

Numerische Mathematik

Übungen

1. Stelle folgende Dezimalzahlen im Dual-, Hexadezimalsystem sowie im Zwölfersystem dar (d.h. $p = 2, 16, 12$, für $p = 12$ verwende man A, B als Ziffern für 10, 11)

$$275, 0.1, 1/3, 12.125$$

2. a_1, \dots, a_n seien Maschinenzahlen und es sei $f(a_1, \dots, a_n) = \sum_{i=1}^n a_i$. Der Computer habe die Maschinengenauigkeit eps . Auf dem Computer wird f in der Form $\tilde{f}(a_1, \dots, a_n) = (\dots((a_1 \oplus a_2) \oplus a_3) \oplus \dots) \oplus a_n$ berechnet.

- Mache eine Rückwärtsanalyse für f . D.h. $\tilde{f}(a_1, \dots, a_n) = f(a_1(1+\delta_1), \dots, a_n(1+\delta_n))$ und $|\delta_1|, \dots, |\delta_n|$ sind geeignet abzuschätzen.
- Mache eine Vorwärtsanalyse für f . D.h. $\tilde{f}(a_1, \dots, a_n) = f(a_1, \dots, a_n) + \varepsilon$ und $|\varepsilon|$ ist geeignet abzuschätzen.
- Zeige, daß für den Fehler zwischen \tilde{f} und f gilt:

$$|\tilde{f}(a_1, \dots, a_n) - f(a_1, \dots, a_n)| \leq \text{eps} \sum_{i=1}^n |n - i + 1| |a_i|.$$

Terme in der Größenordnung $\text{eps}^2, \text{eps}^3, \dots$ dürfen als näherungsweise 0 vernachlässigt werden.

3. Ermittle in **MATLAB** die Realisierung von underflow/overflow sowie $x_{\min}, x_{\max}, \text{eps}$.
4. Schreibe einen **MATLAB**-Algorithmus zur Lösung eines Gleichungssystems $Lx = b$ mit einer unteren Dreiecksmatrix L (Algorithmus 2.2 Lösung von Dreieckssystemen lemma.2.2 der Vorlesung). Verfahre analog bei $Ux = b$ wobei U obere Dreiecksmatrix ist (Algorithmus 2.5 Lösung von Dreieckssystemen lemma.2.5 der Vorlesung).
5. Schreibe einen **MATLAB**-Algorithmus zur Berechnung der Determinante einer $n \times n$ -Matrix mit Hilfe des Laplaceschen Entwicklungssatzes. Der Laplacesche Entwicklungssatz zur Determinantenentwicklung (nach der ersten Zeile) lautet:
 $\det A = \sum_{j=1}^n (-1)^{j+1} a_{1j} \det A(2 : n, [1 : j - 1, j + 1 : n])$. Diese Formel soll natürlich rekursiv ohne 'det' aus **MATLAB** angewendet werden.

6. Sei $A \in \mathbb{R}^{m,n}$. Zeige:

- (a) $\|A\|_2 \leq \|A\|_F \leq \sqrt{n} \|A\|_2$
- (b) $\|AB\|_F \leq \|A\|_2 \|B\|_F \leq \|A\|_F \|B\|_F$.

7. Sei $A \in \mathbb{R}^{m,n}$. Zeige:

- (a) Ist $B \in \mathbb{R}^{n,l}$ so gilt $\|AB\|_p \leq \|A\|_p \|B\|_p$
 Tipp: Zeige die Behauptung zuerst für $l = 1$ und nutzte die Def.

$$(b) \|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^m |a_{ij}|.$$

Tipp: Zeige zunächst $\|Ax\|_1 \leq (\max \Sigma \dots) \|x\|_1$ und konstruiere dann ein x für welches Gleichheit gilt.

$$(c) \|A\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}|.$$

(d) Ist $\rho(A) = \max\{|\lambda| : \lambda \text{ Eigenwert von } A\}$ eine Matrixnorm?

8. Sei $A \in \mathbb{R}^{m,n}, B \in \mathbb{R}^{k,l}$. Zeige:

(a) Falls $k = m, l = n$, dann gilt $|A + B| \leq |A| + |B|$.

(b) Falls $k = n$, dann gilt $|AB| \leq |A| |B|$.

(c) Falls $k = m, l = n$ und $A \leq B$, dann gilt für alle $C \in \mathbb{R}^{p,m}$ mit $C \geq O$: $CA \leq CB$.

(d) Falls $k = m, l = n$ und $|A| \leq |B|$, dann gilt $\|A\|_\infty \leq \|B\|_\infty$.

9. Vergleiche folgende vier Verfahren in **MATLAB** zur Lösung von $Ax = b, A \in \mathbb{R}^{n,n}, b \in \mathbb{R}^n$. Messe die Rechenzeit für $n = 4, 6, 8, 10$ und zufällig erzeugte A, b .

(a) den `\` aus **MATLAB**. (Hinter diesem steckt der Gauß-Alg.)

(b) Einen selbstgeschriebenen Gauß-Algorithmus entsprechend der Vorlesung

(c) Die Cramersche Regel mit Hilfe der eingebauten `'det'`-Funktion

(d) Die Cramersche Regel mit Hilfe des Laplaceschen Entwicklungssatzes

Die Cramersche Regel für $Ax = b$ ist wie folgt definiert:

$$x_i = \det[A(:, 1 : i - 1), b, A(:, i + 1 : n)] / \det A, \quad i = 1, \dots, n$$

Der Laplacesche Entwicklungssatz zur Determinantenentwicklung (nach der ersten Zeile) lautet:

$\det A = \sum_{j=1}^n (-1)^{j+1} a_{1j} \det A(2 : n, [1 : j - 1, j + 1 : n])$. Diese Formel soll natürlich rekursiv ohne `'det'` aus **MATLAB** angewendet werden.

10. Die exakte Lösung des Gleichungssystems

$$\begin{pmatrix} 1 & 0.99 \\ 0.99 & 0.98 \end{pmatrix} x = \begin{pmatrix} 1.99 \\ 1.97 \end{pmatrix}$$

ist $x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$. Mit $y = \begin{pmatrix} 1.94 \\ 0.050459 \end{pmatrix}$ gilt aber auch

$$\begin{pmatrix} 1 & 0.99 \\ 0.99 & 0.98 \end{pmatrix} y = \begin{pmatrix} 1.98995441 \\ 1.97004982 \end{pmatrix}.$$

Erkläre dieses Verhalten anhand der Störungstheorie der Vorlesung.

11. Schreibe ein **MATLAB**-Programm, welches den Gauß-Algorithmus der Vorlesung ohne Pivotisierung auf folgendes Gleichungssystem (Hilbertmatrix, in **MATLAB** `'hilb(n)'`) anwendet.

$$Hx = b, \quad H = \left(\frac{1}{i+j-1} \right)_{i,j=1,\dots,n}, \quad b = H(1, \dots, 1)^T.$$

Vergleiche die Lösung des Algorithmus mit der exakten Lösung $x = (1, \dots, 1)^T$ sowie der von **MATLAB** mittels `\` berechneten Lösung für $n = 2, 3, 4, \dots, 30$. Was stellst Du fest? Erklärung!

12. Eine Matrix $A = (a_{ij})_{i,j=1,\dots,n}$ heißt strikt diagonaldominant, wenn gilt

$$|a_{jj}| - \sum_{\substack{i=1 \\ i \neq j}}^n |a_{ij}| > 0, \text{ für alle } j = 1, \dots, n.$$

Zeige:

- (a) Ist A strikt diagonaldominant, dann ist A nicht singulär.
- (b) Ist A strikt diagonaldominant, dann gilt das auch für alle Hauptabschnittsmatrizen.
- (c) Ist A strikt diagonaldominant, dann besitzt A eine LR -Zerlegung.

13. Sei

$$A = \begin{pmatrix} a_{11} & a_{12} & & & \\ a_{21} & \ddots & \ddots & & \\ & \ddots & \ddots & & \\ & & & a_{n-1,n} & \\ & & & a_{n,n-1} & a_{n,n} \end{pmatrix}$$

tridiagonal. Wieviel flops (+, −, *, /) braucht der Gauß-Algorithmus, falls nicht permutiert werden muß und der Algorithmus durchgeht? Wie teuer sind Vorwärts- und Rückwärtseinsetzen?

14. Zeige Lemma 2.30 Fehleranalyse der Gauß-Eliminationlemma.2.30 aus der Vorlesung: Sei

$$\begin{aligned} Ax &= b, & A \in \mathbb{R}^{n,n}, 0 \neq b \in \mathbb{R}^n \\ (A + \Delta A)y &= b + \Delta b, & \Delta A \in \mathbb{R}^{n,n}, \Delta b \in \mathbb{R}^n, \end{aligned}$$

mit $|\Delta A| \leq \delta|A|, |\Delta b| \leq \delta|b|$. Falls $\delta \| |A^{-1}| |A| \|_{\infty} = r < 1$, so ist $A + \Delta A$ nichtsingulär und

$$\frac{\|y\|_{\infty}}{\|x\|_{\infty}} \leq \frac{1+r}{1-r}.$$

Tip: Verfahre analog zum Beweis des entsprechenden Lemmas 2.26 Fehleranalyse der Gauß-Eliminationlemma.2.26 über Normen.

15. Sei $1 > \varepsilon > 0$,

$$A = \begin{pmatrix} \varepsilon & -1 & & & \\ & \ddots & \ddots & & \\ & & \ddots & & \\ & & & -1 & \\ & & & & \varepsilon \end{pmatrix} \in \mathbb{R}^{n,n}.$$

- (a) Zeige, es existiert eine singuläre Matrix B mit $\|B - A\|_{\infty} \leq \varepsilon^n$. D.h. der Abstand von A zu den singulären Matrizen ist höchstens ε^n . Tip: man füge in der Position $(n, 1)$ einen geeigneten Eintrag ein, so dass bei anschließender Determinantenbildung eine Null entsteht.
- (b) Berechne A^{-1}
- (c) Bestimme für $\|\bullet\|_{\infty}$ die Normen von A, A^{-1} , die Konditionszahl sowie die Skeel-Konditionszahl $\| |A| |A^{-1}| \|$.
- (d) Wende für $\varepsilon \leq \frac{1}{4}$ und $\delta = \varepsilon^n/4, b = (1, \dots, 1)^T, \Delta b = \delta \cdot b$ und $\Delta A = 0$ die Sätze 2.27 Fehleranalyse der Gauß-Eliminationlemma.2.27, 2.31 Fehleranalyse der Gauß-Eliminationlemma.2.31 der Vorlesung an. Wie scharf sind die Ergebnisse?

16. Sei

$$L = \begin{pmatrix} 1 & & & 0 \\ l_{21} & \ddots & & \\ \vdots & \ddots & \ddots & \\ l_{n1} & \cdots & l_{n,n-1} & 1 \end{pmatrix}, \quad l_k = \left. \begin{pmatrix} 0 \\ \vdots \\ 0 \\ l_{k+1,k} \\ \vdots \\ l_{n,k} \end{pmatrix} \right\}^k \quad k = 1, \dots, n-1.$$

Zeige:

- (a) $(I + l_k e_k^T)(I + l_m e_m^T) = I + l_k e_k^T + l_m e_m^T$ für alle $k \leq m$. Hier bezeichnen e_1, \dots, e_n die Einheitsvektoren.
- (b) $L = (I + l_1 e_1^T) \cdots (I + l_{n-1} e_{n-1}^T)$.
- (c) $L^{-1} = (I - l_{n-1} e_{n-1}^T) \cdots (I - l_1 e_1^T)$.

17. Zähle die genaue Anzahl flops in der GAXPY-Version des Gauß-Algorithmus (Algorithmus 2.20 GAXPY Form der Gauß-Elimination Lemma 2.20 der Vorlesung). Zeige dass $\frac{2}{3}n^3 + \mathcal{O}(n^2)$ flops benötigt werden.

18. Sei $A \in \mathbb{R}^{n,n}$ und $\varepsilon > 0$. Zeige:

- (a) Für jede nicht singuläre Matrix $X \in \mathbb{R}^{n,n}$ und jede Matrixnorm $\|\bullet\|$ ist $\|AX\| := \|X^{-1}AX\|$ wieder eine Matrixnorm.
- (b) Ist $A = (\lambda I)$, dann existiert zu jedem $\varepsilon > 0$ eine Diagonalmatrix D , so dass $\|D^{-1}AD\|_\infty \leq \rho(A) + \varepsilon$.
- (c) Es existiert zu A eine Norm $\|\bullet\|$, so dass $\|A\| \leq \rho(A) + \varepsilon$.
- (d) Sei $\rho(A) < 1$. Dann konvergiert für jeden Startwert $x_0 \in \mathbb{R}^n$ die Iteration $x_{k+1} = Ax_k$.

19. Eine Matrix $A \in \mathbb{R}^{n,n}$ heisst M -Matrix, falls es eine nichtnegative Matrix $N \in \mathbb{R}^{n,n}$ und ein $s > \rho(N)$ gibt, mit $A = sI - N$ ($\rho(N) = \max\{|\lambda| : \lambda \text{ Eigenwert von } N\}$ ist der Spektralradius von N). Zeige: A ist nicht singulär und A^{-1} ist nicht negativ.

20. Schreibe **MATLAB**-Programme, welche den Gauß-Algorithmus der Vorlesung mit partieller Pivotisierung sowie vollständiger Pivotisierung realisieren. Verwende als Testbeispiel $Hx = b$ wieder die Hilbertmatrix H , $x = (1, \dots, 1)^T$ für $n = 2, 3, 4, \dots, 30$.

21. Bestimme den Wachstumsfaktor $\rho = \max_{i,j,k} \frac{|a_{ij}^{(k)}|}{\|A\|_\infty}$ aus der Vorlesung bei exakter Arithmetik und partieller Pivotisierung für die Matrix

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & 1 \\ -1 & 1 & \ddots & \vdots & 1 \\ -1 & -1 & \ddots & 0 & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ -1 & \cdots & \cdots & -1 & 1 \end{pmatrix}.$$

22. Sei $A \in \mathbb{R}^{n,n}$, $A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$ und nehme an, daß A und $A_{11} \in \mathbb{R}^{k,k}$ nicht singulär seien. Das Schur–Komplement ist gegeben durch $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$. Zeige:
- A läßt sich schreiben als

$$A = \begin{pmatrix} I & O \\ A_{21}A_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} A_{11} & O \\ O & S \end{pmatrix} \begin{pmatrix} I & A_{11}^{-1}A_{12} \\ O & I \end{pmatrix}.$$
 - S ist nicht singulär.
 - A^{-1} läßt sich schreiben als $A^{-1} = \begin{pmatrix} * & * \\ * & S^{-1} \end{pmatrix}$ (* heißt, die Einträge interessieren uns hier nicht).
 - Ist A symmetrisch, so ist S auch symmetrisch.
 - Ist A positiv definit d.h. $x^T Ax > 0$, $\forall x \neq 0$, so ist S auch positiv definit, d.h. $y^T Sy > 0$, $\forall y \neq 0$.
 - Ist A invers nicht negativ, dann ist auch S invers nicht negativ (A heißt invers nicht negativ, falls A^{-1} nur nicht negative Einträge hat).
23. Sei $A \in \mathbb{R}^{n,n}$, $A = (a_{ij})_{i,j=1,\dots,n}$, $A = A^T$. Zeige folgende Äquivalenzen:
- A ist positiv definit, d.h. $x^T Ax > 0$, $\forall x \neq 0$.
 - Alle Eigenwerte von A sind positiv.
 - Alle Hauptabschnittsmatrizen $(a_{ij})_{i,j=1,\dots,k}$, $k \leq n$ sind positiv definit.
 - Alle Hauptabschnittsdeterminanten $\det \left((a_{ij})_{i,j=1,\dots,k} \right)$, $k \leq n$ sind positiv.
24. Sei $A \in \mathbb{R}^{n,n}$, $A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$ Falls $A_{11} \in \mathbb{R}^{k,k}$ nicht singulär ist, dann ist das Schur–Komplement gegeben durch $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$. Zeige:
- Ist A strikt diagonaldominant, dann (ist A_{11} nicht singulär und) S ist strikt diagonaldominant. Was bedeutet das für den Gauß–Algorithmus mit partieller Pivotisierung?
 - Ist A eine M –Matrix, dann ist auch S eine M –Matrix. (A heißt M –Matrix, falls A invers nicht negativ ist und alle Einträge von A außerhalb der Diagonalen kleiner oder gleich 0 sind).
Tip: Aus der Definition M –Matrix folgt bereits, daß die Diagonaleinträge von A positiv sind. Warum?
25. Sei $A \in \mathbb{R}^{n,n}$ und sei $F \in \mathbb{R}^{n,r}$ mit $r \leq n$. Zeige:
- Ist A positiv definit (d.h. $x^T Ax > 0$ für alle $x \neq 0$), dann ist $F^T AF$ positiv definit genau dann, wenn F vollen Rang hat.
 - Welchen Rang haben Matrizen der Form $\begin{pmatrix} X \\ I_r \end{pmatrix}$, $\begin{pmatrix} I_r \\ Y \end{pmatrix}$?
 - Ist A positiv definit, dann ist auch A^{-1} positiv definit.

26. (a) Schreibe einen **MATLAB**-Algorithmus zur Berechnung der Cholesky-Zerlegung. Vergleiche den Algorithmus bezüglich Rechengenauigkeit und Rechenzeit mit der Funktion 'chol' und dem '\ ' aus **MATLAB** für $n = 10, 20, 30, 40, \dots, 100$ und zufällig erzeugten positiv definiten Matrizen $A = MDM^T$, $M = rand(n)$, $D = diag(1, 2, 4, 9, \dots, n^2)$.
- (b) Schreibe einen **MATLAB**-Algorithmus zur Berechnung der Cholesky-Zerlegung für tridiagonale Matrizen. Vergleiche den Algorithmus bezüglich Rechengenauigkeit und Rechenzeit mit der Funktion 'chol' und dem '\ ' aus **MATLAB** für $n = 10, 20, 30, 40, \dots, 100$ anhand der Matrix $A =$
- $$\begin{pmatrix} 2 & -1 & & & \\ -1 & \ddots & \ddots & & \\ & \ddots & \ddots & -1 & \\ & & & -1 & 2 \end{pmatrix}.$$
27. (a) Schreibe ein **MATLAB**-Programm, welches den Gauß-Algorithmus der Vorlesung mit vollständiger Pivotisierung sowie Zeilenskalierung und iterativer Verbesserung realisiert. Verwende als Testbeispiel $Hx = b$ wieder die Hilbertmatrix H , $x = (1, \dots, 1)^T$ für $n = 2, 3, 4, \dots, 30$. (Siehe Übung 11).
- (b) Untersuche die Genauigkeitsverbesserung mittels iterativer Verbesserung dadurch, dass Daten und Rechenoperationen mit zufälligen Werten der Form $(1+\varepsilon)$ gestört werden, wobei $|\varepsilon| \leq \sqrt{\text{eps}}$ ist (einfache statt doppelte Genauigkeit). Vergleiche die iterative Verbesserung mit und ohne gestörte Berechnung des Residuums.
28. Zeige:
- (a) Für jedes $V \in \mathbb{R}^{n,m}$, $\text{Rang } V = m$ ist $S = I - 2V(V^TV)^{-1}V^T$ symmetrisch und orthogonal und es ist $S^2 = I$.
- (b) S ist eine Spiegelung am orthogonalen Komplement von Bild V .
- (c) S hat nur Eigenwerte -1 m -fach, 1 $n - m$ -fach.
- (d) $P = I - V(V^TV)^{-1}V^T$ ist symmetrisch und erfüllt $P^2 = P$.
- (e) P ist eine Projektion auf das orthogonalen Komplement von Bild V .
- (f) P hat nur Eigenwerte 0 m -fach, 1 $n - m$ -fach.
- (g) Jede orthogonale Matrix obere Dreiecksmatrix ist bereits eine Diagonalmatrix.
- (h) Jede orthogonale Matrix läßt sich als Produkt von Elementarspiegelungen S (d.h. $m = 1$ in (a)) schreiben.
29. Sei $G = \begin{pmatrix} c & -s \\ s & c \end{pmatrix}$ orthogonal und nehme an, dass wir für vorgegebenes $v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$, c, s so bestimmen können, dass $Gv = \begin{pmatrix} * \\ 0 \end{pmatrix}$ ist. Zeige:
- (a) Ist A eine obere Hessenbergmatrix (d.h. $a_{i,j} = 0$ für alle $i > j + 1$), dann besitzt A eine QR -Zerlegung, die in $3n^2 + \mathcal{O}(n)$ flops berechnet werden kann (ohne explizite Berechnung von Q).
Welches Belegungsmuster hat Q ?
- (b) Ist A eine beliebige $n \times n$ -Matrix, dann braucht die QR -Zerlegung mit Hilfe von G , $2n^3 + \mathcal{O}(n^2)$ flops (ohne explizite Berechnung von Q).
Ist das besser oder schlechter als die QR -Zerlegung mit Hilfe von Householder-Transformationen?

30. Zeige:

- (a) Ist eine orthogonale Matrix gleichzeitig eine obere Dreiecksmatrix, dann ist sie bereits diagonal. Wie sehen die Diagonaleinträge aus? Gilt ein analoges Ergebnis auch für unitäre Matrizen?
- (b) Jede orthogonale Matrix lässt sich als Produkt von Elementarspiegelungen (Householdertransformationen) schreiben. Gilt das auch, wenn man Elementarspiegelungen durch Rotationen ersetzt?

31. Betrachte für $c > 0, s = \sqrt{1 - c^2}$ die Matrix

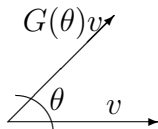
$$T_n = \text{diag}(1, s, \dots, s^{n-1}) \begin{pmatrix} 1 & -c & \cdots & -c \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -c \\ 0 & \cdots & 0 & 1 \end{pmatrix} \in \mathbb{R}^{n,n}.$$

Zeige:

- (a) $\max_{1 \leq k \leq n} \|T_n(:, k)\|_2 = \|T_n(:, 1)\|_2$.
- (b) $\sigma_n(T_n) \equiv \min_{\|x\|_2=1} \|T_n x\|_2 \leq \frac{s^{n-1}}{c(c+1)^{n-2}}$.
- (c) Was bedeutet das für den QR -Algorithmus mit Spaltenpivotisierung (d.h. in der Restmatrix wird die Spalte mit maximaler 2-Norm mit der führenden Spalte vertauscht) und für die Rangbestimmung von T_n ? (Betrachte z.B. $n = 100, c = 0.2$)

32. Sei $G(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \equiv \begin{pmatrix} c & -s \\ s & c \end{pmatrix}$. Zeige:

- (a) die Abbildung G beschreibt eine Drehung um den Winkel θ gegen den Uhrzeigersinn in der Ebene.



- (b) Bestimme für vorgegebenes $v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$, c, s so, dass $Gv = \begin{pmatrix} * \\ 0 \end{pmatrix}$ ist. Wie müssen c, s bestimmt werden, damit ein Höchstmaß an numerischer Stabilität gesichert ist? Begründung!

33. Schreibe einen **MATLAB**-Algorithmus zur Berechnung der QR -Zerlegung mittels Householder-Transformationen.

Vergleiche den Algorithmus bezüglich Rechengenauigkeit und Rechenzeit mit der Funktion 'qr' aus **MATLAB** für $n = 10, 20, 30, 40, \dots, 100$ und zufällig erzeugten Matrizen

34. Sei $Q = [q_1, \dots, q_n] \in \mathbb{R}^{m,n}$ mit $Q^T Q = I$.

- (a) Was bedeutet das für q_1, \dots, q_n ?
- (b) Zeige:

$$I - QQ^T = \prod_{i=1}^n (I - q_i q_i^T),$$

dabei ist die Reihenfolge, in der das Produkt gebildet wird, beliebig.

(c) Sei $a \in \mathbb{R}^n$. Zeige $(I - QQ^T)a$ steht senkrecht auf q_1, \dots, q_n

(d) Berechne $b = (I - QQ^T)a$ auf zwei Weisen

1. for $j = 1 : n$
 $R_j := q_j^T a$
 end
 $b := a$
 for $j = 1 : n$
 $b := b - q_j * R_j$
 end
2. $b := a$
 for $j = 1 : n$
 $R_j := q_j^T b$
 $b := b - q_j * R_j$
 end

Nach einer etwaigen Normierung von b am Ende, welcher der beiden Algorithmen entspricht dem klassischen und welcher dem modifizierten Gram-Schmidt-Verfahren. Wie kann man das hinsichtlich Teil (b) interpretieren?

35. Seien A und $B = QA$ gegeben, wobei Q orthogonal. Nehme an, dass $\nu_k = \|(a_{i,k})_{i=1,\dots,m}\|_2^2$, für $k = 1, \dots, n$ gegeben ist. Wie lassen sich $\nu'_k = \|(b_{i,k})_{i=2,\dots,m}\|_2^2$, für $k = 2, \dots, n$ möglichst schnell aus ν_2, \dots, ν_n berechnen? Welche Auswirkungen hat das auf Anzahl flops bei der QRP -Zerlegung?

36. Schreibe einen **MATLAB**-Algorithmus zur Berechnung

- (a) der QRP -Zerlegung (Spaltenpivotisierung) mittels Householder-Transformationen (hierzu wird in jedem Schritt vor der Elimination die Spalte mit der größten 2-Norm in die führende Position gebracht. D.h., ist $A(j : m, j : n)$ die Restmatrix im j -ten Schritt, so berechne μ mit $\|A(:, \mu)\|_2 = \max_{j \leq l \leq n} \|A(:, l)\|_2$ und vertausche $A(:, j)$ mit $A(:, \mu)$),
- (b) der QR -Zerlegung mittels klassischem Gram-Schmidt Verfahren.
- (c) der QR -Zerlegung mittels modifiziertem Gram-Schmidt Verfahren.

Untersuche die Algorithmen für das lineare Ausgleichsproblem $\|Ax - b\|_2 \stackrel{!}{=} \min$ und vergleiche sie bezüglich Rechengenauigkeit und Rechenzeit mit dem Cholesky-Verfahren für die Normalgleichungen $A^T A x = A^T b$. Zur Generierung der Beispielaufgaben kann auf ein entsprechendes **MATLAB**-Scriptfile im WWW zurückgegriffen werden.

37. Sei

$$A = \begin{pmatrix} * & & & & & & & * \\ * & & * & & & & & \\ * & & & & & & * & \\ * & & * & & & & & \\ * & & & & & & * & \\ * & * & & & & & & \\ * & & & & * & & & \\ * & * & * & * & * & * & * & * \end{pmatrix}$$

Bestimme Permutationen U, V , so dass bei einer QR -Zerlegung von UAV möglichst wenig Einträge in R entstehen.

38. Sei

$$A = \begin{pmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix} \in \mathbb{R}^{n,n}.$$

Zeige

- (a) Die Eigenwerte von A sind $\lambda_j = 2 - 2 \cos \frac{j\pi}{n+1}$, und die zugehörigen Eigenvektoren lauten $v_j = (v_{ij})_{i=1,\dots,n} = \left(\sin \frac{ij\pi}{n+1}\right)_{i=1,\dots,n}$ für alle $j = 1, \dots, n$.
Hinweis: $\sin(x+y) = \sin x \cos y + \cos x \sin y$, $\cos(-x) = \cos x$, $\sin(-x) = -\sin x$.
- (b) Für jede mit der Matrixmultiplikation verträgliche Norm $\|\bullet\|$ gilt: $\rho(A) \leq \|A\|$.
- (c) In der Standard-Zerlegung $A = D - L - U$ ist das Einzelschrittverfahren definiert durch $M = D - L$, $N = U$ und die Iterationsmatrix $T_E = M^{-1}N$. Zeige: $T_E = (I + D^{-1}L + \dots + (D^{-1}L)^{n-1})D^{-1}U$.
- (d) Bestimme/schätze den Spektralradius von Gesamt-/Einzelschrittverfahren. Was kann man über die Konvergenzgeschwindigkeit des cg-Verfahrens sagen?
39. Sei $A \in \mathbb{R}^{n,n}$, $A = D - L - U$ Standardzerlegung, d.h. D ist Diagonale von A , $-L$ striktes unteres Dreieck und $-U$ striktes oberes Dreieck von A . Sei $\omega \in \mathbb{R}$ und betrachte die Zerlegung $\omega A = M - N$ mit $M = D - \omega L$, $N = (1 - \omega)D + \omega U$.
Zeige: Ist $\rho(M^{-1}N) < 1$, so ist $0 < \omega < 2$.
40. Sei $\Phi : K \rightarrow \mathbb{R}^n$, $K \subseteq \mathbb{R}^n$ offen, konvex, Φ differenzierbar und nehme an, daß das totale Differential $D\Phi$ auf K beschränkt ist. Zeige:

$$\|\Phi(x) - \Phi(y)\|_\infty \leq \|D\Phi\|_{\infty|_K} \|x - y\|_\infty, \quad \forall x, y \in K.$$

41. Schreibe **MATLAB**-Programme für das Jacobi-Verfahren, Gauß-Seidel-Verfahren, sowie das cg-Verfahren ohne Vorkonditionierung. Vergleiche die Verfahren bezüglich Rechenzeit und Iterationszahl anhand des Beispiels $Ax = b$,

$$A = \begin{pmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ \vdots \\ \vdots \\ 1 \end{pmatrix},$$

$x^{(0)} = 0$, $n = 10, 100, 1000$. Als Abbruchkriterium nehme man $\|r_{k+1}\|_2 = \|b - Ax_{k+1}\|_2 < 10^{-8}$, $k_{\max} = 600$.

42. Sei $A \in \mathbb{R}^{n,n}$ symmetrisch, positiv definit und $b, x_0 \in \mathbb{R}^n$. Betrachte das cg-Verfahren ohne Vorkonditionierung aus der Vorlesung (zur besseren Analyse mit Indizes).

```

k = 0, r_0 = b - Ax_0, rho_0 = r_0^T r_0
while sqrt(rho_k) > epsilon * sqrt(rho_0) and k < k_max
    k = k + 1
    if k = 1
        p_1 = r_0
    else

```

$$\beta_k = \rho_{k-1}/\rho_{k-2}$$

$$p_k = r_{k-1} + \beta_k p_{k-1}$$

end

$$w_k = Ap_k, \alpha_k = \rho_{k-1}/w_k^\top p_k$$

$$x_k = x_{k-1} + \alpha_k p_k$$

$$r_k = r_{k-1} - \alpha_k w_k$$

$$\rho_k = r_k^\top r_k$$

end

Zeige:

(a) $p_k^\top Ap_l = 0$, für alle $k > l \geq 0$.

Tipp: Zeige durch Induktion über k , $r_{k-1}^\top r_{l-1} = 0$, $p_k^\top Ap_l = 0$, für alle $k > l > 0$.

(b) Nehme an, es sei $p_k^\top Ap_l = 0$, für alle $k > l > 0$, dann ist

$$\min_{v_1, \dots, v_k \in \mathbb{R}} \Phi(x_0 + \sum_{l=1}^k p_l v_l) = \min_{v_k \in \mathbb{R}} \left(\min_{v_1, \dots, v_{k-1} \in \mathbb{R}} \Phi(x_0 + \sum_{l=1}^k p_l v_l) \right),$$

wobei $\Phi(x) = \frac{1}{2}x^\top Ax - x^\top b$. Was passiert mit dem cg-Verfahren nach maximal n Schritten?

43. Für welche der folgenden Gleichungen und welche Startwerte konvergiert die zugehörige Fixpunktiteration?

(a) $x = e^x - \sin x + x$

(b) $x = \sin x - e^x + x$

(c) $x = \arcsin e^x, x < 0$

(d) $x = \ln \sin x, x \in (0, \pi)$

44. Sei $p(x) = a_n x^n + \dots + a_1 x + a_0$, $a_n \neq 0$, Polynom n -ten Grades. Zeige:

(a) Die Nullstellen von p sind die Eigenwerte der Begleitmatrix

$$A = \begin{pmatrix} 0 & 1 & & & \\ & \ddots & \ddots & & \\ & & 0 & 1 & \\ -\frac{a_0}{a_n} & -\frac{a_1}{a_n} & \dots & -\frac{a_{n-1}}{a_n} & \end{pmatrix}$$

(b) Die Nullstellen ξ_1, \dots, ξ_n von p erfüllen für alle $i = 1, \dots, n$ die Ungleichungen

$$|\xi_i| \leq \max\left\{1, \sum_{k=0}^{n-1} \left|\frac{a_k}{a_n}\right|\right\}, \quad |\xi_i| \leq \max\left\{\left|\frac{a_0}{a_n}\right|, 1 + \left|\frac{a_1}{a_n}\right|, \dots, 1 + \left|\frac{a_{n-1}}{a_n}\right|\right\},$$

(Hinweis: Für jeden Eigenwert λ von A und jede mit der Matrixmultiplikation verträgliche Norm gilt: $|\lambda| \leq \|A\|$)

45. Sei $A \in \mathbb{R}^{n,n}$ und

$$f(x_1, \dots, x_n, x_{n+1}) = \begin{pmatrix} (A - x_{n+1}I) \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \\ \frac{1}{2}(\sum_{i=1}^n |x_i|^2 - 1) \end{pmatrix}.$$

Zeige:

- (a) (x_1, \dots, x_{n+1}) ist Nullstelle von f , genau dann wenn x_{n+1} Eigenwert und $(x_1, \dots, x_n)^T$ normierter Eigenvektor von A sind.
- (b) Sei (x_1, \dots, x_{n+1}) Nullstelle von f . Dann ist die Ableitungsmatrix $Df(x_1, \dots, x_{n+1})$ nicht singulär, genau dann wenn x_{n+1} ein einfacher Eigenwert von A ist (Tip: Jordan-Form). Was bedeutet das für das Newton-Verfahren?

46. Schreibe ein **MATLAB**-Programm für das Newtonverfahren zur Bestimmung einer Nullstelle einer Funktion. Wende das Verfahren auf die Funktion $e^x - \sin x$ an und vergleiche das Programm bezüglich Rechenaufwand und Genauigkeit mit der Fixpunktiteration angewandt auf die Funktionen aus Aufgabe 43

47. a) Zeige den Satz von Gerschgorin: Sei $A = (a_{ij})_{i,j=1,\dots,n} \in \mathbb{C}^{n,n}$. Dann sind die Eigenwerte von A enthalten in der Vereinigung der Kreisscheiben

$$\bigcup_{i=1}^n K_i, \quad K_i = \{z : |z - a_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|\}.$$

- b) Zeige folgende verschärfte Variante des Satzes von Gerschgorin: Sind i_1, \dots, i_r (paarweise verschieden) so gewählt, daß

$$\left(\bigcup_{k=1}^r K_{i_k} \right) \cap \left(\bigcup_{\substack{i=1 \\ i \neq i_1, \dots, i_r}}^n K_i \right),$$

dann enthält $\bigcup_{k=1}^r K_{i_k}$ bereits r (nicht zwingend verschiedene) Eigenwerte von A

Hinweis: Die Eigenwerte einer Matrix hängen stetig von den Koeffizienten der Matrix ab.

48. Sei $p(x) = a_0x^n + a_1x^{n-1} + \dots + a_n$. Bestimme den Rechenaufwand für die Berechnung von $p(x)$, falls man gerade durchrechnet, d.h. ohne Horner-Schema.

49. Seien $A \in \mathbb{R}^{n,m}$, $v, w \in \mathbb{R}^n$. Vergleiche den Rechenaufwand folgender Algorithmen zur Berechnung von $B = (I - vw^T)A$.

(a) $P := I - vw^T, B := PA$

(b) $z := w^T A, B := A - vz$

Unterscheidet sich der Rechenaufwand wesentlich?

50. Sei p Polynom n -ten Grades. Zeige:

(a) Zu gegebenem $x \in \mathbb{R}$ lässt sich $p(x)$ mit dem Horner-Schema in $2n$ flops berechnen.

(b) Zu gegebenem $x \in \mathbb{R}$ lassen sich $p(x), p'(x)$ mit einem angepassten Horner-Schema in $4n - 2$ flops berechnen, d.h. es werden gegenüber der Berechnung von $p(x)$ zusätzlich $2n - 2$ flops für die Berechnung von $p'(x)$ benötigt (Tip: Satz 3.43 Nullstellenbestimmung für Polynomelemente.3.43 der Vorlesung).

51. Schreibe ein **MATLAB**-Programm für das Newtonverfahren zur Bestimmung der reellen Nullstellen einer rationalen Funktion $f(x) = p(x)/q(x)$, wobei p, q Polynome seien. Vergleiche das Programm bezüglich Rechenaufwand und Genauigkeit mit der Funktion 'roots' aus **MATLAB** anhand des Beispiels $p_n(x) = \prod_{i=1}^n (x - i)$, $n = 2, 4, 6, \dots, 20$, $q(x) = 1$.

52. Sei $p(x) = a_0x^n + a_1x^{n-1} + \dots + a_n$ Polynom n -ten Grades mit $n \geq 3$. Zeige:

(a) Bestimme in Abhängigkeit von gegebenen $r, q \in \mathbb{R}$ nacheinander die Koeffizienten b_0, \dots, b_n , so dass

$$p(x) = q(x)(x^2 - rx - q) + b_{n-1}(x - r) + b_n.$$

Dabei ist $q(x) = b_0x^{n-2} + b_1x^{n-3} + \dots + b_{n-2}$. Was stellst Du fest?

(b) Setze $c_{j-1} = \frac{\partial b_j}{\partial r}$, $d_{j-2} = \frac{\partial b_j}{\partial q}$, $j = 0, \dots, n$. Dann gilt für die partiellen Ableitungen der Koeffizienten b_0, \dots, b_n :

$$c_{-1} = 0, \quad c_0 = b_0, \quad c_{j-1} = b_{j-1} + rc_{j-2} + qc_{j-3}, \quad j = 2, \dots, n$$

sowie

$$d_j = c_j, \quad j = -1, \dots, n.$$

Was kostet die Berechnung der partiellen Ableitungen von b_{n-1}, b_n ?

(c) Das 2-dimensionale Newton-Verfahren zur Bestimmung der Nullstellen von $\Phi(r, q) = (b_{n-1}, b_n)$ lautet

$$\begin{pmatrix} r_{k+1} \\ q_{k+1} \end{pmatrix} = \begin{pmatrix} r_k \\ q_k \end{pmatrix} - \begin{pmatrix} c_{n-2} & c_{n-3} \\ c_{n-1} & c_{n-2} \end{pmatrix}^{-1} \begin{pmatrix} b_{n-1} \\ b_n \end{pmatrix}.$$

53. (a) Sei $A \in \mathbb{R}^{n,n}$, $A = S - fg^T$ mit $f, g \in \mathbb{R}^n$ und $S \in \mathbb{R}^{n,n}$ nicht singulär. Setze $S_c = 1 - g^T S^{-1} f$. Dann ist A nicht singulär genau dann, wenn $S_c \neq 0$ ist und es gilt (Sherman-Morrison-Formel):

$$A^{-1} = S^{-1} + S^{-1} f \frac{1}{S_c} g^T S^{-1}$$

Tip: multipliziere die rechte Seite an A heran.

(b) Was kostet das Lösen von m linearen Gleichungssystemen mit A verglichen mit dem Lösen m linearer Gleichungssysteme mit S ?

(c) Lässt sich die Formel auf Blöcke verallgemeinern? D.h. $F, G \in \mathbb{R}^{n,r}$.

54. Ist $A(x)$ stetige nichtsinguläre Matrixwertige Funktion, dann gilt das auch für ihre Inverse. Tip: Cramersche Regel.

55. Seien $x_0, \dots, x_n \in \mathbb{R}$ paarweise verschieden und $f_0, \dots, f_n \in \mathbb{R}$ zugehörige Funktionswerte. Definiere die dividierten Differenzen $f[x_i, x_{i+1}, \dots, x_{i+k}]$ rekursiv durch

$$f[x_i] = f_i, \quad i = 0, \dots, n,$$

sowie

$$f[x_i, x_{i+1}, \dots, x_{i+k}] := \frac{f[x_{i+1}, x_{i+2}, \dots, x_{i+k}] - f[x_i, x_{i+1}, \dots, x_{i+k-1}]}{x_{i+k} - x_i},$$

$i = 0, \dots, n, k = 0, \dots, n - i$. Nehme an, daß

$$f[x_k], f[x_{k-1}, x_k], f[x_{k-2}, x_{k-1}, x_k], \dots, f[x_0, \dots, x_k]$$

bereits vorhanden sind. Wie kann man daraus $f[x_0, \dots, x_{k+1}]$ berechnen? Was folgt daraus insgesamt für die Berechnung von $f[x_n], f[x_{n-1}, x_n], f[x_{n-2}, x_{n-1}, x_n], \dots, f[x_0, \dots, x_n]$?

56. Seien $x_0, \dots, x_n \in \mathbb{R}$ paarweise verschieden und $f_0, \dots, f_n \in \mathbb{R}$ zugehörige Funktionswerte und $p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0$ das eindeutige Polynom n -ten Grades mit $p(x_i) = f_i, i = 0, \dots, n$.

- Bestimme den Rechenaufwand (flops) zur Bestimmung von p für die Newtondarstellung von p (d.h. die dividierten Differenzen) sowie für die Berechnung der Koeffizienten von p durch Lösen des linearen Gleichungssystems $Va = f$. Dabei ist V die Matrix aus Aufgabe 3 und $a = (a_0, \dots, a_n)^T, f = (f_0, \dots, f_n)^T$.
- Wie teuer ist die Auswertung von p in der Newtondarstellung bei gegebenen dividierten Differenzen, bzw. bei gegebenen Koeffizienten a_0, \dots, a_n verglichen mit der Lagrange-Darstellung von p ?
- Welche der drei Varianten ist die billigste, wenn man $p(x)$ ein einziges Mal auswerten will. Wie sieht es für eine große Anzahl von Werten x (Größenordnung n) aus?

57. Zeige, daß

$$\frac{(b-x)^2}{(b-a)^2} \left[f_1 \frac{2x+b-3a}{b-a} + s_1(x-a) \right] + \frac{(a-x)^2}{(a-b)^2} \left[f_2 \frac{2x+a-3b}{a-b} + s_2(x-b) \right]$$

das eindeutige Polynom dritten Grades ist, welches für $a < b$ und vorgegebene Werte $f_1, s_1, f_2, s_2 \in \mathbb{R}$ die Interpolationsaufgabe

$$p(a) = f_1, p'(a) = s_1, p(b) = f_2, p'(b) = s_2$$

erfüllt ('Hermite-Interpolierende').

58. Seien $x_0, \dots, x_n \in \mathbb{R}$ und

$$V = \begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix}.$$

Zeige, dass V (Vandermonde-Matrix) invertierbar ist genau dann, wenn x_0, \dots, x_n paarweise verschieden sind.

59. Zu vorgegebenen Stützstellen $t_0 < t_1 < \dots < t_n$ mit Funktionswerten f_0, \dots, f_n ist das Neville-Schema zur Berechnung des zugehörigen Interpolationspolynoms p an der Stelle x wie folgt definiert: $P_{i,0} := f_i, n \geq i \geq 0$.

Für $k = 1, \dots, n$ setze $P_{i,k} := \frac{(x - t_{i-k})P_{i,k-1} - (x - t_i)P_{i-1,k-1}}{t_i - t_{i-k}}, n \geq i \geq k$.

Dann ist $p(x) = P_{n,n}$.

- Lässt sich das Neville-Schema in einem ähnlichen Schema wie die dividierten Differenzen aufschreiben?

- b) Wie teuer ist die Berechnung von $P_{n,n}$ im günstigsten Fall und ist dies günstiger oder teurer als die Newton-Darstellung mittels dividierter Differenzen?
- c) Nehme an, dass $x = 0$, $t_i = h^2 4^{-i}$, $i = 0, \dots, n$. Schreibe das Neville-Schema für diesen Fall um.

60. Seien $t_{-3} < t_{-2} < \dots < t_{n+2} < t_{n+3}$ und $t_{i+1} - t_i = h$ für alle $i = -3, \dots, n+2$. Definiere für $i = -1, \dots, n+1$ die Funktion

$$B_i(t) = \frac{1}{h^3} \begin{cases} (t - t_{i-2})^3 & t \in [t_{i-2}, t_{i-1}) \\ h^3 + 3h^2(t - t_{i-1}) + 3h(t - t_{i-1})^2 - 3(t - t_{i-1})^3 & t \in [t_{i-1}, t_i) \\ h^3 + 3h^2(t_{i+1} - t) + 3h(t_{i+1} - t)^2 - 3(t_{i+1} - t)^3 & t \in [t_i, t_{i+1}) \\ (t_{i+2} - t)^3 & t \in [t_{i+1}, t_{i+2}) \\ 0 & \text{sonst} \end{cases}$$

Zeige:

- (a) $B_i(t)$ ist zweimal stetig differenzierbar.
- (b) Skizziere $B_i(t)$.
- (c) Betrachte für $p(t) = \sum_{i=-1}^{n+1} a_i B_i(t)$ und Funktionswerte f_0, \dots, f_n die Interpolationsaufgabe $p(t_i) = f_i$, $i = 0, \dots, n$, $p''(t_0) = 0 = p''(t_n)$. Lässt sich die Aufgabe eindeutig lösen und wenn ja, wie teuer ist die Berechnung von a_0, \dots, a_n ? Wie teuer ist die Auswertung von $p(x)$ bei gegebenen a_0, \dots, a_n ?
- (d) Was ändert sich bei der abgewandelten Aufgabe $p(t_i) = f_i$, $i = 0, \dots, n$, $p'(t_0) = f'_0$, $p'(t_n) = f'_n$ bei zusätzlichen Funktionswerten f'_0, f'_n ?

61. Schreibe ein **MATLAB**-Programm, welches für beliebige Vorgabe $t_0 < t_1 < \dots < t_n$ von Stützstellen mit Funktionswerten f_0, \dots, f_n das zugehörige Interpolationspolynom p an der Stelle x mit Hilfe des Neville-Schemas auswertet, welches wie folgt definiert ist:

$$P_{i,0} := f_i, \quad n \geq i \geq 0.$$

$$\text{Für } k = 1, \dots, n \text{ setze } P_{i,k} := \frac{(x - t_{i-k})P_{i,k-1} - (x - t_i)P_{i-1,k-1}}{t_i - t_{i-k}}, \quad n \geq i \geq k.$$

Dann ist $p(x) = P_{n,n}$. Wende das Programm auf $f(t) = \log_{10}(t) - \frac{t-1}{t}$, $x = 5.25$ und folgende Stützstellenverteilung an und interpretiere die Ergebnisse.

- a) 1.0,2.0,4.0,8.0,10.0
- b) 2.0,4.0,8.0,10.0
- c) 4.0,8.0,10.0
- d) 2.0,4.0,8.0

62. Bestimme mit dem Newtonschema die Interpolationspolynome p zu den Funktionen $f(t) = \frac{1}{1+25t^2}$ und $g(t) = \sqrt{|t|}$. Als Stützstellen verwende jeweils

- i) $t_i = -1 + ih$, $i = 0, \dots, n$; $h = \frac{2}{n}$,
- ii) $t_i = \cos \frac{(2i+1)\pi}{2(n+1)}$, $i = 0, \dots, n$,

für $n = 2, 4, 6, \dots, 20$. Werte das Polynom an den Stellen $y_j^i = t_i + j \frac{t_{i+1} - t_i}{21}$, $j = 1, \dots, 20$ aus. Vergleiche als Schätzung für den maximalen Fehler $\|p - f\|_{\infty [-1,1]}$

$$\max_{i \in \{0, \dots, n-1\}, j \in \{1, \dots, 20\}} |p(y_j^i) - f(y_j^i)|.$$

Plotte die Interpolationspolynome zu den Stützstellen in i), ii) für $n = 20$. Erläutere die Ergebnisse.

63. Gegeben sei die Funktion $f : \mathbb{R} \mapsto \mathbb{R}$ durch

$$f|_{[0,2\pi)}(t) := \begin{cases} 1 & \text{für } 0 \leq t < \pi \\ -1 & \text{für } \pi \leq t < 2\pi \end{cases}$$

und $f(t + 2k\pi) = f(t)$ für alle $k \in \mathbb{Z}$. (Rechteckimpuls erster Art)

i) Bestimme die zu f gehörige Fourierreihe

$$\frac{1}{2}a_0 + \sum_{k=1}^{\infty} (a_k \cos kx + b_k \sin kx), \quad a_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos kx \, dx, \quad b_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \sin kx \, dx.$$

ii) Zeichne (oder Plotte) die ersten fünf Partialsummen und vergleiche mit f .

64. Seien $x_0, x_1, x_2 \in \mathbb{R}$ mit $x_0 \neq x_2$. Zeige:

(a) es gibt eindeutige Polynome p_1, p_2, p_3, p_4 dritten Grades, welche den Bedingungen

$$\begin{array}{cccc} p_1(x_0) = 1 & p_2(x_0) = 0 & p_3(x_0) = 0 & p_4(x_0) = 0 \\ p_1(x_2) = 0 & p_2(x_2) = 1 & p_3(x_2) = 0 & p_4(x_2) = 0 \\ p'_1(x_1) = 0 & p'_2(x_1) = 0 & p'_3(x_1) = 1 & p'_4(x_1) = 0 \\ p''_1(x_1) = 0 & p''_2(x_1) = 0 & p''_3(x_1) = 0 & p''_4(x_1) = 1 \end{array}$$

genügen.

(b) Es existiert ein eindeutiges Polynom p dritten Grades gibt, daß für die Werte f_0, f_2, f'_1, f''_1 die Interpolationsaufgabe

$$p(x_0) = f_0, \quad p(x_2) = f_2, \quad p'(x_1) = f'_1, \quad p''(x_1) = f''_1$$

erfüllt.

65. Bestimme für eine auf dem Intervall $[a, b]$ stetig differenzierbare Funktion f eine Quadraturformel für $\int_a^b f(t) \, dt$, indem Du Stützstellen $a = x_0 < x_1 < \dots < x_n = b$ einführst und auf dem Intervall $[x_i, x_{i+1}]$ die Funktion f durch die (stückweise) Hermite-Interpolierende aus Übung 12, Nr.2 ersetzt (neben $f(x_0), \dots, f(x_n)$ sollen auch $f'(x_0), \dots, f'(x_n)$ gegeben sein).

66. Sei $f : [a, b] \rightarrow \mathbb{R}$ eine zweimal stetig differenzierbare Funktion.

a) Sei F Stammfunktion zu f , d.h. $F' = f$. Seien $x, h \in \mathbb{R}$ mit $[x-h, x+h] \subseteq [a, b]$. Zeige:

$$F(x + \frac{h}{2}) - F(x - \frac{h}{2}) = hf(x) + \frac{h^3}{24} \frac{f''(\theta_+) + f''(\theta_-)}{2}$$

für geeignet zu wählende $\theta_+, \theta_- \in [a, b]$.

b) Zeige: Sei $l \in \mathbb{N} \setminus \{0\}$, $\theta_1, \dots, \theta_l \in [a, b]$. Dann existiert ein $\theta \in [a, b]$, so daß

$$f''(\theta) = \frac{1}{l} \sum_{k=1}^l f''(\theta_k)$$

ist.

c) Sei $z_i = a + ih$, $i = 0, \dots, N$, $h = \frac{b-a}{N}$. Zeige:

$$\int_{z_i}^{z_{i+1}} f(t) dt - hf\left(\frac{z_i + z_{i+1}}{2}\right) = \frac{h^3}{24} f''(\theta_i)$$

für ein geeignetes $\theta_i \in [a, b]$, $i = 0, \dots, N-1$.

d) Sei $M(h) = h \sum_{i=0}^{N-1} f\left(\frac{z_i + z_{i+1}}{2}\right)$. (Summierte Mittelpunkregel) Zeige:

$$\int_a^b f(t) dt - M(h) = (b-a) \frac{h^2}{24} f''(\theta),$$

wobei $\theta \in [a, b]$ geeignet zu wählen ist.

67. Bestimme für eine auf dem Intervall $[a, b]$ stetig differenzierbare Funktion f eine Quadraturformel für $\int_a^b f(t) dt$, indem Du Stützstellen $a = x_0 < x_1 < \dots < x_n = b$ einführst und f durch den kubischen Spline S ersetzt (neben $f(x_0), \dots, f(x_n)$ sollen auch $f'(x_0)$ und $f'(x_n)$ gegeben sein). Tipp: Interpretiere S als Hermite-Interpolierende, bei der $s_i = S'(x_i)$, $0 < i < n$ als Unbekannte auftritt.

68. Verwende zur Berechnung der Integrale

$$\int_0^{\frac{\pi}{2}} \sin x dx, \int_0^1 \sqrt{x} dx, \int_0^1 x\sqrt{x} dx, \int_0^1 x^2\sqrt{x} dx$$

a) die summierte Trapezregel für $N = 1, 2, 4, 8, 16, \dots, 2^9$ und folgende Unterteilung:

$$x_i = i \cdot h, \quad i = 0, 1, \dots, N, \quad h = \frac{\pi}{2N}.$$

b) die summierte Mittelpunkregel für $N = 1, 2, 4, 8, 16, \dots, 2^9$ und der Unterteilung aus a).

c) die summierte Simpsonregel für $N = 2, 4, 8, 16, \dots, 2^9$ und der Unterteilung aus a).

Vergleiche die Verfahren bezüglich Genauigkeit und Rechenaufwand.

69. Verwende zur Berechnung des Integrals aus Aufgabe 68 das Extrapolationsverfahren zur summierten Trapezregel für $k = 0, 1, 2, \dots, 9$ mit der Rombergfolge

$$h_l = \frac{\pi}{2^{l+1}}, \quad l = 0, \dots, k.$$

Untersuche das Verfahren bezüglich Genauigkeit und Rechenzeit.

70. Zeige den Satz von Bauer und Fike: Sei $A = (a_{ij})_{i,j=1,\dots,n} \in \mathbb{C}^{n,n}$ diagonalisierbar, d.h., nehme an, daß $A = X\Lambda X^{-1}$ ist, wobei $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ ist und X nicht

singulär ist. Sei $E \in \mathbb{C}^{n,n}$ irgend eine Matrix. Dann existiert zu jedem Eigenwert μ von $A + E$ ein Eigenwert λ_j von A derart, daß

$$|\lambda_j - \mu| \leq \|X\|_p \|X^{-1}\|_p \|E\|_p.$$

Dabei ist $\|\bullet\|_p$ die übliche induzierte Matrixnorm zur p -Norm für Vektoren.

Hinweis: Betrachte o.B.d.A. $\mu \neq \lambda_j$ und betrachte $(\mu I - \Lambda)X^{-1}v$ für ein geeignetes v .

71. Schreibe ein **MATLAB**-Programm für den QR -Algorithmus für symmetrische Matrizen einmal mit Wilkinson-Shift und einmal ohne Shift (Der Wilkinson-Shift einer symmetrischen Matrix A ist der Eigenwert von $\begin{pmatrix} a_{n-1,n-1} & a_{n-1,n} \\ a_{n-1,n} & a_{nn} \end{pmatrix}$, welcher dichter bei a_{nn} gelegen ist). Für die Reduktion auf symmetrische Tridiagonalgestalt darf die entsprechende Routine 'hess' aus **MATLAB** verwendet werden. Für die Iteration reicht die explizite Variante.

Vergleiche beide Varianten bezüglich Rechenzeit und Genauigkeit anhand der Matrix

$$A = \begin{pmatrix} 2 & -1 & & & \\ -1 & \ddots & \ddots & & \\ & \ddots & \ddots & -1 & \\ & & & -1 & 2 \end{pmatrix} \text{ für Dimension } n = 10, 20, \dots, 100.$$

72. Erläutere den Begriff Kondition eines Problems. Was bedeutet das speziell bei linearen Gleichungssystemen? Wie wirkt sich eine schlechte Kondition auf die Genauigkeit des Ergebnisses aus?
73. Erläutere den Gaußschen Eliminationsalgorithmus. Wozu macht man partielle Pivottisierung und wie wirkt sich das auf die Genauigkeit des Ergebnisses aus?
74. Erläutere die Grundidee des Cholesky-Algorithmus. Muss man hier pivotisieren? Wenn nein, wie sehen die Fehlerschranken aus?
75. Erläutere die QR -Zerlegung. Welche Varianten gibt es und wie unterscheiden sie sich? Was sind Vorteile/Nachteile gegenüber Gaußelimination?
76. Erläutere die Grundidee iterativer Verfahren. Wann liegt Konvergenz vor? Gibt es Situationen, wann iterative Verfahren den direkten Verfahren vorzuziehen sind?
77. Beschreibe das Newtonverfahren. Was weiss man über Konvergenz? Was ist grundsätzliche Unterschied beim modifizierten Newtonverfahren und wie wirkt sich das auf die Konvergenz aus?
78. Erläutere den Begriff Lagrange-Interpolation. Wie kann man die Interpolierende in der Praxis berechnen? Welche Vorteile/Nachteile hat die Polynominterpolation und was kann man dagegen machen?
79. Erkläre die Idee der trigonometrischen Interpolation. Wie kann man die Koeffizienten effizient berechnen? Wo ist der Unterschied zu Fourierreihen/Fourierpolynomen?
80. Beschreibe die Spline-Interpolation. Wie kann man die kubische Splineinterpolierende berechnen? Wo liegt der wesentliche Unterschied zur Polynominterpolation und was weiss man über die Approximationseigenschaften?

81. Beschreibe die Grundidee der numerischen Integration. Wie unterscheiden sich Integration mit Hilfe von Polynomen gegenüber stückweise Polynomen. Wie kann man die Qualität der Approximation verbessern?
82. Gib für das Beispiel aus der Vorlesung zur Berechnung von $x^* = f(a_1, a_2) = a_1 - \sqrt{a_1^2 - a_2}$ einen besseren Algorithmus an. Begründung!

83. Zeige Satz 2.31 Fehleranalyse der Gauß-Elimination Lemma.2.31 aus der Vorlesung: Unter den Voraussetzungen von Lemma 2.30 Fehleranalyse der Gauß-Elimination Lemma.2.30 gilt

$$\frac{\|y - x\|}{\|x\|} \leq \frac{2r}{1 - r},$$

wobei $\|\bullet\| = \|\bullet\|_\infty, \|\bullet\|_1, \|\bullet\|_F$. Tip: Verfahre analog zum Beweis des entsprechenden Satzes 2.27 Fehleranalyse der Gauß-Elimination Lemma.2.27 über Normen.

84. Sei $c \in \mathbb{C}^n, J \in \mathbb{C}^{n \times n}$ definiert durch

$$c := \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{pmatrix}, \quad J := \begin{pmatrix} 0 & & & 1 \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & & 1 & 0 \end{pmatrix}$$

Definiere die Matrix $C \in \mathbb{C}^{n \times n}$ durch

$$C := (c, Jc, J^2c, \dots, J^{n-1}c)$$

(C bezeichnet man auch als Zirkulante).

Sei $\omega_n := e^{-i\frac{2\pi}{n}}$ und $W \in \mathbb{C}^{n \times n}$ definiert als $W := \frac{1}{\sqrt{n}} (\omega_n^{kl})_{k,l=0,\dots,n-1}$. Zeige:

- i) $W^*W = I$, d.h. W ist unitär.
- ii) W^*CW ist eine Diagonalmatrix. (Tip: Zeige zuerst, daß W^*JW diagonal ist und nutze dann die Def. von C).
- iii) Was sind die Eigenwerte von C und wie lassen sie sich berechnen?
- iv) Sei $n = 2^k$. Zeige, daß sich das Gleichungssystem $Cx = b, b \in \mathbb{C}^n$ gegeben, durch drei FFT berechnen läßt. Bestimme die wesentliche Rechenzeit.

85. Gegeben die Werte

$$\begin{array}{cccc} y_0 = 0.580 & y_1 = 0.951 & y_2 = 0.786 & y_3 = 0.298 \\ y_4 = 0.454 & y_5 = 0.006 & y_6 = 0.276 & y_7 = 0.306 \end{array}$$

Führe eine FFT durch, d.h. berechne

$$c_k = \sum_{m=0}^{n-1} \omega_n^{km} y_m, \quad \text{für } k = 0, \dots, n-1$$

mittels FFT, wobei $\omega_n := e^{-i\frac{2\pi}{n}}$.

86. Gegeben sei eine nichtsinguläre untere Dreiecksmatrix $L \in \mathbb{R}^{n,n}$.

- (a) Nehme an, dass $y \in \mathbb{R}^n$ gegeben ist und x die Gleichung $Lx = y$ erfüllt. Zeige: falls $c\|y\|_p \leq \|x\|_p$ gilt, so ist $\|L^{-1}\|_p \geq c$.
- (b) Nutze (86a) zur Entwicklung eines Konditionsschätzers. Wähle ein y der Form $y^\top = (\pm 1 \ \dots \ \pm 1)$ und betrachte folgende Variante des Vorwärtseinsetzens

$p = 0 \in \mathbb{R}^n$

for $k = 1 : n$

 Wähle das Vorzeichen von $y(k)$

$x(k) = (y(k) - p(k))/L(k, k)$

$p(k+1 : n) = p(k+1 : n) + L(k+1 : n, k)x(k)$

end

$\kappa = \|x\|_\infty \|L\|_\infty$

- i. Schreibe ein MATLAB-Programm, in dem y so gewählt wird, dass für $k = 1, \dots, n$, $|x(k)|$ sukzessive maximiert wird.
- ii. Schreibe ein MATLAB-Programm, in dem y so gewählt wird, dass für $k = 1, \dots, n$, $|x(k)| + \|p(k+1 : n) + L(k+1 : n, k)x(k)\|_1$ sukzessive maximiert wird.
- iii. Vergleiche beide Varianten in MATLAB mit der eingebauten Funktion **cond** sowie der exakten Kondition **cond**. Was stellst du fest?

87. Gegeben sei eine symmetrische Tridiagonalmatrix

$$A = \begin{pmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_n \\ & & \beta_n & \alpha_n \end{pmatrix} \in \mathbb{R}^{n,n}.$$

Zeige: Das charakteristische Polynom $p_n(t) = \det(A - tI)$ von A lässt sich rekursiv berechnen durch

$$p_0(t) = 1$$

$$p_1(t) = \alpha_1 - t$$

$$p_j(t) = (\alpha_j - t)p_{j-1}(t) - \beta_j^2 p_{j-2}(t), \quad j = 2, 3, \dots, n.$$

88. Für eine Matrix $A \in \mathbb{R}^{m,n}$ ist die Matrix p -Norm definiert durch

$$\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p} = \max_{\|x\|_p=1} \|Ax\|_p$$

- (a) $\|A\|_p$ ist eine Norm
- (b) $\|Ax\|_p \leq \|A\|_p \|x\|_p$
- (c) Ist c eine Konstante, so dass $\|Ax\|_p \leq c\|x\|_p$ für alle x gilt, so ist $c \geq \|A\|_p$, d.h. $c = \|A\|_p$ ist die kleinst mögliche Konstante, so dass $\|Ax\|_p \leq c\|x\|_p$ gilt.

89. Sei $A \in \mathbb{R}^{n,n}$ eine obere Hessenbergmatrix, d.h. $a_{ij} = 0$ für alle $i > j + 1$. Zeige: die Elemente der Matrix $A^{(k)}$ die durch k Schritte des Gauß-Algorithmus mit partieller Pivotisierung entsteht, erfüllen die Ungleichung

$$\max_{i,j} |a_{ij}^{(k)}| \leq (k+1) \max_{i,j} |a_{ij}|.$$

Was bedeutet das für den Wachstumsfaktor $\rho = \max_{i,j,k} \frac{|a_{ij}^{(k)}|}{\|A\|_\infty}$?

Tipp: Verwende Induktion über die Anzahl Eliminationsschritte k und zeige, dass nach dem k -ten Eliminationsschritt gilt $\max_{i \leq k+1, j} |a_{ij}^{(k)}| \leq (k+1) \max_{i,j} |a_{i,j}|$; $a_{ij}^{(k)} = a_{ij}$, $i > k+1$; $a_{i+1,i}^{(k)} = 0$, $i = 1, \dots, k$.

90. Schreibe ein **MATLAB**-Programm für die GAXPY-Version des Gauß-Algorithmus der Vorlesung mit partieller Pivotisierung. Verwende als Testbeispiel $Hx = b$ die Hilbertmatrix H , $x = (1, \dots, 1)^\top$ für $n = 2, 3, 4, \dots, 30$.

91. Sei

$$A = \begin{pmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix} \in \mathbb{R}^{n,n}.$$

Zeige:

- (a) A besitzt eine LU -Zerlegung
- (b) partielle Pivotisierung führt zu keiner Änderung.
- (c) Es besteht die Beziehung $U = DL^\top$. Was bedeutet das für die LU -Zerlegung?

92. Sei $A \in \mathbb{R}^{n,n}$, symmetrisch positiv definit, $A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$ und nehme an, dass $A_{11} \in \mathbb{R}^{k,k}$ ist. Das Schur-Komplement ist gegeben durch $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$. Zeige:

- (a) A läßt sich schreiben als

$$A = L \begin{pmatrix} A_{11} & O \\ O & S \end{pmatrix} L^\top, \text{ wobei } L = \begin{pmatrix} I & O \\ A_{21}A_{11}^{-1} & I \end{pmatrix}.$$

- (b) Ist $M \in \mathbb{R}^{n,m}$ mit $\text{rang } M = m$, dann ist $M^\top AM$ symmetrisch positiv definit.
- (c) S ist symmetrisch positiv definit

93. Sei $A \in \mathbb{R}^{n,n}$ symmetrisch positiv definit und $A = LL^\top$ die Cholesky-Zerlegung von A . Zeige:

- (a) Für $i = 1, \dots, n$ gilt

$$l_{ii}^2 \geq \min_{x \neq 0} \frac{x^\top Ax}{x^\top x} = \frac{1}{\|L^{-\top}\|_2^2}.$$

- (b) Für $i = 1, \dots, n$ gilt

$$\|L^\top\|_2^2 = \max_{x \neq 0} \frac{x^\top Ax}{x^\top x} \geq l_{ii}^2.$$

- (c) Für die Konditionszahl in der 2-Norm gilt

$$\text{cond}_2(L^\top) \geq \max_{1 \leq i, k \leq n} \frac{|l_{ii}|}{|l_{kk}|}.$$

94. Untersuche in **MATLAB** die Genauigkeitsverbesserung mittels iterativer Verbesserung bei der GAXPY-Version des Gauß-Algorithmus mit partieller Pivotisierung dadurch, dass Daten und Rechenoperationen mit zufälligen Werten der Form $(1 + \varepsilon)$ gestört werden, wobei $|\varepsilon| \leq \sqrt{\text{eps}}$ ist (einfache statt doppelte Genauigkeit). Vergleiche die iterative Verbesserung mit und ohne gestörte Berechnung des Residuums. Verwende als Testbeispiel $Hx = b$ wieder die Hilbertmatrix H , $x = (1, \dots, 1)^T$ für $n = 2, 3, 4, \dots, 30$.

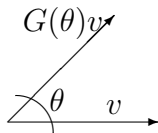
95. Schreibe einen **MATLAB**-Algorithmus zur Berechnung der Cholesky-Zerlegung. Vergleiche den Algorithmus bezüglich Rechengenauigkeit und Rechenzeit mit der Funktion 'chol' und dem '\' aus **MATLAB** für $n = 10, 20, 30, 40, \dots, 100$ und zufällig erzeugten positiv definiten Matrizen $A = MDM^T$, $M = \text{rand}(n)$, $D = \text{diag}(1, 2, 4, 9, \dots, n^2)$.

96. Zeige:

- (a) Für jedes $V \in \mathbb{C}^{n,m}$, $\text{Rang } V = m$ ist $S = I - 2V(V^H V)^{-1}V^H$ Hermitesch und unitär und es ist $S^2 = I$.
- (b) S ist eine Spiegelung am orthogonalen Komplement von Bild V .
- (c) S hat nur die Eigenwerte -1 m -fach, 1 $n - m$ -fach.
- (d) Sei $x \in \mathbb{C}^n \setminus \{0\}$ und schreibe die erste Komponente x_1 von x in Polarkoordinaten als $x_1 = |x_1| e^{i\theta}$. Setze $v = x + e^{i\theta} \|x\|_2 e_1$. Dann ist $S = I - \frac{2}{v^H v} v v^H$ eine Spiegelung mit $Sx = -e^{i\theta} \|x\|_2 e_1$.

97. Sei $G(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \equiv \begin{pmatrix} c & -s \\ s & c \end{pmatrix}$. Zeige:

- (a) die Abbildung G beschreibt eine Drehung um den Winkel θ gegen den Uhrzeigersinn in der Ebene.



- (b) Für vorgegebenes $v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$ bewirkt die Wahl

$$c = \frac{v_1}{\sqrt{v_1^2 + v_2^2}}, \quad s = \frac{-v_2}{\sqrt{v_1^2 + v_2^2}},$$

dass $Gv = \begin{pmatrix} * \\ 0 \end{pmatrix}$ ist (Givens-Rotation). Wie sollten c, s berechnet werden, um ein Höchstmaß an numerischer Stabilität zu sichern? Begründung!

- (c) Sei $v \in \mathbb{R}^n$. Dann lässt sich v durch Anwendung von $n - 1$ eingebetteten Givensrotationen aus Teil 97b auf ein Vielfaches von e_1 transformieren.

98. Schreibe einen **MATLAB**-Algorithmus zur Berechnung der QR-Zerlegung mittels Givens-Rotationen aus Aufgabe 97b.

Vergleiche den Algorithmus bezüglich Rechengenauigkeit und Rechenzeit mit der Funktion 'qr' aus **MATLAB** für $n = 10, 20, 30, 40, \dots, 100$ und zufällig erzeugten Matrizen

99. Sei $G = \begin{pmatrix} c & -s \\ s & c \end{pmatrix}$ orthogonal und nehme an, dass wir für vorgegebenes $v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$, c, s so bestimmen können, dass $Gv = \begin{pmatrix} * \\ 0 \end{pmatrix}$ ist. Zeige:

Ist A eine obere Hessenbergmatrix (d.h. $a_{i,j} = 0$ für alle $i > j + 1$), dann besitzt A eine QR -Zerlegung, die in $3n^2 + \mathcal{O}(n)$ flops berechnet werden kann (ohne explizite Berechnung von Q).

Welches Belegungsmuster hat Q ?

100. Sei $A \in \mathbb{R}^{n,n}$ symmetrisch, positiv definit und $b \in \mathbb{R}^n$. Definiere $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}$ durch $\Phi(x) = \frac{1}{2}x^\top Ax - x^\top b$. Zeige:

(a) Φ besitzt ein eindeutiges Minimum in $x = A^{-1}b$

(b) Sei $x_0 \in \mathbb{R}^n$ und $P = [p_1, \dots, p_k] \in \mathbb{R}^{n,k}$ so, dass $P^\top AP = D = \text{diag}(d_1, \dots, d_k)$ mit von Null verschiedenen d_1, \dots, d_k . Dann besitzt die Funktion $\Psi : \mathbb{R}^k \rightarrow \mathbb{R}$, $\Psi(v) = \Phi(x_0 + Pv)$ ein eindeutiges Minimum in $v = D^{-1}P^\top(b - Ax_0) = \left(\frac{p_i^\top (b - Ax_0)}{d_i} \right)_{i=1, \dots, k}$

101. Sei $A \in \mathbb{R}^{n,n}$ symmetrisch, positiv definit und $b, x_0 \in \mathbb{R}^n$. Betrachte das cg-Verfahren ohne Vorkonditionierung aus der Vorlesung (zur besseren Analyse mit Indizes).

$k = 0, r_0 = b - Ax_0, \rho_0 = r_0^\top r_0$

while $\sqrt{\rho_k} > \varepsilon \sqrt{\rho_0}$ **and** $k < k_{\max}$

$k = k + 1$

if $k = 1$

$p_1 = r_0$

else

$\beta_k = \rho_{k-1} / \rho_{k-2}$

$p_k = r_{k-1} + \beta_k p_{k-1}$

end

$w_k = Ap_k$

$\alpha_k = \rho_{k-1} / w_k^\top p_k$

$x_k = x_{k-1} + \alpha_k p_k$

$r_k = r_{k-1} - \alpha_k w_k$

$\rho_k = r_k^\top r_k$

end

Zeige:

(a) $\text{span}\{r_0, \dots, r_{k-1}\} = \text{span}\{p_1, \dots, p_k\}$

(b) $\text{span}\{r_0, \dots, r_{k-1}\} = \text{span}\{r_0, Ar_0, \dots, A^{k-1}r_0\}$

für alle $k = 1, \dots, n$, sofern der Algorithmus nicht vorher abbricht.

102. Betrachtet wird die Fixpunktaufgabe $x = \phi(x)$, wobei $x = (\xi, \eta, \zeta)^\top$ mit

$$\phi(x) = \frac{1}{6} \begin{pmatrix} \xi e^{-\eta^2} + \xi\zeta + 3 \\ \sin(\pi\xi) + \frac{1}{1+\eta} \\ \ln(1 + \eta^2 + \zeta^2) - 1 \end{pmatrix}$$

auf dem Würfel $I = [0, 1] \times [0, 1] \times [-1, 1]$. Zeige:

(a)

$$D\phi(x) = \frac{1}{6} \begin{pmatrix} e^{-\eta^2} + \zeta & -2\eta\xi e^{-\eta^2} & \xi \\ \pi \cos(\pi\xi) & \frac{-1}{(1+\eta)^2} & 0 \\ 0 & \frac{2\eta}{1+\eta^2+\zeta^2} & \frac{2\zeta}{1+\eta^2+\zeta^2} \end{pmatrix}.$$

- (b) Die Voraussetzung des Banachschen Fixpunktsatzes sind erfüllt für die Lipschitzkonstante $\alpha = 5/6$ bezüglich der Maximumnorm.
- (c) Es seien $x^{(k)}$ die Iterierten der Fixpunktiteration $x^{(k+1)} = \phi(x^{(k)})$ mit Startwert $x^{(0)} = (1/2, 0, 0)^\top$. x^* bezeichne den Fixpunkt von ϕ in I . Wieviele Iterationsschritte sind hinreichend, um

$$\|x^{(k)} - x^*\|_\infty \leq 10^{-3}$$

garantieren zu können?

103. Schreibe ein **MATLAB**-Algorithmus zur Lösung einer nichtlinearen Gleichung $\Phi(x) = 0$ mit Hilfe des Newton-Verfahrens bei gegebenem Startwert $x^{(0)}$ und Ableitungsmatrix $D\Phi(x)$. Teste das Verfahren für die Funktion $\Phi(x) = \phi(x) - x$ mit $\phi(x)$ aus Aufgabe 102 sowie an folgender Funktion Ψ für $n = 10$ mit Startwert $(1, \dots, 1, 0)^\top$.

$$\Psi(x_1, \dots, x_n, x_{n+1}) = \begin{pmatrix} (A - x_{n+1}I) \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \\ \frac{1}{2}(1 - \sum_{i=1}^n |x_i|^2) \end{pmatrix}, \quad A = \begin{pmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix}.$$

Was stellst du fest?

104. Schreibe ein **MATLAB**-Algorithmus zur Lösung einer nichtlinearen Gleichung $\Phi(x) = 0$ mit Hilfe des MODIFIZIERTEN Newton-Verfahrens bei gegebenem Startwert $x^{(0)}$ und Ableitungsmatrix $D\Phi(x)$. Teste das Verfahren für die Funktion $\Phi(x) = \phi(x) - x$ mit $\phi(x)$ aus Aufgabe 102 sowie an folgender Funktion Ψ für $n = 10$ mit Startwert $(1, \dots, 1, 0)^\top$.

$$\Psi(x_1, \dots, x_n, x_{n+1}) = \begin{pmatrix} (A - x_{n+1}I) \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \\ \frac{1}{2}(1 - \sum_{i=1}^n |x_i|^2) \end{pmatrix}, \quad A = \begin{pmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix}.$$

Was stellst du fest?

105. Es sei

$$f(x, y) = \begin{pmatrix} \frac{3x}{1+x} + \frac{1}{y} \\ \frac{3y}{1+y} + \frac{2}{x} \end{pmatrix}.$$

- (a) Man finde eine Teilmenge $I \subseteq \mathbb{R}^2$, in der f bzgl. der $\|\bullet\|_\infty$ -Norm den Voraussetzungen des Banachschen Fixpunktsatzes genügt.
- (b) Man führe mit $(x_0, y_0) = (2, 2)$ zwei Iterationen durch und gebe eine a-posteriori Fehlerschranke für (x_2, y_2) an.
- (c) Wieviele Iterationsschritte sind hinreichend, um ausgehend von $(x_0, y_0) = (2, 2)$ den Fixpunkt mit einer Genauigkeit von 10^{-4} zu berechnen

106. Sei $A \in \mathbb{R}^{n,n}$.

(a) Sei $B \in \mathbb{R}^{n,n}$, $B = \begin{pmatrix} 0 & 0 & b_{1n} \\ 0 & 0 & 0 \\ b_{n1} & 0 & 0 \end{pmatrix}$ Finde $U, V \in \mathbb{R}^{n,2}$, so dass $B = UV^\top$.

(b) Zeige die Sherman–Morrison–Woodbury Formel, d.h. sind A und $A+UV^\top$ nicht singulär, dann ist auch $C = I + V^\top A^{-1}U$ nichtsingulär und es gilt:

$$(A + UV^\top)^{-1} = A^{-1} - A^{-1}UC^{-1}V^\top A^{-1}.$$

(c) Betrachte das lineare Gleichungssystem $(A + B)x = b$, wobei A und $A + B$ nichtsingulär seien mit B aus Teil 106a. Nehme an, dass A tridiagonal ist und A und $A + B$ eine LU -Zerlegung besitzen (ohne Pivotisierung). Wie teuer ist die Lösung von $(A + B)x = b$ wenn man einmal eine LU -Zerlegung von $A + B$ macht verglichen mit der Variante, bei der man zunächst eine LU -Zerlegung von A macht und dann die Sherman–Morrison–Woodbury Formel verwendet.

107. Bestimme die kubischen Splines S_Δ zu den Funktionen $f(t) = \frac{1}{1+25t^2}$ und $g(t) = \sqrt{|t|}$, wobei $S''_\Delta(-1) = S''_\Delta(1) = 0$ (natürliche Splineinterpolation). Als Stützstellen verwende $t_i = -1 + ih$, $i = 0, \dots, n$; $h = \frac{2}{n}$, für $n = 2, 4, 6, \dots, 20$. Werte den Spline an den Stellen $y_j^i = t_i + j \frac{t_{i+1} - t_i}{21}$, $j = 1, \dots, 20$ aus. Vergleiche als Schätzung für den maximalen Fehler $\|S_\Delta - f\|_\infty$ auf $[-1, 1]$

$$\max_{i \in \{0, \dots, n-1\}, j \in \{1, \dots, 20\}} |S_\Delta(y_j^i) - f(y_j^i)|.$$

Plotte die Splines zu den Stützstellen für $n = 20$. Erläutere die Ergebnisse.

108. Zu einer gegebenen stetigen Funktion $p : [-1, 1] \rightarrow \mathbb{R}$ betrachte folgende Quadraturformel

$$\int_{-1}^1 p(x) dx \approx p\left(-\frac{1}{\sqrt{3}}\right) + p\left(\frac{1}{\sqrt{3}}\right).$$

Zeige: Die Formel ist für Polynome p vom Grade höchstens 3 exakt.

Tipp: Zerlege p als $p(x) = L(x)q(x) + r(x)$, wobei $L(x)$ ein quadratisches Polynom ist, welches $\pm \frac{1}{\sqrt{3}}$ als Nullstelle hat und r ein höchstens lineares Polynom ist.

109. Sei $A \in \mathbb{C}^{n,n}$ eine nichtzerfallende obere Hessenbergmatrix und $\mu \in \mathbb{C}$. Zeige:

(a) Falls $A - \mu I = QR$ eine QR -Zerlegung von $A - \mu I$ ist, dann ist Q eine obere Hessenbergmatrix (d.h. $q_{ij} = 0, i > j + 1$). Ist μ ein Eigenwert von A , dann ist $r_{nn} = 0$.

(b) $H = Q^H A Q$ ist eine obere Hessenbergmatrix. Ist μ ein Eigenwert von A , dann ist $h_{n,n-1} = 0$. Was bedeutet das für die letzte Spalte von Q ?

(c) Nehme an A sei reell. Dann ist auch $(A - \mu I)(A - \bar{\mu} I)$ reell.

(d) Sei A reell und $\mu = \alpha + i\beta$ ein Eigenwert von A . Falls $A^2 - 2\alpha A + (\alpha^2 + \beta^2)I = QR$ eine QR -Zerlegung ist, dann ist $\begin{pmatrix} r_{n-1,n-1} & r_{n-1,n} \\ 0 & r_{nn} \end{pmatrix}$ singular.

110. Sei $A \in \mathbb{C}^{n,n}$ obere Hessenbergmatrix und p ein Polynom. Nehme an, dass $p(A)$ nichtsingulär ist. Zeige: Ist $p(A) = QR$ eine QR -Zerlegung von $p(A)$, dann ist $B = Q^H A Q$ eine obere Hessenbergmatrix. Gibt es eine Möglichkeit B zu berechnen, ohne das ganze Polynom $p(A)$ auszurechnen?

111. Schreibe ein **MATLAB**-Programm für die Reduktion auf obere Hessenberggestalt mittels Householder-Transformationen. Vergleiche das Programm bezüglich Genauigkeit und Rechenaufwand mit der **MATLAB**-Routine 'hess' anhand zufällig erzeugter Beispiele der Dimensionen $n = 10, 20, \dots, 100$.

112. Schreibe ein **MATLAB**-Programm für die Reduktion symmetrischer Matrizen auf Tridiagonalgestalt mittels Householder-Transformationen. Vergleiche das Programm bezüglich Genauigkeit und Rechenaufwand mit der **MATLAB**-Routine 'hess' anhand zufällig erzeugter Beispiele der Dimensionen $n = 10, 20, \dots, 100$.

113. Schreibe ein **MATLAB**-Programm für den QR -Algorithmus für symmetrische Matrizen einmal mit Wilkinson-Shift und einmal ohne Shift (Der Wilkinson-Shift einer symmetrischen Matrix A ist der Eigenwert von $\begin{pmatrix} a_{n-1,n-1} & a_{n-1,n} \\ a_{n-1,n} & a_{nn} \end{pmatrix}$, welcher dichter bei a_{nn} gelegen ist). Für die Reduktion auf symmetrische Tridiagonalgestalt darf die entsprechende Routine 'hess' aus **MATLAB** verwendet werden. Für die Iteration soll die implizite Variante verwendet werden.

Vergleiche beide Varianten bezüglich Rechenzeit und Genauigkeit anhand der Matrix

$$A = \begin{pmatrix} 2 & -1 & & & \\ -1 & \ddots & \ddots & & \\ & \ddots & \ddots & -1 & \\ & & & -1 & 2 \end{pmatrix} \text{ für Dimension } n = 10, 20, \dots, 100.$$

114. Schreibe ein **MATLAB**-Programm für den QR -Algorithmus für allgemeine reelle Matrizen mit implizitem Francis-Shift (Eigenwerte der Matrix $\begin{pmatrix} a_{n-1,n-1} & a_{n-1,n} \\ a_{n,n-1} & a_{nn} \end{pmatrix}$). Für die Reduktion auf Hessenberggestalt darf die entsprechende Routine 'hess' aus **MATLAB** verwendet werden.

Vergleiche das Verfahren bezüglich Rechenzeit und Genauigkeit mit der Funktion 'schur' aus **MATLAB** anhand zufällig erzeugter Matrizen der Dimensionen $n = 10, 20, \dots, 100$.

115. Seien $\alpha, \beta \in \mathbb{R}$ und die Funktion $s_{\alpha,\beta} : [-1, 2] \rightarrow \mathbb{R}$ definiert durch

$$s_{\alpha,\beta}(x) = \begin{cases} (x+1)^4 + \alpha(x-1)^4 + 1 & x \in [-1, 0] \\ -x^3 - 8\alpha x + 1 & x \in (0, 1] \\ \beta x^3 + 8x^2 + \frac{11}{3} & x \in (1, 2] \end{cases}$$

Bestimme α, β so, dass $s_{\alpha,\beta}$ ein kubischer Spline bezüglich der Knotenverteilung $\Delta = \{-1, 0, 1, 2\}$ ist.

116. Gegeben sei eine Quadraturformel $\int_{-1}^1 f(x) dx = Q(f) + R_Q(f)$, wobei $Q(f) = \sum_{i=1}^s \omega_i f(\xi_i)$, $-1 \leq \xi_1 < \dots < \xi_s \leq 1$ und $\omega_i \geq 0$, $\sum_{i=1}^s \omega_i = 2$. Sei ausserdem eine stetige $F : [-1, 1]^2 \rightarrow \mathbb{R}$ gegeben.

(a) Konstruiere mit Hilfe der Quadraturformel Q eine Kubaturformel K , d.h. eine Vorschrift zur Berechnung von

$$\int_{-1}^1 \int_{-1}^1 F(x, y) dx dy = K(F) + R_K(F),$$

wobei $R_K(F)$ zugehöriger Approximationsfehler.

- (b) Zeige, dass für den Fehler $R_K(F)$ der Kubaturformel die folgende Abschätzung gilt.

$$R_K(F) \leq 2 \left(\max_{x \in [-1,1]} |R_Q(F(x, \bullet))| + \max_{y \in [-1,1]} |R_Q(F(\bullet, y))| \right).$$