2.1 Motivation: Bildverarbeitung

Sei $B = (B(n, m))$ ein $N \times M$ stochastisches Feld mit ZVen

$$B(n, m) : \Omega \to \{0, \ldots, 255\}, \quad n = 1, \ldots, N, \quad m = 1, \ldots, M.$$

$\to$ dig. Camera $\to$

Realisierung von B

Realisierung von
$B_B = \text{Blur}(B) + \eta$

η ist additives (weißes) Rauschen

Ziel: Umkehrung von Blur und Bildentrauschen

Linearisiertes Wiener-Filter (h_W): Bestimme eine Matrix

$$h_W \in \mathbb{R}^{K \times K}, \quad K << \min\{N, M\},$$

so dass

$$E\left(\|B - B_B * h_W\|^2\right) \to \min.$$

Berechnung der Extrema führt zum linearen Gleichungssystem für h_W.

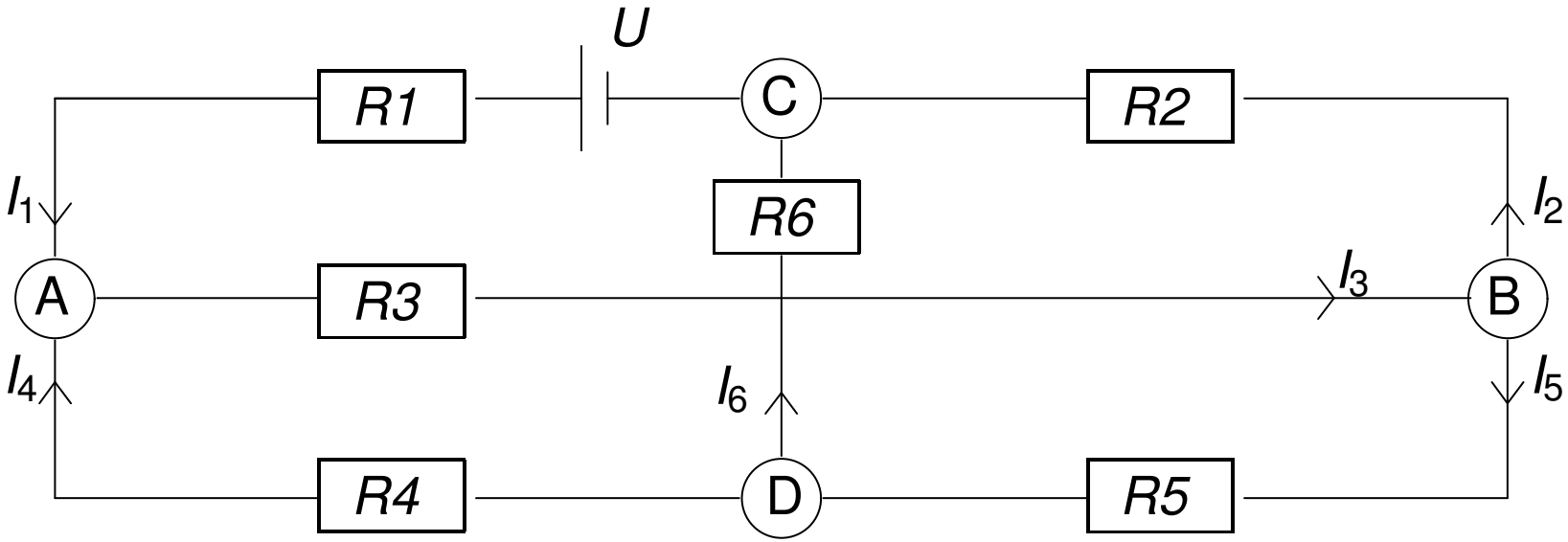


Realisierungen: B $\quad B_B = \text{Blur}(B) + \eta$ $\quad B_B * h_W \approx B$

2.1 Motivation: Elektrische Netzwerke

Gegeben: Widerstände R_j, $j = 1, \ldots, 5$, und Spannung U

Gesucht: Ströme I_j, $j = 1, \ldots, 6$.



Gesetze von Kirchhoff und Ohm ergeben das Gleichungssystem

$$I_1 - I_3 + I_4 = 0,\ I_2 + I_3 - I_5 = 0,\ -I_1 + I_2 + I_6 = 0,\ -I_4 + I_5 - I_6 = 0$$
$$R_1 I_1 - R_4 I_4 + R_6 I_6 = U$$
$$R_2 I_2 - R_5 I_5 - R_6 I_6 = 0$$
$$R_3 I_3 + R_4 I_4 + R_5 I_5 = 0$$
$$R_1 I_1 + R_2 I_2 + R_3 I_3 = U$$

Problem 2: Seien $A \in \mathbb{R}^{n\times n}$ regulär und $b \in \mathbb{R}^n$. Löse $Ax = b$.

2.2 Konditionierung des Problems 2

Sei $\|\cdot\|$ eine Vektornorm bzw. die zugehörige natürliche Matrixnorm.

2.2.1 Hilfsatz: Sei $B \in \mathbb{R}^{n\times n}$ mit $\|B\| < 1$. Dann ist die Matrix $I + B$ regulär, und es gilt $\|(I + B)^{-1}\| \leq \dfrac{1}{1 - \|B\|}$.

2.2.2 Störungssatz: Die Matrix $A \in \mathbb{R}^{n\times n}$ sei regulär, und es sei $\|A - \widetilde{A}\| < \dfrac{1}{\|A^{-1}\|}$. Dann ist die Matrix $\widetilde{A} \in \mathbb{R}^{n\times n}$ regulär und es gilt

$$\frac{\|x - \tilde{x}\|}{\|x\|} \leq \frac{\operatorname{cond}(A)}{1 - \operatorname{cond}(A) \cdot \frac{\|A-\tilde{A}\|}{\|A\|}} \left(\frac{\|b - \tilde{b}\|}{\|b\|} + \frac{\|A - \tilde{A}\|}{\|A\|} \right)$$

mit der Konditionszahl $\operatorname{cond}(A) = \|A\| \cdot \|A^{-1}\|$von A.

Bemerkungen:

(i) Ist $\text{cond}(A) \cdot \frac{\|\Delta A\|}{\|A\|} << 1$, so wird

$$\frac{\|x - \tilde{x}\|}{\|x\|} \lesssim \text{cond}(A) \left(\frac{\|b - \tilde{b}\|}{\|b\|} + \frac{\|A - \tilde{A}\|}{\|A\|} \right),$$

d.h. $\text{cond}(A)$ ist der Verstärkungsfaktor, mit dem sich die relativen Eingabefehler auf den relativen unvermeidbaren Problemfehler auswirken.

(ii) Die Abschätzung im Satz 2.2.2 ist scharf.

2.3 Direkte Eliminationsverfahren

Um $Ax = b$ mit der Cramerschen Regel zu lösen, brauchen die Rechner mit 10^8 Gleitkommaoperationen pro Sekunde

n	10	12	14	16	18	20
Rechenzeit	0.4 s	1 min	36 Stunden	41 Tage	38 Jahre	16000 Jahre

2.3.1 Dreiecksmatrizen, Rückwärtseinsetzen

Sei $A \in \mathbb{R}^{n\times n}$ eine reguläre obere Dreiecksmatrix.

$$Ax = b \quad \Leftrightarrow \quad \left\{ \begin{array}{rcrcrcl} a_{11}x_1 & + & a_{12}x_2 & + & \ldots & + & a_{1n}x_n = b_1 \\ & & a_{22}x_2 & + & \ldots & + & a_{2n}x_n = b_2 \\ & & & & \ldots & & \\ & & & & & & a_{nn}x_n = b_n \end{array} \right.$$

Die Lösung durch Rückwärtseinsetzen: $\frac{n^2}{2}$ Rechenoperationen

$$x_n = \frac{b_n}{a_{nn}}; \qquad x_j = \frac{1}{a_{jj}} \underbrace{\left(b_j - \sum_{k=j+1}^{n} a_{jk}x_k \right)}_{n-j \text{ Mult./Add.}}, \quad j = n-1, \ldots, 1.$$

2.3.2 Gauß-Elimination mit Pivotierung, $LR-$Zerlegungvon A

Idee: $Ax = b \;\Leftrightarrow\; CAx = Cb$, $\det(C) \neq 0$, CA obere Dreiecksmatrix.

Definitionen+Eigenschaften:

Seien $e_1, \ldots, e_n \in \mathbb{R}^n$ die standard Einheitsvektoren.

(i) Frobeniusmatrizen

$$G_k := \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -g_{k+1,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & -g_{n,k} & & & 1 \end{pmatrix} = I - \begin{pmatrix} 0 \\ \vdots \\ g_{k+1,k} \\ \vdots \\ g_{n,k} \end{pmatrix} e_k^T,$$

$k = 1, \ldots, n-1$ sind regulär.

(ii) Permutationsmatrizen

$$P_{ij} = I - (e_i - e_j)(e_i - e_j)^T, \quad 1 \leq i \leq j \leq n,$$

sind symmetrisch und unitär, d.h. $P_{i,j}^2 = I$.

(iii) $G_1 G_2 \cdots G_{n-1}$ und $P_{ij} G_k P_{ij}$, $1 \leq k < i \leq j \leq n$ sind untere Dreiecksmatrizen.

Algorithmus: (Gauß-Elimination mit Pivotierung)

Gegeben: $A \in \mathbb{R}^{n\times n}$ regulär, $b \in \mathbb{R}^n$.

Initialisierung: $[A^{(0)}; b^{(0)}] = [A; b]$

Für $i = 1, \ldots, n-1$:

Pivotsuche: $\left|a_{ji}^{(i-1)}\right| = \max\limits_{i\le k\le n}\left|a_{k,i}^{(i-1)}\right|, \quad j \ge i$

Pivotierung: $[\tilde{A}^{(i-1)}; \tilde{b}^{(i-1)}] = P_{ij} \cdot [A^{(i-1)}; b^{(i-1)}]$

Gauß-Elimination: mit $g_{k,i} = \tilde{a}_{k,i}^{(i-1)} / \tilde{a}_{i,i}^{(i-1)}$, $i+1 \le k \le n$

$$[A^{(i)}; b^{(i)}] = G_i \cdot [\tilde{A}^{(i-1)}; \tilde{b}^{(i-1)}]$$

end

Ausgabe: obere Dreiecksmatrix $A^{(n-1)}$ und $b^{(n-1)}$

Bemerkungen:

(i) Man braucht a priori nicht zu wissen, ob A regulär ist. Ist $\left|a_{ji}^{(i-1)}\right| = 0$ für ein $i = 1, \ldots, n-1$, dann ist A singulär und der Algorithmus muss abgebrochen werden.

(ii) Man kommt ohne Pivotierung nicht aus, z.B. bei $A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$.

Satz:(LR-Zerlegung von A) Zu jeder regulären $A \in \mathbb{R}^{n\times n}$ existiert eine Permutationsmatrix P, eine eindeutige untere Dreiecksmatrix $L = (\ell_{ij})$, $\ell_{ii} = 1$, $i = 1, \ldots, n$, und eine eindeutige obere Dreiecksmatrix R, so dass

$$P \cdot A = L \cdot R.$$

Bemerkung: Läßt sich die Gauß-Elimination ohne Pivotierung durchführen, so gilt

$$A = L \cdot R, \quad R := A^{(n-1)}, \quad L := (G_{n-1} \cdots G_1)^{-1}.$$

Bemerkungen: (*i*) Rechenaufwand der Gauß-Elimination mit Pivotierung ist $\frac{n^3}{3} + O(n^2)$.

(*ii*) Sei $A \in \mathbb{R}^{n\times n}$ regulär. Dann gilt

$$Ax = b \quad \Leftrightarrow \quad LRx = Pb \quad \Leftrightarrow \quad Ly = Pb, \quad Rx = y.$$

(*iii*) Bei Verwendung von Gleitkommaarithmetik erhält man die Matrizen $\widetilde{L}$ und $\widetilde{R}$ mit

$$F := PA - \widetilde{L}\widetilde{R} \neq 0,$$

wobei

$$|F| \leq 2 \cdot a \cdot \frac{\text{eps}}{1 - \text{eps}} \begin{bmatrix} 0 & 0 & \dots & \dots & 0 \\ 1 & 1 & \dots & \dots & 1 \\ 1 & 2 & \dots & \dots & 2 \\ 1 & 2 & 3 & \dots & 3 \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 2 & 3 & \dots & n-1 \end{bmatrix}, \quad a \leq 2^{n-1} \max_{i,j=1,\dots,n} \left|a_{i,j}\right|.$$

(*iv*) Nachiteration (Variante der Gauß-Elimination): Wird $Ax = b$ mit stark verfälschter (z.B. durch Abspeichern von L und R in kurzem Zahlenformat und späteres Einlesen) LR-Zerlegung $PA \neq \tilde{L}\,\tilde{R}$

$$\frac{\|\Delta_A\|}{\|A\|} = \epsilon_1 >> 1, \quad \Delta_A = PA - \tilde{L}\,\tilde{R},$$

gelöst, so gilt für $\tilde{L}\,\tilde{R}\tilde{x} = Pb$

$$\frac{\|x - \tilde{x}\|}{\|x\|} \leq k_A \epsilon_1, \quad k_A := \frac{\mathrm{cond}(A)}{1 - \mathrm{cond}(A)\frac{\|\Delta_A\|}{\|A\|}}.$$

Die verbesserte Lösung $x^{(1)} = \tilde{x} + \delta^{(1)}, \quad \tilde{L}\,\tilde{R}\delta^{(1)} = P(b - A\tilde{x})$, erfüllt

$$\frac{\|x - x^{(1)}\|}{\|x\|} \leq k_A^2 \epsilon_1^2. \quad \text{(Verdopplung der exakten Dezimalstellen).}$$

Weitere Schritte der Nachiteration liefern Verbesserung bis zur Maschinengenauigkeit *eps*. In der Praxis genügen 1 - 3 Schritte.

2.3.3 Cholesky-Zerlegung

Definition: Eine symmetrische Matrix $A \in \mathbb{R}^{n\times n}$ heißt positiv definit, falls

$$x^T Ax > 0, \quad x \in \mathbb{R}^n \setminus \{0\}.$$

Satz (Cholesky-Zerlegung): Eine symmetrische Matrix $A \in \mathbb{R}^{n\times n}$ ist positiv definit genau dann, wenn es eine untere Dreiecksmatrix $G = (g_{j,k})_{1\le j,k\le n} \in \mathbb{R}^{n\times n}$ gibt, so dass

$$A = G \cdot G^T \quad \text{und} \quad g_{j,j} > 0, \quad j = 1, \ldots, n.$$

Bemerkung: Der Rechenaufwand der Cholesky-Zerlegung ist

$$\frac{n^3}{6} + O\left(n^2\right).$$

Algorithmus (Cholesky-Zerlegung):

Eingabe: $A \in \mathbb{R}^{n \times n}$ positiv definit

Für $j = 1, \ldots, n$

Berechne $g_{j,j} = \sqrt{a_{j,j} - \sum_{i=1}^{j-1} g_{j,i}^2}$

Für $k = j+1, \ldots, n$ berechne $g_{k,j} = \frac{1}{g_{j,j}} \left(a_{k,j} - \sum_{i=1}^{j-1} g_{k,i} \cdot g_{j,i} \right)$

end

Bemerkungen: (*i*) Man kommt bei der Cholesky-Zerlegung ohne Pivotierung aus, da alle Elemente auf der Hauptdiagonale von $A^{(j)}, \quad j = 0, \ldots, n-1$, positiv sind.

(*ii*) Gleichzeitig stellt die Cholesky-Zerlegung einen Test dar, ob eine gegebene symmetrische Matrix positiv definit ist. Ist sie nicht positiv definit, dann ist einer der Einträge $g_{j,j} \leq 0$. In diesem Fall bricht der Algorithmus ab.

2.4 Iterationsverfahren

- Das Gaußsche Eliminationsverfahren erfordert bei Bandmatrizen mit der Bandbreite M den Aufwand $\frac{1}{3}nM^2$. Für große Matrizen ($n > 10^6$, $M > 10^2$) sind dies bereits 10^{10} Rechenoperationen.
- Bei vielen Aufgaben für zwei- und dreidimensionale Modelle treten Matrizen mit sehr vielen Nullen in den Matrixeinträgen auf, die keine echte Bandstruktur aufweisen. Das Gaußsche Eliminationsverfahren würde zum "Auffüllen" vieler dieser Stellen führen.

Iterationsverfahren bestimmen die Lösung x^* näherungsweise und nutzen dabei die Struktur der Matrix A aus, indem sie nur auf die von Null verschiedenen Einträge der Matrix A zugreifen. Dies ermöglicht eine kompakte Speicherung dünn besetzter Matrizen (matlab "sparse") und spart Rechenoperationen. Gute iterative Verfahren benötigen nM Rechenoperationene zum Lösen von $Ax = b$, wobei m die durchschnittliche Anzahl der von Null verschiedenen Einträge pro Zeile der Matrix ist.

Mit einer beliebigen invertierbaren Matrix $C \in \mathbb{R}^{n\times n}$ gilt die Äquivalenz

$$\boxed{Ax = b \iff Cx = Cx - Ax + b \iff x = (I - C^{-1}A)x + C^{-1}b.}$$

Man versucht C so zu wählen, dass

- zu gegebenem Vektor $x^{[k]}$, $k \geq 0$, der neue Vektor

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

  mit wenig Aufwand zu berechnen ist,
- die Folge $(x^{(k+1)})_{k\in\mathbb{N}}$ gegen die Lösung von $Ax = b$ konvergiert.

Zuerst betrachten wir zwei Verfahren, die wenig Aufwand zur Berechnung von $x^{(k+1)}$ erfordern.

2.4.1 Gesamtschritt- und Einzelschrittverfahren

Gesamtschrittverfahren (Jacobi-Verfahren 1845)

Die Matrix A habe Diagonalelemente $a_{ii} \neq 0$.

1. Wähle beliebigen Startvektor $x^{(0)} \in \mathbb{R}^n$, z.B. $x_i^{(0)} = \frac{b_i}{a_{ii}}$.
2. Für $k = 0, 1, 2, \ldots$
   für $i = 1, 2, \ldots, n$: ("löse die i-te Gleichung nach x_i")

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j^{(k)} \right).$$

Einzelschrittverfahren (Gauß-Seidel-Verfahren 1820)

Die Matrix A habe Diagonalelemente $a_{ii} \neq 0$.

1. Wähle beliebigen Startvektor $x^{(0)} \in \mathbb{R}^n$, z.B. $x_i^{[0]} = \frac{b_i}{a_{ii}}$.
2. Für $k = 0, 1, 2, \ldots$
   für $i = 1, 2, \ldots, n$: (“löse die i-te Gleichung nach x_i”)

$$x_i^{[(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right).$$

Beispiel:

Zur numerischen Lösung des Randwertproblems 2. Ordnung

$$-u''(x) = f(x), \qquad x \in [0,1],$$

mit $u(0) = u(1) = 0$ führt man die übliche Diskretisierung

$$-u''(x_k) \approx \frac{2u(x_k) - u(x_{k-1}) - u(x_{k+1})}{h^2}, \qquad h = \frac{1}{n+1}, \quad x_k = kh, \quad 0 \le k \le n+1,$$

durch und löst das lineare Gleichungssystem $Au = b$ mit der Tridiagonalmatrix

$$A = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{n \times n}$$

zur rechten Seite $b_k = h^2 f(x_k)$, $1 \le k \le n$. Für spätere Zwecke halten wir fest: die Eigenwerte von A sind

$$\lambda_k = 2\left(1 - \cos\frac{k\pi}{n+1}\right), \qquad 1 \le k \le n,$$

und zugehörige Eigenvektoren sind

$$v_k = \left(\sin\frac{jk\pi}{n+1}\right)_{1 \le j \le n}.$$

Zu $f(x) \equiv 1$ sowie $n = 4$ ist die exakte Lösung $u = (2, 3, 3, 2)^T$. Gesamt- und Einzelschrittverfahren liefern die Werte der folgenden Tabellen.

Tabelle: GSV zum 1-dim. Modellproblem, $h = \frac{1}{5}$

$x^{(0)}$	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$	$x^{(5)}$	$x^{(6)}$	$x^{(7)}$	$\ldots$	$x^{(20)}$
0.5000	0.7500	1.0000	1.1875	1.3438	1.4688	1.5703	1.6523		1.9779
0.5000	1.0000	1.3750	1.6875	1.9375	2.1406	2.3047	2.4375		2.9642
0.5000	1.0000	1.3750	1.6875	1.9375	2.1406	2.3047	2.4375		2.9642
0.5000	0.7500	1.0000	1.1875	1.3438	1.4688	1.5703	1.6523		1.9779

Tabelle: ESV zum 1-dim. Modellproblem, $h = \frac{1}{5}$

$x^{(0)}$	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$	$x^{(5)}$	$x^{(6)}$	$x^{(7)}$	$\ldots$	$x^{(20)}$
0.5000	0.7500	1.0625	1.3438	1.5664	1.7158	1.8140	1.8782		1.9995
0.5000	1.1250	1.6875	2.1328	2.4316	2.6279	2.7565	2.8406		2.9994
0.5000	1.3125	1.9219	2.2969	2.5400	2.6990	2.8030	2.8710		2.9995
0.5000	1.1563	1.4609	1.6484	1.7700	1.8495	1.9015	1.9355		1.9997

Die Matrix $A = (a_{ij})$ wird zerlegt in

$$L = \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ a_{21} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ a_{n1} & \cdots & a_{n,n-1} & 0 \end{pmatrix}, \quad D = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{nn} \end{pmatrix}, \quad R = \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & a_{n-1,n} \\ 0 & \cdots & \cdots & 0 \end{pmatrix}$$

(*i*) Die Iterationsfunktion des Gesamtschrittverfahrens lautet

$$\phi_{\mathrm{GSV}}(x) = (I - D^{-1}A)x + D^{-1}b = -D^{-1}(L+R)x + D^{-1}b.$$

(*ii*) Die Iterationsfunktion des Einzelschrittverfahrens lautet

$$\phi_{\mathrm{ESV}}(x) = (I-(L+D)^{-1}A)x+(L+D)^{-1}b = -(L+D)^{-1}Rx+(L+D)^{-1}b.$$

Verglichen mit der allgemeinen Form

$$\phi(x) = (I - C^{-1}A)x + C^{-1}b = Bx + c,$$

werden hier also die folgenden Iterationsmatrizen verwendet:

$$\begin{aligned} B &= -D^{-1}(L+R), && \text{(Gesamtschrittverfahren)} \\ B &= -(L+D)^{-1}R, && \text{(Einzelschrittverfahren)} \end{aligned}$$

Die Konvergenz der Fixpunktiteration

$$x^{(k+1)} = (I - C^{-1}A)x^{(k)} + C^{-1}b = Bx^{[k]} + c \qquad (F)$$

bei beliebigem Startwert $x^{(0)} \in \mathbb{R}^n$:

Satz: Die Matrizen A und C seien reguär. Dann sind die folgenden Aussagen äquivalent:

(i) Die Fixpunktiteration (F) konvergiert bei beliebigem Startvektor $x^{[0]} \in \mathbb{R}^n$ gegen die Lösung x^* des linearen Gleichungssystems $Ax = b$.

(ii) Die Iterationsmatrix $B = I - C^{-1}A$ hat den Spektralradius

$$\rho(B) = \max\{|\lambda| \ : \ \lambda \text{ Eigenwert von } B\} < 1.$$

(iii) Es gibt eine natürliche Matrixnorm $\|\cdot\|$ auf $\mathbb{R}^{n\times n}$ mit $\|B\| < 1$.

Bemerkung: Matrixnormen $\|\cdot\|_p$, $p = 1, 2, \infty$ sind natürliche Matrixnormen.

Äquivalenz von (*ii*) und (*iii*) folgt aus

Hilfssatz: Gegeben sei die Matrix $B \in \mathbb{R}^{n\times n}$.

(*i*) Für jede natürliche Matrixnorm $\|\cdot\| : \mathbb{R}^{n\times n} \to \mathbb{R}$ gilt

$$\rho(B) \leq \|B\|.$$

(*ii*) Zur Matrix $B \in \mathbb{R}^{n\times n}$ und beliebigem $\epsilon > 0$ existiert eine natürliche Matrixnorm $\|\cdot\| : \mathbb{R}^{n\times n} \to \mathbb{R}$ so, dass für den Spektralradius $\rho(B)$ gilt

$$\|B\| \leq \rho(B) + \epsilon.$$

Bemerkungen:

(*i*) Für hermitesche Matrizen $B \in \mathbb{C}^{n\times n}$ stimmen der Spektralradius und die Spektralnorm überein: $\rho(A) = \|A\|_2$.

(*ii*) Der Satz liefert die fundamentale Beziehung

$$\rho(B) = \limsup_{k\to\infty} \left(\|B^k\|\right)^{1/k},$$

wobei $\|\cdot\|$ eine beliebige Matrixnorm ist.

Satz: lineare Konvergenz der Fixpunktiteration Falls die Fixpunktiteration (F) bei beliebigem Startwert $x^{[0]}$ gegen die Lösung x^* von $Ax = b$ konvergiert, so gilt

$$\limsup_{k\to\infty} \left(\frac{\|x^{[k]} - x^*\|}{\|x^{[0]} - x^*\|} \right)^{1/k} \leq \rho(B).$$

Bemerkung: Selbst wenn $\|B\| \geq 1$, aber $q = \rho(B) < 1$ gilt, so wird nach einigen Schritten die lineare Konvergenz $\|x^{[k]} - x^*\| \to 0$ mit der Geschwindigkeit q^{k-k_0} sichtbar.

Definition und Satz: Eine Matrix $A \in \mathbb{R}^{n\times n}$ heißt *stark diagonaldominant*, wenn

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|$$

für alle $1 \leq i \leq n$ gilt. Falls A stark diagonaldominant ist, so gilt

$$\begin{aligned} \rho(-D^{-1}(L+R)) &\leq \|-D^{-1}(L+R)\|_\infty < 1, \\ \rho(-(L+D)^{-1}R) &\leq \|-(L+D)^{-1}R\|_\infty < 1. \end{aligned}$$

Satz: Falls A positiv definit ist, so konvergiert das Einzelschrittverfahren.

Um die Konvergenz beim ESV zu beschleunigen, wird in jedem Berechnungsschritt (innere Schleife des ESV) eine Verlängerung der Korrektur vorgenommen:

2.4.2 SOR (=successive over-relaxation)-Verfahren

Die Matrix A habe Diagonalelemente $a_{ii} \neq 0$. Weiter sei $\omega \in \mathbb{R}$. (ω wird *Relaxationsparameter* genannt.)

1. Wähle beliebigen Startvektor $x^{[0]} \in \mathbb{R}^n$, z.B. $x_i^{[0]} = \frac{b_i}{a_{ii}}$.
2. Für $k = 0, 1, 2, \ldots$
   für $i = 1, 2, \ldots, n$

$$\widetilde{x}_i^{[k+1]} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{[k+1]} - \sum_{j=i+1}^{n} a_{ij} x_j^{[k]} \right)$$

$$x_i^{[k+1]} = \omega \widetilde{x}_i^{[k+1]} + (1 - \omega) x_i^{[k]}.$$

Beispiel:

Tabelle: SOR zum 1-dim. Modellproblem, $h = \frac{1}{5}$, $\omega = 1.2$

$x^{[0]}$	$x^{[1]}$	$x^{[2]}$	$x^{[3]}$	$x^{[4]}$	$x^{[5]}$	$x^{[6]}$	$x^{[7]}$	...	$x^{[13]}$
0.5000	0.8000	1.2080	1.5642	1.8175	1.9148	1.9611	1.9825		1.9998
0.5000	1.2800	2.0096	2.5506	2.7971	2.9068	2.9578	2.9808		2.9998
0.5000	1.5680	2.3566	2.6945	2.8623	2.9375	2.9715	2.9871		2.9999
0.5000	1.4408	1.7258	1.8715	1.9431	1.9739	1.9881	1.9946		2.0000

Tabelle: SOR zum 1-dim. Modellproblem, $h = \frac{1}{5}$, $\omega = 1.3$

$x^{[0]}$	$x^{[1]}$	$x^{[2]}$	$x^{[3]}$	$x^{[4]}$	$x^{[5]}$	$x^{[6]}$	$x^{[7]}$	...	$x^{[9]}$
0.5000	0.8250	1.2873	1.6871	1.9540	1.9889	2.0012	2.0019		2.0002
0.5000	1.3613	2.1898	2.7848	2.9617	2.9967	3.0034	3.0016		3.0002
0.5000	1.7098	2.6078	2.8878	2.9884	3.0026	3.0022	3.0011		3.0001
0.5000	1.6114	1.8617	1.9686	2.0019	2.0011	2.0011	2.0004		2.0000

Iterationsfunktion des SOR-Verfahrens Mit $A = L + D + R$ lautet die Iterationsfunktion des SOR-Verfahrens

$$\phi_{\mathrm{SOR}}(x) = H_\omega x + \omega(D + \omega L)^{-1} b, \quad \omega \in RR,$$

mit der Iterationsmatrix $H_\omega = (D + \omega L)^{-1}((1 - \omega)D - \omega R)$.
Herleitung:

$$a_{ii} x_i^{[k+1]} + \omega \sum_{j=1}^{i-1} a_{ij} x_j^{[k+1]} = \omega b_i - \omega \sum_{j=i+1}^{n} a_{ij} x_j^{[k]} + (1 - \omega) a_{ii} x_i^{[k]}.$$

Dies lautet

$$(D + \omega L) x^{[k+1]} = \omega b + ((1 - \omega)D - \omega R) x^{[k]}.$$

Bemerkung: Für $\omega = 1$ ist $\phi_{\mathrm{SOR}} = \phi_{\mathrm{ESV}}$.

Satz:

(*i*) *von Kahan:* Für beliebiges A mit Diagonalelementen $a_{ii} \neq 0$ und beliebiges $\omega \in \mathbb{R}$ ist

$$\rho(H_\omega) \geq |\omega - 1|.$$

Die Konvergenz des SOR-Verfahrens kann also höchstens für $\omega \in (0, 2)$ eintreten.

(*ii*) *von Reich und Ostrowski:* Für jede positiv definite Matrix A und jedes $\omega \in (0, 2)$ gilt $\rho(H_\omega) < 1$, also ist das SOR-Verfahren konvergent.

2.4.3 CG-Verfahren (konjugierte Gradienten)

Problem 2': Seien $A \in \mathbb{R}^{n \times n}$ positiv definit und $b \in \mathbb{R}^n$. Löse $Ax = b$.

Bemerkung: Problem 2' ist äquivalent zur Minimierungsaufgabe für die konvexe Funktion $\Phi : \mathbb{R}^n \to \mathbb{R}$

$$\Phi(x) = \frac{1}{2} x^T A x - x^T b \to \min!$$

Denn der Gradient $\nabla \Phi(x) = Ax - b = 0$ genau dann, wenn x das Problem 2' löst, und die Hessematrix A positiv definit ist.

Definition: Zwei Vektoren $v, w \in \mathbb{R}^n \setminus \{0\}$ heißen konjugiert bzgl. A, falls

$$(v, w)_A = v^T A w = 0.$$

Algorithmus: CG-Verfahren

Die Matrix $A \in \mathbb{R}^{n \times n}$ sei positiv definit und $b \in \mathbb{R}^n$.

Wähle beliebigen Startvektor $x^{(0)} \in \mathbb{R}^n$.

1. $r^{(0)} = A\,x^{(0)} - b, \quad d^{(0)} = -r^{(0)}$
2. Für $k = 0, 1, 2, \ldots$

$$x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}, \qquad \alpha_k = \frac{\|r^{(k)}\|_2^2}{(d^{(k)}, d^{(k)})_A}$$

$$r^{(k+1)} = r^{(k)} + \alpha_k\, A\, d^{(k)}$$

$$d^{(k+1)} = -r^{(k+1)} + \beta_k d^{(k)}, \qquad \beta_k = \frac{\|r^{(k+1)}\|_2^2}{\|r^{(k)}\|_2^2}$$

Konvergenzgeschwindigkeit ist linear

$$\|x^{(k)} - x\|_A \leq 2 \left(\frac{\sqrt{\mathrm{cond}_2(A)} - 1}{\sqrt{\mathrm{cond}_2(A)} + 1} \right)^k \|x^{(0)} - x\|_A, \qquad \|y\|_A := \sqrt{y^T A y}.$$