

Große Gleichungssysteme fast richtig lösen

Teilnehmer:

7 Schülerinnen und Schüler

Andreas-Gymnasium
Herder-Gymnasium
Immanuel-Kant-Gymnasium
Käthe-Kollwitz-Gymnasium

mit tatkräftiger Unterstützung durch:

Anika Meyer

Humboldt-Universität zu Berlin

Gruppenleiter:

Falk Ebert

Herder-Gymnasium, Humboldt-Universität zu Berlin, MATHEON

1. Einleitung

Viele Modellierungsaufgaben, wie zum Beispiel das Beschreiben einer Temperaturverteilung in einem Eisenstab, lassen sich letztendlich auf lineare Gleichungssysteme (LGS) zurückführen.

Je nach Problem ergeben sich hier schnell LGS mit tausenden oder mehr Variablen. Gleichungssysteme dieser Größe bringen schnell Probleme in der Auswertung mit sich. Vor allem die exakte Auswertung ist oft rechenaufwändig.

Um den Rechenaufwand erheblich zu minimieren und einige Probleme erst lösbar zu machen, bietet es sich an, die LGS numerisch zu lösen.

Im Rahmen unserer Projektarbeit haben wir uns mit solchen numerischen Verfahren zum Lösen großer LGS beschäftigt und schließlich das oben beschriebene Beispiel in Octave simuliert.

2. Wärmeleitung

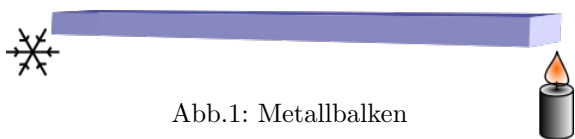


Abb.1: Metallbalken

Es wird ein Metallbalken betrachtet, dessen Temperatur an beiden Seiten konstant gehalten wird. Ziel ist es, die Temperatur $T(x)$ an jedem Ort x im Metallbalken in Abhängigkeit der Randbedingungen bestimmen zu können.

Dazu diskretisiert man das Phänomen. D.h. anstatt unendlich viele Werte bestimmen zu müssen, reduziert man die Anzahl der zu bestimmenden Werte, indem der Metallstab in äquidistante Intervalle aufgeteilt wird. Die Temperatur muss also nur für alle T_i bestimmt werden. Die Randbedingungen T_0 und T_n sind gegeben.

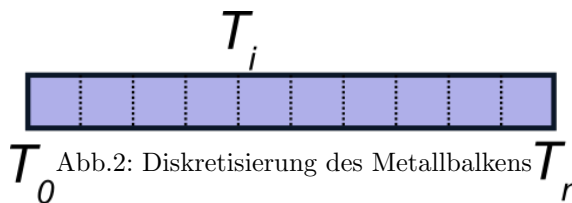


Abb.2: Diskretisierung des Metallbalkens

Die Wärmeleitungsgleichung liefert nun folgende Differentialgleichung für T mit den Randbedingungen $T(0) = T_0$ und $T(1) = T_1$. Der Balken wird hier mit der Länge 1 LE angenommen.

$$T''(x) = 0$$

Um diese Gleichung als LGS zu schreiben, wird der Term $T''(x)$ diskretisiert.

Satz 1. Für eine genügend glatte Funktion $f : \mathbb{R} \rightarrow \mathbb{R}$ gilt:

$$f''(x) = \frac{f(x-h) - 2f(x) + f(x+h)}{h^2} + \mathcal{O}(h^2).$$

Beweis:

Der Beweis erfolgt durch die Taylorentwicklung der Funktionen $f(x+h)$ und $f(x-h)$ am Entwicklungspunkt x .

$$\begin{aligned} f(x+h) &= \sum_{k=0}^{\infty} \frac{f^{(k)}(x)}{k!} (x+h-x)^k = \sum_{k=0}^{\infty} \frac{f^{(k)}(x)}{k!} h^k \\ &= f(x) + f'(x) \cdot h + \frac{f''(x)}{2!} \cdot h^2 + \frac{f'''(x)}{3!} \cdot h^3 + \frac{f^{(4)}(x)}{4!} \cdot h^4 + \dots \\ f(x-h) &= \sum_{k=0}^{\infty} \frac{f^{(k)}(x)}{k!} (x-h-x)^k = \sum_{k=0}^{\infty} \frac{f^{(k)}(x)}{k!} (-h)^k \\ &= f(x) - f'(x) \cdot h + \frac{f''(x)}{2!} \cdot h^2 - \frac{f'''(x)}{3!} \cdot h^3 + \frac{f^{(4)}(x)}{4!} \cdot h^4 - \dots \end{aligned}$$

Es ergibt sich mit den Taylorentwicklungen:

$$\frac{f(x+h) + f(x-h) - 2f(x)}{h^2} = \frac{2f(x) + 2\frac{f''(x)}{2!} \cdot h^2 + 2\frac{f^{(4)}(x)}{4!} \cdot h^4 + \dots - 2f(x)}{h^2} = f''(x) + \frac{h^2 \cdot f^{(4)}(x)}{12} + \dots$$

Für $h \rightarrow 0$ streben die Taylorsummanden nach $f''(x)$ gegen 0 und damit gilt für kleine h :

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

Da h gegen 0 strebt, läuft der Term der niedrigsten Potenz (h^2) am langsamsten gegen Null. $\frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$ konvergiert also quadratisch gegen $f''(x)$. Dies lässt sich auch schreiben als

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + \mathcal{O}(h^2).$$

□

In Bezug auf die Intervallunterteilung lässt sich $h = \frac{1}{n}$ wählen (Der Metallbalken ist 1 LE lang). Damit lässt sich die Wärmeleitungsgleichung folgendermaßen diskretisieren:

$$\begin{aligned} T''(x) &= 0 \\ \rightsquigarrow \frac{T(x+h) - 2T(x) + T(x-h)}{h^2} &= 0 \\ \rightsquigarrow \frac{T_{k+1} - 2T_k + T_{k-1}}{h^2} &= 0. \end{aligned} \quad (1)$$

Zusammen mit gegebenen Randbedingungen $T(0) = T_0$, $T(1) = T_n$ stellt (1) ein lineares Gleichungssystem mit den $n - 1$ Unbekannten T_1 bis T_{n-1} dar.

3. Numerische Verfahren zum Lösen von LGS

3.1. Gauß-Verfahren

Beim Gauß-Verfahren wird ein lineares Gleichungssystem mit n Gleichungen und n Variablen der Form

$$\begin{aligned} a_{11} \cdot x_1 + \dots + a_{1n} \cdot x_n &= b_1 \\ \vdots & \\ a_{n1} \cdot x_1 + \dots + a_{nn} \cdot x_n &= b_n \end{aligned} \quad (2)$$

durch Skalierung und Addition von Zeilen in Dreiecksform

$$\begin{aligned} \tilde{a}_{11} \cdot x_1 + \tilde{a}_{12} \cdot x_2 + \dots + \tilde{a}_{1n} \cdot x_n &= \tilde{b}_1 \\ 0 + \tilde{a}_{22} \cdot x_2 + \dots + \tilde{a}_{2n} \cdot x_n &= \tilde{b}_2 \\ \vdots & \\ 0 + \dots + 0 + \tilde{a}_{nn} \cdot x_n &= \tilde{b}_n \end{aligned}$$

gebracht. Dies erfordert in etwa in der Größenordnung von n^3 Rechenoperationen. Gerade für sehr große Gleichungssysteme ist das sehr aufwändig.

3.2. Jacobi-Verfahren

Löst man jeweils die i -te Gleichung eines linearen Gleichungssystems (2) nach x_i auf, ergibt dies

$$\begin{aligned} x_1 &= \frac{1}{a_{11}}(b_1 - a_{12}x_2 - a_{13}x_3 - \dots - a_{1n}x_n) \\ x_2 &= \frac{1}{a_{22}}(b_2 - a_{21}x_1 - a_{23}x_3 - \dots - a_{2n}x_n) \\ \vdots &= \vdots \\ x_n &= \frac{1}{a_{nn}}(b_n - a_{n1}x_1 - a_{n2}x_2 - \dots - a_{n-1}x_{n-1}). \end{aligned}$$

Ausgehend von Startwerten $x_1^{[0]}, \dots, x_n^{[0]}$ können daraus mit dem folgenden Iterationsschritt neue Werte $x_i^{[k+1]}$ aus den $x_i^{[k]}$ bestimmt werden.

$$\begin{aligned}
 x_1^{[k+1]} &= \frac{1}{a_{11}}(b_1 - a_{12}x_2^{[k]} - a_{13}x_3^{[k]} - \dots - a_{1n}x_n^{[k]}) \\
 x_2^{[k+1]} &= \frac{1}{a_{22}}(b_2 - a_{21}x_1^{[k]} - a_{23}x_3^{[k]} - \dots - a_{2n}x_n^{[k]}) \\
 &\vdots \\
 x_n^{[k+1]} &= \frac{1}{a_{nn}}(b_n - a_{n1}x_1^{[k]} - a_{n2}x_2^{[k]} - \dots - a_{n-1}x_{n-1}^{[k]})
 \end{aligned} \tag{3}$$

Unter geeigneten Bedingungen, die wir im weiteren untersuchen werden, sind diese $x_i^{[k]}$ zunehmend bessere Näherungen für die Lösungen x_i von (2). Dieses Iterationsverfahren lässt sich folgendermaßen in eine Matrix-Vektor-Schreibweise überführen:

$$\begin{aligned}
 a_{11} \cdot x_1^{[k+1]} + \dots + a_{1n} \cdot x_n^{[k]} &= b_1 \\
 \vdots &\vdots \\
 a_{n1} \cdot x_1^{[k]} + \dots + a_{nn} \cdot x_n^{[k+1]} &= b_n
 \end{aligned}$$

$$\Leftrightarrow \underbrace{\begin{bmatrix} a_{11} & & 0 \\ & \ddots & \\ 0 & & a_{nn} \end{bmatrix}}_D \cdot \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}}_{x^{[k+1]}} + \underbrace{\begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ a_{21} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1,n} \\ a_{n1} & \dots & a_{n,n-1} & 0 \end{bmatrix}}_{A-D} \cdot \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}}_{x^{[k]}} = \underbrace{\begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}}_b$$

Die Koeffizientenmatrix A des LGS lässt sich dabei in eine Diagonalmatrix D und eine Matrix $(A - D)$ zerlegen, sodass sich mit weiteren Äquivalenzumformungen folgende Kurzschreibweise für das Jacobi-Verfahren ergibt:

$$\begin{aligned}
 Dx^{[k+1]} + (A - D)x^{[k]} &= b \\
 D^{-1} \cdot | \quad Dx^{[k+1]} &= b - (A - D)x^{[k]} \\
 x^{[k+1]} &= D^{-1}(b - (A - D)x^{[k]}) \\
 x^{[k+1]} &= \underbrace{D^{-1}b}_{c_J} + \underbrace{(-D^{-1}(A - D))}_{M_J} x^{[k]} \\
 &= M_J x^{[k]} + c_J.
 \end{aligned}$$

Dabei gehen wir stillschweigend davon aus, dass $a_{ii} \neq 0$ für alle i und D damit invertierbar ist.

3.3. Gauß-Seidel-Verfahren

Geht man wie beim Jacobi-Verfahren (3) vor und berechnet die neuen Iterierten $x_i^{[k+1]}$ der Reihe nach für $i = 1, 2, \dots, n$, dann kann man ab $i = 2$ bereits auf die Iterierten $x_j^{[k+1]}$ für $j < i$ zurückgreifen. Damit ist zu erwarten, dass die Verbesserung der Näherungslösung beschleunigt wird. Allerdings ist eine Parallelisierung, d.h. die verteilte Berechnung auf mehreren Rechnern, im Gegensatz zum Jacobi-

Verfahren ausgeschlossen. Für das Gauß-Seidel-Verfahren ergibt sich also:

$$\begin{aligned}
 x_1^{[k+1]} &= \frac{1}{a_{11}}(b_1 - a_{12}x_2^{[k]} - a_{13}x_3^{[k]} - \dots - a_{1n}x_n^{[k]}), \\
 x_2^{[k+1]} &= \frac{1}{a_{22}}(b_2 - a_{21}x_1^{[k+1]} - a_{23}x_3^{[k]} - \dots - a_{2n}x_n^{[k]}), \\
 &\vdots \\
 x_n^{[k+1]} &= \frac{1}{a_{nn}}(b_n - a_{n1}x_1^{[k+1]} - a_{n2}x_2^{[k+1]} - \dots - a_{n-1n}x_{n-1}^{[k+1]}).
 \end{aligned} \tag{4}$$

In Matrix-Vektor-Schreibweise ergibt sich dann

$$\underbrace{\begin{bmatrix} a_{11} & 0 & \dots & 0 \\ a_{21} & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_{n1} & \dots & a_{nn-1} & a_{nn} \end{bmatrix}}_{L+D} \cdot \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}}_{x^{[k+1]}} + \underbrace{\begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & a_{n-1n} \\ 0 & \dots & \dots & 0 \end{bmatrix}}_R \cdot \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}}_{x^{[k]}} = \underbrace{\begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}}_b.$$

Als Kurzschreibweise ergibt sich aus $(L + D) \cdot x^{[k+1]} + R \cdot x^{[k]} = b$ durch äquivalentes Umformen:

$$\begin{aligned}
 x^{[k+1]} &= \underbrace{-(L + D)^{-1} \cdot R}_{M_{GS}} \cdot x^{[k]} + \underbrace{(L + D)^{-1} \cdot b}_{b_{GS}} \\
 &= M_{GS} \cdot x^{[k]} + b_{GS}.
 \end{aligned}$$

Beispiel 1. Exaktes und numerisches Lösen eines LGS

$$\begin{aligned}
 3x_1 + x_1 + x_1 &= 5 \\
 3x_1 + 3x_2 + x_3 &= 7 \\
 3x_1 + 3x_2 + 3x_3 &= 9
 \end{aligned}$$

Exakte Lösung nach Gauß:
 $x_1 = 1, x_2 = 1, x_3 = 1$

Jacobi-Verfahren:

Iterationsschritt	0	1	2	3
x_1	0	1,67	-0,11	2,11
x_2	0	2,33	-0,33	2,78
x_3	0	3	-1	4,11

Gauß-Seidel:

Iterationsschritt	0	1	2	3
x_1	0	1,67	1,22	1,07
x_2	0	0,67	0,89	0,96
x_3	0	0,67	0,89	0,96

Es ist zu erkennen, dass das Jacobi-Verfahren für diese Verfahrensmatrix nicht konvergiert, das Gauß-Seidel-Verfahren jedoch schon.

4. Konvergenz der numerischen Lösungsverfahren

Bei den rekursiven Vorschriften der Jacobi- und Gauß-Seidel-Verfahren handelt es sich um Fixpunktiterationen.

Definition 1 (Fixpunktiteration). Sei $\varphi : \mathbb{G} \rightarrow \mathbb{G}$ eine Abbildung. Ein x^* mit $\varphi(x^*) = x^*$ heißt **Fixpunkt** der Abbildung. Die rekursive Vorschrift

$$x_{n+1} = \varphi(x_n)$$

mit gegebenem x_0 heißt **Fixpunktiteration**.

Überlegung. Das Jacobi- und Gauß-Seidel-Verfahren lassen sich schreiben als

$$\begin{aligned}
 x^{[k+1]} &= M_J \cdot x^{[k]} + C_J = \varphi(x^{[k]}) \\
 \text{bzw.} \quad x^{[k+1]} &= M_{GS} \cdot x^{[k]} + C_{GS} = \varphi(x^{[k]}).
 \end{aligned}$$

Die Bedingungen für die Konvergenz einer solchen Fixpunktiteration ergeben sich aus dem Fixpunktsatz von Banach. Vorab muss dazu die Kontraktion definiert werden.

Definition 2 (Kontraktion). Sei $\varphi : \mathbb{I} \rightarrow \mathbb{I}$ eine Funktion. φ ist eine Kontraktion auf \mathbb{I} wenn gilt:

- a) $\varphi(\mathbb{I}) \subseteq \mathbb{I}$ (Selbstabbildung)
- b) $|\varphi(x) - \varphi(y)| \leq L \cdot |x - y| \quad \forall x, y \in \mathbb{I}$ mit $L < 1$

Damit gilt für die Konvergenz von Fixpunktiterationen der Fixpunktsatz von Banach.

Satz 2 (Fixpunktsatz von Banach).

Sei $\varphi : \mathbb{I} \rightarrow \mathbb{I}$ eine Kontraktion auf \mathbb{I} . Dann existiert genau ein Fixpunkt $x^* \in \mathbb{I}$ und die Fixpunktiteration $x_{n+1} = \varphi(x_n)$ mit beliebigem Startwert $x_0 \in \mathbb{I}$ liefert eine gegen x^* konvergierende Folge (x_n) .

Da der Fixpunktsatz von Beträgen, also Abständen ausgeht und diese nicht ohne weiteres auf Vektoren und Matrizen angewendet werden können, muss der Abstandsbegriff für Vektoren und Matrizen geklärt werden.

4.1. Normen

Definition 3. Norm eines Vektors $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}^+ \cup \{0\}$

- a) $\|x\| \geq 0 \quad \forall x \in \mathbb{R}^n, \|x\| = 0 \Leftrightarrow x = 0 \in \mathbb{R}^n$
- b) $\|\alpha x\| = |\alpha| \|x\| \quad \forall x \in \mathbb{R}^n, \alpha \in \mathbb{R}$
- c) $\|x + y\| \leq \|x\| + \|y\|$

Eine Abbildung, die diese Bedingungen erfüllt, heißt **Norm** eines Vektors.

Im Folgenden werden einige gängige Beispiele für Vektornormen angeführt.

Beispiel 2. Normen von Vektoren

- 1.) $\|x\|_1 = \sum_{k=1}^n |x_k|$
- 2.) $\|x\|_2 = \sqrt{\sum_{k=1}^n |x_k|^2}$ (Euklidische Norm)
- 3.) $\|x\|_\infty = \lim_{p \rightarrow \infty} \sqrt[p]{\sum_{k=1}^n |x_k|^p} = \max_{k=1, \dots, n} |x_k|$

Prinzipiell wäre eine Eigenschaft wie die *absolute Homogenität* (Def. 3b) auch für Matrix-Vektor-Produkte wünschenswert.

$$\|A \cdot x\| = \|A\| \cdot \|x\|$$

Es lässt sich aber leicht sehen, dass diese Gleichung im Allgemeinen für kein $\|A\|$ erfüllbar ist. Wir schwächen die Forderung ab und suchen nur nach einer reellen Zahl, die die *Submultiplikativität* erfüllt:

$$\|A \cdot x\| \leq \|A\| \cdot \|x\|$$

$$\Leftrightarrow_{\|x\| \neq 0} \|A\| \geq \frac{\|A \cdot x\|}{\|x\|} \quad \forall x \in \mathbb{R}^n, x \neq 0.$$

Diese Zahl wird nicht eindeutig sein. Entsprechend wählen wir das kleinstmögliche $\|A\|$.

Definition 4. (Matrixnorm)

Die von einer Vektornorm induzierte Matrixnorm ist wie folgt definiert:

$$\|A\| := \sup_{x \neq 0} \frac{\|A \cdot x\|}{\|x\|} = \sup_{\|x\|=1} \|A \cdot x\|.$$

Insbesondere ergibt sich aus der ∞ -Norm eines Vektors die ∞ -Norm einer Matrix, bzw. die sogenannte Zeilensummennorm.

Definition 5. (Zeilensummennorm für $A \in \mathbb{R}^{m,n}$)

$$\|A\|_\infty = \max_{i=1 \dots m} \sum_{j=1}^n |a_{ij}| \quad \Rightarrow \quad \|A \cdot x\|_\infty \leq \|A\|_\infty \cdot \|x\|_\infty$$

Die definierten Normen erlauben eine Anwendung des Fixpunktsatzes auf $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}^n$.

Satz 3. (Konvergenz des Jacobi-Verfahrens)

Die Iteration

$$x^{[k+1]} = -D^{-1}(A - D)x^{[k]} + D^{-1}b$$

konvergiert für beliebige $x^{[0]}, b \in \mathbb{R}^n$, wenn A **strikt diagonaldominant** ist, d.h. $|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}|$ (Diagonale ist größer als Summe des Betrags der restlichen Zeile).

Beweis: Um den Fixpunktsatz von Banach anwenden zu können, muss φ eine Kontraktion sein. Da $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}^n$, ist φ eine Selbstabbildung.

Betrachte

$$\|\varphi(x) - \varphi(y)\|_\infty = \|M_J \cdot x + c_J - M_J \cdot y - c_J\|_\infty = \|M_J \cdot (x - y)\|_\infty \leq \|M_J\|_\infty \cdot \|x - y\|_\infty,$$

mit

$$\begin{aligned} \|M_J\|_\infty &= \|D^{-1}(A - D)\|_\infty = \left\| \begin{bmatrix} a_{11}^{-1} & \cdot & (& 0 & a_{12} & a_{13} & \dots & a_{1n} &) \\ a_{22}^{-1} & \cdot & (& a_{21} & 0 & a_{23} & \dots & a_{2n} &) \\ \vdots & \cdot & (& \vdots & \vdots & \vdots & \vdots & \vdots &) \\ a_{nn}^{-1} & \cdot & (& a_{n1} & a_{n2} & \dots & a_{nn-1} & 0 &) \end{bmatrix} \right\|_\infty \\ &= \left\| \begin{bmatrix} 0 & \frac{a_{12}}{a_{11}} & \frac{a_{13}}{a_{11}} & \dots & \frac{a_{1n}}{a_{11}} \\ \frac{a_{21}}{a_{22}} & 0 & \frac{a_{23}}{a_{22}} & \dots & \frac{a_{2n}}{a_{22}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{a_{n1}}{a_{nn}} & \frac{a_{n2}}{a_{nn}} & \dots & \frac{a_{nn-1}}{a_{nn}} & 0 \end{bmatrix} \right\|_\infty \\ &= \max_{i=1 \dots n} \sum_{j=1, i \neq j}^n \frac{|a_{ij}|}{|a_{ii}|} \stackrel{!}{<} 1 \\ &\Rightarrow \sum_{j=1, i \neq j}^n \frac{|a_{ij}|}{|a_{ii}|} < 1 \quad \forall i = 1 \dots n \\ &\Leftrightarrow \sum_{j=1, i \neq j}^n |a_{ij}| < |a_{ii}| \quad \forall i = 1 \dots n \quad (\text{strikte Diagonaldominanz}) \end{aligned}$$

Damit konvergiert φ , wenn A strikt diagonaldominant ist. □

Bemerkung 1. Die gleiche Bedingung gilt für das Gauß-Seidel-Verfahren. (ohne Beweis)

5. Optimierungen der numerischen Lösungsverfahren (Jacobi)

5.1. Fehlerabschätzung

Betrachtet man die Abweichung der ermittelten Näherungslösung von der exakten Lösung $x^* = A^{-1}b$ des LGS $Ax = b$, so lässt sich diese folgendermaßen abschätzen:

$$\begin{aligned}\|x^{[k]} - x^*\|_\infty &= \|x^{[k]} - A^{-1}b\|_\infty \\ &= \|A^{-1}(Ax^{[k]} - b)\|_\infty \leq \|A^{-1}\|_\infty \cdot \underbrace{\|Ax^{[k]} - b\|_\infty}_{r^{[k]}}.\end{aligned}$$

Das sogenannte *Residuum* $r^{[k]} = Ax^{[k]} - b$ liefert ein Maß für den Fehler, denn für $r^{[k]} = 0$ ist auch $x^{[k]} - x^* = 0$. Mithilfe des Residuums lässt sich die Iterationsvorschrift des Jacobi-Verfahrens auch folgendermaßen schreiben:

$$\begin{aligned}x^{[k+1]} &= M_J x^{[k]} + C_J \\ &= x^{[k]} - D^{-1}(Ax^{[k]} - b) \\ &= x^{[k]} - D^{-1}r^{[k]}.\end{aligned}$$

Damit ergibt sich der neue Jacobi-Schritt $x^{[k+1]}$ als Summe des alten Schritts $x^{[k]}$ und einer Korrektur $-D^{-1}r^{[k]}$. Diese Korrektur wird umso kleiner, je näher $x^{[k]}$ an x^* kommt.

5.2. Relaxiertes Jacobi-Verfahren

Überlegung. Die Korrektur der Näherungslösung kann in jedem erneuten Iterationsschritt durch den Parameter α skaliert werden.

$$x^{[k+1]} = x^{[k]} - \alpha D^{-1}r^{[k]} \quad \alpha \in]0, 2[$$

Für $\alpha > 1$ spricht man von einer Überrelaxation, für $\alpha < 1$ von einer Unterrelaxation. Mittels Abschwächung des Werts der Korrektur ($\alpha < 1$) kann ein Oszillieren um die Lösung x^* vermindert werden. Im konkreten Fall unseres Modellproblems können hochfrequente Schwingungen gefiltert werden, wodurch eine schnellere Konvergenz erreicht werden kann.

Überlegung. Im konkreten Fall sogenannter Laplace- oder Poisson-Gleichungen wie z.B. im Einführungsbeispiel (1) erhält man als Systemmatrix des LGS

$$A = \frac{1}{h^2} \cdot \begin{bmatrix} -2 & 1 & & & \\ 1 & \ddots & \ddots & & \\ & \ddots & \ddots & 1 & \\ & & & 1 & -2 \end{bmatrix}. \quad (5)$$

Das Jacobi-Verfahren für diese Probleme hat die Form:

$$\begin{aligned}x^{[k+1]} &= x^{[k]} - D^{-1}r^{[k]} \\ &= x^{[k]} + \frac{h^2}{2}r^{[k]}.\end{aligned}$$

Dies kann wie folgt umgeformt werden.

$$\begin{aligned} Ax^{[k+1]} &= Ax^{[k]} + \frac{h^2}{1} Ar^{[k]}, \\ Ax^{[k+1]} - b &= Ax^{[k]} - b + \frac{h^2}{2} Ar^{[k]}, \\ r^{[k+1]} &= r^{[k]} + \frac{h^2}{2} Ar^{[k]}. \end{aligned}$$

Genauer betrachten von $\frac{h^2}{2} A \cdot r^{[k]}$ liefert

$$\frac{h^2}{2} A = \frac{h^2}{2} \frac{1}{h^2} \cdot \begin{bmatrix} -2 & 1 & & & \\ 1 & \ddots & \ddots & & \\ & \ddots & \ddots & 1 & \\ & & & 1 & -2 \end{bmatrix} = \begin{bmatrix} -1 & \frac{1}{2} & & & \\ \frac{1}{2} & \ddots & \ddots & & \\ & \ddots & \ddots & \frac{1}{2} & \\ & & & \frac{1}{2} & -1 \end{bmatrix}.$$

Bei Simulationen lässt sich beobachten, dass manche Formen des Residuums bei einem Jacobi-Schritt nur um einen sehr geringen Faktor kleiner werden. Dies soll hier genauer untersucht werden.

Überlegung. (Ansatz: Sinus als Residuum) Wir betrachten das Residuum r als eine Überlagerung einzelner Sinusschwingungen

$$r_0 = 0, \quad r_n = 0, \quad r_i = \sin(x_i \cdot \pi \cdot \omega) = \sin\left(\frac{i}{n} \cdot \pi \cdot \omega\right).$$

Die Bedingung $r_0 = r_n = 0$ ist dadurch gerechtfertigt, dass die Randwerte $T = 0$ und T_n vorgegeben sind, an diesen Stellen also keine Abweichung von der exakten Lösung auftritt und das Residuum damit 0 ist.

Eine Zeile von $A \cdot r$ lautet

$$\frac{1}{2} r_{i-1} - r_i + \frac{1}{2} r_{i+1}.$$

Einsetzen von r_i führt auf:

$$\begin{aligned} & \sin\left(\frac{i}{n} \cdot \pi \cdot \omega\right) \cdot \left(\cos\left(\frac{\pi}{n} \cdot \omega\right) - 1\right) \\ &= \sin(r_i) \cdot \underbrace{\left(\cos\left(\frac{\pi}{n} \cdot \omega\right) - 1\right)}_{\lambda}. \end{aligned}$$

Es ergibt sich

$$\begin{aligned} r^{[k+1]} &= r^{[k]} + \frac{h^2}{2} A \cdot r^{[k]} \\ &= r^{[k]} + \cos\left(\left(\frac{\pi}{n} \cdot \omega\right) - 1\right) \cdot r^{[k]} \\ &= \underbrace{\cos\left(\frac{\pi}{n} \cdot \omega\right)}_{\lambda} r^{[k]}. \end{aligned}$$

Daraus sind mehrere Effekte erkennbar. Zuerst ist $|\lambda| < 1$ für alle $\omega = 1, \dots, n-1$. Damit konvergiert das Jacobi-Verfahren auch für Matrizen der Form (5), auch wenn diese nicht strikt diagonaldominant sind. Zum anderen ist $|\lambda| \approx 1$ für sehr kleine und sehr große ω . Ein Residuum, welches also entweder sehr wenig oder sehr stark schwingt, wird durch das Verfahren fast gar nicht gedämpft. Residuen die einer

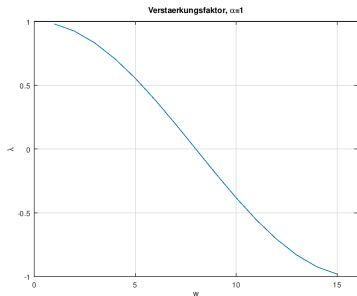
Schwingung mit $\omega \approx \frac{\pi}{2}$ entsprechen werden hingegen innerhalb eines Schrittes gedämpft. Wir bringen den Relaxationsfaktor α ins Spiel und erhalten

$$\begin{aligned} r^{[k+1]} &= r^{[k]} + \alpha \frac{h^2}{2} A \cdot r^{[k]} \\ &= \left(1 + \alpha \left(\cos\left(\frac{\pi}{n} w\right) - 1 \right) \right) r^{[k]}. \end{aligned}$$

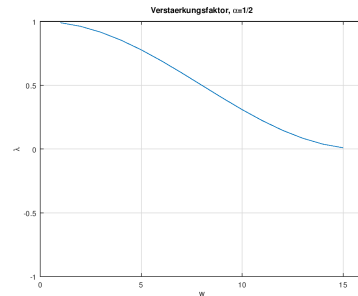
Und im konkreten Fall von $\alpha = \frac{1}{2}$ ergibt sich

$$r^{[k+1]} = \underbrace{\frac{1 + \cos\left(\frac{\pi}{n} w\right)}{2}}_{\lambda} r^{[k]}.$$

Damit ist auch in diesem Fall der Dämpfungsfaktor aus $(0, 1)$. Auch hier werden niederfrequente Lösungsanteile kaum gedämpft. Anders als bei $\alpha = 1$ werden hier die hochfrequenten Anteile sehr stark gedämpft.



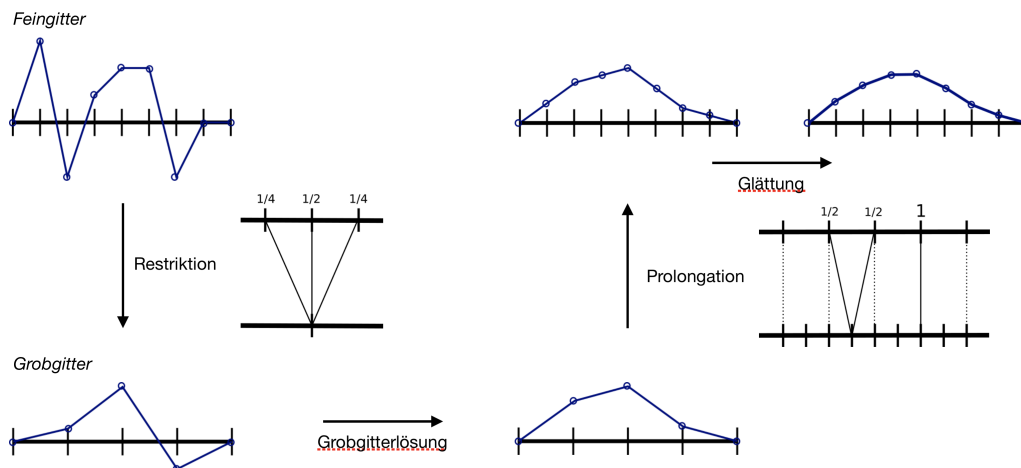
Das normale Jacobi-Verfahren ($\alpha = 1$) dämpft die sehr niederfrequenten und sehr hochfrequenten Lösungsanteile kaum. Ein Schritt des mit $\alpha = \frac{1}{2}$ unterrelaxierten Verfahrens dämpft vor allem hochfrequente Lösungsanteile. Dies wird beim Mehrgitterverfahren genutzt.



5.3. Mehrgitterverfahren

Überlegung. Der Grundgedanke des Mehrgitterverfahrens besteht darin, dass man eine sehr grob diskretisierte Lösung als Grundlage für eine Lösung auf einem feineren Gitter nutzt. Niederfrequente Anteile des Residuums werden durch eine Vergrößerung (**Restriktion**) des Gitters hochfrequent. In einem größeren Gitter können die hochfrequenten Unbekannten mit weniger Schritten an eine Lösung angenähert werden. Anschließend wird das Gitter wieder verfeinert (**Prolongation**). Die dadurch entstehenden Näherungslösungen auf dem Feingitter sind ein guter Ausgangspunkt für eine Nachiteration mit Jacobi. Zuerst sollte aber ein Glättungsschritt erfolgen, da die Prolongation einen sehr hochfrequenten Fehler in das Residuum einbringt.

Schematisch ist das Ganze im Folgenden dargestellt.



6. Beispielrechnung (Temperaturverteilung)

Wie in Abschnitt 2 wird ein Metallbalken betrachtet, dessen Temperatur an beiden Seiten konstant gehalten wird. Ziel ist es, die Temperatur an jedem Ort im Metallbalken in Abhängigkeit der Randbedingungen bestimmen zu können.

Anstatt der Wärmeleitungsgleichung wie in (1), welche Wärmeabgabe an die Umgebung ignoriert, betrachten wir einen Balken, der sich in einer Umgebung von $T_u = 20^\circ C$ befindet und mit der Umgebung Wärme tauschen kann.

$$T''(x) - \frac{T - T_u}{\lambda} = 0 \Leftrightarrow T'' - \frac{T}{\lambda} = -\frac{T_u}{\lambda}.$$

Nach Diskretisierung nimmt diese Gleichung die folgende Form an:

$$\begin{aligned} T''(x) - \frac{T(x)}{\lambda} &= -\frac{T_u}{\lambda} \\ \rightsquigarrow \frac{T(x+h) - 2T(x) + T(x-h)}{h^2} - \frac{T(x)}{\lambda} &= -\frac{T_u}{\lambda} \\ \rightsquigarrow \frac{T_{k+1} - 2T_k + T_{k-1}}{h^2} - \frac{T_k}{\lambda} &= -\frac{T_u}{\lambda}. \end{aligned}$$

Diese rekursive Vorschrift lässt sich dann als LGS schreiben.

$$\begin{aligned} \frac{1}{h^2}T_2 - \frac{2}{h^2}T_1 - \frac{1}{\lambda}T_1 &= -\frac{T_u}{\lambda} - T_0 \\ \frac{1}{h^2}T_3 - \frac{2}{h^2}T_2 + \frac{1}{h^2}T_1 - \frac{1}{\lambda}T_2 &= -\frac{T_u}{\lambda} \\ &\vdots \\ -\frac{2}{h^2}T_{n-1} + \frac{1}{h^2}T_{n-2} - \frac{1}{\lambda}T_{n-1} &= -\frac{T_u}{\lambda} - \frac{1}{h^2}T_n. \end{aligned}$$

In Matrix-Vektor-Schreibweise ergibt das dann:

$$\begin{aligned} \frac{1}{h^2} \cdot \begin{bmatrix} -2 & 1 & & & \\ 1 & \ddots & \ddots & & \\ & \ddots & \ddots & 1 & \\ & & & 1 & -2 \end{bmatrix} \cdot \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_{n-1} \end{bmatrix} - \frac{1}{\lambda} \cdot \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_{n-1} \end{bmatrix} &= \begin{bmatrix} -\frac{T_u}{\lambda} - \frac{1}{h^2}T_0 \\ -\frac{T_u}{\lambda} \\ \vdots \\ -\frac{T_u}{\lambda} - \frac{1}{h^2}T_n \end{bmatrix} \\ \Leftrightarrow \frac{1}{h^2} \cdot \underbrace{\begin{bmatrix} -2 - \frac{h^2}{\lambda} & 1 & & & \\ 1 & \ddots & \ddots & & \\ & \ddots & \ddots & 1 & \\ & & & 1 & -2 - \frac{h^2}{\lambda} \end{bmatrix}}_{A_W} \cdot \underbrace{\begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_{n-1} \end{bmatrix}}_t &= \underbrace{\begin{bmatrix} -\frac{T_u}{\lambda} - \frac{1}{h^2}T_0 \\ -\frac{T_u}{\lambda} \\ \vdots \\ -\frac{T_u}{\lambda} - \frac{1}{h^2}T_n \end{bmatrix}}_{b_W} \end{aligned}$$

Auch dieses LGS können wir in der Form $A_W \cdot t = b_W$ mit strikt diagonaldominantem A_W schreiben. Es ergibt sich damit die folgende Fixpunktiteration für das Jacobi-Verfahren

$$x^{[k+1]} = \varphi(x^{[k]}) = \underbrace{-D^{-1}(A_W - D)}_{M_W} x^{[k]} + \underbrace{D^{-1}b}_{c_W}.$$

Die numerische Lösung dieses Gleichungssystems bzw. die obige Iteration haben wir in Octave programmiert und konnten damit die Temperaturverteilung in dem Metallbalken bestimmen. Graphische Darstellungen der einzelnen Schritte finden sich in den Abbildungen 3 bis 10. Dabei ist das obere Bild

jeweils der aktuelle Lösungsverlauf mit eingeblendeter Referenzlösung. Darunter ist das zugehörige Residuum dargestellt. Aus Gründen der Übersichtlichkeit wurde eine Diskretisierung mit $n = 16$ Intervallen, also $n - 1 = 15$ Unbekannten betrachtet.

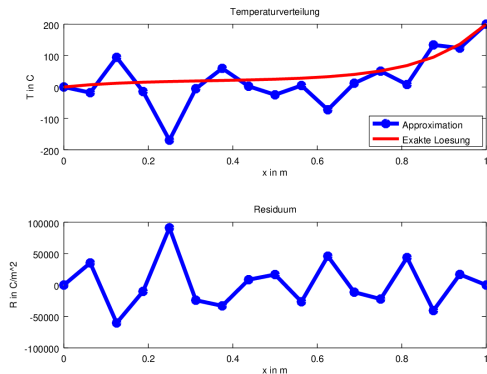


Abb.3: Zufällig initialisierte Startwerte

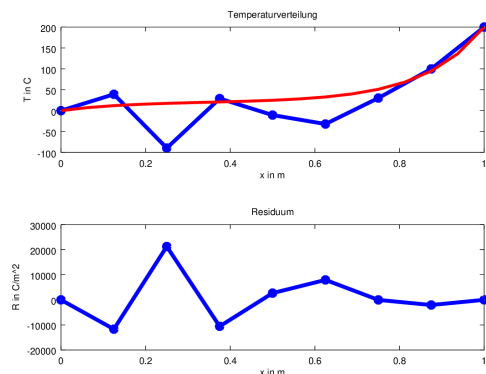


Abb.4: Erster Restriktionsschritt

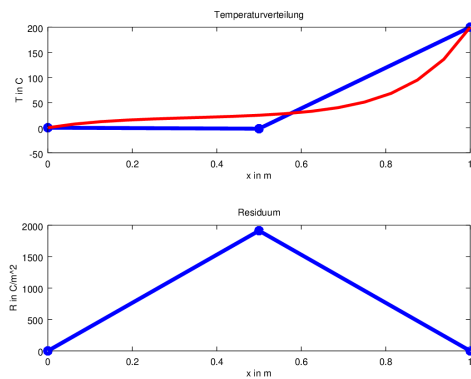


Abb.5: Maximale Restriktionsstufe

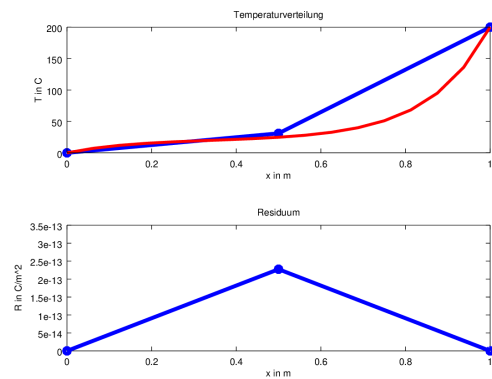


Abb.6: Jacobi-Verfahren

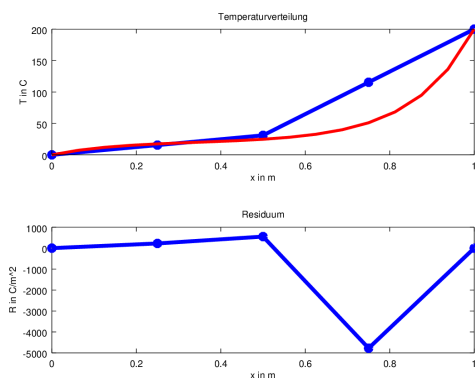


Abb.7: Erste Prolongation

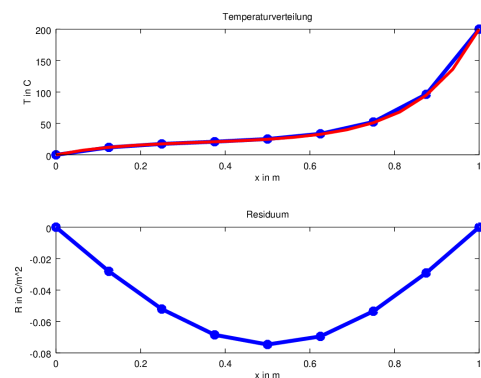


Abb.8: Näherung für sieben Unbekannte

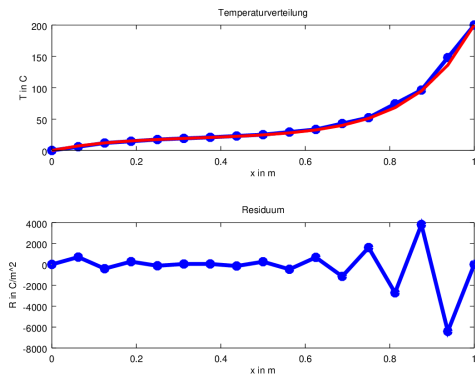


Abb.9: Oszillationen nach Prolongation

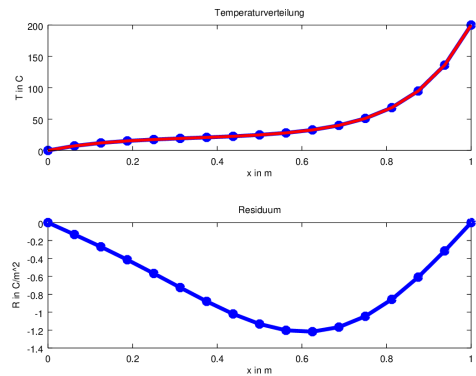


Abb.10: Finale Lösung

Zuerst wurde maximal restringiert, bis die Anzahl der Unbekannten auf 1 reduziert worden ist (Abb. 5). Diese Gleichung wurde mit Jacobi exakt gelöst (Abb. 6). Die Prolongation auf drei Unbekannte (Abb. 7) weist an den interpolierten Stellen deutliche Abweichungen auf. Diese wurden aber mit zwei relaxierten ($\alpha = \frac{1}{2}$) Jacobi Schritten weitestgehend behoben. Schrittweise wurde weiter prolongiert und jeweils zweimal nachgeglättet. In Abb. 9 ist das Resultat nach der letzten Prolongation dargestellt. Deutlich sichtbar ist auch das stark oszillierende Residuum, welches durch Glättung fast vollständig verschwindet (Abb. 10).

Es ist anzumerken, dass nur zwei Jacobi-Schritte auf dem feinsten Gitter durchgeführt wurden. Alle weiteren Schritte erfolgten auf größeren Gittern mit jeweils deutlich weniger Unbekannten. Damit liegt der Gesamtaufwand für die Berechnung bei etwa 4 Jacobi-Schritten auf dem Feingitter. Diese Betrachtung ist auch für deutlich größere n noch zutreffend. Damit liegt der Aufwand für ein vollständig durchgeführtes Mehrgitterverfahren in der Größenordnung einiger n^2 im Gegensatz zu n^3 bei Gauß.

7. Zusammenfassung

Zum Lösen von großen Gleichungssystemen nutzen wir das Gauß-Seidel-Verfahren und das Jacobi-Verfahren. Beide Verfahren lösen Gleichungssysteme iterativ. Das verringert den Rechenaufwand.

So ist das exakte Lösen eines LGS mit n Unbekannten mittels Gauß-Verfahren etwa n mal so teuer wie das numerische Annähern durch das Jacobi Verfahren. Bei einem LGS mit 1000 Unbekannten könnten so statt einer Gauß-Lösung 1000 Jacobi-Iterationen gemacht werden.

Wir konnten für das Jacobi-Verfahren ein einfach zu überprüfendes Konvergenzkriterium des Iterationsverfahrens herleiten.

Weiterhin haben wir das Konzept der Mehrgitterverfahren erfolgreich auf ein realitätsnahes Beispiel angewandt und damit zeigen können, dass mit insgesamt geringem Aufwand bei geschickter Restriktion und Prolongation ein relativ großes Gleichungssystem schnell gelöst werden kann.