \times n}, b \in \mathbb{R}^n$.
Finde $x \in \mathbb{R}^n$, sodass $Ax = b$.
Dabei ist A *sehr groß*, dafür aber dünnbesetzt.

Bisher: Iterative Verfahren. Funktionieren, im Prinzip, aber

- konvergieren nicht für jede invertiere Matrix A,
- Konvergenzgeschwindigkeit eventuell gering; hängt von der Kondition ab,
- Abbruchkriterien nötig, algebraischer Fehler

Alternative: direkte Löser:

- Bestimmen die exakte Lösung x nach endlich vielen Schritten (exakte Arithmetik vorausgesetzt)
- Beispiel: Gauß-Elimination; funktioniert, nützt aber die Dünnbesetztheit von A nicht aus. Deshalb:
  - Hohe Zeitkomplexität ($\mathcal{O}(n^3)$ Schritte)
  - Großer Speicheraufwand

Das Beste beider Welten: Direkte Löser für dünn besetzte Gleichungssysteme.

- Varianten von Gauß- und Cholesky-Elimination
- teilweise sehr kompliziert
- Zeit- und Speicheraufwand deutlich besser als bei Gauß-Verfahren
- Werden in der Praxis sehr häufig verwendet.
- Geeignet für Systeme bis ca. 10^5 Unbekannte, danach zu großer Speicheraufwand

5.1 Die Multifrontale Methode

Direktes Lösungsverfahren für dünnbesetzte Matrizen.

- Verfahren nach Duff und Reid [2] (1983)
- Implementiert z.B. in Matlab, Octave, UMFPack
- Wir behandeln nur den Fall das A symmetrisch und positiv definit ist.
- Unsere Darstellung folgt Liu: "The Multifrontal Method for Sparse Matrix Solution: Theory and Practice", SIAM Review, 1992 [4]

5.1.1 Cholesky-Zerlegung

Die multifrontale Methode basiert auf der Cholesky-Zerlegung.

Sei A s.p.d. und dünnbesetzt.

Ziel: Berechne die Cholesky-Zerlegung

$$A = LL^T, \qquad L \text{ untere Dreiecksmatrix}, \qquad L_{ii} > 0 \quad \forall i = 1, \dots, n.$$

Wir wiederholen kurz die Cholesky-Zerlegung für vollbesetzte Matrizen.

Schreibe dafür A in Blockform

$$A = \begin{pmatrix} B & V^T \\ V & C \end{pmatrix}, \qquad B \in \mathbb{R}^{(j-1)\times(j-1)}.$$

Dann existiert die Zerlegung

$$A = \begin{pmatrix} L_B & 0 \\ VL_B^{-T} & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & C - VB^{-1}V^T \end{pmatrix} \begin{pmatrix} L_B^T & L_B^{-1}V^T \\ 0 & I \end{pmatrix}.$$

Dabei ist L_B der Cholesky-Faktor von B.

- Dieser existiert, da B ja s.p.d. ist.

Lemma 5.1. *Die $n \times (j-1)$-Matrix $\begin{pmatrix} L_B \\ VL_B^{-T} \end{pmatrix}$ besteht aus den ersten $j-1$ Spalten von L.*

Die Zerlegung lässt sich rekursiv fortsetzen, denn $C - VB^{-1}V^T$ ist ebenfalls s.p.d.

Beweis der Definitheit. Sei $u \in \mathbb{R}^{n-j+1}$, $u \neq 0$. Dann ist

$$u^T(C - VB^{-1}V^T)u = \begin{pmatrix} -B^{-1}V^Tu \\ u \end{pmatrix}^T \begin{pmatrix} B & V^T \\ V & C \end{pmatrix} \begin{pmatrix} -B^{-1}V^Tu \\ u \end{pmatrix} > 0. \qquad \square$$

Wenn man $j = 2$ wählt erhält man

$$A = \begin{pmatrix} \sqrt{B} & 0 \\ \frac{V}{\sqrt{B}} & I \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & C - \frac{VV^T}{B} \end{pmatrix} \begin{pmatrix} \sqrt{B} & \frac{V^T}{\sqrt{B}} \\ 0 & I \end{pmatrix}$$

- Hier ist $B \in \mathbb{R}^{1\times 1}$ eine Zahl. Wurzel und Divisions sind also wohldefiniert.
- Da $\sqrt{B}$ und $V/\sqrt{B}$ billig auszurechnen sind ist also *eine Spalte* von L billig auszurechnen.

Wegen Lemma 5.1 gilt

$$\begin{aligned} -VB^{-1}V^T &= -(VL_B^{-T})(L_B^{-1}V^T) \\ &= -\sum_{k=1}^{j-1} \begin{pmatrix} L_{jk} \\ \vdots \\ L_{nk} \end{pmatrix} \begin{pmatrix} L_{jk} & \dots & L_{nk} \end{pmatrix}. \end{aligned}$$

Daraus konstruieren wir einen Algorithmus:

for *alle Spalten* $j = 1, \dots, n$ **do**

Definiere

$$F_j := \begin{pmatrix} A_{jj} & \dots & A_{jn} \\ \vdots & & \vdots \\ A_{nj} & \dots & A_{nn} \end{pmatrix} - \sum_{k=1}^{j-1} \begin{pmatrix} L_{jk} \\ \vdots \\ L_{nk} \end{pmatrix} \begin{pmatrix} L_{jk} & \dots & L_{nk} \end{pmatrix}$$

Faktorisiere

$$F_j = \begin{pmatrix} L_{jj} & 0 & \dots & 0 \\ \vdots & & I & \\ L_{nj} & & & \end{pmatrix} \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & \widetilde{U}_j & \\ 0 & & & \end{pmatrix} \begin{pmatrix} L_{jj} & \dots & L_{nj} \\ 0 & & \\ \vdots & I & \\ 0 & & \end{pmatrix}$$

end

- In jedem Schritt wird eine weitere Spalte von L bestimmt.
- Die Matrix $\widetilde{U}_j$ ergibt sich aus der Faktorisierung. Sie wird aber bis auf weiteres nicht weiter verwendet.

Wir wollen diesen Algorithmus jetzt so modifizieren, dass er die Dünnbesetztheit von A ausnutzt.

5.1.2 Die Struktur von L

Zunächst machen wir uns klar, dass man die Struktur von L bestimmen kann, ohne L selbst bestimmen zu müssen.

Definition. *Die Struktur $\mathcal{S}_X$ einer Matrix $X \in \mathbb{R}^{n\times n}$ ist*

$$\mathcal{S}_X := \{(i,j)\colon X_{ij} \neq 0\}.$$

- In der tatsächlichen Umsetzung von Matrixalgorithmen im Computer wird man statt der ganzen Matrix A nur deren Struktur $\mathcal{S}_A$ speichern, sowie die dazugehörigen Einträge.
- Wie genau, dazu gibt es wieder verschiedene Varianten!

Wie sieht die Struktur des Cholesky-Faktors aus?

Satz 5.1. *Sei $A \in \mathbb{R}^{n\times n}$ s.p.d. und L der Cholesky-Faktor.*

1. *Falls $i \leq j$ und $(j,i) \in \mathcal{S}_A$, so ist auch $(j,i) \in \mathcal{S}_L$.*
2. *Falls $i < j < k$ und $(j,i) \in \mathcal{S}_L$ und $(k,i) \in \mathcal{S}_L$, dann ist auch $(k,j) \in \mathcal{S}_L$.*

Damit sind alle Einträge von $\mathcal{S}_L$ beschrieben.

Beispiel.

$$A = \begin{pmatrix} \bullet & & & \bullet & & & \bullet \\ & \bullet & & & & & \bullet \\ & & \bullet & & \bullet & \bullet & \\ \bullet & & & \bullet & & & \\ & & \bullet & & \bullet & & \bullet \\ & & \bullet & & & \bullet & \\ \bullet & \bullet & & & \bullet & & \bullet \end{pmatrix}, \quad L = \begin{pmatrix} \bullet & & & & & & \\ & \bullet & & & & & \\ & & \bullet & & & & \\ \bullet & & & \bullet & & & \\ & & \bullet & & \bullet & & \\ & & \bullet & & \circ & \bullet & \\ \bullet & \bullet & & \circ & \bullet & \circ & \bullet \end{pmatrix}$$

Beweis. Betrachte wieder Darstellung

$$A = \begin{pmatrix} B & V^T \\ V & C \end{pmatrix}, \qquad B \in \mathbb{R}^{1\times 1} = \mathbb{R}.$$

1.
   - Da $L_{11} = \sqrt{B}$ und $B > 0$ enthält $\mathcal{S}_L$ alle Diagonaleinträge.
   - Da die erste Spalte von L unterhalb der Diagonalen gleich $\frac{V}{\sqrt{B}}$ ist enthält $\mathcal{S}_L$ alle Einträge von $\mathcal{S}_A$ unterhalb der Diagonalen.
2.
   - Der Algorithmus berechnet die Cholesky-Zerlegung spaltenweise.
   - Im i-ten Schritt wird die i-te Spalte von L berechnet. Danach ändert sie sich nicht mehr.

$$\left(\begin{array}{cc} & 0 \\ \hline L_{ii} & \\ \hline \tilde{l}_i & \tilde{A}_i - \tilde{l}_i\tilde{l}_i^T \end{array}\right)$$

- Angenommen, $(j,i) \in \mathcal{S}_L$ und $(k,i) \in \mathcal{S}_L$
- Dann sind auch $(\tilde{l}_i)_j$ und $(\tilde{l}_i)_k$ Struktureinträge.
- Da $(\tilde{l}_i\tilde{l}_i^T)_{jk} = (\tilde{l}_i)_j \cdot (\tilde{l}_i)_k$ ist dann auch $(\tilde{l}_i\tilde{l}_i^T)_{jk}$ Struktureintrag.
- Also ist auch $(\tilde{A}_i - \tilde{l}_i\tilde{l}_i^T)_{jk}$ Struktureintrag.
- Mit 1) folgt die Aussage rekursiv. □

Was passiert wenn $(\tilde{A}_i - \tilde{l}_i\tilde{l}_i^T)_{jk}$ zufällig gerade Null ergibt?

- Entgegen der Definition von S_L bezeichnet man (j,k) dann dennoch als Struktureintrag von L.
- Vorteil: Man kann dann die Struktur von L nur anhand der Struktur von A bestimmen.
- Vermutung: Dieser Fall kommt ohnehin selten vor.

5.1.3 Ausnutzen der Dünnbesetztheit, Teil 1

Wir wollen den Algorithmus jetzt so modifizieren, dass er die Dünnbesetztheit von A ausnutzt.

Idee: Die j-te Spalte von L hängt nur von der ersten Zeile und Spalte von F_j ab.
Modifizierter Algorithmus:

1 **for** *alle Spalten* $j = 1, \dots, n$ **do**

2 Definiere

$$F_j := \begin{pmatrix} A_{jj} & \dots & \dots & A_{jn} \\ \vdots & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ A_{nj} & 0 & \dots & 0 \end{pmatrix} - \sum_{k=1}^{j-1} \begin{pmatrix} L_{jk} \\ \vdots \\ L_{nk} \end{pmatrix} \begin{pmatrix} L_{jk} & \dots & L_{nk} \end{pmatrix}$$

3 Faktorisiere

$$F_j = \begin{pmatrix} L_{jj} & 0 & \dots & 0 \\ \vdots & & I & \\ L_{nj} & & & \end{pmatrix} \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & \widehat{U}_j & \\ 0 & & & \end{pmatrix} \begin{pmatrix} L_{jj} & \dots & L_{nj} \\ 0 & & \\ \vdots & I & \\ 0 & & \end{pmatrix}$$

4 **end**

Aus $\widetilde{U}_j$ ist eine andere Matrix $\widetilde{U}_j$ geworden.

- Macht nichts: $\widetilde{U}_j$ wurde ohnehin nicht weiter verwendet.

Idee: Auch vom zweiten Summanden

$$-\sum_{k=1}^{j-1} \begin{pmatrix} L_{jk} \\ \vdots \\ L_{nk} \end{pmatrix} \begin{pmatrix} L_{jk} & \dots & L_{nk} \end{pmatrix}$$

ist nur die erste Zeile und Spalte relevant.

Es reicht also, über die Spalten $k = 1, \dots, j-1$ zu addieren, für die $L_{jk} \neq 0$ ist.

Neuer Algorithmus:

for *alle Spalten* $j = 1, \dots, n$ **do**

Definiere

$$F_j := \begin{pmatrix} A_{jj} & \dots & \dots & A_{jn} \\ \vdots & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ A_{nj} & 0 & \dots & 0 \end{pmatrix} - \sum_{\substack{k=1 \\ L_{jk} \neq 0}}^{j-1} \begin{pmatrix} L_{jk} \\ \vdots \\ L_{nk} \end{pmatrix} \begin{pmatrix} L_{jk} & \dots & L_{nk} \end{pmatrix}$$

Faktorisiere

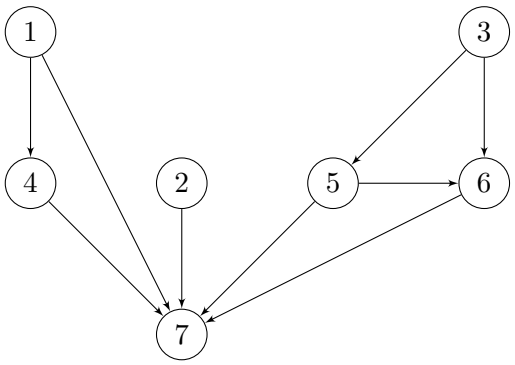
$$F_j = \begin{pmatrix} L_{jj} & 0 & \dots & 0 \\ \vdots & & I & \\ L_{nj} & & & \end{pmatrix} \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & \overline{U}_j & \\ 0 & & & \end{pmatrix} \begin{pmatrix} L_{jj} & \dots & L_{nj} \\ 0 & & \\ \vdots & I & \\ 0 & & \end{pmatrix}$$

end

5.1.4 Graphen- und Baumdarstellung

Die Struktur der Dreiecksmatrix L lässt sich als gerichteter Graph $G = (V, E)$ darstellen mit $V = \{1, \dots, n\}$ und $E = \{(i, j) \colon (j, i) \in \mathcal{S}_L, i > j\}$.

Für die Beispielmatrix von oben erhält man:



Wir können also die letzte Version des Algorithmus umschreiben:

for *alle Spalten* $j = 1, \dots, n$ **do**
  Definiere

$$F_j := \big(\dots\big) - \sum_{\substack{k=1 \\ \exists \text{ Kante von } k \text{ nach } j}}^{j-1} \begin{pmatrix} L_{jk} \\ \vdots \\ L_{nk} \end{pmatrix} \begin{pmatrix} L_{jk} & \dots & L_{nk} \end{pmatrix}$$

  Faktorisiere [...]
end

Der Eliminierungsbaum

Andererseits ist diese Darstellung teilweise redundant: zwischen zwei Knoten kann es mehr als einen Weg geben.

Denn: Satz 5.1 sagt, dass falls es die Kanten (i, j) und (i, k) gibt mit $j < k$, dann gibt es auch (j, k).

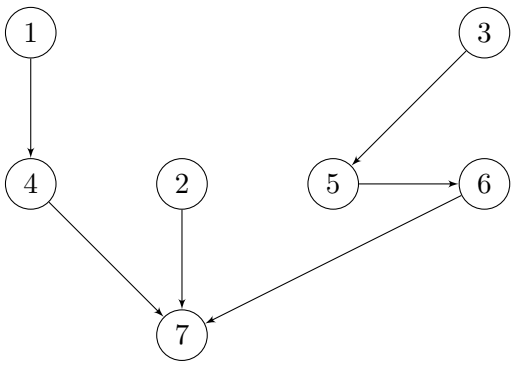
Entferne Kanten solange, bis jeder Knoten nur noch höchstens einen Nachfolger hat. Und zwar auf eine bestimmte Art:

Definition. *Der Elimiminierungsbaum $T(A)$ von A ist der Baum mit den Knoten $V = \{1, \dots, n\}$ und Kanten*

$$E = \{(j, p) \colon \text{falls } p = \min\{i > j \colon (i, j) \in S_L\}\}.$$

Anschaulich: p ist die Zeile des ersten Eintrags von L in der j-ten Spalte (unter der Diagonalen).

Für das Beispiel von eben erhält man:



Satz 5.2. *Falls $(j, k) \in \mathcal{S}_L$, dann existiert ein Pfad $k \rightsquigarrow j$ in $T(A)$.*

Definition. *Ein Knoten i heißt Vorgänger von j, falls ein Weg $i \rightsquigarrow j$ in $T(A)$ existiert. Schreibe $T(j)$ für die Menge aller Vorgänger von j.*

Mit dieser Terminologie können wir den Algorithmus nochmals umschreiben:

for *alle Spalten* $j = 1, \dots, n$ **do**
  Definiere

$$F_j := \Big(\dots\Big) - \sum_{j \in T(j) \setminus \{j\}}^{j-1} \begin{pmatrix} L_{jk} \\ \vdots \\ L_{nk} \end{pmatrix} \begin{pmatrix} L_{jk} & \dots & L_{nk} \end{pmatrix}$$

  Faktorisiere [...]
end

Die Umkehrung von Satz 5.2 gilt nicht unbedingt. So ist im Beispiel Knoten 3 Vorgänger von Knoten 7, der Eintrag $(3, 7)$ existiert aber nicht in L. Der Wechsel zum Eliminierungsbaum führt also erstmal wieder überflüssige Rechenoperationen ein. Im endgültigen Algorithmus sind die aber wieder verschwunden (s.u.).

5.1.5 Ausnutzen der Dünnbesetztheit, Teil 2

Idee: Jeder Summand in F_j ist für sich dünnbesetzt.

Aber: Jeder Summand hat doch vermutlich eine andere Besetzungsstruktur? Die *Summe* ist doch vermutlich dann doch relativ dicht?

Nein!

Satz 5.3. *Seien i und j Knoten in $T(A)$, sodass ein Pfad $i \rightsquigarrow j$ existiert (also $i < j$). Dann sind alle Struktureinträge der i-ten Spalte von L (unterhalb der j-ten Zeile) in der Struktur der j-ten Spalte enthalten*

$$(k, i) \in \mathcal{S}_L \implies (k, j) \in \mathcal{S}_L (\textit{falls } k \geq j \textit{ und } i \rightsquigarrow j).$$

Seien also

$$j = i_0 < i_1 < \dots < i_r$$

die Zeilen der Einträge der j-ten Spalte von L.

Neuer Algorithmus:

for *alle Spalten* $j = 1, \ldots, n$ **do**

Definiere

$$F_j := \begin{pmatrix} A_{i_0 i_0} & \ldots & \ldots & A_{i_0 i_r} \\ \vdots & 0 & \ldots & 0 \\ \vdots & \vdots & & \vdots \\ A_{i_r i_0} & 0 & \ldots & 0 \end{pmatrix} - \sum_{k \in T(j) \setminus \{j\}}^{j-1} \begin{pmatrix} L_{i_0 k} \\ \vdots \\ L_{i_r k} \end{pmatrix} \begin{pmatrix} L_{i_0 k} & \ldots & L_{i_r k} \end{pmatrix}$$

Faktorisiere

$$F_j = \begin{pmatrix} L_{i_0 j} & 0 & \ldots & 0 \\ \vdots & & I & \\ L_{i_r j} & & & \end{pmatrix} \begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & & & \\ \vdots & & U_j & \\ 0 & & & \end{pmatrix} \begin{pmatrix} L_{i_0 j} & \ldots & L_{i_r j} \\ 0 & & \\ \vdots & I & \\ 0 & & \end{pmatrix}$$

end

Die Matrizen $F_j \in \mathbb{R}^{(r+1)\times(r+1)}$ und $U_j \in \mathbb{R}^{r \times R}$ sind jetzt vollbesetzt.

- F_j nennt man j-te *Frontal-Matrix*
- U_j nennt man j-te *Update-Matrix*.

5.1.6 Matrix-Superposition (Der extend-add Operator)

Wir brauchen noch ein Werkzeug zum Arbeiten mit den neuen Matrizen.

- Die aktuelle Definition von F_j sagt, dass man den kompletten Teilbaum von $T(A)$ in j abläuft und für jeden Knoten Matrizen aufstellt.
- Das kann immer noch ziemlich teuer sein.
- Hier kommt der Trick: Man kann F_j effizient aus den Update-Matrizen der *direkten* Vorgänger zusammenbauen!
- Sei $R \in \mathbb{R}^{n \times n}$ mit $r \leq n, S \in \mathbb{R}^{s \times s}$ mit $s \leq n$.
- Jede Zeile/Spalte von R und S soll zu einer Zeile/Spalte der gegebenen Matrix A gehören
- Indexmengen: $i_1 < \ldots < i_r$ für R,
  $j_1 < \ldots < j_s$ für S

1) Sei $k_1 < \ldots < k_t$ die Vereinigung der beiden Indexmengen
2) Passe R und S an die Indexmenge $k_1 < \ldots < k_t$ an, indem Nullzeilen und Nullspalten eingefügt werden.

3) Definiere $R \leftrightarrow\!\!\updownarrow\!\!\rightarrow S \in \mathbb{R}^{t\times t}$ als Summe der erweiterten Matrizen R, S.

Der Operator $\leftrightarrow\!\!\updownarrow\!\!\rightarrow$ wird in der englischsprachigen Literatur als „extend–add“ bezeichnet.

Beispiel.

$$R = \begin{pmatrix} p & q \\ u & v \end{pmatrix}, \qquad S = \begin{pmatrix} w & x \\ y & z \end{pmatrix}$$

Indexmengen $\{5, 8\}$ bzw. $\{5, 0\}$.

Dann hat $R \leftrightarrow\!\!\updownarrow\!\!\rightarrow S$ die Indexmenge $\{5, 8, 9\}$, und

$$R \leftrightarrow\!\!\updownarrow\!\!\rightarrow S = \begin{pmatrix} p & q & 0 \\ u & v & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} w & 0 & x \\ 0 & 0 & 0 \\ y & 0 & z \end{pmatrix} = \begin{pmatrix} p+w & q & x \\ u & v & 0 \\ y & 0 & z \end{pmatrix}$$

Damit kann man eine billigere Formel für F_j finden.

Satz 5.4 (Liu [4, Thm. 4.1])**.** *Seien $c_1, \ldots, c_s$ die* direkten *Vorgänger des Knotens j im Eliminierungsbaum $T(A)$ von A. Dann ist*

$$F_j = \begin{pmatrix} A_{i_0 i_0} & A_{i_0 i_1} & \ldots & A_{i_0 i_r} \\ A_{i_1 i_0} & & & \\ \vdots & & 0 & \\ A_{i_r i_0} & & & \end{pmatrix} \leftrightarrow\!\!\updownarrow\!\!\rightarrow U_{c_1} \leftrightarrow\!\!\updownarrow\!\!\rightarrow U_{c_2} \leftrightarrow\!\!\updownarrow\!\!\rightarrow \ldots \leftrightarrow\!\!\updownarrow\!\!\rightarrow U_{c_s}.$$

Zum Beweis braucht man:

Satz 5.5 (Liu, Thm. 3.3)**.** *Es gilt*

$$U_j = -\sum_{k\in T(j)} \begin{pmatrix} L_{i_1 k} \\ \vdots \\ L_{i_r k} \end{pmatrix} \begin{pmatrix} L_{i_1 k} & \ldots & L_{i_r k} \end{pmatrix}.$$

Beweis. Umformen der Zerlegung von F_j gibt

$$F_j = \begin{pmatrix} L_{i_0 j} \\ \vdots \\ L_{i_r j} \end{pmatrix} \begin{pmatrix} L_{i_0 j} & \ldots & L_{i_r j} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & U_j \end{pmatrix}$$

Aber F_j und U_j unterscheiden sich nur in der ersten Zeile und Spalte.

Deshalb

$$\begin{aligned} &[F_j \text{ ohne erste Zeile und Spalte}] \\ &= [\overline{U}_j \text{ ohne erste Zeile und Spalte}] \\ &= \left[-\sum_{k\in T(j)\setminus\{j\}} \begin{pmatrix} L_{i_0 k} \\ \vdots \\ L_{i_r k} \end{pmatrix} \begin{pmatrix} L_{i_0 k} & \ldots & L_{i_r k} \end{pmatrix} \text{ ohne erste Zeile und Spalte} \right] \\ &= -\sum_{k\in T(j)} \begin{pmatrix} L_{i_1 k} \\ \vdots \\ L_{i_r k} \end{pmatrix} \begin{pmatrix} L_{i_1 k} & \ldots & L_{i_r k} \end{pmatrix} = \begin{pmatrix} L_{i_1 j} \\ \vdots \\ L_{i_r j} \end{pmatrix} \begin{pmatrix} L_{i_1 j} & \ldots & L_{i_r j} \end{pmatrix} + U_j. \end{aligned}$$

Daraus folgt die Behauptung. □

Es fehlt noch der (kurze) Beweis von Satz 5.4.

5.1.7 Der endgültige Algorithmus

Berechne zunächst die Struktur von L.

Danach:

for *alle Spalten* $j = 1, \ldots, n$ **do**
  Seien $i_0, \ldots, i_r$ die Zeilenindizes der Einträge von L_{*j}. Nicht vergessen: $i_0 = j$
  Seien $c_1, \ldots, c_s$ die direkten Vorgänger von j im Eliminationsbaum $T(A)$ von A.
  Bilde Frontal-Matrix F_j wie in Satz 5.4

$$F_j = \begin{pmatrix} A_{i_0 i_0} & A_{i_0 i_1} & \ldots & A_{i_0 i_r} \\ A_{i_1 i_0} & & & \\ \vdots & & 0 & \\ A_{i_r i_0} & & & \end{pmatrix} \leftrightarrow U_{c_1} \leftrightarrow U_{c_2} \leftrightarrow \ldots \leftrightarrow U_{c_s}.$$

  Faktorisiere

$$F_j = \begin{pmatrix} L_{i_0 i_0} & 0 & \ldots & 0 \\ L_{i_1 i_0} & & & \\ \vdots & & I & \\ L_{i_r i_0} & & & \end{pmatrix} \begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & & & \\ \vdots & & U_j & \\ 0 & & & \end{pmatrix} \begin{pmatrix} L_{i_0 i_0} & L_{i_1 i_0} & \ldots & L_{i_r i_0} \\ 0 & & & \\ \vdots & & I & \\ 0 & & & \end{pmatrix}$$

  Man merke sich U_j, falls nötig.
end

Das Ergebnis im j-ten Schritt ist also: $L_{i_0 i_0}, \ldots, L_{i_r i_0}$ und U_j.

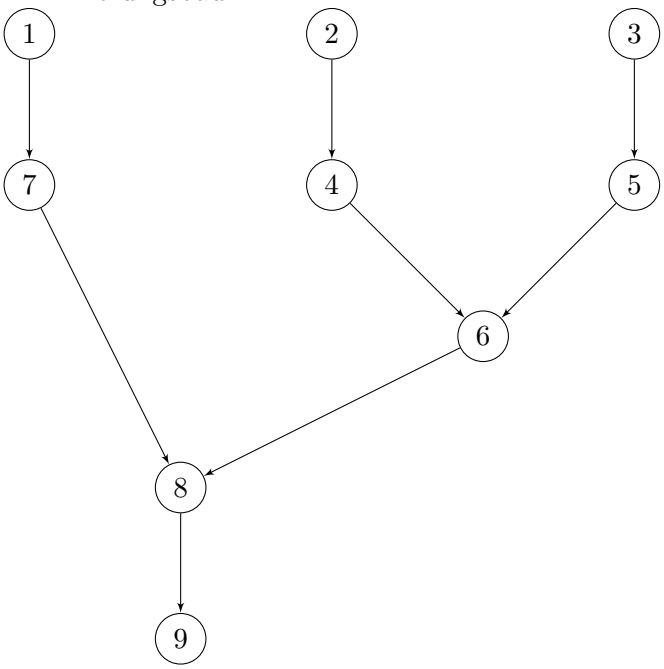
5.1.8 Umsortierungen der Matrix

- Die Matrixspalten werden von 1 bis n behandelt
- Die Update-Matrizen werden in eben dieser Reihenfolge erzeugt
- Wird eine Matrix U_j nicht für F_{j+1} gebraucht, so muss sie zwischengespeichert werden.
- Das verbraucht Speicher!

Sei beispielsweise

$$A = \begin{pmatrix} a & & & & & & \bullet & \bullet & \bullet \\ & b & & \bullet & & \bullet & & & \\ & & c & & \bullet & & & \bullet & \\ & \bullet & & d & & & & \bullet & \bullet \\ & & \bullet & & e & \bullet & & \bullet & \\ & \bullet & & & \bullet & f & & & \bullet \\ \bullet & & & & & & g & \bullet & \bullet \\ \bullet & & \bullet & \bullet & \bullet & & \bullet & h & \\ \bullet & & & \bullet & & \bullet & \bullet & & i \end{pmatrix}, \qquad L = \begin{pmatrix} a & & & & & & & & \\ & b & & & & & & & \\ & & c & & & & & & \\ & \bullet & & d & & & & & \\ & & \bullet & & e & & & & \\ & \bullet & & \circ & \bullet & f & & & \\ \bullet & & & & & & g & & \\ \bullet & & \bullet & \bullet & \bullet & \circ & \bullet & h & \\ \bullet & & & \bullet & & \bullet & \bullet & \circ & i \end{pmatrix}$$

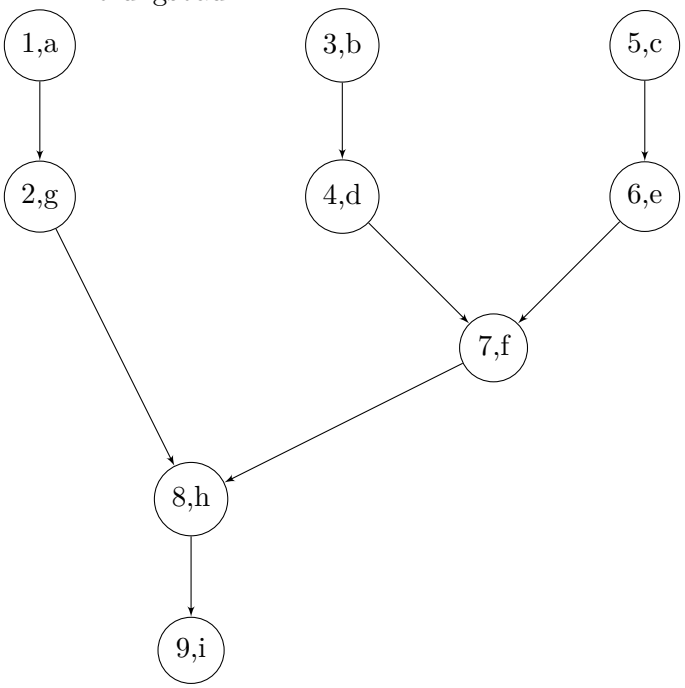
Eliminierungsbaum:



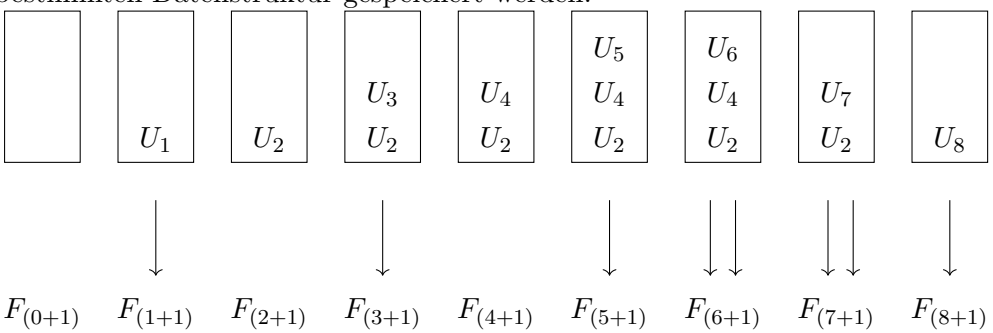
Z.B.: F_4 benutzt nur U_2, d.h. U_1 und U_3 müssen temporär zwischengespeichert werden.
Idee: Sortiere die Zeilen und Spalten von A so um, dass zu jedem Zeitpunkt möglichst wenig Matrizen U_j gespeichert werden müssen.

$$\overline{A} = \begin{pmatrix} a & \bullet & & & & & \bullet & \bullet \\ \bullet & g & & & & & \bullet & \bullet \\ & & b & \bullet & & \bullet & & \\ & & \bullet & d & & & \bullet & \bullet \\ & & & & c & \bullet & & \bullet & \\ & & & & \bullet & e & \bullet & \bullet & \\ & & \bullet & & & \bullet & f & & \bullet \\ \bullet & \bullet & & \bullet & \bullet & \bullet & & h & \\ \bullet & \bullet & & \bullet & & & \bullet & & i \end{pmatrix}, \qquad \overline{L} = \begin{pmatrix} a & & & & & & & & \\ \bullet & g & & & & & & & \\ & & b & & & & & & \\ & & \bullet & d & & & & & \\ & & & & c & & & & \\ & & & & \bullet & e & & & \\ & & \bullet & \circ & & \bullet & f & & \\ \bullet & \bullet & & \bullet & \bullet & \bullet & \circ & h & \\ \bullet & \bullet & & \bullet & & & \bullet & \circ & i \end{pmatrix}$$

Eliminierungsbaum:



Mit dieser Nummerierung können die Matrizen U_j in einem Stapel (engl.: Stack), einer bestimmten Datenstruktur gespeichert werden.



Geht das immer?

Definition. *Eine Ordnung, auf der Knotenmenge eines gerichteten Graphen heißt* topologisch*, wenn jeder Knoten vor seine Nachfolger einsortiert wird.*

Satz 5.6. *Sei $T(A)$ der Eliminationsbaum einer Matrix A. Jede topologische Ordnung von $T(A)$ erzeugt eine Umsortierung von A, die den E.-baum invariant lässt.*

Definition. *Eine topologische Ordnung eines Baumes $T(A)$ heißt* Post-Ordnung*, wenn alle Teilbäume konsekutiv durchnummeriert sind.*

Die Zahlen $1, \ldots, 9$ bilden in unserem Beispiel eine Post-Ordnung: Die Indexmengen jedes Teilbaums sind konsekutiv.

6 Numerik von gewöhnlichen Differentialgleichungen

Der Inhalt dieses Kapitels ist größtenteils dem Buch von Deuflhard und Bornemann [1] entnommen.

Gewöhnliche Differentialgleichung:

$$x_i' = f_i(t, x_1, \ldots, x_d), \quad i = 1, \ldots, d$$

wobei $(t, x) \in \mathbb{R} \times \mathbb{R}^d$ und $f_i \colon \Omega \to \mathbb{R}^d$, $\Omega \subseteq \mathbb{R} \times \mathbb{R}^d$ offen.

- Die Variable t ist häufig als Zeit interpretierbar, man spricht daher häufig von *Evolutionsproblemen*.
- x heißt *Zustandsvektor*.
- $\mathbb{R}^d$ mit $x \in \mathbb{R}^d$ heißt *Zustandsraum*.
- $\mathbb{R} \times \mathbb{R}^d$ heißt *erweiterter Zustandsraum*.

Beispiele

1. (Radioaktiver Zerfall) Finde $x : \mathbb{R} \to \mathbb{R}$ so dass

   $$x' = -kx$$

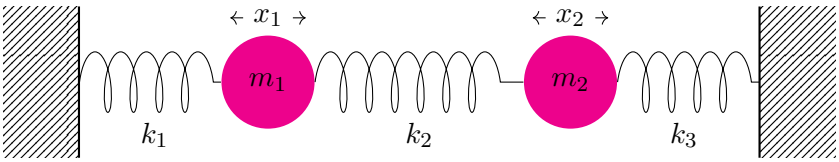
   Achtung: Traditionell verwendet man das gleiche Symbol für Zustände $x \in \mathbb{R}^d$ und Funktionen in den Zustandsraum $x : \mathbb{R} \to \mathbb{R}^d$. Nicht verwirren lassen!

2. (Zwei-Massen-Schwinger) $d = 4$

   $$\begin{aligned} x_1' &= x_3 \\ x_2' &= x_4 \\ x_3' &= \frac{1}{m_1}\big(- k_1 x_1 + k_2(x_2 - x_1)\big) \\ x_4' &= \frac{1}{m_2}\big(k_2(x_1 - x_2) - k_3 x_2\big) \end{aligned}$$

   Dürfen denn keine höheren Ableitungen vorkommen?

3. (Zwei-Massen-Schwinger physikalisch).



Massen m_1, m_2, horizontale Positionen x_1, x_2. Auf eine Masse wirken zwei Arten von Kräften:

- Trägheitskräfte: $m_i \ddot{x}_i(t)$
- Federkräfte: Für jede Feder Auslenkung × Federkonstante

Mechanisches Prinzip: Alle wirkenden Kräfte addieren sich zu Null.

$$m_1 \ddot{x}_1(t) = -k_1 x_1 + k_2(x_2 - x_1)$$
$$m_2 \ddot{x}_2(t) = k_2(x_1 - x_2) - k_3 x_2$$

Kann auf obiges System erster Ordnung reduziert werden:

- Zusätzliche Variablen x_3, x_4
- Zusätzliche Gleichungen $\dot{x}_1 = x_3$, $\dot{x}_2 = x_4$

6.1 Anfangswertprobleme

Lösungen von gewöhnlichen Differentialgleichungen sind normalerweise nicht eindeutig.

Beispiel. Für alle $c \in \mathbb{R}$ löst $x(t) = ce^{-kt}$ die Gleichung

$$x' = -kx.$$

Deshalb: *Zusatzinformation.* Schreibe Anfangsbedingung

$$x(t_0) = x_0$$

vor. $x_0 \in \mathbb{R}^d$ heißt *Anfangswert.*

Bezeichnung:

Differentialgleichung + Anfangsbedingung = Anfangswertproblem (AWP/IVP)

6.2 Existenz und Eindeutigkeit

Sei die Notation wie oben.

- Der Definitionsbereich Ω von f sei offen,
- $(t_0, x_0) \in \Omega$.

Was meinen wir genau mit „Lösung des Anfangswertproblems“?

Definition. *Sei $J \subset \mathbb{R}$ ein Intervall mit nichtleerem Inneren, und $t_0 \in J$. Eine Abbildung $x \in C^1(J, \mathbb{R}^d)$ heißt Lösung des AWPs genau dann, wenn*

$$\dot{x}(t) = f(t, x(t)) \qquad \text{für alle } t \in J,$$

und $x(t_0) = x_0$ gilt.

- Es reichen schon Funktionen auf einem „kleinen“ Intervall.
- Wir wollen „große“ Intervalle I.

Es reichen schon wenige Zusatzinformationen, um zu erreichen, dass I größtmöglich ist.

Was heißt „größtmöglich“?

Definition (Maximale Fortsetzbarkeit)**.** *Eine Lösung $x \in C^1([t_0, t_1), \mathbb{R}^d)$ heißt (in der Zukunft) fortsetzbar bis an der Rand von Ω, wenn es eine Funktion $x^* \in C^1([t_0, t_+), \mathbb{R}^d)$ mit $t_1 \le t_+ \le \infty$ gibt, sodass*

- *$x(t) = x^*(t)$ für alle $t \in [t_0, t_1)$*
- *x^* ist ebenfalls Lösung,*

und einer der drei folgenden Fälle vorliegt:

1) $t_+ = \infty$

2) $t_+ < \infty$ und $\lim_{t \uparrow t_+} |x^*(t)| = \infty$

3) $t_+ < \infty$ und $\lim_{t \uparrow t_+} \operatorname{dist}((t, x^*(t)), \partial\Omega) = 0$

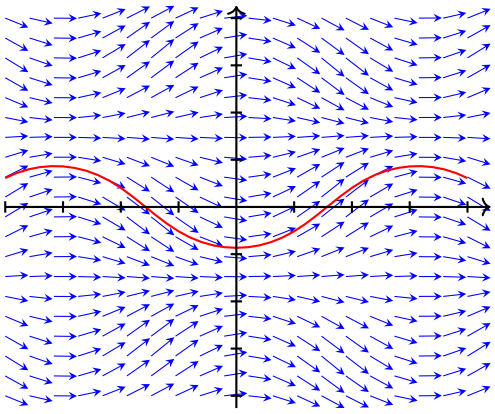
Beispiel (Fortsetzbarkeit). Betrachte das AWP $x' = -kx, x(0) = 1$. Eine Lösung davon ist

$$x \colon [0, 1] \to \mathbb{R},\ x(t) = e^{-kt}.$$

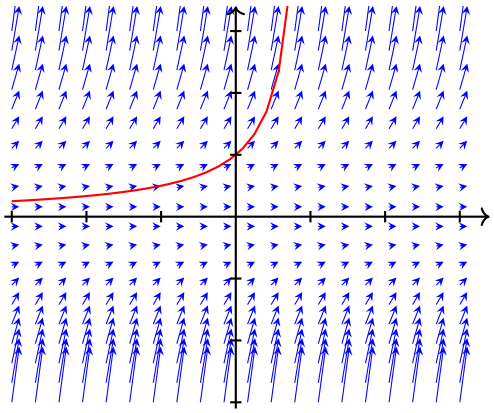
Die maximal fortgesetzte Lösung ist

$$x^* \colon [0, \infty) \to \mathbb{R},\ x^*(t) = e^{-kt}.$$

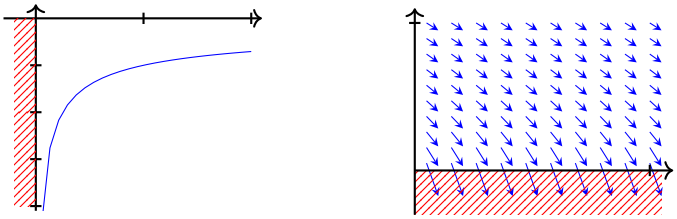
Beispiel. Für a) $f(t, x) = \sin t \cos x$



Für b) $f(t,x) = x^2$



Für c) $f(t,x) = -\frac{1}{\sqrt{x}}$



Wann sind Lösungen bis an den Rand fortsetzbar? Die Antwort ist erstaunlich einfach! Gegeben sei das AWP

$$\dot{x} = f(t,x), \qquad x(t_0) = x_0.$$

Satz 6.1 (Peano,1890)**.** *Sei $f\colon \Omega \subseteq \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$ stetig im zweiten Argument. Dann hat das AWP für alle $(t_0, x_0) \in \Omega$ mindestens eine Lösung. Jede Lösung lässt sich bis an den Rand von Ω fortsetzen.*

Beweisskizze.

1. Konstruiere eine numerische Approximation der vermuteten Lösung, mit Genauigkeitsparameter h.
2. Die Folge dieser Approximationen für $h \to 0$ hat eine konvergente Teilfolge.
3. Der Grenzwert dieser Teilfolge löst das AWP. □

Eindeutigkeit: Es kann mehr als eine Lösung geben.

Beispiel. Betrachte $x' = \sqrt{|x|}$, $x(0) = 0$.

- $f(x) = \sqrt{|x|}$ ist stetig auf $\mathbb{R}$, es existiert also eine Lösung.
- Z.B.: $x(t) = 0\ \forall t$ ist Lösung
- Eine Lösung ist aber auch $x(t) = \frac{1}{4}t^2$ für $t > 0$

- Es kommt noch schlimmer:
  - Wähle ein $c > 0$
  - Definiere
$$\tilde{x}(t) := \begin{cases} 0 & \text{falls } 0 \le t \le c \\ \frac{1}{4}(t-c)^2 & \text{falls } c < t \end{cases}$$
  - $\tilde{x}$ löst das Problem.

  Es gibt also unendlich viele Lösungen!

Für die Eindeutigkeit braucht man noch ein bisschen mehr.

Definition. *Die Abbildung $f \in C(\Omega, \mathbb{R}^d)$ heißt auf Ω bzgl. x lokal Lipschitz-stetig, wenn zu jedem $(t_0, x_0) \in \Omega$ ein offener Zylinder*

$$Z \colon (t_0 - \tau, t_0 + \tau) \times B_\rho(x_0) \subset \Omega$$

existiert, in dem eine Lipschitzbedingung

$$|f(t,x) - f(t,\overline{x})| \le L|x - \overline{x}| \qquad \forall (t,x), (t,\overline{x}) \in Z$$

mit Konstante L gilt.

Bemerkung. Falls $f(t,x)$ nach x ableitbar ist, dann ist es auch bzgl. x lokal Lipschitz-stetig.

Jetzt kommt der zentrale Satz zur Eindeutigkeit.
Benannt nach Émile Picard (1890) und Ernst Lindelöf (1894).

Satz 6.2 (Picard, Lindelöf)**.** *Betrachte das Anfangswertproblem*

$$x' = f(t,x), \qquad x(t_0) = x_0$$

auf dem erweiterten Zustandsraum $\Omega \subset \mathbb{R} \times \mathbb{R}^d$ mit $(t_0, x_0) \in \Omega$. f sei stetig, und bzgl. x lokal Lipschitz-stetig.

Dann besitzt das AWP eine bis an den Rand von Ω fortgesetzte Lösung. Sie ist eindeutig bestimmt, d.h. Fortsetzung jeder weiteren Lösung.

Beweisskizze.
- Schreibe AWP als
$$x(t) = x_0 + \int_{t_0}^{t} f(s, x(s))\, ds \qquad \forall t \ge t_0$$
- Konstruiere dafür eine Fixpunktiteration
$$\varphi_0(t) = x_0 \qquad \forall t \ge t_0$$
$$\varphi_{k+1}(t) = x_0 + \int_{t_0}^{t} f(s, \varphi_k(s))\, ds \qquad \text{„Picard-Iteration“}$$
- Banachscher Fixpunktsatz:
  1) Die Iteration konvergiert gegen einen Fixpunkt.
  2) Der Fixpunkt ist eindeutig.
- Der Fixpunkt löst das AWP. □

6.3 Evolution und Phasenfluss

Falls die Bedingungen des Satzes von Picard–Lindelöf gelten, so kann man eine elegante neue Notation einführen.

Sei $(t_0, x_0) \in \Omega$. Bezeichne mit $J_{\max}(t_0, x_0)$ das maximale Zeitintervall, auf dem eine Lösung des dazugehörigen AWPs existiert.

Zu jedem Anfangswert (t_0, x_0) gibt es eine eindeutige Lösung, d.h. zu jedem AW (t_0, x_0) ist der Wert $x(t)$ für alle $J_{\max}(t_0, x_0)$ eindeutig bestimmt.

Definition. *Für alle $t_0, t \in J_{\max}(t_0, x_0)$ heißt*

$$\Phi^{t,t_0} \colon x_0 \mapsto x(t)$$

Evolution *der Differentialgleichung $x' = f(t, x)$.*

- Wohldefiniert, weil das AWP für jedes x_0 eine eindeutige Lösung hat.

Man kann also schreiben: $x(t) = \Phi^{t,t_0} x_0$.

Der Satz von Picard–Lindelöf erhält folgende schöne Form: Deuflhard und Bornemann [1, Lemma 2.9]

Satz 6.3 (Picard–Lindelöf)**.** *Es mögen die Bedingungen des Satzes von Picard–Lindelöf gelten. Für alle $(t_0, x_0) \in \Omega$ gilt*

$$J_{max}(t_0, x_0) = J_{max}(t, \Phi^{t,t_0} x_0) \qquad \forall t \in J_{max}(t_0, x_0).$$

Außerdem

1. $\Phi^{t_0,t_0} x_0 = x_0$
2. $\Phi^{t,s} \Phi^{s,t_0} x_0 = \Phi^{t,t_0} x_0$ *für alle $t, s \in J_{max}(t_0, x_0)$.*

Für autonome Gleichungen $x' = f(x)$ kann man die Abhängigkeit von t_0 weglassen (wähle immer $t_0 = 0$). Der Evolutionsoperator $\Phi^t x_0 = x(t)$ heißt dann „Phasenfluss“.

Aus den zwei Eigenschaften des vorigen Satzes wird dann

1. $\Phi^0 x_0 = x_0$
2. $\Phi^t \Phi^s x_0 = \Phi^{t+s} x_0$ (damit auch $\Phi^{-t} \Phi^t x_0 = \Phi^0 x_0 = x_0$).

Der Phasenfluss Φ hat also eine Gruppenstruktur.

6.4 Explizite Einschrittverfahren für AWP

Ziel: Finde eine numerische Approximation der Lösung $x \in C^1([t_0, T], \mathbb{R}^d)$ des AWPs

$$x' = f(t, x), \qquad x(t_0) = x_0$$

Vorgehensweise:

- Unterteile das Intervall $[t_0, T]$ durch $n + 1$ Zeitpunkte

$$t_0 < t_1 < t_2 < \ldots < t_n = T. \tag{6.1}$$

- Die Menge der Zeitpunkte heißt *Gitter* $\Delta := \{t_0, t_1, \ldots, t_n\}$
- Schrittweite: $\tau_j := t_{j+1} - t_j$ für $j = 0, \ldots, n-1$
- Maximale Schrittweite: $\tau_\Delta = \max_{j=0,\ldots,n-1} \tau_j$

Wir suchen eine Gitterfunktion

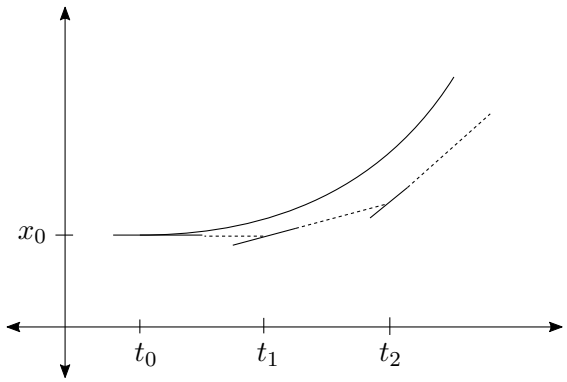
$$x_\Delta \colon \Delta \to \mathbb{R}^d,$$

welche die Lösung des AWPs an den Gitterpunkten möglichst gut approximiert.

(Manchmal interpretieren wir so ein x_Δ auch als eine Funktion $[t_0, T] \to \mathbb{R}^d$, die die Werte an den Gitterpunkten linear interpoliert.)

6.4.1 Das explizite Euler-Verfahren

Nach L. Euler (1786), auch Eulersches Polygonzugverfahren genannt.



1. $x_\Delta(t_0) = x_0$
2. Für $t \in [t_j, t_{j+1}]$:

$$x_\Delta(t) = x_\Delta(t_j) + (t - t_j) f(t_j, x_\Delta(t_j))$$

3. Insbesondere:
$$x_\Delta(t_{j+1}) = x_\Delta(t_j) + \tau_j f(t_j, x_\Delta(t_j))$$
Keine Gleichungsysteme zu lösen $\longrightarrow$ das Verfahren ist explizit.

Beachte: Berechnung durch eine Zweiterm-Rekursion

1. $x_\Delta(t_0) = x_0$
2. $x_\Delta(t_{j+1}) = \Psi^{t_{j+1},t_j} x_\Delta(t_j), j = 0, 1, \ldots, n-1$

mit Ψ unabhängig von Δ.

- Die Funktion Ψ heißt *diskrete Evolution* des expliziten Euler-Verfahrens.

Hoffnung natürlich: Wenn das Gitter immer feiner wird (wenn also τ_Δ immer kleiner wird), wird der Unterschied zwischen x und x_Δ immer kleiner.

6.5 Konsistenz

Der Fehler der Lösung, also der Unterschied $x - x_\Delta$, besteht aus zwei Beiträgen:

- Jeder einzelne Schritt produziert einen Fehler.
- Da man nach dem ersten Schritt immer von einem fehlerbehafteten Wert startet, bekommt man auch falsche Ableitungen f.

Mit *Konsistenz* bezeichnet man das *lokale* Verhältnis zwischen der Evolution Φ und der diskreten Evolution Ψ.

Definition. *Eine diskrete Evolution Ψ heißt konsistent, falls*

$$\Psi^{t,t} x = x \qquad \text{für alle } (t,x) \in \Omega,$$
$$\frac{d}{d\tau} \Psi^{t+\tau,t} x \Big|_{\tau=0} = f(x,t) \qquad \text{für alle } (t,x) \in \Omega.$$

Definition. *Sei $(t,x) \in \Omega$. Die Differenz*

$$\varepsilon(t,x,\tau) = \Phi^{t+\tau,t} x - \Psi^{t+\tau,t} x$$

heißt Konsistenzfehler von Ψ.

Man kann konsistente Evolutionen einfach charakterisieren.

Lemma 6.1 (Deuflhard und Bornemann [1, Lemma 4.4])**.** *Die diskrete Evolution $\Psi^{t+\tau,t} x$ sei für jedes $(t,x) \in \Omega$ und hinreichend kleines τ bzgl. τ stetig differenzierbar. Dann ist äquivalent:*

1. Ψ *ist konsistent*
2. Ψ *hat die Darstellung*
$$\Psi^{t+\tau,t}x = x + \tau\psi(t,x,\tau)$$
ψ heißt Inkrementfunktion. ψ ist stetig in $\tau = 0$, und
$$\psi(t,x,0) = f(t,x).$$
3. *Für den Konsistenzfehler gilt*
$$\varepsilon(t,x,\tau) = o(\tau), \qquad \tau \to 0 \qquad (\textit{also} \lim_{\tau\to 0} \frac{\varepsilon(\tau)}{\tau} = 0).$$

Beweis. Zunächst 1) $\implies$ 3):

- Taylor-Entwicklung von Φ um $\tau = 0$
$$\begin{aligned}\Phi^{t+\tau,t}x &= \Phi^{t,t}x + \tau \cdot . \frac{d}{d\tau}\Phi^{t+\tau,t}x|_{\tau=0} + o(\tau)\\ &= x + \tau \cdot f(t,x) + o(\tau)\end{aligned}$$
- Ebenso für Ψ:
$$\Psi^{t+\tau,t}x = x + \tau \cdot f(t,x) + o(\tau) \tag{6.2}$$
da Ψ konsistent ist.
- Subtraktion zeigt 1) $\implies$ 3). Ebenso 3) $\implies$ 1)

Zu 2):

- (6.2) bedeutet:
$$\Psi^{t+\tau,t}x = x + \tau f(t,x) + \eta(t,x,\tau)$$
mit einer Funktion η für die $\lim_{\tau\to 0} \frac{\eta(\tau)}{\tau} = 0$.
- Umformen ergibt
$$\frac{\Psi^{t+\tau,t}x - x}{\tau} = f(t,x) + \frac{\eta(t,x,\tau)}{\tau}.$$
Die Funktion
$$\psi(t,x,\tau) := f(t,x) + \frac{\eta(t,x,\tau)}{\tau}$$
ist also gerade die Inkrementfunktion aus 2).
- ψ stetig und $\psi(t,x,0) = f(t,x)$, da $\eta \in o(\tau)$. □

Das vorige Lemma beschreibt Äquivalenzen. Insbesondere gilt: Wenn sich Ψ mit einer Inkrementfunktion schreiben lässt, so ist Ψ konsistent.

Beispiel. Das explizite Euler-Verfahren

$$\Psi^{t+\tau,t}x = x + \tau f(t,x) = x + \tau\psi(t,x,\tau)$$

ist konsistent.

Wir wollen eine quantitative Version von Konsistenz.

Definition. *Eine diskrete Evolution Ψ besitzt Konsistenzordnung p, wenn es eine Konstante $C > 0$ unabhängig von t und x gibt, so dass*

$$\varepsilon(t,x,\tau) \le C\tau^{p+1}.$$

Ja, der Exponent ist wirklich $p+1$!

Beispiel (Konsistenzordnung des Euler-Verfahrens). Das Euler-Verfahren ist

$$\Psi^{t+\tau,t}x = x + \tau f(t,x).$$

- Sei $f \in C^1(\Omega)$.
- Konsistenzfehler: $\varepsilon(t,x,\tau) = \Phi^{t+\tau,t}x - x - \tau f(t,x)$

Wir machen eine Taylor-Entwicklung von $\Phi^{t+\tau,t}x$ in $\tau = 0$.

- Erste Ableitung von Φ nach τ:

$$\frac{d}{d\tau}\Phi^{t+\tau,t}x = f(t+\tau, \Phi^{t+\tau,t}x)$$

- Warum? Sei $x(t) = \Phi^{t,t_0}x_0$ die Lösung des AWPs.
  Dann ist $\Phi^{t+\tau,t}x = x(t+\tau)$, und

$$\begin{aligned}\frac{d}{d\tau}\Phi^{t+\tau,t}x &= x'(t+\tau)\big|_{\tau=0} = f(t+\tau, x(t+\tau)) \\ &= f(t+\tau, \Phi^{t+\tau,t}x).\end{aligned}$$

- Zweite Ableitung (Kettenregel)

$$\frac{d^2}{d\tau^2}\Phi^{t+\tau,t}x = f_t\big(t+\tau, \Phi^{t+\tau,t}x\big) + f_x\big(t+\tau, \Phi^{t+\tau,t}x\big) \cdot \underbrace{\frac{d}{d\tau}\Phi^{t+\tau,t}x}_{=f(t+\tau,\Phi^{t+\tau,t}x)}$$

- Taylor-Entwicklung um $\tau = 0$:

$$\begin{aligned}\Phi^{t+\tau,t}x &= \Phi^{t+0,t}x + \tau \cdot \frac{d}{d\tau}\Phi^{t+\tau,t}x\Big|_{\tau=0} + \tau^2 \int_0^1 (1-\sigma)\frac{d^2}{d\tau^2}\Phi^{t+\sigma\tau,t}x\, d\sigma \\ &= x + \tau f(t,x) + \tau^2 \int_0^1 (1-\sigma)\Big[f_t\big(t+\sigma\tau, \Phi^{t+\sigma\tau,t}x\big) \\ &\quad + f_x\big(t+\sigma\tau, \Phi^{t+\sigma\tau,t}x\big) f\big(t+\sigma\tau, \Phi^{t+\sigma\tau,t}x\big)\Big] d\sigma\end{aligned}$$

- Deshalb ist

$$\begin{aligned}|\varepsilon(t,x,\tau)| &= |\Phi^{t+\tau,t}x - x - \tau f(t,x)| \\ &= \tau^2 \Big| \int_0^1 \dots\, d\sigma \Big| \\ &\le \tau^2 \frac{1}{2} \max_{(s,z)\in K} |f_t(s,z) + f_x(s,z) f(s,z)|.\end{aligned}$$

- Das Maximum existiert, denn es wird über eine kompakte Menge K maximiert. Da wir nur an kleinen τ interessiert sind können wir immer solch ein Kompaktum K, das alle relevanten (t,x) enthält (siehe Deuflhard und Bornemann [1, Beispiel 4.8]).
- Die Konsistenzordnung ist also 1.

6.6 Konvergenz

- Konsistenz ist ein lokales Phänomen, d.h. es betrachtet den Fehler nur in der Nähe eines festen t.
- Wir betrachten jetzt, wie gut eine Lösung $x \in C^1([t_0, T])$ insgesamt approximiert wird.
- Wir hoffen, dass für kleinere $\tau_\Delta = \max \tau_j$ die Approximation immer besser wird
- und das schnell!

Definition. *Der Vektor der Approximationsfehler auf dem Gitter* Δ

$$\varepsilon_\Delta \colon \Delta \to \mathbb{R}^d, \qquad \varepsilon_\Delta(t) = x(t) - x_\Delta(t)$$

heißt Gitterfehler. *Seine Norm*

$$\|\varepsilon_\Delta\|_\infty = \max_{t\in\Delta} |\varepsilon_\Delta(t)|$$

heißt Diskretisierungsfehler.

Definition. *Zu jedem Gitter Δ auf $[t_0, T]$ sei eine Gitterfunktion x_Δ gegeben. Die Familie dieser Gitterfunktionen konvergiert mit Ordnung $p \in \mathbb{N}$ gegen $x \in C^1([t_0, T])$, falls eine Konstante $C > 0$ existiert, so dass*

$$\|\varepsilon_\Delta\|_\infty \leq C\tau_\Delta^p$$

für alle τ_Δ klein genug.

Alternative Notation: $\|\varepsilon_\Delta\|_\infty = \mathcal{O}(\tau_\Delta^p)$.

- Konvergenz hängt eng mit dem Konsistenzfehler zusammen.
- Das ist praktisch: Der Konsistenzfehler lässt sich häufig direkt am Verfahren ablesen.

Satz 6.4 (Deuflhard und Bornemann [1, Satz 4.10])**.** *Sei ψ lokal Lipschitz-stetig in x. Die diskrete Evolution sei konsistent mit Ordnung p, d.h.*

$$\varepsilon(t, x, \tau) := x(t + \tau) - \Psi^{t+\tau,t}x(t) = \mathcal{O}(\tau^{p+1}).$$

Dann definiert die diskrete Evolution Ψ für alle Gitter Δ mit hinreichend kleiner Zeitschrittweite τ_Δ eine Gitterfunktion x_Δ zum Anfangswert $x_\Delta(t_0) = x_0$. Die Familie dieser Gitterfunktionen konvergiert mit Ordnung p gegen die Lösung x des AWPs, d.h.

$$\|\varepsilon_\Delta\|_\infty := \max_{t\in\Delta} |x(t) - x_\Delta(t)| = \mathcal{O}(\tau_\Delta^p).$$

Beweis.

- Sei $K \subset \Omega$ kompakte Umgebung des Graphen der Lösung x
- ψ ist lokal Lipschitz-stetig in x, d.h. es gibt $\tau_K, \Lambda_K > 0$ so dass

  $$|\psi(t, x, \tau) - \psi(t, \overline{x}, \tau)| \leq \Lambda_K |x - \overline{x}|$$

  für alle $(t, x), (t, \overline{x}) \in K$ und $\tau \in [0, \tau_K]$.
- Insbesondere ist dann $\Psi^{t+\tau,t}x$ für alle $(t, x) \in K, 0 \leq \tau \leq \tau_K$ definiert.
- Es gibt einen „Schlauch“ der Dicke $2\delta_K > 0$ um die Lösung x, der in K enthalten ist.

  Genauer: Es gibt ein $\delta_K > 0$ so dass für alle $t \in [t_0, T]$

  $$\text{Falls } |y - x(t)| \leq \delta_K, \text{ so folgt } (t, y) \in K.$$

- Sei Δ Gitter für $[t_0, T]$ mit $\tau_\Delta \leq \tau_K$.
- *Setze voraus*, dass x_Δ existiert und

  $$|\varepsilon_\Delta(t)| = |x(t) - x_\Delta(t)| \leq \delta_K \qquad \forall t \in \Delta.$$

- Betrachte ε_Δ genauer. Der Gitterfehler im Gitterpunkt t_{j+1} besteht aus zwei Teilen:

$$\begin{aligned}\varepsilon_\Delta(t_{j+1}) &= x(t_{j+1}) - x_\Delta(t_{j+1}) \\ &= x(t_{j+1}) - \Psi^{t_{j+1},t_j} x_\Delta(t_j) \\ &= \underbrace{x(t_{j+1}) - \Psi^{t_{j+1},t_j} x(t_j)}_{\text{Konsistenzfehler}} + \underbrace{\Psi^{t_{j+1},t_j} x(t_j) - \Psi^{t_{j+1},t_j} x_\Delta(t_j)}_{=:\varepsilon_j}.\end{aligned}$$

- ε_j kann als Propagation des Fehlers $\varepsilon_\Delta(t_j)$ durch Ψ zum Zeitpunkt t_{j+1} interpretiert werden.

  Denn: Angenommen, Ψx sei linear in x (ist es nicht!). Dann wäre

$$\varepsilon_j = \Psi^{t_{j+1},t_j}(x(t_j) - x_\Delta(t_j)) = \Psi^{t_{j+1},t_j} e_\Delta(t_j).$$

  (*Achtung:* ε_j beschreibt eine Größe zur Zeit t_{j+1}!)

- Darstellung von ε_j mit der Inkrementfunktion:

$$\varepsilon_j = x(t_j) - x_\Delta(t_j) + \tau\Big[\psi(t_j, x(t_j), \tau_j) - \psi(t_j, x_\Delta(t_j), \tau_j)\Big]$$

- Lipschitz-Stetigkeit von ψ:

$$|\varepsilon_j| \le |\varepsilon_\Delta(t_j)| + \tau_j \Lambda_K |x(t_j) - x_\Delta(t_j)| = (1 + \tau_j \Lambda_K)|\varepsilon_\Delta(t_j)|$$

Da wir die Abschätzung für alle j machen können folgt:

1. $|\varepsilon_\Delta(t_0)| = 0$
2. $|\varepsilon_\Delta(t_{j+1})| \le C\tau_j^{p+1} + (1 + \tau_j \Lambda_K)|\varepsilon_\Delta(t_j)|$

Behauptung: Daraus folgt

$$|\varepsilon_\Delta(t)| \le \tau_\Delta^p \frac{C}{\Lambda_K}(e^{\Lambda_K(t-t_0)} - 1) \tag{6.3}$$

für alle $t \in \Delta$.

Damit haben wir

$$\|\varepsilon_\Delta\|_\infty \le \tau_\Delta^p \frac{C}{\Lambda_K}(e^{\Lambda_K(T-t_0)} - 1) = O(\tau_\Delta^p).$$

Achtung: wir mussten *voraussetzen*, dass $\|\varepsilon_\Delta\|_\infty$ klein ist! □

.

Beweis der Behauptung (6.3). Induktion:

- Die Behauptung stimmt für $j = 0$.

- Angenommen die Behauptung ist richtig für $j < n$. Dann folgt

$$\begin{aligned}|\varepsilon_\Delta(t_{j+1})| &\le C \underbrace{\tau_j^{p+1}}_{\le \tau_\Delta^p \cdot \tau_j} + (1+\tau_j\Lambda_K)\tau_\Delta^p \frac{C}{\Lambda_K}(e^{\Lambda_K(t_j-t_0)}-1) \\ &\le \tau_\Delta^p \frac{C}{\Lambda_K}[\tau_j\Lambda_K + (1+\tau_j\Lambda_K)(e^{\Lambda_K(t_j-t_0)}-1)] \\ &= \tau_\Delta^p \frac{C}{\Lambda_K}[(1+\tau_j\Lambda_K)e^{\Lambda_K(t_j-t_0)}-1].\end{aligned}$$

- Es gilt aber $1+\alpha \le e^\alpha$, bzw. $1+\tau_j\Lambda_K \le e^{\tau_j\Lambda_K}$, also

$$(1+\tau_j\Lambda_K)e^{\Lambda_K(t_j-t_0)} \le e^{\Lambda_k\tau_j}e^{\Lambda_k(t_j-t_0)} = e^{\Lambda_K(t_{j+1}-t_0)}.$$ □

Für die Profis hier noch die Argumentation, warum die Annahme dass $\|\varepsilon_\Delta\|_\infty$ klein ist, weggelassen werden kann.

- Bisher mussten wir voraussetzen, dass

$$|\varepsilon_\Delta(t)| \le \delta_K \qquad \text{für alle } t \in \Delta.$$

  Jetzt haben wir aber (6.3)!

  Damit zeigen wir, dass die Bedingung gilt, wenn τ_Δ klein genug ist.

- Wähle $\tau_* > 0$ so klein dass

$$\tau_*^p \frac{C}{\Lambda_K}(e^{\Lambda_K(T-t_0)}-1) \le \delta_K \qquad \text{und} \qquad \tau_* \le \tau_K.$$

- Wähle ein Gitter Δ auf $[t_0, T]$ mit $\tau_\Delta \le \tau_*$.
- Zeige mit der gleichen Induktion wie oben für alle j dass
  - die Abschätzung

$$|\varepsilon_\Delta(t_j)| \le \tau_\Delta^p \frac{C}{\Lambda_K}(e^{\Lambda_K(t_j-t_0)}-1) \le \delta_K$$

    gilt, und deshalb
  - $x_\Delta(t_{j+1})$ existiert.

Beispiel (Explizites Euler-Verfahren).

$$x_{j+1} = x_j + \tau_j f(t_j, x_j)$$

- Konsistenzordnung 1 (der lokale Fehler verhält sich wie τ_Δ)
- Inkrementfunktion $\psi(t, x, \tau) = f(t, x)$ ist lokal Lipschitz-stetig in x.

$\Longrightarrow$ Verfahren konvergiert mit Ordnung 1, d.h. $\|\varepsilon_\Delta\|_\infty = \mathcal{O}(\tau_\Delta^1)$.

$\Longrightarrow$ halber Fehler bedeutet doppelte Anzahl von Zeitschritten.

$\Longrightarrow$ doppelte Anzahl von f-Auswertungen $\Longrightarrow$ doppelter Aufwand.

6.7 Explizite Runge–Kutta-Verfahren

[Nach: Carl Runge, 1856 (Bremen) – 1927 (Göttingen), Martin Wilhelm Kutta, 1867 (Pitschen/Oberschlesien) – 1944 (Fürstenfeldbruck)]

Können wir Verfahren mit einer höheren Konsistenzordnung konstruieren?

Wiederholung: Wie lief der Beweis, dass das explizite Euler-Verfahren Konsistenzordnung 1 hat?

- Konsistenzfehler:

$$\begin{aligned}\varepsilon(t,x,\tau) &= \Phi^{t+\tau,t}x - \Psi^{t+\tau,t}x \\ &= (\Phi^{t+\tau,t}x - x) - \tau\psi(t,x,\tau)\end{aligned}$$

- Entwickle den Term in Klammern

$$\Phi^{t+\tau,t}x - x = \tau f(t,x) + \mathcal{O}(\tau^2)$$

- Euler-Verfahren: $\psi(t,x,\tau) = f(t,x)$ eliminiert gerade den ersten Term!

arrow Konsistenzfehler ist $\mathcal{O}(\tau^2)$

6.7.1 Taylor-Verfahren

Idee: Berechne weiteren Term der Taylor-Reihe:

$$\Phi^{t+\tau,t}x - x = \tau f(t,x) + \frac{\tau^2}{2}[\frac{df(t,x)}{dt} + \frac{df(t,x)}{dx} \cdot f(t,x)] + \mathcal{O}(\tau^3)$$

Inkrementfunktion für ein Verfahren mit Ordnung 2:

$$\psi^* = f(t,x) + \frac{\tau}{2}\Big[f_t(t,x) + f_x(t,x) \cdot f(t,x)\Big]$$

- So konstruierte Verfahren heißen „Taylor-Verfahren“.
- Im Prinzip für jede Ordnung möglich, solange f glatt genug ist.
- Man muss Ableitungen von f berechnen.
- Das geht sogar automatisch!
  - Schlagwort: Automatisches Differenzieren (AD)
  - de facto aber nur für Ableitungen niedriger Ordnung
- Wird deshalb in der Praxis kaum verwendet.

6.7.2 Idee der Runge-Kutta-Verfahren

Ziel: Hohe Ordnung ohne Ableitungen von f.
Idee: (Runge, 1893)

- Hauptsatz der Integralrechnung:

$$x\colon [t_0, T] \to \mathbb{R}^d \qquad \text{löst} \qquad x' = f(t, x),$$

also

$$x(t+\tau) = x(t) + \int_0^\tau f(t+\sigma, x(t+\sigma))\, d\sigma$$

bzw.

$$\Phi^{t+\tau,t}x - x = \int_0^\tau f(t+\sigma, \Phi^{t+\sigma,t}x)\, d\sigma$$

- Approximiere das Integral numerisch, z.B. Mittelpunktsregel / 1-Punkt Gauß–Legendre:

$$\int_0^\tau f(t+\sigma, \Phi^{t+\sigma,t}x)\, d\sigma = \tau f(t+\frac{\tau}{2}, \Phi^{t+\frac{\tau}{2},t}x) + \mathcal{O}(\tau^3)$$

- Beachte: auch hier ist der Fehler in $\mathcal{O}(\tau^3)$!

Wie berechnet man aber $\Phi^{t+\frac{\tau}{2},t}x$?

- Auf den ersten Blick nicht einfacher zu berechnen als $\Phi^{t+\tau,t}x$.
- *Glück*: $\Phi^{t+\frac{\tau}{2},t}x$ taucht in einem Term auf, der mit τ multipliziert wird.
- Es reicht deshalb, $\Phi^{t+\frac{\tau}{2},t}x$ bis auf $\mathcal{O}(\tau^2)$ zu berechnen.
- Das geht mit dem expliziten Euler-Verfahren:

$$\Phi^{t+\frac{\tau}{2},t}x = x + \frac{\tau}{2} f(t, x) + \mathcal{O}(\tau^2)$$

- Man erhält das *Verfahren von Runge*:

$$\Psi^{t+\tau,t}x = x + \tau f\Big(t + \frac{\tau}{2}, x + \frac{\tau}{2} f(t, x)\Big)$$

mit Konsistenzordnung 2.

- Zwei Auswertungen von f pro Gitterpunkt, gegenüber 4 Auswertungen (von f oder seinen Ableitungen) beim entsprechenden Taylor-Verfahren.

Den Ansatz von Runge kann man verallgemeinern. Das war der Beitrag von Kutta 1901.

- Schreibe Verfahren von Runge in drei Schritten:

1. $k_1 := f(t, x)$
2. $k_2 := f(t + \frac{\tau}{2}, x + \frac{\tau}{2} k_1)$
3. $\Psi^{t+\tau,t} x = x + \tau k_2$

- Allgemein: s-stufiges explizites Runge-Kutta-Verfahren:

1. $k_i := f\left(t + c_i\tau, x + \tau \sum_{j=1}^{i-1} a_{ij} k_j\right) \quad \forall i = 1, \ldots, s$
2. $\Psi^{t+\tau,t} x = x + \tau \sum_{i=1}^{s} b_i k_i.$

- Die Größen $k_i = k_i(t, x, \tau)$ heißen *Stufen* des Verfahrens.

Koeffizienten:

$$b = (b_1, \ldots, b_s)$$
$$c = (c_1, \ldots, c_s)$$
$$A = \begin{pmatrix} 0 & & & & \\ a_{21} & 0 & & & \\ \vdots & \ddots & & & \\ a_{s1} & a_{s2} & \ldots & a_{s,s-1} & 0 \end{pmatrix} \in \mathbb{R}^{s \times s}$$

Traditionell notiert man die Koeffizienten im Butcher-Schema
(nach John C. Butcher, 1933 in Auckland)

$$\begin{array}{c|c} c^T & A \\ \hline & b \end{array}$$

Beispiele:

expl. Euler-Verfahren $\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$

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

- Eine Funktionsauswertung pro Stufe
- $s + s + \frac{(s-1)s}{2}$ Parameter bei s Stufen

Wie müssen die Koeffizienten gewählt werden, um eine möglichst hohe Konsistenzordnung zu erreichen?

Zuerst: Konsistenz an sich (d.h., Konsistenz erster Ordnung)

- Einschrittverfahren
$$\Psi^{t+\tau,t}x = x + \tau\psi(t,x,\tau)$$
ist konsistent, wenn $\psi(t,x,0) = f(t,x)$ ist.

- Runge–Kutta-Verfahren:
$$k_i(t,x,0) = f\Big(t + c_i \cdot 0, x + 0 \cdot \sum_{j=1}^{i-1} a_{ij}k_j\Big) = f(t,x)$$
$$\psi(t,x,0) = \sum_{i=1}^{s} b_i k_i = f(t,x)\sum_{i=1}^{s} b_i$$

Lemma 6.2. *Ein explizites Runge–Kutta-Verfahren ist genau dann konsistent für alle $f \in C(\Omega, \mathbb{R}^d)$, wenn $\sum_{i=1}^{s} b_i = 1$.*

6.7.3 Autonomisierung

- Taylor- und Runge–Kutta-Verfahren brauchen $f \in C^p(\Omega, \mathbb{R}^d$, also gleiche Glattheit in Zeit und Zustand.
- Wir wollen die Notation vereinfachen, und nur noch autonome Gleichungen betrachten.
- Das geht erstaunlich einfach! Ersetze
$$x' = f(t,x), \qquad x(t_0) = x_0$$
durch das erweiterte System
$$\begin{pmatrix} x'(t) \\ s'(t) \end{pmatrix} = \begin{pmatrix} f(s(t),x(t)) \\ 1 \end{pmatrix}, \qquad \begin{pmatrix} x(0) \\ s(0) \end{pmatrix} = \begin{pmatrix} x_0 \\ t_0 \end{pmatrix}.$$
- Beide Systeme haben die gleiche Lösung.
- Runge–Kutta-Verfahren liefern möglicherweise *unterschiedliche Lösungen* für die beiden Gleichungen!
- Das ist natürlich häßlich!

Lemma 6.3. *Ein explizites Runge–Kutta-Verfahren ist genau dann invariant unter Autonomisierung, wenn es konsistent ist, und*
$$c_i = \sum_{j=1}^{i-1} a_{ij} \qquad \textit{für } i = 1, \ldots, s$$
erfüllt.

Wir betrachten bis auf weiteres nur noch solche Runge–Kutta-Verfahren, und schreiben sie

$$(b, A).$$

- Nur noch autonome Probleme

$$x' = f(x), \qquad x(0) = x_0$$

  mit Phasenfluss Φ^t.

- Diskreter Fluss $\Psi^{t+\tau,t}$ vereinfacht sich zu

$$\Psi^t x = x + \tau\psi(x, \tau).$$

6.7.4 Konstruktion von Runge-Kutta-Verfahren

Wir wollen ein Runge–Kutta-Verfahren (b, A) der Ordnung p konstruieren.

- Wieviele Stufen brauche ich?
  - Mindestens p Stück
- Nach Autonomisierung sind noch $\frac{(s+1)s}{2}$ Koeffizienten zu bestimmen.

Schritt 1: Aufstellung von Bedingungsgleichungen an die Koeffizienten, die die gewünschte Ordnung liefern.
Schritt 2: Lösen dieser Gleichungen.

Wir behandeln den Fall $p = 4$.

- Autonome Gleichung $x' = f(x)$ mit $f \in C^4(\Omega_0)$.

Ansatz: Entwickle den Konsistenzfehler

$$\varepsilon(x, \tau) = \Phi^\tau x - \Psi^\tau x$$

in Taylor-Reihen bis auf $\mathcal{O}(\tau^5)$.

Konstruiere Ψ so, dass die ersten p Terme von ε verschwinden.

Taylor-Entwicklung des Phasenflusses Φ^τ

Taylor-Entwicklung von Φ^τ nach τ.
Vom Euler-Verfahren wissen wir schon dass

$$\Phi^\tau x = x + \tau f(x) + \frac{\tau^2}{2} f'(x) \cdot f(x) + \mathcal{O}(\tau^3).$$

Wie bekommen wir die nächsthöhere Ordnung?

Trick:

$$\frac{d}{d\tau}\Phi^\tau x = x'(\tau) = f(x(\tau),\tau) = f(\Phi^\tau x)$$

Einsetzen der Reihe für Φ^τ:

$$\frac{d}{d\tau}\Phi^\tau x = f\Big(x + \tau f(x) + \frac{\tau^2}{2} f'(x)\cdot f(x) + \mathcal{O}(\tau^3)\Big)$$

Taylor-Entwicklung davon:

$$\begin{aligned}\frac{d}{d\tau}\Phi^\tau x &= f(x) + f'(x)\cdot\Big(\tau f(x) + \frac{\tau^2}{2} f'(x)f(x) + \mathcal{O}(\tau^3)\Big) + \frac{1}{2!}f''(x)(\tau f(x), \tau f(x)) + \mathcal{O}(\tau^3)\\ &= f + f'f + \frac{\tau^2}{2}\Big(f'f'f + f''(f,f)\Big) + \mathcal{O}(\tau^3)\end{aligned}$$

Wir wollen aber eine Reihendarstellung von $\Phi^\tau x$.

- Hauptsatz:

$$\begin{aligned}\Phi^\tau x &= x + \int_0^\tau \frac{d}{d\sigma}\Phi^\sigma x\, d\sigma\\ &= x + \int_0^\tau \Big[f + \sigma f'f + \frac{\sigma^2}{2}(f'f'f + f''(f,f)) + \mathcal{O}(\sigma^3)\Big]\, d\sigma\\ &= x + \tau f + \frac{\tau^2}{2}f'f + \frac{\tau^3}{3!}(f'f'f + f''(f,f)) + \mathcal{O}(\tau^4)\end{aligned}$$

- Eine Ordnung mehr!
- Und nochmal!

$$\begin{aligned}f(\Phi^\tau x) = f &+ \tau f'f + \frac{\tau^2}{2}(f'f'f + f''(f,f))\\ &+ \frac{\tau^3}{3!}\Big[f'''(f,f,f) + 3f''(f'f,f) + f'f''(f,f) + f'f'f'f\Big] + \mathcal{O}(\tau^4)\end{aligned}$$

- Nochmal den Hauptsatz benutzen

$$\begin{aligned}\Phi^\tau x = x &+ \tau f + \frac{\tau^2}{2}f'f + \frac{\tau^3}{3!}(f'f'f + f''(f,f))\\ &+ \frac{\tau^4}{4!}\Big[f'''(f,f,f) + 3f''(f'f,f) + f'f''(f,f) + f'f'f'f\Big] + \mathcal{O}(\tau^5)\end{aligned}$$

Taylorentwicklung des diskreten Flusses Ψ^τ

Ähnliches Spiel mit den Runge–Kutta Gleichungen

$$k_i := f\Big(x + \tau\sum_{j=1}^{i-1} a_{ij}k_j\Big) \qquad \text{für } i = 1,\ldots,s.$$

1. f ist stetig, also auf Kompakta beschränkt
   $\Longrightarrow$ k_i beschränkt, $k_i = \mathcal{O}(1)$

2. Einsetzen:

$$k_i = f\Big(x + \tau \sum_{j=1}^{i-1} a_{ij}\mathcal{O}(1)\Big) = f(x + \tau\mathcal{O}(1)) = f(x + \mathcal{O}(\tau))$$

   Taylor-Entwicklung um x:

$$k_i = f(x) + \mathcal{O}(\tau)$$

3. Einsetzen:

$$k_i = f\Big(x + \tau \sum_{j=1}^{i-1} a_{ij}(f + \mathcal{O}(\tau))\Big) = f(x + \tau \sum_{j=1}^{i-1} a_{ij} f + \mathcal{O}(\tau^2))$$

$$= f(x + \tau c_i f + \mathcal{O}(\tau^2)) \qquad \text{mit } c_i = \sum_{j=1}^{i-1} a_{ij}.$$

   Taylor-Entwicklung:

$$k_i = f + \tau c_i f' f + \mathcal{O}(\tau^2) \qquad \forall i = 1, \ldots, s$$

4. Nochmal:

$$k_i = f\Big(x + \tau \sum_{j=1}^{i-1} a_{ij}(f + \tau c_j f' f) + \mathcal{O}(\tau^3)\Big)$$

$$= f + \tau c_i f' f + \tau^2 \sum_j a_{ij} c_j f' f' f + \frac{\tau^2}{2} c_i^2 f''(f, f) + \mathcal{O}(\tau^3)$$

5. Nochmal:

$$k_i = f\Big[x + \tau c_i f + \tau^2 \sum_{j=1}^{i-1} a_{ij} c_j f' f + \tau^3 \sum_{jk} a_{ij} a_{jk} c_k f' f' f$$
$$\frac{\tau^3}{2} \sum_j a_{ij} c_j^2 f''(f, f) + \mathcal{O}(\tau^4)\Big]$$

$$= f + \tau c_i f' f + \tau^2 \sum_j a_{ij} c_j f' f' f + \frac{\tau^2}{2} c_i^2 f''(f, f)$$
$$+ \tau^3 \sum_{jk} a_{ij} a_{jk} c_k f' f' f' f + \frac{\tau^3}{2} \sum_j a_{ij} c_j^2 f' f''(f, f)$$
$$+ \tau^3 \sum_j c_i a_{ij} c_j f''(f' f, f) + \frac{\tau^3}{6} c_i^3 f'''(f, f, f) + \mathcal{O}(\tau^4)$$

Einsetzen in $\Psi^\tau x = x + \tau \sum_i b_i k_i$:

$$\begin{aligned}\Psi^\tau x = x &+ \tau \sum_{i=1}^{s} b_i f + \frac{\tau^2}{2}\Big(2\sum_i b_i c_i f' f\Big) \\ &+ \frac{\tau^3}{3!}\Big[3\sum_i b_i c_i^2 f''(f,f) + 6\sum_{i,j} b_i a_{ij} c_j f' f' f\Big] \\ &+ \frac{\tau^4}{4!}\Big[4\sum_i b_i c_i^3 f'''(f,f,f) + 24\sum_{i,j} b_i c_i a_{ij} c_j f''(f' f, f) \\ &+ 12\sum_{i,j} b_i a_{ij} c_j^2 f' f''(f,f) + 24\sum_{i,j,k} b_i a_{ij} a_{jk} c_k f' f' f' f\Big] + \mathcal{O}(\tau^5)\end{aligned}$$

Koeffizientenvergleich ergibt die Ordnungsbedingungen:

Satz 6.5 (Deuflhard und Bornemann [1, Satz 4.18])**.** *Ein Runge–Kutta-Verfahren* (b, A) *besitzt genau dann*

- *Konsistenzordnung* $p = 1$*, falls*
$$\sum_{i=1}^{s} b_i = 1,$$
- *Konsistenzordnung* $p = 2$*, falls zusätzlich*
$$\sum_{i=1}^{s} b_i c_i = \frac{1}{2},$$
- *Konsistenzordnung* $p = 3$*, falls zusätzlich*
$$\sum_i b_i c_i^2 = \frac{1}{3} \qquad \textit{und} \qquad \sum_{i,j} b_i a_{ij} c_j = \frac{1}{6},$$
- *Konsistenzordnung* 4*, falls zusätzlich*
$$\sum_i b_i c_i^3 = \frac{1}{4} \qquad \sum_{i,j} b_i c_i a_{ij} c_j = \frac{1}{8}$$
$$\sum_{i,j} b_i a_{ij} c_j^2 = \frac{1}{12} \qquad \sum_{i,j,k} b_i a_{ij} a_{jk} c_k = \frac{1}{24}$$

  gelten.

(Eigentlich muss man noch zeigen, dass diese Bedingungen notwendig und nicht nur hinreichend sind.)

Lösen der Gleichungen

Wir wollen ein Verfahren vierter Ordnung.

- Reichen $s = 3$ Stufen?
  - Nein! 8 Gleichungen, aber nur $\frac{(s+1)s}{2} = 6$ Unbekannte.
  - Das Gleichungssystem ist überbestimmt $\longrightarrow$ keine Lösung
- $s = 4$? $\frac{(s+1)s}{2} = 10$ Unbekannte, $\longrightarrow$ könnte gehen!

Unbekannte: $b_1, b_2, b_3, b_4, a_{21}, a_{31}, a_{32}, a_{41}, a_{42}, a_{43}$

1. $b_1 + b_2 + b_3 + b_4 = 1$
2. $b_1c_1 + b_2c_2 + b_3c_3 + b_4c_4 = \frac{1}{2}$
3. $b_2c_2^2 + b_3c_3^2 + b_4c_4^2 = \frac{1}{3}$
4. $b_3a_{32}c_2 + b_4(a_{42}c_2 + a_{43}c_3) = \frac{1}{6}$
5. $b_2c_2^3 + b_3c_3^3 + b_4c_4^3 = \frac{1}{4}$
6. $b_3c_3a_{32}c_2 + b_4c_4(a_{42}c_2 + a_{43}c_3) = \frac{1}{8}$
7. $b_3a_{32}c_2^2 + b_4(a_{42}c_2^2 + a_{43}c_3^2) = \frac{1}{12}$
8. $b_4a_{43}a_{32}c_2 = \frac{1}{24}$

Außerdem $c_i = \sum_j a_{ij}$, $i = 1, \ldots, 4$, insbesondere $c_1 = 0$.

Was nun?

- Gute Idee:
$$\int_0^1 1\,dx = 1, \qquad \int_0^1 x\,dx = \frac{1}{2}, \qquad \int_0^1 x^2\,dx = \frac{1}{3}, \qquad \int_0^1 x^3 dx = \frac{1}{4}.$$
- Approximiere Integral durch 4-Punkt-Formel mit Stützstellen c_1, c_2, c_3, c_4, Gewichten b_1, b_2, b_3, b_4
- Falls Formel exakt für kubische Polynome ist, erhält man die Gleichungen 1, 2, 3, und 5.

Simpson-Regel: Stützstellen $0, \frac{1}{2}, 1$ und Gewichte $\frac{1}{6}, \frac{2}{3}, \frac{1}{6}$.

Verdopple mittleren Punkt:

$$c = (0, \frac{1}{2}, \frac{1}{2}, 1), \qquad b = (\frac{1}{6}, \frac{1}{3}, \frac{1}{3}, \frac{1}{6})$$

Damit kann man die fehlenden a_{ij} bestimmen.

Man erhält das klassische Runge-Kutta-Verfahren:

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

6.8 Lineare Mehrschrittverfahren

6.8.1 Einführung

Gegeben sei ein AWP

$$x'(t) = f(t, x(t)), \qquad t \in [t_0, T], \qquad x(t_0) = x_0.$$

Approximation auf Gitter $\Delta = \{t_0, \ldots, t_n\}$ mit

$$t_0 < t_1 < \ldots < t_n = T$$

durch Gitterfunktion x_Δ mit dem Ziel $x_\Delta(t_i) \approx x(t_i)$ für $i = 1, \ldots, n$.

Bisher: Einschrittverfahren

- Berechne $x_\Delta(t_{j+1})$ ausschließlich aus der Kenntnis des vorangegangenen Zustands $x_\Delta(t_j)$.
- Formalisierung: Diskrete Evolution

$$x_\Delta(t_{j+1}) = \Psi^{t_{j+1}, t_j} x_\Delta(t_j).$$

Eigenschaften der Einschrittverfahren:

- Konsistenztheorie eher schwierig
- Konvergenztheorie eher einfach
- Änderung der Zeitschrittweite jederzeit möglich.
- Verfahren hoher Ordnung sind teuer: Für Konsistenzordnung p braucht man (z.B. mit einem explizitien Runge–Kutta-Verfahren) mindestens $s \geq p$ Auswertungen der Funktion f.

Idee der Mehrschrittverfahren: Berechne den neuen Wert $x_\Delta(t_{j+1})$ aus den letzten k Werten, also

$$x_\Delta(t_{j-k+1}), \ldots, x_\Delta(t_j) \mapsto x_\Delta(t_{j+1}).$$

Vorteile:

- nur eine einzige Auswertung von f pro Schritt, für beliebig hohe Ordnung.

Nachteile:

- Konvergenztheorie komplizierter,
- nicht so flexibel bei Änderungen der Schrittweite, schwierig auf nicht uniformen Gittern,
- k-Schritt-Verfahren benötigen neben Startwert $x_\Delta(t_0)$ noch $(k-1)$ zusätzliche Startwerte $x_\Delta(t_1), \ldots, x_\Delta(t_{k-1})$.

6.8.2 Mehrschrittverfahren für äquidistante Gitter

Wir beschränken uns auf äquidistante (auch: uniforme) Gitter

$$t_j = t_0 + j\tau, \qquad j = 0, 1, \ldots, n$$

also

$$\tau = \frac{T - t_0}{n}.$$

Dies ist nicht nur zur Bequemlichkeit. Mehrschrittverfahren auf unregelmäßigen Gittern sind deutlich komplizierter!

Beispiel: Die explizite Mittelpunktsregel

- Integriere die Differentialgleichung über $[t_{j-1}, t_{j+1}]$:

$$x(t_{j+1}) = x(t_{j-1}) + \int_{t_{j-1}}^{t_{j+1}} x'(\sigma)\, d\sigma = x(t_{j-1}) + \int_{t_{j-1}}^{t_{j+1}} f(\sigma, x(\sigma))\, d\sigma$$

- Approximiere das Integral durch die Mittelpunktsregel

$$\int_{t_{j-1}}^{t_{j+1}} f(\sigma, x(\sigma))\, d\sigma = 2\tau f(t_j, x(t_j)) + \mathcal{O}(\tau^3).$$

- Mehrschrittverfahren

$$x_\Delta(t_{j+1}) = x_\Delta(t_{j-1}) + 2\tau f(t_j, x_\Delta(t_j)), \qquad j = 1, \ldots, n-1$$

- Verfahren ist explizit: Die gesuchte Größe $x_\Delta(t_{j+1})$ taucht nur links des Gleichheitszeichens auf.
- Fehler der Integralapproximation ist $\mathcal{O}(\tau^3)$ (für f hinreichend glatt). Wir vermuten deshalb Konsistenzordnung $p = 2$.

Wie bekommt man höhere Konsistenzordnung?

Idee: Nimm genauere Quadraturformel!

Beispiel: Die Simpson-Regel

$$\int_{t_{j-1}}^{t_{j+1}} f(\sigma, x(\sigma))\, d\sigma = \frac{\tau}{3}\Big[f(t_{j+1}, x_(t_{j+1})) + 4f(t_j, x(t_j)) + f(t_{j-1}, x(t_{j-1}))\Big] + \mathcal{O}(\tau^5)$$

- Damit konstruiert man das Milne–Simpson-Verfahren

$$x_\Delta(t_{j+1}) = x_\Delta(t_{j-1}) + \frac{\tau}{3}\Big[f(t_{j+1}, x_\Delta(t_{j+1})) + 4f(t_j, x_\Delta(t_j)) + f(t_{j-1}, x_\Delta(t_{j-1}))\Big]$$

- Konsistenzordnung $p = 4$
- Dennoch nur *eine* f-Auswertung pro Zeitschritt!
- Verfahren ist *implizit*: Zur Bestimmung von $x_\Delta(t_{j+1})$ muss ein Gleichungssystem gelöst werden.

Allgemeine lineare k-Schritt-Verfahren:

- Beide Beispiel-Verfahren sind linear in f.
- Definiere $f_\tau(t_i)$ als Abkürzung von $f(t_i, x_\delta(t_i))$.

Definite das Verfahren

$$\alpha_k x_\tau(t_{j+k}) + \alpha_{k-1} x_\tau(t_{j+k-1}) + \ldots + \alpha_0 x_\tau(t_j) \\ = \tau\Big[\beta_k f_\tau(t_{j+k}) + \beta_{k-1} f_\tau(t_{j+k-1}) + \ldots + \beta_0 f_\tau(t_j)\Big] \quad (6.4)$$

mit $|\alpha_0| + |\beta_0| > 0$ und $\alpha_k \neq 0$, (sonst ist es ein $(k-1)$-Schritt-Vefahren).

- Das Verfahren ist explizit, falls $\beta_k = 0$. Ansonsten ist es implizit.

Existiert eine eindeutige Gitterlösung?

- Explizites lMSV, also $\beta_k = 0$ klar (sogar ohne Einschränkung an τ).

Lemma 6.4. *Sei $\beta_k \neq 0$ und $f : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$ genüge der Lipschitz-Bedingung*

$$\|f(t,x) - f(t,\bar{x})\| \leq L\|x - \bar{x}\| \qquad \forall x, \bar{x} \in \mathbb{R}^d, t \in \mathbb{R}.$$

Dann existiert für $\tau < \frac{|\alpha_k|}{|\beta_k|L}$ zu beliebigen Startwerten $x_\Delta(t_0), \ldots, x_\Delta(t_{k-1})$ eine eindeutige Gitterfunktion x_Δ des lMSV.

Beweis.

- Löse (6.4) nach $x_\Delta(t_{j+k})$ auf:

$$x_\Delta(t_{j+k}) = \tau \frac{\beta_k}{\alpha_k} f(t_{j+k}, x_\Delta(t_{j+k})) + \text{sonstige Terme}$$

- Dies ist eine Fixpunktgleichung.
- $\tau \frac{\beta_k}{\alpha_k} f$ ist Lipschitz-stetig mit Konstante $L^* = \tau \frac{\beta_k}{\alpha_k} L$
- Der Banachsche Fixpunktsatz liefert die Existenz eines eindeutigen Fixpunkts, wenn diese Lipschitzkonstante L^* kleiner 1 ist, wenn also

$$\tau < \frac{|\alpha_k|}{|\beta_k|L}.$$

□

Darstellung durch Polynome: Zum Mehrschrittverfahren (6.4) definiere die Polynome

$$\rho(\xi) = \alpha_k \xi^k + \alpha_{k-1}\xi^{k-1} + \ldots + \alpha_0$$
$$\sigma(\xi) = \beta_k \xi^k + \beta_{k-1}\xi^{k-1} + \ldots + \beta_0.$$

Zu den Beispielen:

- explizite Mittelpunktsregel: $\rho(\xi) = \xi^2 - 1$ und $\sigma(\xi) = 2\xi$
- Milne-Simpson: $\rho(\xi) = \xi^2 - 1$ und $\sigma(\xi) = \frac{1}{3}(\xi^2 + 4\xi + 1)$

6.8.3 Konsistenz

Wir brauchen einen neuen Konsistenzbegriff.

- Beschreibt den Zusammenhang zwischen Differenzengleichung und Differentialgleichung.

Plan:

- Ersetzen von x_τ durch die Funktion x, die die DGL erfüllt.
- Lokaler Diskretisierungsfehler $L(x,t,\tau)$ (hier wegen $f(t_j, x(t_j)) = x'(t_j)$)

$$\begin{aligned} L(x,t,\tau) := \alpha_k x(t+k\tau) + \alpha_{k-1} x(t+(k-1)\tau) + \ldots + \alpha_0 x(t) \\ - \tau\Big[\beta_k x'(t+k\tau) + \beta_{k-1} x'(t+(k-1)\tau) + \ldots + \beta_0 x'(t)\Big] \end{aligned} \quad (6.5)$$

Definition. *Das lMSV* (6.4) *besitzt die Konsistenzordnung p, wenn*

$$L(x,t,\tau) = \mathcal{O}(\tau^{p+1})$$

für alle $x \in C^\infty([t_0,T],\mathbb{R}^d)$ *und* $\tau \to 0$ *gleichmäßig für alle* t, τ *gilt.*

Diese Definition ist eine Verallgemeinerung des bisherigen Konvergenzbegriffs.

Beispiel: Explizites Euler-Verfahren

- Lineares 1-Schritt-Verfahren mit definierenden Polynomen $\rho(\xi) = \xi - 1$ und $\sigma(\xi) = 1$.
- Zugehöriger lokaler Diskretisierungsfehler

$$\begin{aligned} L(x,t,\tau) &= x(t+\tau) - x(t) - \tau x'(t) \\ &= x(t+\tau) - \big[x(t) + \tau f(t, x(t))\big] \\ &= \Phi^{t+\tau,t} x - \Psi^{t+\tau,t} x. \end{aligned}$$

Lemma 6.5. *Das lineare Mehrschrittverfahren* (6.4) *hat genau dann die Konsistenzordnung p, wenn eine der folgenden äquivalenten Bedingungen erfüllt ist:*

1. *Für beliebige* $x \in C^{p+1}([t_0,T],\mathbb{R}^d)$ *gilt* $L(x,t,\tau) = \mathcal{O}(\tau^{p+1})$ *gleichmäßig in allen zulässigen* t *und* τ*.*
2. $L(Q,0,\tau) = 0$ *für alle Polynome* Q *von Grad höchstens* p
3. $L(\exp,0,\tau) = \rho(e^\tau) - \tau\sigma(e^\tau) = \mathcal{O}(\tau^{p+1})$

4. Für alle $l = 1, \ldots, p$ gilt

$$\rho(1) = \sum_{j=0}^{k} \alpha_j = 0, \qquad \sum_{j=0}^{k} \alpha_j j^l = l \sum_{j=0}^{k} \beta_j j^{l-1}$$

(dabei gelte $0^0 = 1$).

Beweis. Wir zeigen 1) $\Longrightarrow$ 2) $\Longrightarrow$ 3) $\Longrightarrow$ 4) $\Longrightarrow$ 1).

1) $\Longrightarrow$ 2):
- Für $Q \in \Pi_p$ ist $L(Q, 0, \tau)$ ein Polynom in τ.
- Dessen Grad ist durch den von Q, also p, beschränkt.
- Gleichzeitig ist $L(Q, 0, \tau) = \mathcal{O}(\tau^{p+1})$.
- Also ist $L(Q, 0, \tau) = 0$.

- 2) $\Longrightarrow$ 3)
$$\exp(\tau) = Q(\tau) + \mathcal{O}(\tau^{p+1})$$
mit Q Taylorpolynom von exp an der Stelle 0.
Aus der Linearität von L bzgl. des ersten Arguments folgt
$$L(\exp, 0, \tau) = L(Q, 0, \tau) + \mathcal{O}(\tau^{p+1})$$
Wegen 2) ist $L(Q, 0, \tau) = 0$

- 3) $\Longrightarrow$ 4) Taylor-Entwicklung (nach τ) an der Stelle 0 liefert
$$\begin{aligned} L(\exp, 0, \tau) &= \sum_{l=0}^{p} \frac{1}{l!} \sum_{j=0}^{k} \alpha_j j^l \tau^l - \sum_{l=0}^{p-1} \frac{1}{l!} \sum_{j=0}^{k} \beta_j j^l \tau^{l+1} + \mathcal{O}(\tau^{p+1}) \\ &= \sum_{l=0}^{p} \frac{1}{l!} \sum_{j=0}^{k} \alpha_j j^l \tau^l - \sum_{l=1}^{p} \frac{1}{(l-1)!} \sum_{j=0}^{k} \beta_j j^{l-1} \tau^l + \mathcal{O}(\tau^{p+1}) \end{aligned}$$
Koeffizientenvergleich bzgl. der τ-Potenzen!

- 4) $\Longrightarrow$ 1) Taylor-Entwicklung (nach τ) an der Stelle 0 liefert für $x \in C^{p+1}$:
$$L(x, t, \tau) = \sum_{l=0}^{p} \frac{1}{l!} \sum_{j=0}^{k} \alpha_j j^l \tau^l x^{(l)}(t) + \mathcal{O}(\tau^{p+1}) - \tau[\sum_{l=0}^{p-1} \frac{1}{l!} \sum_{j=0}^{k} \beta_j j^l \tau^l x^{(l+1)}(t) + \mathcal{O}(\tau^p)]$$
Koeffizientenvergleich liefert mit 4) $(= 0)$, also folgt die Aussage. □

Beispiel: $p = 0$.

- Wegen ii) bedeutet das, dass konstante Funktionen exakt behandelt werden.

- D.h., das Anfangswertproblem

$$x' = 0, \qquad x(t_0) = x_0$$

  wird exakt gelöst.

- Nach iv) ist die dazugehörige Bedingungsgleichung

$$\sum_{j=0}^{k} \alpha_j = 0, \qquad \text{bzw.} \qquad \rho(1) = 0.$$

Beispiel: $p = 1$.

- Zusätzlich die Bedingungsgleichung

$$\sum_{j=0}^{k} \alpha_j j = \sum_{j=0}^{k} \beta_k$$

- In Polynomform

$$\rho'(1) = \sigma(1).$$

Diese Bedingungen entsprechen dem $\sum_{j=1}^{s} b_j = 1$ bei Runge–Kutta-Verfahren.

6.8.4 Stabilität

Bei Einschrittverfahren galt:

- Wenn ein Verfahren mit einer gewissen Ordnung konsistent ist, dann konvergiert es auch mit dieser Ordnung.

So einfach ist es bei Mehrschrittverfahren leider nicht!

Beispiel. Betrachte das (bis auf Normierung $\alpha_k = 1$) eindeutige explizite Zweischrittverfahren der Konsistenzordnung $p = 3$

$$\rho(\xi) = \xi^2 + 4\xi - 5, \qquad \sigma(\xi) = 4\xi + 2. \tag{6.6}$$

- Anwendung auf

$$\dot{x}(t) = 0, \qquad x(0) = 1$$

  mit den Startwerten

$$x_\Delta(0) = 1, \qquad x_\Delta(\tau) = 1 + \tau\varepsilon.$$

- Man erhält die Gitterfunktion

$$x_\Delta(n\tau) = 1 + \tau\varepsilon\frac{(1 - (-5)^n)}{6}$$

  (Beachte: 1 und -5 sind gerade die Nullstellen von ρ!)

- Es folgt für alle $\varepsilon \neq 0$ dass
$$\lim_{n\to\infty} |x_\Delta(n\tau)| = \infty$$
- obwohl für den Startwert gilt, dass
$$\lim_{\tau\to 0} x_\Delta(\tau) = 1$$
und die exakte Lösung $x(t) = 1$ beliebig glatt ist.

Das heißt: keine Konvergenz, sogar ganz anderes Verhalten.

- Das Mehrschrittverfahren ist instabil.
- Es ist komplett unbenutzbar!

Definition. *Ein lineares Mehrschrittverfahren heißt stabil (oder nullstabil oder D-stabil), wenn die lineare homogene Differenzengleichung*

$$\sum_{j=0}^{k} \alpha_j x_\tau(t_{j+k}) = 0, \qquad k = 0, 1, \ldots$$

bei beliebigen Startwerten stabil ist. Das wiederum heißt: Die Folge bleibt für alle Startwerte beschränkt.

Beachte: Die homogene Differenzengleichung ist gerade das Mehrschrittverfahren, angewandt auf $x' = 0$.

Satz 6.6 (Dahlquistsche Wurzelbedingung)**.** *Ein lineares Mehrschrittverfahren ist genau dann stabil, wenn die Nullstellen ξ von ρ die Dahlquist'sche Wurzelbedingung erfüllen, d.h.*

- $|\xi| \leq 1$,
- *wenn $|\xi| = 1$, dann ist ξ einfache Nullstelle.*

Beispiel. Für das lMSV (6.6) ist $\rho(\xi) = \xi^2 + 4\xi - 5 = (\xi - 1)(\xi + 5)$, d.h. dieses Verfahren ist nicht stabil.

Beispiel. Für die explizite Mittelpunktsregel sowie für das Milne–Simpson-Verfahren ist $\rho(\xi) = \xi^2 - 1 = (\xi + 1)(\xi - 1)$. Beide sind also stabil.

Beispiel. Einschrittverfahren:

- Sofern sie konsistent sind muss $\rho(1) = 0$, und deshalb $\rho(\xi) = \xi - 1$ gelten.
- Alle konsistenten Einschrittverfahren sind deshalb automatisch stabil.

- Beim Konvergenzbeweis für Einschrittverfahren mussten wir auf Stabilität deshalb gar nicht eingehen!

Wenn man sich die Beziehungsgleichungen in iv) genauer anschaut, erkennt man:

- Es gibt k-Schritt-Verfahren mit Ordnung $2k$, aber keine mit höherer Ordnung.
- Es gibt explizite k-Schritt-Verfahren mit Ordnung $2k-1$, aber keine mit höherer Ordnung.

Wenn man sich auf *stabile* Verfahren beschränkt verliert man in etwa die halbe Ordnung.

Satz 6.7. *Die Konsistenzordnung p eines stabilen linearen k-Schrittverfahrens unterliegt der Beschränkung*

1. *$p \leq k+2$, wenn k gerade ist,*
2. *$p \leq k+1$, wenn k ungerade ist,*
3. *$p \leq k$, wenn $\frac{\beta_k}{\alpha_k} \leq 0$ ist, also insbesondere auch für explizite Verfahren.*

Diese Aussage nennt man die erste Dahlquistsche Schranke.
Man kann zeigen dass sie scharf ist.

6.8.5 Konvergenz

Definition. *Sei $x \in C^1([t_0,T],\mathbb{R}^d)$ die Lösung eines Anfangswertproblems $x' = f(t,x)$, $x(t_0) = x_0$. Ein Mehrschrittverfahren* konvergiert *gegen diese Lösung, wenn*

$$\lim_{\tau \to 0} x_\Delta(t) = x(t) \qquad \text{für alle} \qquad t \in \Delta_\tau \cap [t_0,T]$$

gilt, sobald die Startwerte

$$\lim_{\tau \to 0} x_{\Delta_\tau}(t_0 + j\tau) = x_0 \qquad j = 0,\dots,k-1$$

erfüllen. Wenn ein lineares Mehrschrittverfahren für beliebige Anfangswertprobleme mit hinreichend glatter rechter Seite konvergiert, so heißt es konvergent.

Das Phänomen aus dem obigen Beispiel ist von grundsätzlicher Natur, wie folgender Satz zeigt.

Satz 6.8. *Ein konvergentes lineares Mehrschrittverfahren ist notwendigerweise stabil und konsistent, speziell gilt*

$$\rho'(1) = \sigma(1) \neq 0.$$

Die Umkehrung gilt aber auch:

Satz 6.9. *Ein stabiles und konsistentes lineares Mehrschrittverfahren ist konvergent.*

Dies ist ein gutes Beispiel dessen, was manche den Hauptsatz der Numerik nennen:

Konsistenz und Stabilität implizieren Konvergenz.

Aussagen zur Konvergenzordnung sind natürlich auch möglich, gehen aber über den Rahmen dieser Vorlesung hinaus.

7 Steife Differentialgleichungen und implizite Verfahren

7.1 Steife Differentialgleichungen

Explizites Euler-Verfahren

$$x_{k+1} = x_k + \tau f\left(t_k, x_k\right)$$

- Konvergent mit Ordnung 1.

Löse damit das Anfangswertproblem

$$\dot{x} = \lambda x, \qquad x(0) = 1, \qquad \lambda \in \mathbb{R}.$$

Bilder!

- Keine Überraschungen, falls $\lambda \geq 0$
- Falls $\lambda < 0$, dann produziert das explizite Euler-Verfahren nur dann qualitativ richtige Ergebnisse, falls der Zeitschritt $\tau < 1/|\lambda|$ ist.
- Für $\lambda < 0$, $\tau > 1/|\lambda|$ ist das Verfahren instabil.

Nachrechnen:
Expliziter Euler:

$$x_{k+1} = x_k + \tau f\left(t_k, x_k\right) = x_k + \tau\lambda x_k = \left(1 + \lambda\tau\right) x_k = \left(1 + \lambda\tau\right)^{k+1} x_0$$

Fall 1: $\lambda > 0$

- Lösung $x(t) = \exp\left(\lambda t\right)$ monoton steigend in t
- Diskrete Lösung: monoton steigend in k, da $1 + \lambda\tau > 1$.

Fall 2: $\lambda < 0$

- Lösung $x(t) = \exp\left(\lambda t\right)$ monoton fallend und positiv
- x_k monoton fallend und positiv nur dann, wenn $0 < 1+\lambda\tau < 1 \iff \tau \leq 1/|\lambda|$.

Falls $\lambda < 0$ und $\tau > \frac{1}{|\lambda|}$

- Diskrete Lösung oszilliert.

Falls $\lambda < 0$ und sogar $\tau > \frac{2}{|\lambda|}$

- Diskrete Lösung ist unbeschränkt.

Differentialgleichungen, die dieses Verhalten zeigen, heißen *steif*.

7.1.1 Steifheit und Kondition

Für Euler- und Runge-Kutta-Verfahren hatten wir Konvergenzaussagen der Form

$$\|\varepsilon_\Delta\|_\infty \leq C\tau_\Delta^p$$

bewiesen.

Problem: Diese Aussagen sind *asymptotisch*.

- Der Fehler wird kleiner, wenn wir ein hinreichend kleines τ_Δ weiter verkleinern.
- Die Konstante C ist aber unbekannt: Wir wissen nicht, *wie* klein τ_Δ sein muss, um eine vernünftige Genauigkeit zu erzielen.

Man kann C nur in seltenen Fällen exakt ausrechnen.

Stattdessen: Qualitativ verstehen, wann C groß sein kann.

Angenommen, wir erhalten für ein gegebenes τ_Δ eine gute Approximation der Lösung.

Dann können wir davon ausgehen, dass das nicht zufällig so ist: Für einen leicht gestörten Anfangswert $x_0 + \delta x_0$ erwarten wir dann auch eine gute Approximation der gestörten Lösung.

Erinnerung: Intervallweise Kondition eines AWPs:

$$x' = f(t, x) \qquad x(t_0) = x_0 \qquad t \in [t_0, T].$$

Störung der Eingabedaten $x_0 \mapsto x_0 + \delta x_0$ führt zu einer Störung der Lösung $x(t) \mapsto x(t) + \delta x(t)$ für alle $t \in [t_0, T]$.
Die intervallweise Kondition $\kappa[t_0, T]$ ist die kleinste Zahl, für die

$$\|\delta x\|_\infty \leq \kappa[t_0, T] \cdot \|\delta x_0\|.$$

Analog führen wir eine diskrete Kondition κ_Δ ein: die Auswirkung einer Störung des Anfangswerts auf eine von einem numerischen Verfahren erzeugte Gitterfunktion

$$\|\delta x_\Delta\|_\infty \leq \kappa_\Delta \cdot \|\delta x_0\|.$$

Wenn ein Verfahren für x_0 und $x_0 + \delta x_0$ (mit δx_0 klein) vernünftige Lösungen liefert, dann muss

$$\kappa_\Delta \approx \kappa[t_0, T]$$

gelten.
Umgekehrt bedeutet $\kappa_\Delta \gg \kappa[t_0, T]$ dass das Verfahren völlig unbrauchbar ist, denn es reagiert auf kleine Störungen völlig anders als das eigentliche Problem.

$\Longrightarrow$ Das Gitter ist dann noch zu grob, da für jedes konvergente Verfahren

$$\kappa_\Delta \to \kappa[t_0, T] \qquad \text{für } \tau_\Delta \to 0.$$

Die Beziehung

$$\kappa_\Delta \approx \kappa[t_0, T]$$

ist eine qualitative Minimalforderung an ein Verfahren + Wahl des Zeitschritts.

- Für die bisher vorgestellten Verfahren gibt es Anfangswertprobleme, für die $\kappa_\Delta \approx \kappa[t_0, T]$ erst für sehr kleine τ_Δ gilt.
- Solche Probleme nennt man *steif*.

Ungewöhnlich:

- Es gibt keine mathematisch präzise Definition des Begriffs „steif“.
- Eine Verfahrensklasse klassifiziert die Probleme!

7.1.2 Beispiel: Wieder das Modellproblem

Wieder das Modellproblem

$$x' = \lambda x, \qquad x(0) = 1.$$

Wie ist die Kondition dieses AWPs? Wir betrachten einen etwas allgemeineren Fall

Beispiel. Betrachte das skalare AWP

$$\dot{x} = \lambda\,(x - g(t)) + g'(t) \qquad \text{mit} \qquad x(0) = g(0).$$

- $\lambda \in \mathbb{R}$ ein Parameter
- $g\colon [t_0, \infty) \to \mathbb{R}$ stetig differenzierbar und beschränkt
- Lösung: $x(t) = g(t) \quad \forall t$.

Betrachte jetzt stattdessen einen gestörten Startwert: $x(0) = g(0) + \delta$

- Lösung davon: $x(t) = g(t) + \delta \exp(\lambda t)$ mit $\delta \exp(\lambda t)$ Störung des Resultats

Wie verhält sich die Störung $\delta \exp(\lambda t)$? Drei Fälle:

1. $\lambda > 0$: Die Störung wächst exponentiell mit t. Lösen der Gleichung für große t ist kaum sinnvoll, bzw. sehr schwierig.
2. $\lambda = 0$: Die Störung bleibt für alle t in konstanter Größe erhalten.
3. $\lambda < 0$: Für große t wird die Störung „von alleine“ immer kleiner!

Aus diesem Beispiel folgt mit $g(t) = e^{\lambda t}$

- $\kappa[0,T] = e^{\lambda T}$ falls $\lambda \geq 0$,
- $\kappa[0,T] = 1$ falls $\lambda \leq 0$.

Diskrete Kondition des expliziten Euler-Verfahrens:

$$x_\Delta(t_{j+1}) = (1+\tau\lambda)x_\Delta(t_j) = (1+\tau\lambda)^{j+1}x_0$$

ist linear in x_0, deshalb gilt

$$\kappa_\Delta = \max_{0\leq k\leq n_\Delta - 1} \prod_{j=0}^{k} |1+\tau\lambda|$$

Fall 1: $\lambda \geq 0$. Wegen $1+\tau\lambda \leq e^{\tau\lambda}$ gilt

$$\kappa_\Delta = (1+\tau\lambda)^{n_\Delta} \leq \exp(n_\Delta\tau\lambda) = e^{\lambda T}.$$

Also ist $\kappa_\Delta \approx \kappa[0,T]$, das AWP ist nichtsteif.

Fall 2: $\lambda < 0$

$$\kappa_\Delta = \max_{1\leq k\leq n_\Delta} |1-\tau_\Delta|\lambda||^k.$$

Falls $\tau_\Delta < 2/|\lambda|$ so ist $\kappa_\Delta \leq 1 = \kappa[0,T]$.

Andererseits gilt für $\tau_\Delta \gg 2/|\lambda|$

$$\kappa_\Delta = |1-\tau_\Delta|\lambda||^k \gg 1 = \kappa[0,T].$$

Das Problem ist steif.

7.1.3 Stabilität

Betrachtet man die Auswirkung von Störungen nicht auf ein beschränktes Intervall $[t_0, T]$, sondern für alle Zeiten $[t_0, \infty)$, dann spricht man statt von Kondition meistens von *Stabilität*.

Die obige Dreiteilung ist typisch. Wir machen daraus eine Definition.

Definition. *Sei (t_0, x_0) so, dass $\Phi^{t,t_0}x_0$ für alle $t \geq t_0$ existiert. Die Lösung des AWPs heißt*

2) (Lyapunov)-stabil, falls zu jedem $\varepsilon > 0$ ein $\delta > 0$ existiert, so dass

$$\left\|\Phi^{t,t_0}(x_0+\delta) - \Phi^{t,t_0}x_0\right\| \leq \varepsilon$$

für alle $t \geq t_0$ und $\|x - x_0\| \leq \delta$,

3) asymptotisch stabil, falls es zusätzlich ein $\delta_0 > 0$ gibt, so dass

$$\lim_{t\to\infty} \left\|\Phi^{t,t_0}x - \Phi^{t,t_0}x_0\right\| = 0$$

falls $\|x - x_0\| \leq \delta_0$,

1) instabil, falls weder 2) noch 3) gelten.

Achtung: Dieser Stabilitätsbegriff hat nichts mit der Stabilität von Algorithmen zu tun.

Es kann anspruchsvoll bis zu schwierig sein, die Stabilität von DGLn zu bestimmen.

7.1.4 Das implizite Euler-Verfahren

Expliziter Euler:

$$x_{k+1} = x_k + \tau f\left(t_k, x_k\right)$$

Impliziter Euler:

$$x_{k+1} = x_k + \tau f\left(t_{k+1}, x_{k+1}\right)$$

Implizit bedeutet: In jedem Schritt muss ein Gleichungssystem gelöst werden.

Betrachte wieder das AWP

$$\dot{x} = \lambda x, \qquad x(0) = 1, \qquad \lambda \in \mathbb{R}$$

Implizites Euler-Verfahren:

$$\begin{aligned} x_{k+1} &= x_k + \tau f\left(t_{k+1}, x_{k+1}\right) = x_k + \tau\lambda x_{k+1} \\ \Longrightarrow\ x_{k+1} &= \frac{x_k}{1-\tau\lambda} = \left(\frac{1}{1-\tau\lambda}\right)^{k+1} x_0 \end{aligned}$$

Wenn $\lambda < 0$, so ist

$$0 < \frac{1}{1-\tau\lambda} < 1$$

für alle $\tau > 0$. Das Verfahren ist unbedingt stabil.

7.2 Stabilität von Einschrittverfahren

Das explizite Euler-Verfahren wird für die lineare Gleichung

$$x' = \lambda x, \qquad x(t_0) = x_0$$

instabil, wenn $\lambda < 0$ und der Zeitschritt τ zu groß ist.

Wir verallgemeinern das jetzt und betrachten lineare, autonome, homogene Systeme

$$x' = Ax, \qquad x(0) = x_0 \in \mathbb{R}^d, \qquad A \in \mathbb{R}^{d\times d}$$

Satz 7.1. *Die Lösung dieses AWPs ist*

$$x(t) = \exp(tA)x_0$$

wobei

$$\exp(tA) := \sum_{k=0}^{\infty} \frac{(tA)^k}{k!}$$

Diese Reihe konvergiert gleichmäßig auf jedem kompakten Zeitintervall.

Stabilität heißt: Störungen im Startwert führen auf beschränkte Störungen in der Lösung (für $t \longrightarrow \infty$).

- Für lineare Gleichungen $\dot{x} = Ax, \qquad x(0) = x_0 + \delta_0$ ist die Lösung

$$x_\delta(t) = \exp(tA)\,(x_0 + \delta_0) = \exp(tA)x_0 + \exp(tA)\delta_0$$

  d.h. die Störung löst das AWP $\dot{x} = Ax, \qquad x(0) = \delta_0$

Lemma 7.1 (Deuflhard und Bornemann [1, Lemma 3.20])**.** *Die Lösung x eines* linearen*, homogenen AWPs ist genau dann stabil, wenn*

$$\sup_{t \geq 0} \|x(t)\| < \infty$$

Sie ist asymptotisch stabil, falls $\|x(t)\| \xrightarrow{t\to\infty} 0$

$\dot{x} = \lambda x$ ist stabil, wenn $\lambda \leq 0$ und asymptotisch stabil, wenn $\lambda < 0$.

Satz 7.2 (Deuflhard und Bornemann [1, Satz 3.23])**.** *Die Lösung des AWPs*

$$\dot{x} = Ax, \qquad x(0) = x_0, \qquad A \in \mathbb{C}^{d\times d}$$

ist genau dann stabil, wenn

- *der Realteil aller Eigenwerte nicht positiv ist und*
- *falls λ ein Eigenwert von A mit* $\mathrm{Re}(\lambda) = 0$*, so hat λ die gleiche algebraische und geometrische Vielfachheit.*

Die Lösung ist asymptotisch stabil, falls $\mathrm{Re}(\lambda) < 0$ *für alle Eigenwerte λ von A.*

Wunsch: Die von einem numerischen Verfahren erzeugte Folge x_k soll diese Stabilitätseigenschaften *erben*. Beim expliziten Euler-Verfahren:

$$x_{k+1} = \Psi^\tau x_k = x_k + \tau A x_k = (I + \tau A)\, x_k$$

war das nicht der Fall. Beim impliziten Euler-Verfahren

$$x_{k+1} = x_k + \tau A x_{k+1} \implies x_{k+1} = (I - \tau A)^{-1}\, x_k$$

jedoch schon! Verallgemeinere: Betrachte Verfahren der Form

$$x_{k+1} = \Psi^\tau x_k = R(\tau A)\, x_k$$

mit P, Q Polynomen, sodass

$$R(\tau A) = \frac{P(\tau A)}{Q(\tau A)}$$

x_{k+1} berechnet sich als Lösung des linearen(!) Gleichungssystems

$$Q(\tau A)x_{k+1} = Q(\tau A)(\Psi^\tau x_k) = P(\tau A)x_k$$

Die rationalen Funktionen werden als Approximationen der Evolution $\Phi^\tau = \exp(tA)$ verwendet. Die Funktion R heißt Stabilitätsfunktion

$$R\colon \mathbb{C}^{d\times d} \to \mathbb{C}^{d\times d}$$
$$R\colon \mathbb{C} \to \mathbb{C}$$

Definition. *Die Konsistenzordnung einer durch die rationale Funktion R gegebenen Approximation der Exponentialfunktion ist die größte ganze Zahl p, sodass*

$$R(z) = \exp(z) + \mathcal{O}\left(z^{p+1}\right)$$

für $z \to 0$ in $\mathbb{C}$.

Lemma 7.2. *Die Flüsse aller expliziten Runge-Kutta-Verfahren (für lineare Gleichungen) sind Polynome in τA, also $\Psi^\tau x = P(\tau A)\, x$.*

Beweis. Wir zeigen dass für jedes $i \le s$ der Ausdruck τk_i ein formales Polynom in τA ist.

Dann folgt die Behauptung aus

$$\Psi^{t+\tau,t}x = x + \tau\sum_{i=1}^{s} b_i k_i = (\tau A)^0 x + \sum_{i=1}^{s} b_i \tau k_i.$$

Beweis mit vollständiger Induktion: RK-Verfahren für $x' = Ax$:

$$k_i = f(t + c_i\tau, x + \tau\sum_{j=1}^{i-1} a_{ij}k_j) = A\Big[x + \tau\sum_{j=1}^{i-1} a_{ij}k_j\Big]$$

- $\tau k_1 = \tau A x$ ist Polynom in τA.
- Seien τk_j Polynome für alle $j < 1$. Dann ist

$$k_i = Ax + \sum_{j+1}^{i-1} a_{ij}\tau k_j$$

Polynom in τA. □

7.2.1 Spektren rationaler Funktionen von Matrizen

Stabilitätseigenschaften linearer Systeme werden häufig über Eigenwerte ausgedrückt.

Damit $R(A)$ wohldefiniert ist muss also $Q(A)$ invertierbar sein.

- $Q(A)$ darf also nicht den Eigenwert Null haben.

Verallgemeinerung der entsprechenden Bedingung für rationale Funktionen in $\mathbb{C}$:

Lemma 7.3. *Eine rationale Funktion $r : z \mapsto \frac{p(z)}{q(z)}$ ist genau dort nicht definiert (bzw. hat genau dort Polstellen), wo $q(z) = 0$ ist.*

Satz 7.3 (Deuflhard und Bornemann [1, Satz 3.42])**.** *Für eine Matrix $A \in \mathbb{C}^{d\times d}$ ist $R(A)$ genau dann definiert, wenn kein Eigenwert von A Pol von R ist.*

Der Zusammenhang zwischen den Eigenwerten von R und A ist aber noch viel enger:

Satz 7.4 (Deuflhard und Bornemann [1, Satz 3.42, Forts.])**.** *Sei $\sigma(A)$ das Spektrum von A. Dann ist*

$$\sigma(R(A)) = R(\sigma(A)).$$

7.2.2 Wann sind Einschrittverfahren R stabil?

Satz 7.5 (Deuflhard und Bornemann [1, Satz 3.33])**.** *Die lineare Iteration $x_{k+1} = Bx_k$ mit $B \in \mathbb{C}^{d\times d}$ ist genau dann stabil, wenn*

- *$|\lambda| \leq 1$ für alle Eigenwerte λ von B und*
- *Falls λ Eigenwert von B mit $|\lambda| = 1$, so hat λ gleiche algebraische und geometrische Vielfachheit.*

Die Iteration ist asymptotisch stabil, falls $|\lambda| < 1$ für alle Eigenwerte λ von B.

Spektralradius: $\rho(B) = \max_{\lambda\in\sigma(B)} |\lambda|$ mit $\sigma(B) = \{$ Menge aller Eigenwerte $\}$. Verfahren ist stabil, dann $\rho\left(R\left(\tau A\right)\right) \leq 1$.

- Dabei gilt $\rho\left(R\left(\tau A\right)\right) = \max_{\lambda\in\sigma(A)} \left|R\left(\tau\lambda\right)\right|$

Definition. *Die Menge*

$$S = \{z \in \mathbb{C} \mid |R(z)| \leq 1\}$$

heißt Stabilitätsgebiet von R.

Beispiel. Explizites Euler-Verfahren:

$$x_{k+1} = \underbrace{\left(I + \tau A\right)}_{R(\tau A)} x_k \implies R(z) = 1 + z$$

Stabilitätsgebiet: $S = \{z \in \mathbb{C}\colon\ |1 + z| \leq 1|$.

Damit ein Verfahren stabil ist muss $\tau\lambda \in S$ für alle $\lambda \in \sigma(A)$ sein. Im Falle der skalaren Gleichung mit $\mathbb{R} \ni \lambda < 0$ und dem Euler-Verfahren führt das auf die Bedingung $\tau \leq \frac{2}{|\lambda|}$.
Für eine graphische Darstellung der Stabilitätsgebiete expliziter Runge-Kutta-Verfahren siehe Deuflhard & Bornemann, Numerische Mathematik 2, Seite 238.
Folgendes Detail sticht ins Auge.

Lemma 7.4 (Deuflhard und Bornemann [1, Lemma 6.5])**.** *Für jede konsistente rationale Approximation von* exp *gilt* $0 \in \partial S$.

Also:

- Die Lösung $x(t) = \exp(tA)x_0$ ist stabil, wenn alle Eigenwerte von A in der linken Halbebene von $\mathbb{C}$ liegen.
- Ein numerisches Verfahren $\Psi^\tau = R(\tau A)$ ist stabil, wenn alle Eigenwerte von τA in S liegen (plus Zusatzbedingungen am Rand von S).

Folgende Eigenschaft ist deshalb wünschenswert

Definition. *Ein Einschrittverfahren heißt A-stabil, falls sein Stabilitätsgebiet die negative komplexe Halbebene enthält.*

In diesem Fall gibt es keine Schrittweitenbeschränkung! Explizite Verfahren können aber nicht A-stabil sein.

Lemma 7.5 (Deuflhard und Bornemann [1, Lemma 6.11])**.** *Das Stabilitätsgebiet von Polynomen ist kompakt.*

Beweis. Für jedes Polynom P vom Grad ≥ 1 gilt $|P(z)| \xrightarrow{z\to\infty} \infty$. Also ist S beschränkt. □

Implizite Verfahren können A-stabil sein: z.B. Implizites Euler-Verfahren:

$$R(z) = \frac{1}{1-z}$$

$S = \{z \in \mathbb{C} \mid |1-z| \geq 1\}$. Stabilitätsgebiet S^C. Das Ziel für die Zukunft lautet also, A-stabile Verfahren hoher Ordnung zu konstruieren.

8 Hamilton-Systeme

8.1 Variationelle Integratoren

8.1.1 Variationelle Integratoren höherer Ordnung

Wie können wir variationelle Integratoren höherer Ordnung konstruieren?

Bessere Approximation von

$$L_h(q_k, q_{k+1}) :\approx \int_{t_k}^{t_{k+1}} L(q(t), \dot{q}(t))\, dt$$

heißt:

- Quadraturformel höherer Ordnung
- Approximation von q höherer Ordnung.

Idee: (Marsden und West [5])

$$L_h(q_k, q_{k+1}) := \tau \sum_{i=1}^{s} b_i L(u(c_i\tau), \dot{u}(c_i\tau)) \tag{8.1}$$

- Quadraturformel mit s Stützstellen $c_1, \ldots, c_s$, Gewichten $b_1, \ldots, b_s$
- u ist Polynom vom Grad höchstens s mit
  - $u(0) = q_k, \qquad u(\tau) = q_{k+1}$
  - u macht die rechte Seite von (8.1) stationär.

Tatsächlich werden von u nur die Werte und Ableitungen an den Stützstellen $c_i\tau$ gebraucht.

Definiere deshalb:

$$Q_i := u(c_i\tau), \qquad \dot{Q}_i := \dot{u}(c_i\tau).$$

Die Q_i können durch die $\dot{Q}_i$ ausgedrückt werden:

$$\begin{aligned} Q_i = u(c_i\tau) &= u(0) + \int_0^{c_i} \dot{u}(\sigma\tau)\, d\sigma \\ &= q_k + \tau \int_0^{c_i} \sum_{j=1}^{s} L_j(\sigma)\dot{u}(c_j\tau)\, d\sigma \quad \text{(Lagrange–Darstellung)} \\ &= q_k + \tau \sum_{j=1}^{s} a_{ij}\dot{Q}_i \quad \text{mit} \quad a_{ij} = \int_0^{c_i} L_j(\sigma)\, d\sigma \end{aligned} \tag{8.2}$$

Setze außerdem

$$b_i = \int_0^1 c_i(\sigma)\, d\sigma.$$

Wir wählen die $\dot{Q}_i$ so, dass der Ausdruck

$$L_h(q_0, q_1) = \tau \sum_{i=1}^{s} b_i L(Q_i, \dot{Q}_i)$$

stationär wird.

Allerdings brauchen wir zusätzlich die Nebenbedingung

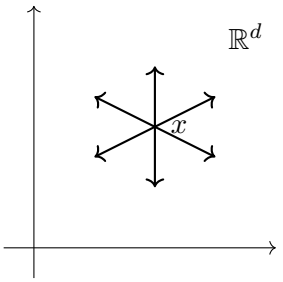
$$q_1 = u(\tau) = q_0 + \tau \sum_{i=1}^{s} b_i Q_i. \tag{8.3}$$

Exkurs: Stationärität unter Gleichheitsnebenbedingungen

Betrachte eine Funktion $f : \mathbb{R}^d \to \mathbb{R}$.

- Wir suchen einen stationären Punkt von f
- D.h., einen Punkt x, in dem die Richtungsableitung in alle Richtungen v verschwindet

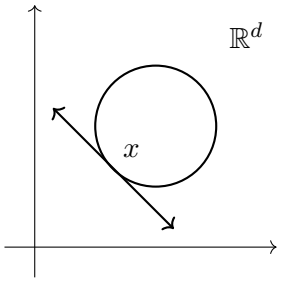
$$\frac{df}{dv} = 0 \quad \forall v \in \mathbb{R}^d,\ v \neq 0.$$



- D.h. $\nabla f(x) = 0$

Betrachte jetzt eine weitere Funktion $g : \mathbb{R}^d \to \mathbb{R}$.

- Wir suchen ein x mit $g(x) = 0$, so dass f in x stationär ist bzgl. der Menge $M = \{y \in \mathbb{R}^d \ : \ g(y) = 0\}$.
- D.h. $\frac{df}{dv} = 0$ for alle Richtungen v, die tangential zu M sind.



- $\nabla f(x)$ ist nicht zwangsläufig null!
- Aber $\nabla f(x)$ steht senkrecht auf M.
- $\nabla g(x)$ steht ebenfalls senkrecht auf M.
- Gesucht werden $x \in \mathbb{R}^d$, $\lambda \in \mathbb{R}$, so dass

$$\nabla f(x) = \lambda \nabla g(x).$$

  Solch eine Variable λ heißt *Lagrange–Multiplikator*.

Umschreiben: Definiere die Lagrange–Funktion

$$\mathcal{L}(x, \lambda) := f(x) - \lambda g(x).$$

- Der Gradient davon ist

$$\nabla \mathcal{L}(x, y) = \begin{pmatrix} \frac{\partial \mathcal{L}}{\partial x} \\ \frac{\partial \mathcal{L}}{\partial \lambda} \end{pmatrix} = \begin{pmatrix} \nabla f(x) - \lambda \nabla g(x) \\ -g(x) \end{pmatrix}$$

- Die gesuchten Punkte sind also gerade die stationären Punkte von $\mathcal{L}$ (ohne Nebenbedingungen).

Exkurs Ende

Diese Technik wenden wir auf die Nebenbedingung

$$q_1 = q_0 + \tau \sum_{i=1}^{s} b_i \dot{Q}_i$$

an. Da diese Nebenbedingung d-wertig ist, ist auch der Lagrange–Multiplikator λ aus $\mathbb{R}^d$.

Wir suchen also stationäre Punkte von

$$\mathcal{L}(\dot{Q}_1, \ldots, \dot{Q}_d, \lambda) = \tau \sum_{i=1}^{s} b_i L(Q_i, \dot{Q}_i) - \Big\langle \lambda, \Big(q_0 - q_1 + \tau \sum_{i=1}^{s} b_i \dot{Q}_i \Big) \Big\rangle.$$

Wir berechnen hiervon die partiellen Ableitungen nach den $\dot{Q}_j$ (die part. Ableitungen nach λ sind klar).

$$\frac{\partial \mathcal{L}}{\partial \dot{Q}_j} = \tau \sum_{i=1}^{s} b_i \left[\frac{\partial L}{\partial q} \cdot \frac{\partial Q_i}{\partial \dot{Q}_j} + \frac{\partial L}{\partial \dot{q}} \cdot \frac{\partial \dot{Q}_i}{\partial \dot{Q}_j} \right] - \underbrace{\lambda^T \left(\tau \sum_{i=1}^{s} b_i \frac{\partial \dot{Q}_i}{\partial \dot{Q}_j} \right)}_{=\tau b_j \lambda}.$$

Da

$$\frac{\partial Q_i}{\partial \dot{Q}_j} = \frac{\partial}{\partial \dot{Q}_j} \left(q_0 + \tau \sum_{k=1}^{s} a_{ik} \dot{Q}_k \right) = \tau a_{ij} I_{d \times d}$$

folgt

$$\frac{\partial \mathcal{L}}{\dot{Q}_j} = \tau \sum_{i=1}^{s} b_i \frac{\partial L}{\partial q} \cdot \tau a_{ij} + \tau b_j \frac{\partial L}{\partial \dot{q}} - \tau b_j \lambda.$$

Stationäre Punkte von $\mathcal{L}$ erfüllen also

$$\sum_{i=1}^{s} b_i \frac{\partial L}{\partial q}(Q_i, \dot{Q}_i) \cdot \tau a_{ij} + b_j \frac{\partial L}{\partial \dot{q}}(Q_j, \dot{Q}_j) = b_j \lambda.$$

Wir führen wieder die konjugierten Impulse ein:

$$P_i = \frac{\partial L}{\partial \dot{q}}(Q_i, \dot{Q}_i), \qquad \text{außerdem} \quad \dot{P}_i = \frac{\partial L}{\partial q}(Q_i, \dot{Q}_i).$$

Damit vereinfacht sich die Bedingung (8.1.1) zu

$$\tau \sum_{i=1}^{s} b_i \dot{P}_i a_{ij} + b_j P_j = b_j \lambda. \tag{8.4}$$

Das allgemeine variationelle Integrationsverfahren hatte die Form

$$p_0 = -\frac{\partial L_h}{\partial x}(q_0, q_1), \qquad p_1 = \frac{\partial L_h}{\partial y}(q_0, q_1).$$

Das rechnen wir jetzt für den konkreten Fall aus.

$$\begin{aligned}
p_0 &= -\frac{\partial L_h}{\partial q_0}(q_0, q_1) \\
&= -\tau \sum_{i=1}^{s} b_i \frac{\partial}{\partial q_0} L(Q_i, \dot{Q}_i) \\
&= -\tau \sum_{i=1}^{s} b_i \left[\underbrace{\frac{\partial}{\partial x} L(Q_i, \dot{Q}_i)}_{=\dot{P}_i} \cdot \frac{\partial Q_i}{\partial q_0} + \underbrace{\frac{\partial}{\partial y} L(Q_i, \dot{Q}_i)}_{=P_i} \cdot \frac{\partial \dot{Q}_i}{\partial q_0} \right] \\
&= -\tau \sum_{i=1}^{s} b_i \left[\dot{P}_i \left(I + \tau \sum_{j=1}^{s} a_{ij} \frac{\partial \dot{Q}_j}{\partial q_0} \right) + P_i \frac{\partial \dot{Q}_i}{\partial q_0} \right] \quad (\text{wg. } Q_i = q_0 + \tau \sum_{j=1}^{s} a_{ij} \dot{Q}_j) \\
&= -\tau \sum_{i=1}^{s} b_i \dot{P}_i - \tau \sum_{j=1}^{s} \underbrace{\tau \sum_{i=1}^{s} b_i \dot{P}_i a_{ij}}_{=b_j \lambda - b_j P_j} \frac{\partial \dot{Q}_j}{\partial q_0} - \tau \sum_{i=1}^{s} b_i P_i \frac{\partial \dot{Q}_i}{\partial q_0} \\
&= -\tau \sum_{i=1}^{s} b_i \dot{P}_i - \tau \sum_{j=1}^{s} (b_j \lambda - b_j P_j) \frac{\partial \dot{Q}_j}{\partial q_0} - \tau \sum_{i=1}^{s} b_i P_i \frac{\partial \dot{Q}_i}{\partial q_0} \\
&= -\tau \sum_{i=1}^{s} b_i \dot{P}_i - \tau \sum_{j=1}^{s} b_j \lambda \frac{\partial \dot{Q}_j}{\partial q_0}
\end{aligned}$$

Differenzieren der Nebenbedingung $q_1 = q_0 + \tau \sum_{i=1}^{s} b_i \dot{Q}_i$ ergibt

$$0 = I + \tau \sum_{i=1}^{s} b_i \frac{\partial \dot{Q}_i}{\partial q_0}.$$

Deshalb ist

$$p_0 = -\tau \sum_{i=1}^{s} b_i \dot{P}_i + \lambda. \tag{8.5}$$

Ganz ähnlich erhält man

$$p_1 = \frac{\partial L_h}{\partial y}(q_0, q_1) = \lambda. \tag{8.6}$$

Lemma 8.1. *Es gilt*

1. $p_1 = p_0 + \tau \sum_{i=1}^{s} b_i \dot{P}_i$
2. $q_1 = q_0 + \tau \sum_{i=1}^{s} b_i \dot{Q}_i$
3. $P_i = p_0 + \tau \sum_{j=1}^{s} \underbrace{(b_j - b_j a_{ji}/b_i)}_{=:\hat{a}_{ij}} \dot{P}_i$

4. $Q_i = q_0 + \tau \sum_{j=1}^{s} a_{ij} \dot{Q}_j$

Beweis. 1. ist (8.5) mit (8.6)

2. ist gerade die Nebenbedingung (8.3), d.h. $u(\tau) = q_1$.

3. Aus 1. und $p_1 = \lambda$ folgt

$$0 = p_0 + \tau \sum_{j=1}^{s} b_j \dot{P}_j - \lambda.$$

Multiplizieren mit einem b_i ($\neq 0$):

$$0 = b_i p_0 + \tau \sum_{j=1}^{s} b_i b_j \dot{P}_j - b_i \lambda$$

Addiere $b_i P_i$ auf beiden Seiten:

$$b_i P_i = b_i p_0 + \tau \sum_{j=1}^{s} b_i b_j \dot{P}_j + \underbrace{b_i P_i - b_i \lambda}_{\substack{= -\tau \sum_{j=1}^{s} b_j a_{ji} \dot{P}_j \\ \text{(wg. (8.4))}}}$$

$$\Longrightarrow \; b_i P_i = b_i p_0 + \tau \sum_{j=1}^{s} (b_j - b_j a_{ji}) \dot{P}_j$$

4. ist (8.2) (Darstellung der Werte Q über Hauptsatz und Lagrange–Darstellung)

□

Die vier Gleichungen aus dem Lemma bilden ein partitioniertes Runge–Kutta-Verfahren $(p_0, q_0) \mapsto (p_1, q_1)$ für die Gleichungen

$$\dot{p} = \frac{\partial L}{\partial q}(q, \dot{q}), \qquad p = \frac{\partial L}{\partial \dot{q}}(q, \dot{q}) \qquad \left(\text{bzw. } \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) = \frac{\partial L}{\partial q} \right).$$

Erinnerung: Partitionierte RK-Verfahren für ein System

$$\begin{aligned} \dot{y} = f(y, z) \quad & \dot{z} = g(y, z) \\ y_1 = y_0 + \tau \sum_{i=1}^{s} \hat{b}_i k_i \quad & z_1 = z_0 + \tau \sum_{i=1}^{s} b_i l_i \\ k_i = f(y_0 + \tau \sum_{j=1}^{s} \hat{a}_{ij} k_j, z_0 + \tau \sum_{j=1}^{s} a_{ij} l_j) \quad & l_i = g(y_0 + \tau \sum_{j=1}^{s} \hat{a}_{ij} k_j, z_0 + \tau \sum_{j=1}^{s} a_{ij} l_j) \end{aligned}$$

Symmetrische Form

$$\begin{aligned} y_1 &= y_0 + \tau \sum_{i=1}^{s} \hat{b}_i f(g_i, h_i) & z_1 &= z_0 + \tau \sum_{i=1}^{s} b_i g(g_i, h_i) \\ g_i &= y_0 + \tau \sum_{i=1}^{s} \hat{a}_{ij} f(g_i, h_i) & h_i &= z_0 + \tau \sum_{i=1}^{s} a_{ij} g(g_i, h_i) \end{aligned}$$

Wir haben ein Verfahren dieser Bauart, mit

$$\begin{aligned} \dot{P}_i &= f(g_i, h_i) & \dot{Q}_i &= g(g_i, h_i) \\ P_i &= g_i & Q_i &= h_i \\ & & \hat{b}_i &= b_i. \end{aligned}$$

Literatur

[1] P. Deuflhard und F. Bornemann. *Numerische Mathematik 2 – Gewöhnliche Differentialgleichungen*. de Gruyter, 2008.

[2] I. S. Duff und J. K. Reid. "The Multifrontal Solution of Indefinite Sparse Symmetric Linear Equations". In: *ACM Transactions on Mathematical Software (TOMS)* 9.3 (1983), S. 302–325.

[3] M. R. Hestenes und E. Stiefel. "Methods of Conjugate Gradients for Solving Linear Systems". In: *Journal of Research of the National Bureau of Standards* 49 (1952), S. 409–436.

[4] J. Liu. "The Multifrontal Method for Sparse Matrix Solution: Theory and Practice". In: *SIAM Review* 34.1 (1992), S. 82–109.

[5] J. E. Marsden und M. West. "Discrete mechanics and variational integrators". In: *Acta Numerica* 10 (2001), S. 357–514. DOI: 10.1017/S096249290100006X.

[6] J. R. Shewchuk. *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. Techn. Ber. Pittsburgh, PA, USA, 1994.