Numeric

Ruben Triwari

Inhaltsverzeichnis

Kapitel 1

Normen und Konditionszahl

1: Matrixnormen

Sei $A \in \mathbb{K}^{N\times N}$ beliebig.
Die Spaltensummennorm ist die maximale Summe der Spalten einer Matrix.

$$||A||_1 = \max_{n=1,2,\dots,N} \sum_{m=1}^{N} |a_{m,n}|$$

Die Zeilensummennorm ist die maximale Summe der Zeilen einer Matrix.

$$||A||_\infty = \max_{n=1,2,\dots,N} \sum_{m=1}^{N} |a_{n,m}|$$

Die Frobeniusnorm ist die Wurzel der Quadratsumme über alle Einträge einer Matrix.

$$||A||_F = \sqrt{\sum_{m=1}^{N} \sum_{n=1}^{N} |a_{n,m}|^2} = \sum_{n=1}^{N} \sqrt{\lambda_n(A^*A)}$$

Die zweite Gleichheit gilt, da

$$\texttt{spur}(A^*A) = \sum_{n=1}^{N} \sum_{m=1}^{N} \overline{a_{n,m}} a_{n,m} = \sum_{n=1}^{N} \sum_{m=1}^{N} |a_{n,m}|^2 = ||A||_F^2$$

Spektralnorm:

$$||A||_2 = \max_{n=1,2,\dots,N} \sqrt{\lambda_n(A^*A)}$$

2: Induzierte Matrixnorm

Sei $||\cdot||$ eine Norm auf $\mathbb{K}^N$, dann ist die induzierte Matrixnorm:

$$||A|| = \sup_{0\neq x\in\mathbb{K}^N} \frac{||Ax||}{||x||} = \sup_{||x||=1} ||Ax||.$$

Es gilt die Submultiplikativität:

$$||A\cdot B|| \leqslant ||A||\cdot||B||.$$

Sowie:

$$||\mathbb{1}|| = \sup_{||x||=1} ||\mathbb{1}x|| = \sup_{||x||=1} ||x|| = 1.$$

Wichtige induzierte Matrixnomen:

$$||x||_1 \leftrightarrow ||A||_1$$

$$||x||_\infty \leftrightarrow ||A||_\infty$$

$$||x||_2 \leftrightarrow ||A||_2$$

3: Konditionszahl

Die Konditionszahl ist der Fehlerverstärker falls eine Störung in einer regulären Matrix $A \in \mathbb{K}^{N\times N}$ in einen linearen Gleichungssystem vorkommt. Sei $||\cdot||$ eine Matrixnorm, dann setze:

$$\texttt{cond}(A) = ||A||\cdot||A^{-1}||.$$

Es gilt $\forall\alpha \in \mathbb{K}\setminus\{0\}$:

$$\texttt{cond}(\alpha A) = ||\alpha A||\cdot||(\alpha A)^{-1}|| = |\alpha\cdot\alpha^{-1}|\cdot||A||\cdot||A^{-1}|| = \texttt{cond}(A)$$

Falls $||\cdot||$ eine induzierte Matrixnorm ist, gilt wegen Submultiplikativit:

$$\texttt{cond}(A) = ||A||\cdot||A^{-1}|| \geq ||A\cdot A^{-1}|| = 1$$

Außerdem für eine induzierte Matrixnorm:

$$\texttt{cond}(A) = \frac{\sup_{||x||=1}||Ax||}{\inf_{||x||=1}||Ax||}$$

Daraus folgt für die Spektralnorm:

$$\texttt{cond}(A) = \sqrt{\frac{\max_n \lambda_n(A^*A)}{\min_n \lambda_n(A^*A)}}$$

4: Fehlerabschätzungen für lineare Gleichungssysteme

Sei $A, \Delta A \in \mathbb{K}^{N\times N}$ und $b, \Delta b, x, \Delta x \in \mathbb{K}^N$.
Mit $Ax = b$ und $A(x + \Delta x) = b + \Delta b$ gilt dann:

$$||\Delta x|| \leqslant ||A^{-1}|| \cdot ||\Delta b|| \quad \text{und} \quad \frac{||\Delta x||}{||x||} \leqslant \texttt{cond}(A)\frac{||\Delta b||}{||b||}$$

Dies motiviert die Konditionszahl.
Falls nun $||\Delta A|| < \frac{1}{||A^{-1}||}$ gilt dann $A + \Delta A$ regulär, mit

$$||(A + \Delta A)^{-1}|| \leqslant ||A^{-1}|| \frac{1}{1 - ||\Delta A|| \cdot ||A^{-1}||}$$

Daraus erhält man falls weiter $||\Delta A|| < \frac{1}{||A^{-1}||}$ gilt und mit

$$(A + \Delta A)(x + \Delta x) = b + \Delta b.$$

$$\frac{||\Delta x||}{||x||} \leqslant \texttt{cond}(A)\frac{1}{1 - ||\Delta A|| \cdot ||A^{-1}||}(\frac{\Delta A}{A} + \frac{\Delta b}{b})$$

5: Strikt diagonaldominante Matrizen

Sei $A \in \mathbb{K}^{N\times N}$ dann heißt die Matrix strikt diagonaldominant, wenn für alle $n = 1, 2, \ldots, N$ gilt:

$$\sum_{m=1}^{N} |a_{n,m}| < |a_{n,n}|$$

Eigenschaften falls A strikt diagonaldominant:

- A regulär
- So ist bei andwendung der L-R Zerlegung $a_{n,n}^{(n)} \neq 0 \; \forall n$
- Die Pivoelementsuche liefert immer $a_{n,n}^{(n)}$

6: LR-Zerlegung

Sei $y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} \in \mathbb{K}^N$ ein Vektor, $n \in \{1, 2, \ldots, N\}$ mit $y_n \neq 0$ und setze

$$l_n := \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \frac{y_{n+1}}{y_n} \\ \vdots \\ \frac{y_N}{y_n} \end{pmatrix}.$$

Wobei die ersten n Einträge null sind. Daraus erhält man die Matrix:

$$L_n := E_n - l_n \cdot e_n^* = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & \frac{-y_{n+1}}{y_n} & 1 & & \\ & & \vdots & & \ddots & \\ & & \frac{-y_N}{y_n} & & & 1 \end{pmatrix}$$

Die y Einträge befinden sich in der n-ten Spalte und in den $(n+1) - N$ Zeilen.
Diese Matrix hat folgende Eigenschaften:

$$L_n y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \text{und} \quad L_n e_m = e_m \text{ für } m \neq n$$

Sei A eine beliebige Matrix mit $y = A_n$, dann setze:

$$R := L_{N-1} \cdots L_2 L_1 A \quad \text{und} \quad L := L_1^{-1} L_2^{-1} \cdots L_{N-1}^{-1}$$

Dann sind L und R Dreiecksmatritzen und für L gilt:

$$L = E_n + l_1 e_1^* + l_2 e_2^* + \cdots + l_{N-1} e_{N-1}^*$$

Jetzt kann das LGS einfach mit L und R gelöst werden:

$$Ax = L(Rx) = b \iff Lz = b \quad \text{und} \quad Rx = z$$

Dieser Algorithmus verwendet folgende Anzahl an Schritten:

$$\texttt{Rechenaufwand} = \frac{1}{3}N^3 - \frac{1}{3}N$$

7: Pivotsuche

Idee: Falls das Pivoelement bei der LR-Zerlegung $a_{n,n}^{(n)} = 0$ ist, tauschen wir die Zeilen oder Spalten um ein Pivotelemnt $a_{n,n}^{(n)} \neq 0$ erhalten.
Die Vertauschungen können mittels einer Permutationsmatrix durchgeführt werden:

$$
P_n := \left(\begin{array}{ccc|ccccc|ccc}
1 & & & & & & & & & & \\
 & \ddots & & & & & & & & & \\
 & & 1 & & & & & & & & \\
\hline
 & & & 0 & 0 & \cdots & 0 & 1 & & & \\
 & & & 0 & 1 & & & 0 & & & \\
 & & & \vdots & & \ddots & & \vdots & & & \\
 & & & 0 & & & 1 & 0 & & & \\
 & & & 1 & 0 & \cdots & 0 & 0 & & & \\
\hline
 & & & & & & & & 1 & & \\
 & & & & & & & & & \ddots & \\
 & & & & & & & & & & 1
\end{array}\right)
\begin{array}{l}
\\ \\ \\ \leftarrow n \\ \\ \\ \\ \leftarrow p \\ \\ \\ \\
\end{array}
$$

$$
\begin{array}{ccc}
\uparrow & \qquad & \uparrow \\
n & & p
\end{array}
$$

Die Matrix heißt elementare Permutationsmatrix und hat folgende Eigenschaften:

- $A \cdot P_n$ vertauscht n-te und p-te Zeile von A
- $P_n \cdot A$ vertauscht n-te und p-te Spalte von A
- $P_n^2 = E_n$

Der n-te Schritt mit Pivotsuche ist daher:

$$A^{(n+1)} = L_n P_n A^{(n)}$$

Dann gilt für R und L:

$$R := \tilde{L}_{N-1}\tilde{L}_{N-2}\cdots\tilde{L}_1 PA \quad \text{und} \quad L := \tilde{L}_1^{-1}\tilde{L}_2^{-1}\cdots\tilde{L}_{N-1}^{-1}$$

mit

$$\tilde{L}_n := P_{N-1}\cdots P_{n+1} L_n P_{n+1}\cdots P_{N-1}$$

Also gilt wieder:

$$PA = LR \quad \text{dann läst man wieder} \quad Ly = Pb \quad \text{und} \quad Rx = y$$

Spaltenpivotsuche:
Das m-te Pivotelement mit dem höchsten Score wird gewählt mit der Sapltenmaximierungstrategie:

$$\texttt{Pivoelement-Zeile} = \texttt{argmax}_{n=m,\ldots,N} = \frac{|a_{n,m}|}{\sum_{l=m}^{N} = |a_{n,l}|}$$

8: LR-Zerlegung Algorithmus

Algorithmus:

1. Start: $A^{(1)} := A$, $L^{(1)} := 0$, $P^{(1)} := E_N$
2. Pivotsuche ergibt neues Element: Vertausche Zeilen von $A^{(n)}$, also berechne: $A'^{(n)} := P_{n,j}A^{(n)}$, $P^{(n)} := P_{n,j}P^{(n-1)}$
3. Pivotsuche gibt kein neues Element: $P^{(n)} := P^{(n-1)}$
4. Berechne: $l_n := \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \frac{a_{n+1,n}^{(n)}}{a_{n,n}^{(n)}} \\ \vdots \\ \frac{a_{N,n}^{(n)}}{a_{n,n}^{(n)}} \end{pmatrix}$, $L_n := \begin{pmatrix} 0 & & & & \\ & \ddots & & & \\ & & 0 & & \\ & & l_{n+1} & 0 & \\ & & \vdots & & \ddots & \\ & & l_N & & & 0 \end{pmatrix}$, $L_-^{(n)} := E_N - L^{(n)}$
5. Berechne: $A^{(n+1)} := L_-^{(n)} A^{(n)}$
6. Schluss: $R := A^{(N)}$, $L = E_N + L^{(1)} + \cdots + L^{(N)}$, $P := P^{(N)}$

9: Cholesky Zerlegung

Sei $A \in \mathbb{K}^{N\times N}$ hermitesch und positiv definit, dann gibt es eine untere Dreiecksmatrix mit positiven Diagonalelementen und

$$A = LL^*.$$

Ansatz zur Berechnung:

$$\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,N} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,N} \\ \vdots & \vdots & & \vdots \\ a_{N,1} & a_{N,2} & \cdots & a_{N,N} \end{pmatrix} = \begin{pmatrix} l_{1,1} & & & \\ l_{2,1} & l_{2,2} & & \\ \vdots & \vdots & \ddots & \\ l_{N,1} & l_{N,2} & \cdots & l_{N,N} \end{pmatrix} \cdot \begin{pmatrix} l_{1,1} & \overline{l_{1,2}} & \cdots & \overline{l_{1,N}} \\ & l_{2,2} & \cdots & \overline{l_{2,N}} \\ & & \ddots & \vdots \\ & & & l_{N,N} \end{pmatrix}$$

Mit Rechenaufwand:

$$\texttt{Rechenaufwand} = \frac{1}{6}N^3 + \mathcal{O}(N^2)$$

10: Householder-Transformation

Die Householder-Transformation bzgl. eines Vektors $v \in \mathbb{K}^N$ ist:

$$P = E_n - \frac{2}{v^*v} vv^* \in \mathbb{K}^{N\times N}$$

Die Householder-Transformation ist eine Spiegelung an der Ebene orthogonal zu v.
Für P gilt folgende Eigenschaften:

- $Pv = -v$
- $Pw = w \ \forall w \perp v$
- P hermitesch, also $P^* = P$
- P unitär, also $P^*P = E_N$

11: QR-Zerlegung

Sei $A \in \mathbb{K}^{M\times N}$ mit $M \geqslant N$ und $\texttt{rang}(A) = N$. Dann gibt es eine unitäre Matrix $Q \in \mathbb{K}^{M\times M}$ und eine obere Dreiecksmatrix $R \in \mathbb{K}^{M\times N}$ mit

$$A = QR$$

Wegen Q unitär kann das Gleichungssystem wie folgt gelöst werden:

$$Rx = Q^* b$$

Algorithmus:
Im n-ten Schritt haben wir $A^{(n)}$ bestimmt und sei $a_n = A_n^{(n)}$ die n-te Spalte von $A^{(n)}$.

1. Berechne $v_n := \frac{a_n}{||a_n||_2} + \delta_n e_1$ mit $\delta_n := \begin{cases} \frac{(a_n)_1}{|(a_n)_1|} & , (a_n)_1 \neq 0 \\ 1 & , \text{sonst} \end{cases}$
2. Berechne $\beta_n := \frac{2}{v^* v}$, $w_n := A^* v_n$
3. Berechne $P_n = E_N - \beta_n v_n v_n^* = \begin{pmatrix} E_n & 0 \\ 0 & P'_n \end{pmatrix} \implies P_n A^{(n)} = A^{(n)} - \beta_n v_n w_n^* = \begin{pmatrix} r_{n,n} & * \\ 0 & A^{(n+1)} \end{pmatrix}$
4. Schluss: $R := P_n \cdots P_1 A$ und $Q := P_1 \cdots P_n$

Bemerkungen:

- QR-Zerlegung nicht eindeutig für $M \neq N$.
- Die Konditionszahl von der QR Zerlegung ist gleich der von A, also $\texttt{cond}_2(A) = \texttt{cond}_2(RQ) = \texttt{cond}_2(R)$.
- Also besser als bei der LR Zerlegung mit $\texttt{cond}_2(A) \leqslant \texttt{cond}_2(L)\texttt{cond}_2(R)$, wobei die Konditionszahlen von L und R viel höher als A sein können.
- Auch besser als die Cholesky Zerlegung, da für die cholesky-Zerlegung gilt: $\texttt{cond}_2(A) \leqslant (\texttt{cond}_2(L'))^2$.
- QR-Zerlegung ist doppelt so aufwendig wie LR-Zerlegung und viermal so aufwendig wie cholesky-Zerlegung.
- Falls der Rang nicht voll ist kann wieder Vertauschungen von Spalten verwendet werden, um den Algorithmus trotzdem anwendenden zu können.
- Der Algorithmus kann leicht abgewandelt werden dass die Pivoelemente immer ungleich Null sind.

$$\texttt{Rechenaufwand} = MN^2 - \frac{1}{3} + \mathcal{O}(MN)$$

12: Lineare Ausgleichsrechnung

Sei $Ax = b$ mit $A \in \mathbb{K}^{M\times N}$, $x \in \mathbb{K}^N$, $b \in \mathbb{K}^M$ und $M > N$. Dieses LGS ist allgemein nicht lösbar, deswegen brauchen wir einen neuen Lösungsbegriff:

$$\inf_{x\in\mathbb{K}^N} ||Ax - b||_2$$

Die Minimierer sind die Lösungen der Gaußschen Normalengleichung:

$$A^* A x = A^* b$$

Also eine Alternative zu der QR-Zerlegung

13: Banachscher Fixpunktsatz

Sei M vollständig bzg. einer Metrik d und sei $\phi : M \to M$ eine Abbildung, sodass es ein $q > 1$ gibt mit

$$\forall x, y \in M : d(\phi(x), \phi(y)) \leqslant q \cdot d(x, y)$$

Dann gibt es **genau ein** $\hat{x} \in M$ mit $\phi(\hat{x}) = \hat{x}$. Außerdem konvergiert für jedes $x^{(0)} \in M$ die Folge

$$x^{(n+1)} := \phi(x^{(n)}), \forall n \in \mathbb{N}_0$$

gegen $\hat{x}$ und für $n \geq 1$ gilt:

(a) Monotonie: $d(x^{(n)}, \hat{x}) \leqslant q \cdot d(x^{(n-1)}, \hat{x})$

(b) A-priori-Schranke: $d(x^{(n)}, \hat{x}) \leqslant \frac{q^n}{1-q} \cdot d(x^{(0)}, x^{(1)})$

(c) A-posteriori-Schranke: $d(x^{(n)}, \hat{x}) \leqslant \frac{q}{1-q} \cdot d(x^{(n)}, x^{(n-1)})$

14: Fixpunktsatz angewendet auf lineare Gleichungssysteme

Anwendung auf LGS $Ax = b$, zerlege in $A = M - N$. Daraus erhalten wir

$$Mx = Nx + b.$$

dass kann nun als ein Fixpunktproblem geschrieben werden:

$$x = Tx + c \qquad \text{mit} \quad T = M^{-1}N \quad \text{und} \quad c = M^{-1}x$$

also $\phi(x) = Tx + c$.
Dieses Problem konvergiert falls $||T|| < 1$, für eine induzierte Matrixnorm.

Gesamtschrittverfahren(Jacobi-Verfahren):
Hier ist $M = D$, $N = L + R$, wobei D Diagonalmatrix mit Einträgen ungleich Null, dann ist das wieder ein Fixpunktproblem mit:

$$x_{\text{neu}} = D^{-1}(b + (L + R)x_{\text{alt}}$$
$$\iff x_{\text{neu}} = \underbrace{D^{-1}b}_{:=c} + \underbrace{D^{-1}(L + R)}_{:=T} x_{\text{alt}}$$

oder konkret:

$$x_n^{(k+1)} = \frac{1}{a_{n,n}}(b_n - \sum_{\substack{m=1 \\ m \neq n}}^{N} a_{n,n} \cdot x_m^{(k)})$$

Einzelschrittverfahren(Gauß-Seidel-Verfahren):
Hier ist $M = D - L$ und $N = R$, daraus:

$$x = (D - L)^{-1}(b - Rx)$$

$$\rightsquigarrow x^{(neu)} = (D - L)^{-1}(b - Rx^{(alt)})$$
$$\iff x^{(neu)} = D^{-1}(b + Lx^{(neu)} - Rx^{alt})$$

konkret:

$$x_n^{k+1} = \frac{1}{a_{n,n}}(b_n - \sum_{k=1}^{n-1} a_{n,m}x_m^{k+1} - \sum_{k=n+1}^{N} a_{n,m}x_m^{k})$$

Falls A strikt diagonaldominant konvergieren beide Verfahren für jeden Startwert $x^{(0)} \in \mathbb{K}^N$

15: Konjugierte Gradienten

Sei $A \in \mathbb{K}^{N\times N}$ hermitesch und positiv definit. Das Verfahren berechnet die exakte Lösung wird aber oft vorher abgebrochen.
Algorithmus:

1. Start: Wähle $x^{(0)} \in \mathbb{K}^N$ und setze $d^{(0)} := r^{(0)} = b - Ax^{(0)}$. Ist $d^{(0)} = 0$ kann abgebrochen werden.

2. Berechne:
$$\alpha_{k-1} := \frac{d^{(k-1)*}r^{(k-1)}}{d^{(k-1)*}Ad^{(k-1)}}, \qquad \beta_{k-1} := \frac{d^{(k-1)*}Ar^{(k)}}{d^{(k-1)*}Ad^{(k-1)}} \quad \text{und} \quad r^{(k)} := b - Ax^{(k)}$$

3. Berechne:
$$x^{(k)} := x^{(k-1)} + \alpha_{k-1}d^{(k-1)} \quad \text{und} \quad d^{(k)} := r^{(k)} + \beta_{k-1}d^{(k-1)}$$

4. Schluss: $d^{(k)} = 0 \implies x^{(k)} = \hat{x}$ mit $A\hat{x} = b$

Das Verfahren konvergiert nach genau N schritten.
Konvergenzrate:
$$||x^{(k)} - \hat{x}||_2 \leqslant 2\mathtt{cond}(A)\left(\frac{\sqrt{\mathtt{cond}(A)} - 1}{\sqrt{\mathtt{cond}(A)} + 1}\right)^k ||x^0 - \hat{x}||_2$$

16: Eigenwerteinschließung

Sei $A, \Delta A \in \mathbb{K}^{N\times N}$, $||\cdot||$ eine induzierte Matrixnorm und $\sigma(A) := \{\lambda \in \mathbb{C} : \lambda \text{ ist Eigenwert von A}\}$. Dann gilt (Bauer-Fike):
$$\min_{\lambda(A)\in\sigma(A)} |\lambda(A) - \mu(\Delta A)| \leqslant \mathtt{cond}(A)\cdot||\Delta A||.$$
Wenn A *normal* ($A^*A = AA^*$) ist, so gilt sogar
$$\min_{\lambda(A)\in\sigma(A)} |\lambda(A) - \mu(\Delta A)| \leqslant ||\Delta A||_2.$$

Gerschgorin:
Sei
$$\mathcal{K}_n := \{\xi \in \mathbb{C} : |a_{n,n} - \xi| \leqslant \sum_{\substack{m=1\\m\neq n}}^N |a_{n,m}|\} \quad \text{und} \quad \mathcal{K}_n^* := \{\xi \in \mathbb{C} : |a_{n,n} - \xi| \leqslant \sum_{\substack{m=1\\m\neq n}}^N |a_{m,n}|\}.$$
Dann ist
$$\sigma(A) \subset \left(\bigcup_{n=1}^N \mathcal{K}_n\right) \cap \left(\bigcup_{n=1}^N \mathcal{K}_n^*\right).$$
Wenn $\mathcal{N}_1 \dot\cup \mathcal{N}_2 = \{1, 2, \dots, N\}$ mit
$$\left(\bigcup_{n\in\mathcal{N}_1} \mathcal{K}_n\right) \cap \left(\bigcup_{n\in\mathcal{N}_2}^N \mathcal{K}_n^*\right) = \emptyset.$$
Dann enthalten $\bigcup_{n\in\mathcal{N}_i}$, $i = 1, 2$ je genau $\#\mathcal{N}_i$ Eigenwerte.
Courant-Fischer:
Falls A hermitesch mit Eigenwerten $\lambda_1 \geqslant \lambda_2 \geqslant \dots \geqslant \lambda_N$. Dann gilt für jeden Unterraum $\mathcal{L} \subset \mathbb{K}^N$ mit $\mathtt{dim}(\mathcal{L}) = n$, dass
$$\min_{0\neq x\in\mathcal{L}} \frac{x^*Ax}{x^*x} \leqslant \lambda_n \quad \text{und} \quad \max_{0\neq x\in\mathcal{L}} \frac{x^*Ax}{x^*x} \geqslant \lambda_{N-n+1}$$
mit Gleichheit $\mathcal{L} = \{x_1, \dots, x_n\}$ bzw. $\mathcal{L} = \{x_{N-n+1}, \dots, x_N\}$, wobei $(x_m)_{m=1}^N$ Eigenvektoren einer Orthonormalbasis sind.

17: Courant-Fischer Folgerung

Sei $A, \Delta A \in \mathbb{K}^{N\times N}$ hermitesch, dann

$$\lambda_n(A) + \lambda_N(\Delta A) \leqslant \lambda_n(A+\Delta A) \leqslant \lambda_n(A) + \lambda_1(\Delta A), \forall n = 1, \ldots, N$$

und insbesondere

$$|\lambda_n(A+\Delta A) - \lambda_n(A)| \leqslant ||\Delta A||_2 \forall n = 1, \ldots, N$$

18: Potenzmethode

Motivation: Sei $A \in \mathbb{K}^{N\times N}$ diagonalisierbar, und $(v_n)_{n=1}^N$ eine Orthonormalbasis aus Eigenvektoren, so gilt für $x = \sum_{n=1}^N \xi_n v_n$, dass

$$A^k x = \sum_{n=1}^N \lambda_n^k \xi v_n.$$

Also dominiert der betragsgrößte Eigenwert für große k.
Sei $|\lambda_1| > |\lambda_2| \geqslant \ldots \geqslant |\lambda_n|$ die Eigenwerte von A, $0 \neq y \in \mathbb{C}^N$, mit $A^* y = \overline{\lambda_1} y$ und sei $\tilde{z}^{(0)} \in \mathbb{C}^N$ mit $y^* \tilde{z}^{(0)} \neq 0$, dann:

$$z^{(k)} := \frac{A^k \tilde{z}^{(0)}}{||A^k \tilde{z}^{(0)}||}, \forall k \geqslant 0$$

so gilt

$$\limsup_{k\to\infty} |||Az^{(k)}|| - |\lambda_1|| \leqslant \left(\frac{|\lambda_2|}{|\lambda_1|}\right)^k.$$

Außerdem gibt es genau ein Eigenvektor von λ_1 mit $||v|| = 1$ und $\mathtt{sgn}(y^* v) = \mathtt{sgn}(y^* \tilde{z}^{(0)})$ und dann gilt:

$$\limsup_{k\to\infty} ||z^{(k)} - \mathtt{sgn}(\lambda_1)^k v|| \leqslant \left(\frac{|\lambda_2|}{|\lambda_1|}\right)^k.$$

Algorithmus:

1. Start: Wähle $\tilde{z}^{(0)}$, sodass möglichst $y^* \tilde{z}^{(0)} \neq 0$
2. Berechne $z^k := \frac{\tilde{z}^{(k)}}{||\tilde{z}^{(k)}||}$
3. Berechne $\tilde{z}^{(k)} := Az^{(k-1)}$
4. Schluss Eigenwert: $|\lambda_1| \approx ||Az^{(k)}||$
5. Schluss Eigenvektor: $\mathtt{sgn}(\lambda_1)^k z^{(k)} \approx v$

Definition 1.0.1: Lokale Konvergenz

Ein Iterationsverfahren $x_{k+1} = \phi(x_k)$, mit $\phi(\hat{x}) = \hat{x}$ heißt *lokal Konvergent* gegen $\hat{x} \in \mathbb{K}$, falls es eine offene Menge $U \subset \mathcal{D}(\phi)$ mit $\hat{x} \in U$ gibt, sodass für jedes $x^{(0)} \in U$ das Iterationsverfahren definiert ist und gegen $\hat{x}$ konvergiert.

Definition 1.0.2: Konvergenz Ordnung

Ein lokal gegen $\hat{x}$ konvergentes Iterationsverfahren heißt konvergent *von der Ordnung* $p \geqslant 1$, falls es ein $\rho > 0$ und $C < \infty$ (mit $C < 1$ für $p = 1$) gibt mit:

$$||x_{k+1} - \hat{x}|| \leqslant C||x_k - \hat{x}||^p, \quad \forall k \geqslant 0 \quad \text{und} \quad ||x_0 - \hat{x}|| < \rho$$

Example 1.0.1

Das Gesamtschrittverfahren(Banach), Einzelschrittverfahren(Banach) und CG-Verfahren konvergieren mit Ordnung 1.

19: Konvergenz von Iterationsverfahren

Falls $\phi : \mathcal{D}(\phi) \subset \mathbb{K}^N \to \mathbb{K}^N$ stetig differenzierbar mit $\hat{x}$ im inneren von $\mathcal{D}(\phi)$ und es gibt eine Induzierte Matrixnorm mit $||\phi'(x)|| < 1$ so ist die Fixpunktiteration lokal konvergent gegen $\hat{x}$

20: Ordnung von Iterationsverfahren

Die Funktion $\phi : \mathcal{D}(\phi) \subset \mathbb{K} \to \mathbb{K}$ sie p-mal stetig differenzierbar mit $p \geqslant 2$ und habe einen Fixpunkt $\hat{x}$ im inneren von $\mathcal{D}(\phi)$. Ist

$$\phi'(x) = \cdots = \phi^{(p-1)} = 0$$

so ist die Fixpunktiteration lokal konvergent gegen $\hat{x}$ mit Konvergenzordnung mindestens p. Ist zusätzlich $\phi^{(p)}(\hat{x}) \neq 0$, ist die Ordnung genau p.

21: Newton-Verfahren

Die Funktion $f : \mathcal{D}(\phi) \subset \mathbb{R} \to \mathbb{R}$ ist differenzierbar und sei $\hat{x} \in \mathbb{R}$ mit $f(\hat{x}) = 0$, dann heißt folgende Vorschrift *Newton-Verfahren*:

$$x_{k+1} := x_k - \frac{f(x_k)}{f'(x_k)}, \forall k \geqslant 0$$

Falls $f \in C^3[a, b]$ und $\hat{x} \in (a, b)$ mit $f(\hat{x}) = 0, f'(\hat{x}) \neq 0$. Dann konvergiert das Newton-Verfahren lokal quadratisch, also $p = 2$, gegen $\hat{x}$.
Sekantenverfahren:

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

Sei $-\infty < a < b \leqslant \infty$ und $f : [a, b) \to \mathbb{R}$ stetig mit $f(a) = 0$ und in (a, b) stetig differenzierbar $f'(x) > 0$ für alle $x \in (a, b)$. Außerdem gelte

$$f(x) \geqslant f'(x)(x - a) \forall x \in (a, b).$$

Dann konvergiert das Newton-Verfahren für alle $x_0 \in (a, b)$ gegen a und die iterierte ist streng monoton fallend.

22: Polynom-Interpolation

Seien $x_0, \ldots, x_N \in \mathbb{R}$ paarweise verschieden. Dann heißen $w(x) := \prod_{n=0}^{N}(x - x_n)$ das knotenpolynom. Falls die Knoten angeordnet sind durch $x_0 < \cdots < x_N$ heißt,

$$l_n(x_k) := \frac{w(x_k)}{(x_k - x_n)w'(x_k)} = \prod_{\substack{m=0\\m\neq n}}^{N} \frac{x_k - x_m}{x_n - x_m} = \delta_{n,k}$$

das Lagrange-Grundpolynom. Dann gibt es für alle $y_0, \ldots, y_N \in \mathbb{K}$, genau ein Polynom p von $\texttt{Grad} \leqslant N$ mit

$$p(x_n) = y_n, \forall n = 0, \ldots, N.$$

Dieses Polynom ist gegeben durch

$$p(x) = \sum_{n=0}^{N} y_n l_n(x)$$

Sei nun $f \in C^{N+1}[a,b]$ und p das Interpolationspolynom vom Grad $\geqslant N$. Dann gibt es zu jedem $x \in [a,b]$ ein $\xi \in \texttt{conv}(x, x_0, \ldots, x_N)$ (Konvexehülle) mit

$$f(x) - p(x) = \frac{f^{(N+1)}(\xi)}{(N+1)1!} w(x).$$

Insbesondere ist

$$|f(x) - p(x)| \leqslant \frac{\max_{\xi\in[a,b]} |f(\xi)|}{(N+1)!} |w(x)|.$$

Es gilt:

$$\min_{x_0,\ldots,x_n\in[a,b]} \max_{x\in[a,b]} |(x - x_0)\ldots(x - x_N)| = \frac{(b-a)^{N+1}}{2 \cdot 4^N}.$$

Die beste Wahl ist dann für die Knoten:

$$x_n = \frac{1}{2}\left((b-a)\cos\frac{(2n-1)\pi}{2(N+1)} + a + b\right), n = 0, \ldots, N$$

Definition 1.0.3: Splines

Sei $[a,b]$ Intervall und Knoten $a = x_0 < x_1 < \cdots < x_n = b$. Sei

$$h_n := x_n - x_{n-1}, n = 1, \ldots, N$$

und

$$h := \max_n h_n$$

Ein Spline von Grad L ist eine Funktion $s \in C^{L-1}[a,b]$, die auf jeden Intervall $[x_{n-1}, x_n]$ mit einen Polynom von Grad $\geqslant L$ übereinstimmt. Die Menge aller Splines vom Grad L wird mit S_L oder $S_{L,\Delta}$ mit $\Delta = \{x_0, \ldots, x_N\}$ bezeichnet.
S_L ist ein $(N+L)$-dimensionaler Vektoraum.

23: Lineare Splines

Hutfunktionen:

$$\Lambda_0(x) = \begin{cases} \frac{x-x_1}{x_0-x_1} & \text{falls } x_0 \leqslant x \leqslant x_1, \\ 0 & \text{falls } x_1 \leqslant x \leqslant x_N \end{cases}$$

$$\Lambda_n(x) = \begin{cases} 0 & \text{falls } x_0 \leqslant x \leqslant x_{n-1}, \\ \frac{x-x_{n-1}}{x_n-x_{n-1}} & \text{falls } x_{n-1} \leqslant x \leqslant x_n \\ \frac{x-x_{n+1}}{x_n-x_{n+1}} & \text{falls } x_n \leqslant x \leqslant x_{n+1}, \\ 0 & \text{falls } x_{n+1} \leqslant x \leqslant x_N \end{cases}$$

$$\Lambda_N(x) = \begin{cases} 0 & \text{falls } x_0 \leqslant x \leqslant x_{N-1}, \\ \frac{x-x_{N-1}}{x_N-x_{N-1}} & \text{falls } x_{N-1} \leqslant x \leqslant x_N \end{cases}$$

24: Trigonomentrische Interpolation

Sei O.B.d.A: $L = 2\pi$, $0 \leqslant \theta_0 < \theta_1 \cdots < \theta_{N-1} \leqslant L$ und $y_0, \ldots, y_{N-1}$. Zur Vereinfachung äquidistante Knoten:

$$\theta_n = \frac{2\pi n}{N}$$