Numerik II

Roland Pulch

Institut für Mathematik und Informatik

Mathematisch-Naturwissenschaftliche Fakultät

Universität Greifswald

Skript zu Iterativer Lösung linearer Gleichungssysteme

Literatur:

Kanzow, Ch.: Numerik linearer Gleichungssysteme: Direkte und iterative Verfahren, Springer Verlag, 2005. (Kapitel 4–6)

Stoer, J.; Bulirsch, R.: Numerische Mathematik 2 (5. Aufl.), Springer Verlag, 2005. (Kapitel 8)

Deuflhard, P.; Hohmann, A.: Numerische Mathematik I (3. Aufl.), de Gruyter Verlag, 2002. (Kapitel 8)

Kapitel

Iterative Lösung großer linearer Gleichungssysteme

1 Motivation

In diesem Kapitel wird die iterative Lösung von linearen Gleichungssystemen

$$Ax = b, \qquad A \in \mathbb{R}^{n\times n}, \quad \det A \neq 0, \quad x, b \in \mathbb{R}^n \tag{1}$$

behandelt. Da eine LR-Zerlegung ca. $\frac{2}{3}n^3$ Rechenoperationen erfordert, wird eine direkte Lösung für großes n extrem aufwendig. In den Anwendungen sind große Matrizen A typischerweise dünnbesetzt (englisch *sparse*), d.h. nur wenige Elemente sind ungleich null. Ein Weg, diese Besetzungsstruktur auszunutzen, führt auf Varianten der LR-Zerlegung, bei denen der Eliminationsprozess die Anzahl der entstehenden, von null verschiedenen Einträge (englisch *fill-in*) minimiert, meist mit Hilfe der Graphentheorie und eingeschränkter Pivotsuche. Da nur Rechenoperationen mit von null verschiedenen Einträgen auch tatsächlich durchgeführt werden, sind diese *direct sparse solver* für den Bereich $1000 \leq n \leq 100\,000$ eine bewährte Wahl, siehe in MATLAB den Befehl `sparse`.

Schwerpunkt im folgenden ist jedoch eine Einführung in iterative Verfahren, die auch bei deutlich höheren Dimensionen zum Erfolg führen. Die Größenordnung zur Anzahl n an Unbekannten ist dann durchaus im Bereich von Millionen oder Milliarden in den praktischen Anwendungen. Die Iteration liefert eine Folge $(x^k)_{k\in\mathbb{N}} \subset \mathbb{R}^n$ aus Näherungen für $x = A^{-1}b$

$$x^0 \quad \to \quad x^1 \quad \to \quad x^2 \quad \to \quad x^3 \quad \to \quad \cdots$$

ausgehend von einem vorzugebendem Startwert $x^0 \in \mathbb{R}^n$.

Dabei soll jeder Iterationsschritt möglichst geringen Rechenaufwand besitzen. Konkreter soll der Aufwand eines einzelnen Schritts den einer Matrix-Vektor-Multiplikation mit A nicht wesentlich übersteigen (wobei die Besetzungsstruktur selbstverständlich ausgenutzt wird und keine unnötigen Operationen mit Nullen durchgeführt werden).

Beispiel zu dünnbesetzter direkter Lösung

Wir betrachten ein lineares Gleichungssystem (1) der Dimension $n = 6900$, welches aus einer Anwendung mit einem dynamischen System entstand. Die Tabelle unten enthält die Anzahl der Einträge ungleich null in der Koeffizientenmatrix A sowie deren relativen Anteil (d.h. nach Division durch die Gesamtanzahl n^2 der Einträge). Abbildung 1 verdeutlicht zudem die Besetzungsstruktur dieser Matrix. Somit ist die Matrix dünnbetzt.

Zum einen wird der übliche Gauß-Algorithmus mit Spaltenpivotsuche (d.h. mit Zeilenvertauschungen) durchgeführt, wobei nur die Elemente ungleich null in der Matrix abgearbeitet werden. Die entstehende LR-Zerlegung wird in einer Matrix abgespeichert. Die Tabelle und Abbildung 2 zeigen, dass hier viele neue Einträge ungleich null entstehen. Der Speicherbedarf und der Rechenaufwand sind somit hoch.

Zum anderen wird ein Algorithmus für eine LR-Zerlegung mit vollständiger Pivotsuche (d.h. mit Zeilen- und Spaltenvertauschungen) verwendet. Dabei erfolgt die Pivotisierung nur zum Teil zur numerischen Stabilität und soll hauptsächlich die Anzahl der neuen Einträge ungleich null klein halten. Die Tabelle und Abbildung 2 demonstrieren, dass dieses Ziel erreicht werden kann. Dadurch sinkt der Speicherbedarf und der Rechenaufwand deutlich.

Vergleich zu Anzahl der Nicht-Null-Einträge in den Matrizen:

Matrix	Nicht-Null-Einträge	Anteil
Koeffizientenmatrix A	64 378	0.14%
LR-Zerlegung mit Spaltenpivot.	3 415 484	7.17%
LR-Zerlegung mit vollständiger Pivot.	212 111	0.45%

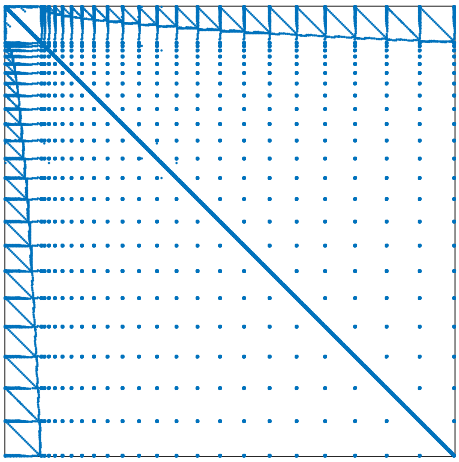


Abbildung 1: Beispiel einer dünnbesetzten Koeffizientenmatrix in einem großen linearen Gleichungssystem. Einträge ungleich null sind mit blauer Farbe dargestellt.

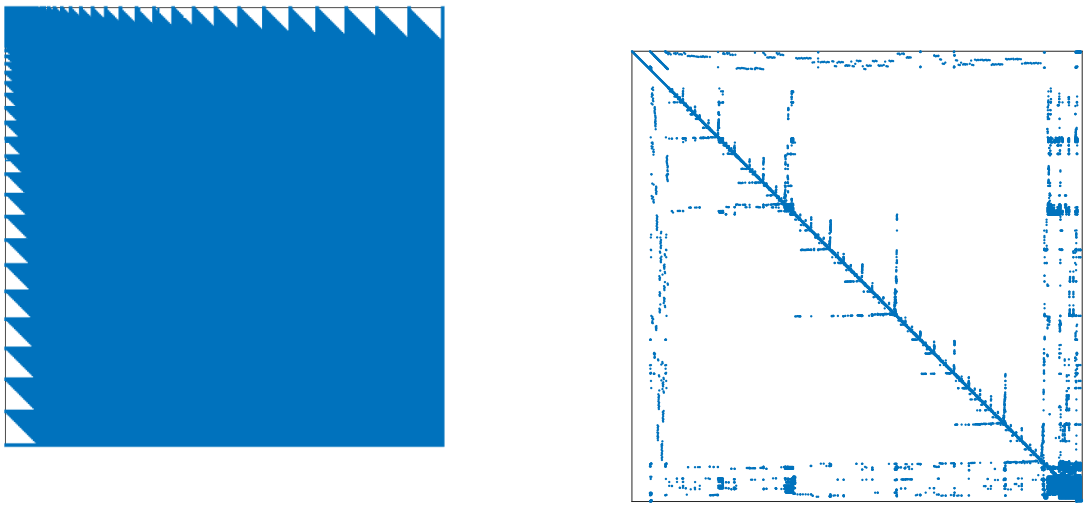


Abbildung 2: LR-Zerlegung mit Spaltenpivotsuche (links) und LR-Zerlegung mit vollständiger Pivotsuche zur Reduzierung der Nicht-Null-Einträge (rechts).

2 Matrixnormen und Spektralradius

Die Größe eines Vektors wird durch eine Vektornorm $\|\cdot\|_V$ quantifizert. Übliche Vektornormen für $x \in \mathbb{R}^n$ sind die Summennorm $\|\cdot\|_1$, die Euklidische Norm $\|\cdot\|_2$ und die Maximumnorm $\|\cdot\|_\infty$

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|\,, \qquad \|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}\,, \qquad \|x\|_\infty = \max_{i=1,\ldots,n} |x_i|\,.$$

Wir betrachten nur quadratische Matrizen in diesem Kapitel. Die Größe einer Matrix kann durch eine Matrixnorm mit analogen Eigenschaften spezifiziert werden.

Def.: Eine Abbildung $\|\cdot\| : \mathbb{R}^{n\times n} \to \mathbb{R}_{\geq 0}$ heißt Matrixnorm, falls die folgenden vier Eigenschaften für $A, B \in \mathbb{R}^{n\times n}$ erfüllt sind:

(i) definit: $A \neq 0 \Rightarrow \|A\| > 0$

(ii) homogen: $\|\lambda A\| = |\lambda| \cdot \|A\|$ für alle $\lambda \in \mathbb{R}$

(iii) subadditiv: $\|A + B\| \leq \|A\| + \|B\|$ (Dreiecksungleichung)

(iv) submultiplikativ: $\|AB\| \leq \|A\| \cdot \|B\|$

Übliche Matrixnormen für $A \in \mathbb{R}^{n\times n}$ sind

- Spaltenbetragssumme: $\|A\|_1 = \max\limits_{j=1,\ldots,n} \sum\limits_{i=1}^{m} |a_{ij}|,$
- Zeilenbetragssumme: $\|A\|_\infty = \max\limits_{i=1,\ldots,m} \sum\limits_{j=1}^{n} |a_{ij}|,$
- Spektralnorm: $\|A\|_2 = \sqrt{\lambda_{\max}(A^\top A)}.$

Die Matrix $A^\top A$ ist stets symmetrisch und positiv semi-definit. Deren Eigenwerte sind somit reell und nichtnegativ. Es bezeichnet $\lambda_{\max}$ den größten Eigenwert der Matrix.

Mit Matrixnormen kann auch die Konditionszahl κ einer regulären Matrix alternativ formuliert werden. Es gilt

$$\kappa(A) = \|A\| \, \|A^{-1}\|.$$

Für eine reguläre symmetrische Matrix gilt bezüglich der Spektralnorm noch

$$\kappa_2(A) = \frac{\max\{|\lambda_1|, \ldots, |\lambda_n|\}}{\min\{|\lambda_1|, \ldots, |\lambda_n|\}}$$

mit deren reellen Eigenwerten $\lambda_1, \ldots, \lambda_n$, welche alle ungleich null sind.

Desweiteren besteht eine Verbindung von Matrixnormen zu Vektornormen.

Def.: Eine Matrixnorm $\|\cdot\|$ auf $\mathbb{R}^{n\times n}$ heißt konsistent zu einer Vektornorm $\|\cdot\|_V$ auf $\mathbb{R}^n$, falls

$$\|Ax\|_V \leq \|A\| \, \|x\|_V$$

für alle $x \in \mathbb{R}^n$ und alle $A \in \mathbb{R}^{n\times n}$ gilt.

Die Spaltenbetragssumme ist konsistent zur Summennorm, die Zeilenbetragssumme zur Maximumnorm und die Spektralnorm zur Euklidischen Norm. Zudem sind dies jeweils die kleinsten Matrixnormen mit dieser Konsistenzeigenschaft.

Eine wichtige Kenngröße für quadratische Matrizen ist der Spektralradius.

Def.: Die Matrix $A \in \mathbb{R}^{n\times n}$ besitze die Eigenwerte $\lambda_1, \ldots, \lambda_n \in \mathbb{C}$. Der Spektralradius der Matrix ist

$$\rho(A) = \max_{i=1,\ldots,n} |\lambda_i|. \tag{2}$$

In der komplexen Zahlenebene ist somit die Menge aller Eigenwerte (d.h. das Spektrum der Matrix A) in einem Kreis um null mit Radius $\rho(A)$ enthalten. Der Spektralradius ist jedoch keine Matrixnorm, da sowohl die Definitheit als auch die Subadditivität verletzt sind.

3 Stationäre Iterationsverfahren

Ein stationäres Iterationsverfahren zur näherungsweisen Lösung des linearen Gleichungssystems (1) besitzt die Form

$$x^{k+1} = \Phi(x^k), \quad k = 0, 1, 2, \ldots \tag{3}$$

bei vorgebenen Startvektor x^0. Stationär bedeutet, dass die Funktion Φ in jedem Iterationsschritt identisch gewählt wird. Die exakte Lösung x von (1) soll der einzige Fixpunkt der Iteration sein.

Eine Iterationsvorschrift entsteht durch die Wahl einer regulären Matrix $B \in \mathbb{R}^{n\times n}$ aus der Zerlegung

$$Bx + (A - B)x = b \quad \Rightarrow \quad Bx^{k+1} + (A - B)x^k = b, \tag{4}$$

welche ein lineares Gleichungssystem für x^{k+1} darstellt. Aufgelöst erhält man

$$x^{k+1} = x^k - B^{-1}(Ax^k - b) = (I - B^{-1}A)x^k + B^{-1}b. \tag{5}$$

Zur Berechnung von x^{k+1} ist ein Matrix-Vektor-Produkt bezüglich A und die Lösung eines linearen Gleichungssystems bezüglich B notwendig. Die Iterationsmatrix $I - B^{-1}A$ dient nur zur Untersuchung der Konvergenz.

Die Matrix B soll in diesem Zusammenhang zwei Eigenschaften besitzen:

1. Lineare Gleichungssysteme mit der Matrix B sollen leicht auflösbar sein, damit die Iteration mit wenig Aufwand durchführbar ist.
2. Die Matrix B soll A gut approximieren, d.h. wesentliche Informationen aus A enthalten, damit die Konvergenz der Iteration gesichert ist.

Diese beiden gewünschten Eigenschaften stehen im Gegensatz zueinander. Es folgt, dass nur für bestimmte Matrizentypen als A eine geeignete Matrix B gefunden werden kann.

Zur Untersuchung der Konvergenz ist der Begriff des Spektralradius aus (2) von zentraler Bedeutung.

Satz 1 (Konvergenz stationärer Verfahren)
Das Iterationsverfahren (5) ist für beliebigen Startwert genau dann konvergent, wenn gilt

$$\rho(I - B^{-1}A) < 1. \tag{6}$$

Hinreichend für die Konvergenz von (5) bei beliebigem Startwert ist die Bedingung

$$||I - B^{-1}A|| < 1, \tag{7}$$

wobei $||\cdot||$ eine beliebige Matrixnorm ist, die konsistent zu einer bestimmten Vektornorm.

Beweis:

Für den Fehler $f^k := x^k - x$ ($x = A^{-1}b$) hat man mit

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

durch Subtraktion die Rekursionsformel

$$f^{k+1} = (I - B^{-1}A)f^k$$

und damit

$$f^k = (I - B^{-1}A)^k f^0, \quad k = 0, 1, 2, \ldots .$$

Sei (5) konvergent. Ist λ ein Eigenwert von $I - B^{-1}A$, so wähle man f^0 als zugehörigen Eigenvektor. Damit gilt $f^k = \lambda^k f^0$. Durch die Konvergenz ist $\lim f^k = 0$ und somit notwendigerweise $|\lambda| < 1$.

Sei umgekehrt $\rho(I - B^{-1}A) < 1$. Zur Iterationsmatrix und gegebenem $\varepsilon > 0$ existiert eine Vektornorm, so dass in der korrespondierenden Matrixnorm

$$||I - B^{-1}A|| < \rho(I - B^{-1}A) + \varepsilon$$

gilt (ohne Beweis). Damit folgt

$$||(I - B^{-1}A)^k|| \leq ||I - B^{-1}A||^k \leq (\rho(I - B^{-1}A) + \varepsilon)^k = \mu^k$$

mit $\mu < 1$ für ε hinreichend klein. Also ist $\lim(I - B^{-1}A)^k = 0$ und somit $\lim f^k = 0$ für alle f^0.

Sei schließlich $||I - B^{-1}A|| < 1$ in einer Matrixnorm, die konsistent zu einer festen Vektornorm ist. Für eine solche Matrixnorm gilt stets $\rho(C) \leq \|C\|$ (ohne Beweis). Damit ist das hinreichende Kriterium aus dem ersten Teil des Satzes erfüllt. □

Dem Beweis entnimmt man insbesondere die Abschätzung

$$||x^{k+1} - x|| \leq ||I - B^{-1}A|| \cdot ||x^k - x|| \tag{8}$$

in einer beliebigen Vektornorm und der induzierten Matrixnorm. Damit gilt für $||I - B^{-1}A|| < 1$ dann globale lineare Konvergenz in dieser Norm. Dies kann man hier auch aus dem *Banach'schen Fixpunktsatz* folgern.

Die Konvergenzgeschwindigkeit des Verfahrens (5) ist umso höher, je kleiner der Spektralradius der Iterationsmatrix ist. Der folgende Satz gibt eine Aussage, die von Matrixnormen unabhängig ist.

Satz 2 (Konvergenzgeschwindigkeit)
Beim Iterationsverfahren (5) gilt für den Fehler $f^k = x^k - x$

$$\sup_{f^0 \neq 0} \limsup_{k \to \infty} \left(\frac{||f^k||}{||f^0||} \right)^{1/k} = \rho(I - B^{-1}A) \tag{9}$$

in einer beliebigen Vektornorm $|| \cdot ||$.

Beweis siehe z.B. Stoer, Bulirsch: Numerische Mathematik 2.

4 Klassische Iterationsverfahren

Im folgenden werden gängige stationäre Iterationsverfahren vorgestellt. Sie entstehen durch die Zerlegung der Gestalt

$$A = D + L + R, \tag{10}$$

wobei D Diagonalmatrix mit der Diagonalen von A und L, R den unteren bzw. oberen Dreiecksanteil (ohne Diagonale) von A enthalten.

Jacobi-Verfahren

Ein einfaches Iterationsverfahren entsteht, wenn man in (5) $B = D$ wählt, also nur die Diagonale von A. Gilt stets $a_{ii} \neq 0$, so kann die Invertierung sofort vorgenommen werden. Wir erhalten als Iteration

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

Es ergibt sich für jede Komponente $i = 1, \ldots, n$ die Formel

$$x_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j^k \right). \tag{11}$$

Da jede Komponente sofort separat berechnet werden kann, nennt man diese Iteration auch *Gesamtschrittverfahren*.

Das Jacobi-Verfahren ist konvergent, wenn die Diagonale von A den Hauptanteil der gesamten Matrix bildet. Die Matrix A erfüllt das *starke Zeilensummenkriterium*, wenn gilt

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \quad \text{für} \quad i = 1, \ldots, n. \tag{12}$$

Analog erfüllt A das *starke Spaltensummenkriterium*, falls

$$|a_{jj}| > \sum_{i \neq j} |a_{ij}| \quad \text{für} \quad j = 1, \ldots, n. \tag{13}$$

Sowohl (12) als auch (13) ist nach Satz 1 hinreichend für die Konvergenz des Jacobi-Verfahrens. Aus (12) folgt nämlich sofort

$$||I - B^{-1}A||_\infty = \max_{i=1,\ldots,n} \frac{1}{|a_{ii}|} \sum_{j \neq i} |a_{ij}| < 1 \tag{14}$$

in der Zeilenbetragssumme als Matrixnorm und aus (13)

$$||I - B^{-1}A||_1 = \max_{j=1,\ldots,n} \frac{1}{|a_{jj}|} \sum_{i \neq j} |a_{ij}| < 1 \tag{15}$$

in der Spaltenbetragssumme als Matrixnorm.

Unter zusätzlichen Voraussetzungen kann die Konvergenz des Jacobi-Verfahrens noch garantiert werden, wenn in (12) oder (13) nur eine kleinergleich Beziehung gilt, also schwache Summenkriterien. Dies ist bei vielen Matrizen, die aus Diskretisierungen entstehen, gegeben (siehe Anwendungsbeispiel am Ende dieses Abschnitts).

Gauß-Seidel-Verfahren

Mehr Information über die Matrix A wird einbezogen, wenn man die Dreiecksmatrix $B = D + L$ ansetzt. Die Invertierung entspricht dann gerade einer Vorwärtssubstitution, wozu ebenfalls $a_{ii} \neq 0$ notwendig ist. Die Iteration lautet

$$x^{k+1} = x^k - (D+L)^{-1}((D+L+R)x^k - b) = -(D+L)^{-1}(Rx^k - b).$$

Es entsteht für die einzelnen Komponenten $i = 1, \ldots, n$ die Formel

$$x_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{j<i} a_{ij} x_j^{k+1} - \sum_{j>i} a_{ij} x_j^k \right). \tag{16}$$

Da die Komponenten nur sukzessive berechnet werden können, wird diese Iteration auch als *Einzelschrittverfahren* bezeichnet.

Für die Konvergenz des Gauß-Seidel-Verfahrens sind ebenfalls die Bedingungen (12) oder (13) hinreichend. Bei bestimmten Matrix-Klassen lässt sich zeigen, dass das Gauß-Seidel-Verfahren schneller konvergiert als das Jacobi-Verfahren.

Im Gegensatz zum Jacobi-Verfahren konvergiert das Gauß-Seidel-Verfahren auch für symmetrisch positiv definite Matrizen:

Zu zeigen ist, dass die Eigenwerte von $K = -(D+L)^{-1}L^\top$ im Einheitskreis liegen. Mit A ist auch D positiv definit. Damit haben K und

$$K' := D^{\frac{1}{2}} K D^{-\frac{1}{2}} = -(I + \tilde{L})^{-1}\tilde{L}^\top \quad \text{mit} \quad \tilde{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$$

das gleiche Spektrum. Zu zeigen ist daher nur $\rho(K') < 1$.

Aus $K'x = \lambda x$ mit $x^*x = 1$ ergibt sich

$$-\tilde{L}^\top x = \lambda(I + \tilde{L})x \quad \Rightarrow \quad -x^*\tilde{L}^\top x = \lambda(1 + x^*\tilde{L}x)\,.$$

Mit $\alpha + \mathrm{i}\beta := x^*\tilde{L}^\top x$ folgt daraus

$$|\lambda|^2 = \left| \frac{-\alpha - \mathrm{i}\beta}{1 + \alpha + \mathrm{i}\beta} \right|^2 = \frac{\alpha^2 + \beta^2}{1 + 2\alpha + \alpha^2 + \beta^2} < 1$$

falls $1 + 2\alpha > 0$.

Wegen $x^* D^{-\frac{1}{2}} A D^{-\frac{1}{2}} x = (D^{-\frac{1}{2}} x)^* A (D^{-\frac{1}{2}} x)$ ist die Matrix $D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ positiv definit. Aus $D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = I + \tilde{L} + \tilde{L}^\top$ folgt sofort

$$0 < x^*(I + \tilde{L} + \tilde{L}^\top)x = 1 + x^*\tilde{L}x + x^*\tilde{L}^\top x = 1 + x^*\tilde{L}x + \overline{x^*\tilde{L}x} = 1 + 2\alpha.$$

Das Gauß-Seidel-Verfahren kann analog auch über $B = D + R$ angesetzt werden. Desweiteren existieren symmetrische Varianten, die abwechselnd je einen Schritt mit dem unteren und dann einen Schritt mit dem oberen Dreiecksanteil von B durchführen.

Relaxationsverfahren

In der Iteration (5) kann man allgemeiner die Matrix B in Abhängigkeit von einem Parameter $\omega \in \mathbb{R}$ wählen. Ziel ist es dann, ω optimal zu wählen dahingehend, dass $\rho(I - B(\omega)^{-1}A)$ minimal wird und somit die Konvergenz des Verfahrens zu beschleunigen. Bei den Relaxationsverfahren wird zum Relaxationsparameter $\omega > 0$ über die Zerlegung (10) die Matrix

$$B(\omega) = \frac{1}{\omega}(D + \omega L) \tag{17}$$

gewählt. Wir erkennen, dass für $\omega = 1$ das Einzelschrittverfahren (16) entsteht. Für die Berechnung der einzelnen Komponenten erhalten wir die Formel für $i = 1, \ldots, n$

$$\begin{aligned} z_i^{k+1} &= \frac{1}{a_{ii}} \left(b_i - \sum_{j<i} a_{ij} x_j^{k+1} - \sum_{j>i} a_{ij} x_j^k \right), \\ x_i^{k+1} &= (1-\omega) x_i^k + \omega z_i^{k+1}. \end{aligned}$$

Die Hilfsgröße z_i^{k+1} wird dabei wie im Einzelschrittverfahren (16) bestimmt. Als neue Näherung der i-ten Komponente wird jedoch eine Linearkombination aus alter Näherung und der Hilfsgröße gebildet. Für $0 < \omega \leq 1$ ist dies eine Konvexkombination. Bei $\omega < 1$ spricht man von Unterrelaxation und bei $\omega > 1$ von Überrelaxation. Dadurch entstand der Name *SOR-Verfahren* (successive overrelaxation).

Laut Satz 1 wird die Konvergenz eines stationären Verfahrens durch den Spektralradius der Iterationsmatrix bestimmt. Zum Relaxationsverfahren mit der Matrix (17) geben die beiden nächsten Sätze eine allgemeine Aussage.

Satz 3 (Kahan)
Bei beliebiger regulärer Matrix $A \in \mathbb{R}^{n\times n}$ mit Diagonalelementen ungleich null gilt im SOR-Verfahren für den Spektralradius der Iterationsmatrix

$$\rho(I - B(\omega)^{-1}A) \geq |\omega - 1|$$

für alle $\omega \in \mathbb{R}$.

Satz 4 (Reich-Ostrowsky)
Ist $A \in \mathbb{R}^{n\times n}$ symmetrisch sowie positiv definit, dann gilt im SOR-Verfahren für den Spektralradius der Iterationsmatrix $\rho(I - B(\omega)^{-1}A) < 1$ falls $0 < \omega < 2$.

Der Satz von Kahan zeigt, dass $\omega \in (0, 2)$ notwendig für die Konvergenz des Verfahrens ist. Der Satz von Reich-Ostrowsky liefert bei symmetrischer positiv definiter Matrix, dass $\omega \in (0, 2)$ auch hinreichend für die Konvergenz des Verfahren ist. Bei symmetrischen positiv definiten Matrizen existiert ein eindeutiger optimaler Relaxationsparameter $1 \leq \omega_{\text{opt}} < 2$, d.h.

$$\rho(I - B(\omega_{\text{opt}})^{-1}A) < \rho(I - B(\omega)^{-1}A) \quad \text{für } \omega \neq \omega_{\text{opt}}.$$

Dieser optimale Parameter läßt sich in Spezialfällen explizit berechnen und muss ansonsten geschätzt werden. Die Konvergenz ist dann erheblich schneller als für das Gauß-Seidel-Verfahren.

Ist nur das starke Zeilen- oder Spaltensummenkriterium erfüllt, dann ist $\rho(I - B(\omega)^{-1}A) < 1$ garantiert für $0 < \omega \leq 1$. Auch hier existiert häufig ein optimaler Relaxationsparameter mit $1 \leq \omega_{\text{opt}} < 2$.

Iterative Nachbesserung

In den Kontext der iterativen Verfahren (5) – jedoch nicht auf der Zerlegung (10) basierend – fällt auch die sogenannte *Nachiteration*. Wird das lineare Gleichungssystem $Ax = b$ über LR-Zerlegung auf dem Rechner direkt gelöst, so erhält man eine durch Rundungsfehler beeinträchtigte Lösung $\tilde{x} \approx A^{-1}b$. Diese kann, falls A nicht zu schlecht konditioniert ist, durch diese Nachiteration bis auf Maschinengenauigkeit verbessert werden.

Der Gauß-Algorithmus liefert auf dem Rechner eine durch Rundungsfehler verfälschte LR-Zerlegung $A \approx \hat{L}\hat{R}$, d.h. es gilt nur

$$A = \hat{L}\hat{R} + E, \tag{18}$$

wobei die unbekannte Matrix E Fehleranteile enthält. Für die Matrix B in (5) wählt man dann $B = \hat{L}\hat{R}$ ($\hat{L}, \hat{R}$ haben hier nicht die Bedeutung von L, R aus (10)) und es entsteht die Vorschrift

$$x^{k+1} = x^k - (\hat{L}\hat{R})^{-1}(Ax^k - b) = x^k - (\hat{L}\hat{R})^{-1}r^k \tag{19}$$

mit dem Residuum r der k-ten Näherung. Als Startwert bietet sich die direkte Lösung $x^0 = (\hat{L}\hat{R})^{-1}b$ an.

Die Iteration (19) lässt sich mit wenig Aufwand durchführen, da die Zerlegung $\hat{L}\hat{R}$ bereits berechnet ist und somit nur Vorwärts- und Rückwärtssubstitution erforderlich sind. Da nur Rundungsfehler in $\hat{L}, \hat{R}$ auftreten, gilt $(\hat{L}\hat{R})^{-1}A \approx I$. Folglich ist der Spektralradius der Iterationsmatrix klein und die Konvergenz der Iteration sehr schnell. Zudem liegt über x^0 bereits ein guter Startwert für die Iteration vor. Dementsprechend erhält man im allgemeinen nach etwa zwei Iterationsschritten die Lösung auf Maschinengenauigkeit.

Bei der Berechnung des Residuums r^k tritt durch Subtraktion Auslöschung wegen $Ax^k \approx b$ auf. Dieser Wert muss daher mit höherer Genauigkeit als der Maschinengenauigkeit bei den anderen Operationen berechnet werden. Nur dann kann das Ergebnis auf die ursprüngliche Maschinengenauigkeit erhalten werden. Ist x^0 nur eine grobe Näherung, so kann die Nachbesserung mit stets gleicher Rechengenauigkeit die Anzahl der Stellen erhöhen, jedoch nicht auf die volle Stellenzahl. Die Problematik der Auslösung in (5) tritt sonst bei den Iterationsverfahren nicht auf, da nur eine relativ geringe Genauigkeit erzielt werden soll (d.h. $Ax^k \approx b$ gilt weniger stark).

Anwendungsbeispiel

Wir betrachten das Dirichletsche Randwertproblem für eine reellwertige Funktion $u(x,y)$ im Einheitsquadrat $\Omega := \{(x,y) : 0 < x, y < 1\}$

$$\begin{array}{rcll} -\Delta u = -u_{xx} - u_{yy} & = & f(x,y) & (x,y) \in \Omega \\ u(x,y) & = & 0, & (x,y) \in \partial\Omega \end{array} \tag{20}$$

bei vorgegebener stetiger Funktion f. Die Differenzenquotienten werden auf einem uniformen Gitter der Schrittweite $h := \frac{1}{M+1}$ für $M \in \mathbb{N}$ diskretisiert

$$\Omega_h := \{(x_i, y_j) = (ih, jh) : i, j = 1, \ldots, M\}. \tag{21}$$

Der symmetrische Differenzenquotient zweiter Ordnung liefert als Näherungsformel für $u_{i,j} := u(x_i, y_j)$

$$-\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2} - \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{h^2} \approx f(x_i, y_j) \tag{22}$$

und äquivalent mit $f_{i,j} := f(x_i, y_j)$

$$4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1} \approx h^2 f_{i,j} \tag{23}$$

bei $i, j = 1, \ldots, M$. Die Unbekannten und die rechten Seiten seien in der Form

$$\begin{array}{rcl} u & = & (u_{1,1}, u_{2,1}, \ldots, u_{M,1}, u_{1,2}, \ldots, u_{M,2}, \ldots, u_{1,M}, \ldots, u_{M,M})^\top \\ b & = & h^2(f_{1,1}, f_{2,1}, \ldots, f_{M,1}, f_{1,2}, \ldots, f_{M,2}, \ldots, f_{1,M}, \ldots, f_{M,M})^\top \end{array} \tag{24}$$

angeordnet, wodurch ein lineares Gleichungssystem $Au = b$ der Dimension $n = M^2$ entsteht. Die Matrix A besitzt die folgende Bandstruktur

$$A = \begin{pmatrix} C & -I & & \\ -I & C & \ddots & \\ & \ddots & \ddots & -I \\ & & -I & C \end{pmatrix} \quad \text{mit} \quad C = \begin{pmatrix} 4 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{pmatrix}. \tag{25}$$

Die Matrix A ist dünnbesetzt, da jede Zeile maximal 5 Elemente ungleich null enthält. Desweiteren ist A symmetrisch und positiv definit. Eine Cholesky-Zerlegung $A = LL^\top$ würde jedoch zahlreiche Einträge ungleich null

in L erzeugen, d.h. L ist nicht dünnbesetzt. Dagegen erfordern Matrix-Vektor-Produkte bezüglich A nur vergleichsweise wenig Operationen. (Nur Elemente ungleich null müssen abgearbeitet werden.)

Für die Matrix (25) kann der Spektralradius der jeweiligen Iterationsmatrix $I - B^{-1}A$ für die obigen drei Verfahren explizit berechnet werden. Zur Wahl des optimalen Relaxationsparameters im SOR-Verfahren siehe auch Abbildung 3. Im einzelnen ergeben sich die folgenden Werte:

$$\begin{aligned}
\text{Jacobi-Verfahren:} \qquad \rho(I - D^{-1}A) &= \cos\left(\tfrac{\pi}{M+1}\right) \\
\text{Gauß-Seidel-Verfahren:} \quad \rho(I - (D+L)^{-1}A) &= \cos^2\left(\tfrac{\pi}{M+1}\right) \\
\text{SOR-Verfahren:} \quad \rho(I - B(\omega_{\text{opt}})^{-1}A) &= \frac{\cos^2\left(\frac{\pi}{M+1}\right)}{\left(1+\sin\left(\frac{\pi}{M+1}\right)\right)^2}
\end{aligned}$$

Wir erkennen die Relationen

$$1 > \rho_{\text{J}} > \rho_{\text{GS}} > \rho_{\text{SOR}} > 0 \qquad \text{für festes } \; M \geq 2.$$

Nach Satz 1 ist damit die Konvergenz in allen drei Iterationsverfahren garantiert. Nach Satz 2 ist die Größenordnung des Spektralradius entscheidend für die Konvergenzgeschwindigkeit, d.h. wieviele Schritte für eine bestimmte Genauigkeit benötigt werden. Mit steigender Problemgröße $M \to \infty$ folgt $\rho \to 1$, wodurch die Konvergenz in allen drei Verfahren immer langsamer wird. Für festes M ist jedoch das Gauß-Seidel-Verfahren etwa doppelt so schnell wie das Jacobi-Verfahren wegen $\rho_{\text{GS}} \approx \rho_{\text{J}}^2$. Es lässt sich desweiteren für großes M annähern

$$\rho_{\text{J}}^{\kappa} = \rho_{\text{SOR}} \quad \Rightarrow \quad \kappa = \frac{\ln \rho_{\text{SOR}}}{\ln \rho_{\text{J}}} \approx \frac{4(M+1)}{\pi},$$

wodurch das SOR-Verfahren mit optimalem Relaxationsparameter mehr als M mal so schnell wie das Jacobi-Verfahren konvergiert.

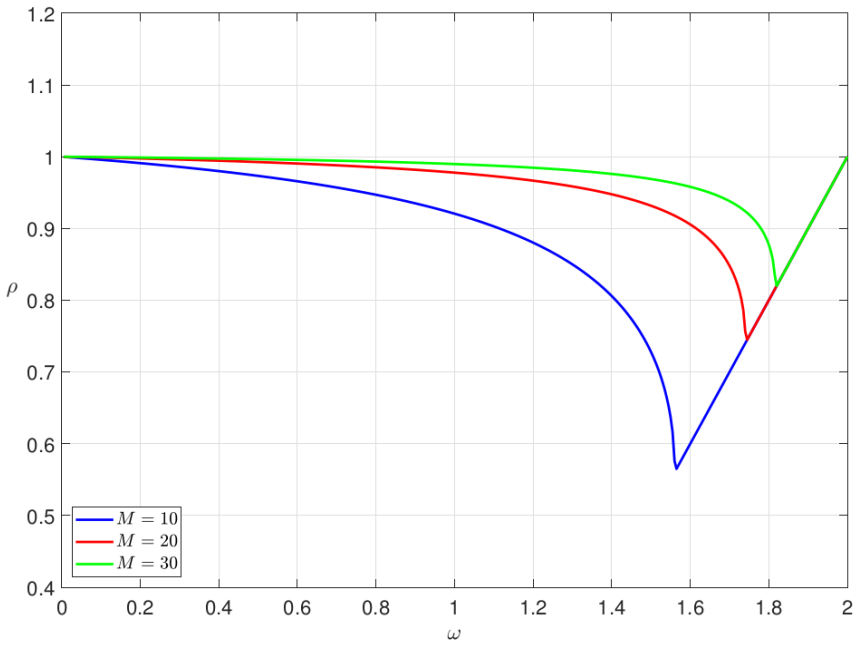


Abbildung 3: Spektralradius der Iterationsmatrix im Relaxationsverfahren für verschiedene Parameter $\omega \in [0, 2]$ bei Matrix (25) mit $M = 10, 20, 30$.

Ausblick:

Es gibt neben den stationären Verfahren noch weitere fortgeschrittene Iterationsmethoden für große lineare Gleichungssysteme.

- *Verfahren der Konjugierten Gradienten* (*CG-Verfahren*)
  nur für symmetrische positiv definite Matrizen; basiert auf sukzessiver Minimierung des Fehlers entlang von eindimensionalen Suchrichtungen; Verallgemeinerungen auf allgemeine Matrizen und nichtlineare Gleichungssysteme existieren.
- *Generalised Minimal Residual method* (*GMRES*)
  für allgemeine Matrizen; basiert auf Minimierung des Residuums; Rechenaufwand steigt in der Iteration jedoch mit der Schrittzahl an.
- *Mehrgitterverfahren*
  bei Matrizen die aus Diskretisierungen von partiellen Differentialgleichungen entstehen; Wechsel zwischen groben und feinen Gittern findet in der Iteration statt.

5 Verfahren der Konjugierten Gradienten

Gegeben sei ein lineares Gleichungssystem $Ax = b$ mit einer symmetrischen, positiv definiten Matrix $A \in \mathbb{R}^{n\times n}$. Dadurch existiert auch die inverse Matrix A^{-1} und ist symmetrisch sowie positiv definit. Wir definieren ein Funktional $F : \mathbb{R}^n \to \mathbb{R}$ durch

$$\begin{aligned} F(z) &:= \tfrac{1}{2}(Az-b)^\top A^{-1}(Az-b) \\ &= \tfrac{1}{2}z^\top Az - b^\top z + \tfrac{1}{2}b^\top A^{-1}b. \end{aligned} \tag{26}$$

Dadurch liegen drei Eigenschaften vor:

- $F(z) \geq 0$ für alle z,
- $F(A^{-1}b) = 0$,
- Für $z \neq A^{-1}b$ gilt $Az \neq b$ und somit $F(z) > 0$.

Aus diesen Eigenschaften folgt sofort, dass $\hat{x} := A^{-1}b$ das eindeutige Minimum von F mit $F(\hat{x}) = 0$ ist. Die Idee besteht nun darin, ein Iterationsverfahren zur Minimierung des Funktionals F einzusetzen, um die Lösung $\hat{x}$ des linearen Gleichungssystems näherungsweise zu erhalten.

Methode des steilsten Abstiegs

Die Methode des steilsten Abstiegs wird auch Gradientenverfahren genannt. Der Gradient des Funktionals (26) als Spaltenvektor geschrieben lautet

$$\nabla F(z) = Az - b \in \mathbb{R}^n.$$

Die Richtung des steilsten Abstiegs in einem Punkt ist gerade der negative Gradient, d.h.

$$-\nabla F(z) = b - Az.$$

Sei eine Näherung $x^k \in \mathbb{R}^n$ gegeben. Das zugehörige Residuum ist dann $r_k := b - Ax^k$. Wir definieren die Funktion $\tilde{F} : \mathbb{R} \to \mathbb{R}$ durch

$$\tilde{F}(\mu) := F(x^k + \mu r_k).$$

Der Parameter $\mu \in \mathbb{R}$ wird jetzt als Lösung des eindimensionalen Minimierungsproblems

$$\min_{\mu} F(x^k + \mu r_k)$$

bestimmt. Dadurch lautet die neue Näherung $x^{k+1} = x^k + \mu_{\min} r_k$ mit diesem optimalen Parameter $\mu_{\min}$. Das entstehende Iterationsverfahren konvergiert im Fall einer symmetrischen, positiv definiten Matrix A gegen das Minimum $\hat{x} = A^{-1}b$. Jedoch ist die Konvergenzordnung nur linear und die Konvergenzgeschwindigkeit oft sehr langsam. Ein Grund hierfür ist, dass die sukzessiven Suchrichtungen orthogonal aufeinander stehen, d.h. $r_{k+1}^\top r_k = 0$ für alle k. Typischerweise gilt in dieser Iteration $x^k \neq \hat{x} = A^{-1}b$ für alle k.

Definition des Verfahrens der konjugierten Gradienten

Das Iterationsverfahren der konjugierten Gradienten (engl. conjugate gradient (CG) method) wird wie folgt rekursiv definiert.

Algorithmus 1 (CG-Verfahren)

Wähle $x^0 \in \mathbb{R}^n$.
Setze $p_0 := r_0 := b - Ax^0$.

Für $k = 0, 1, 2, \ldots$

1.) Falls $p_k = 0$: ENDE
x^k ist Lösung von $Ax = b$.

2.) Berechne

$$\begin{aligned}
\alpha_k &= \frac{{r_k}^\top r_k}{{p_k}^\top A p_k} \\
x^{k+1} &= x^k + \alpha_k p_k \\
r_{k+1} &= r_k - \alpha_k A p_k \\
\beta_k &= \frac{{r_{k+1}}^\top r_{k+1}}{{r_k}^\top r_k} \\
p_{k+1} &= r_{k+1} + \beta_k p_k.
\end{aligned}$$

Der Rechenaufwand pro Iterationsschritt besteht aus:

1. Eine Matrix-Vektor-Multiplikation: Ap_k,
2. Zwei Skalarprodukte: $p_k^\top(Ap_k)$, $r_{k+1}^\top r_{k+1}$
   ($r_k^\top r_k$ liegt aus vorhergehenden Schritt vor),
3. Drei Vektoradditionen: $x^k + \alpha_k p_k$, $r_k - \alpha_k(Ap_k)$, $r_{k+1} + \beta_k p_k$.

Die Matrix-Vektor-Multiplikation ist dabei der wesentliche Anteil des Rechenaufwands.

Eigenschaften des CG-Verfahrens

Wir definieren zunächst den folgenden Begriff.

Def.: *Sei $A \in \mathbb{R}^{n\times n}$ symmetrisch und positiv definit. Zwei Richtungen p, q heißen A-konjugiert, wenn $\langle p, q\rangle_A := p^\top Aq = 0$ gilt.*

Sind zwei Richtungen A-konjugiert, so bedeutet dies, dass sie orthogonal bezüglich des von A induzierten Skalarprodukts sind.

Satz 5 (Eigenschaften des CG-Verfahrens)
Sei $A \in \mathbb{R}^{n\times n}$ symmetrisch und positiv definit sowie $b \in \mathbb{R}^n$. Zu jedem Startvektor $x^0 \in \mathbb{R}^n$ gibt es eine kleinste ganze Zahl ℓ mit $0 \le \ell \le n$, so dass $p_\ell = 0$. Die Vektoren x^k, p_k, r_k für $k \le \ell$ aus dem CG-Verfahren besitzen die folgenden Eigenschaften:

a) $Ax^\ell = b$,

b) $r_i^\top p_j = 0 \quad$ *für* $0 \le j < i \le \ell$,

c) $r_i^\top p_i = r_i^\top r_i \quad$ *für* $i \le \ell$,

d) $p_i^\top Ap_j = 0 \quad$ *für* $0 \le i < j \le \ell$, $\qquad p_j^\top Ap_j > 0 \quad$ *für* $j < \ell$,

e) $r_i^\top r_j = 0 \quad$ *für* $0 \le i < j < \ell$, $\qquad r_j^\top r_j > 0 \quad$ *für* $j < \ell$,

f) $r_i = b - Ax^i \quad$ *für* $i \le \ell$.

Beweis: siehe J. Stoer, R. Bulirsch: Numerische Mathematik 2, Springer 2005, Satz 8.7.1.4.

Folgerungen:

i) Nach spätestens n Iterationsschritten liefert das Verfahren bei exakter Rechnung auch die exakte Lösung $x_\ell = A^{-1}b$, siehe (a). Abgesehen von Spezialfällen für den Startwert werden im allgemeinen genau n Schritte benötigt.

ii) Die Vektoren $r_i = b - Ax^i$, siehe (f), sind die Residuen der Näherungen. Wegen (e) sind die Residuen für $i \neq j$ orthogonal zueinander.

iii) Die Suchrichtungen p_i sind A-konjugiert, siehe (d). Dies begründet den Namen des Verfahrens.

iv) Das CG-Vefahren ist wohldefiniert wegen $p_j^\top A p_j \neq 0$, siehe (d), und $r_j^\top r_j \neq 0$, siehe (e).

Trotz der Eigenschaft (i) besteht bei großen linearen Gleichungssystemen, d.h. hohes n, der Wunsch, dass das Verfahren nach nur relativ wenigen Iterationsschritten $k \ll n$ eine hinreichend genaue Näherung liefert. Für kleine lineare Gleichungssysteme ist wegen der Eigenschaft (i) eine direkte Lösung mit dem CG-Verfahren praktisch möglich. Ein Schritt des CG-Verfahrens wird dominiert vom Aufwand der Matrix-Vektor-Multiplikation. Bei vollbesetzter Matrix sind dazu n^2 Operationen notwendig (der typische Rechenschritt $a + b \cdot c$ zählt dabei als eine Operation). Bei Vergleich mit der Cholesky-Zerlegung zeigen sich jedoch folgende Verhältnisse.

Rechenaufwand:	Cholesky-Algorithmus :	ca. $\frac{1}{6}n^3$ Operationen
	CG-Verfahren :	ca. $n \cdot n^2 = n^3$ Operationen

Somit ist das Cholesky-Verfahren zur direkten Lösung besser geeignet.

Minimalitätseigenschaft

Wir zeigen, dass die Näherung x^{k+1} aus dem CG-Verfahren eine Lösung des Minimierungsproblems

$$\min_{\mu_0,\mu_1,\ldots,\mu_k} F(x^k + \mu_0 r_0 + \mu_1 r_1 + \cdots + \mu_k r_k)$$

mit $r_i := b - Ax^i$ und dem Funktional (26) darstellt.

Es gilt

$$S_k := \operatorname{span}\{r_0, \ldots, r_k\} = \operatorname{span}\{p_0, \ldots, p_k\} \qquad \text{für } k < \ell.$$

Dies folgt induktiv aus $p_0 = r_0$ und $p_k = r_k + \beta_k p_{k-1}$ bzw. $r_k = p_k - \beta_k p_{k-1}$. Wir definieren eine Hilfsfunktion

$$\Phi(\mu_0, \mu_1, \ldots, \mu_k) := F(x^k + \mu_0 p_0 + \mu_1 p_1 + \cdots + \mu_k p_k).$$

Differentiation ergibt mit der Kettenregel

$$\frac{\partial \Phi}{\partial \mu_j}(\mu_0, \ldots, \mu_k) = \nabla F(x^k + \mu_0 p_0 + \cdots + \mu_k p_k)^\top p_j \qquad \text{für } 0 \le j \le k.$$

Sei $x = x^k + \mu_0 p_0 + \cdots + \mu_k p_k$ im weiteren. Dadurch zeigt sich

$$\frac{\partial \Phi}{\partial \mu_j} = (Ax - b)^\top p_j \qquad \text{für } 0 \le j \le k.$$

Für $\mu_0 = \cdots = \mu_{k-1} = 0$ und $\mu_k = \alpha_k$ folgt $x^{k+1} = x^k + \alpha_k p_k$. Wir erhalten

$$\begin{aligned} \frac{\partial \Phi}{\partial \mu_j} &= (Ax^k + \alpha_k A p_k - b)^\top p_j = -r_k^\top p_j + \alpha_k p_k^\top A p_j \\ &= \begin{cases} -0 + 0 = 0 & \text{für } j < k, \\ -r_k^\top r_k + \frac{r_k^\top r_k}{p_k^\top A p_k} p_k^\top A p_k = 0 & \text{für } j = k. \end{cases} \end{aligned}$$

Somit ist x^{k+1} ein kritischer Punkt von Φ. Es gilt $F(z) = z^\top A z - b^\top z + c$ mit einer Konstanten $c \in \mathbb{R}$. Für die Hilfsfunktion ergibt sich

$$\begin{aligned} \Phi(\mu_0, \ldots, \mu_k) &= \tfrac{1}{2} {x^k}^\top A x^k + \tfrac{1}{2} \sum_{i,j=0}^{k} \mu_i \mu_j p_i^\top A p_j + \sum_{j=0}^{k} \mu_j p_j^\top A x^k \\ &\quad + \sum_{j=0}^{k} \mu_j b^\top p_j + b^\top x^k + c. \end{aligned}$$

Die Doppelsumme beschreibt den quadratischen Term, während die anderen Terme linear oder konstant in $\mu := (\mu_0, \mu_1, \ldots, \mu_k)^\top$ sind. Wir erhalten

$$\sum_{i,j=0}^{k} \mu_i \mu_j p_i^\top A p_j = \mu^\top A' \mu$$

mit $A' = (a'_{ij}) \in \mathbb{R}^{(k+1)\times(k+1)}$ und $a'_{ij} := p_i^\top A p_j$. Wegen Satz 5 (d) ist A' eine Diagonalmatrix mit positiven Diagonalelementen. Insbesondere ist A' damit symmetrisch und positiv definit. Somit repräsentiert x^{k+1} das eindeutige globale Minimum der Funktion Φ.

Fehlerabschätzung

Wir definieren den Fehler der k-ten Näherung aus dem Iterationsverfahren als den Vektor

$$e_k := x^k - A^{-1}b \qquad \text{für } \; k = 0, 1, 2, \ldots \; .$$

Damit gilt die folgende Abschätzung.

Satz 6 (Fehlerschranke im CG-Verfahren)
Für die Fehler der Näherungen x^k aus dem CG-Verfahren gilt bezüglich der Energienorm $\|\cdot\|_A$ die obere Schranke

$$\frac{\|e_k\|_A}{\|e_0\|_A} \leq 2 \left(\frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1} \right)^k$$

mit der Konditionszahl $\kappa_2(A)$ zur Euklidischen Vektornorm.

Beweis: siehe J. Stoer, R. Bulirsch: Numerische Mathematik 2, Springer 2005, S. 315-316.

Für eine symmetrische, positiv definite Matrix A gilt

$$\kappa_2(A) = \frac{\lambda_{\max}}{\lambda_{\min}}$$

mit dem größten und kleinsten Eigenwert. Wir führen noch die Wurzel einer Matrix als Hilfsmittel ein. Zu einer symmetrischen, positiv definiten Matrix A existiert die Zerlegung

$$A = SDS^\top \quad \text{mit} \quad D = \operatorname{diag}(\lambda_1, \ldots, \lambda_n) \quad \text{und} \quad S^{-1} = S^\top.$$

Es sei $D^{\frac{1}{2}} := \mathrm{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_n})$. Damit kann die Wurzel der Matrix A definiert werden als

$$A^{\frac{1}{2}} := SD^{\frac{1}{2}}S^\top.$$

Es folgt $(A^{\frac{1}{2}})^2 = A$. Zudem ist auch $A^{\frac{1}{2}}$ symmetrisch und positiv definit.

Lemma 1 *Für die Fehler der Näherungen x^k aus dem CG-Verfahren gilt bezüglich der Euklidischen Norm $\|\cdot\|_2$ die obere Schranke*

$$\frac{\|e_k\|_2}{\|e_0\|_2} \leq 2\sqrt{\kappa_2(A)} \left(\frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1} \right)^k.$$

Beweis:

Wir erhalten

$$\|A^{\frac{1}{2}}x\|_2^2 = (A^{\frac{1}{2}}x)^\top (A^{\frac{1}{2}}x) = x^\top (A^{\frac{1}{2}})^\top A^{\frac{1}{2}}x = x^\top A^{\frac{1}{2}}A^{\frac{1}{2}}x = x^\top Ax = \|x\|_A^2,$$

d.h. $\|A^{\frac{1}{2}}x\|_2 = \|x\|_A$ für beliegiges $x \in \mathbb{R}^n$. Damit schätzen wir ab

$$\|x\|_2 = \|A^{-\frac{1}{2}}A^{\frac{1}{2}}x\|_2 \leq \|A^{-\frac{1}{2}}\|_2 \|A^{\frac{1}{2}}x\|_2 = \|A^{-\frac{1}{2}}\|_2 \|x\|_A,$$

d.h. $\|x\|_2 \leq \|A^{-\frac{1}{2}}\|_2 \|x\|_A$ für beliegiges $x \in \mathbb{R}^n$. Die Konditionszahl erfüllt die Gleichung $\kappa_2(A^{\frac{1}{2}}) = \|A^{\frac{1}{2}}\|_2 \|A^{-\frac{1}{2}}\|_2$. Es folgt mit Satz 6

$$\begin{aligned} \|e_k\|_2 &\leq \|A^{-\frac{1}{2}}\|_2 \|e_k\|_A \leq \|A^{-\frac{1}{2}}\|_2 2 \left(\tfrac{\sqrt{\kappa_2(A)}-1}{\sqrt{\kappa_2(A)}+1} \right)^k \|e_0\|_A \\ &\leq \|A^{-\frac{1}{2}}\|_2 2 \left(\tfrac{\sqrt{\kappa_2(A)}-1}{\sqrt{\kappa_2(A)}+1} \right)^k \|A^{\frac{1}{2}}\|_2 \|e_0\|_2 = 2\kappa_2(A^{\frac{1}{2}}) \left(\tfrac{\sqrt{\kappa_2(A)}-1}{\sqrt{\kappa_2(A)}+1} \right)^k \|e_0\|_2. \end{aligned}$$

Für die Konditionszahl erhalten wir $\kappa_2(A^{\frac{1}{2}}) = \sqrt{\kappa_2(A)}$, da die Eigenwerte von $A^{\frac{1}{2}}$ gerade die Wurzeln der Eigenwerte von A sind. □

Die Abschätzung in Lemma 1 ist jedoch grob im Vergleich zu Satz 6.

Eine wichtige Folgerung aus Satz 6 ist, dass die Konvergenzgeschwindigkeit umso langsamer ist, je höher die Konditionszahl der Matrix ausfällt. Für hohe Konditionszahl ist nämlich

$$\frac{\sqrt{\kappa_2(A)}-1}{\sqrt{\kappa_2(A)}+1} \approx 1,$$

wodurch die k-ten Potenzen nur sehr langsam abnehmen.

Vorkonditionierungstechnik

Ein Konvergenzbeschleunigung soll nun erhalten werden, indem statt des ursprünglichen Gleichungssystems ein äquivalentes Gleichungssystem mit einer Matrix von deutlich niedrigerer Konditionszahl iterativ gelöst wird. Als Ansatz wird $Ax = b$ ersetzt durch

$$A'x' = b' \qquad \text{mit} \qquad A' = B^{-\frac{1}{2}}AB^{-\frac{1}{2}},\ \ x' = B^{\frac{1}{2}}x,\ \ b' = B^{-\frac{1}{2}}b$$

mit einer symmetrischen, positv definiten Matrix $B \in \mathbb{R}^{n\times n}$. Damit gilt

$$A' = B^{-\frac{1}{2}}AB^{-\frac{1}{2}} = B^{\frac{1}{2}}(B^{-1}A)B^{-\frac{1}{2}},$$

wodurch die Eigenwerte von A' und $B^{-1}A$ identisch sind. Das Ziel ist nun

$$\kappa_2(A') \ll \kappa_2(A)$$

zu erreichen. Dabei gilt als untere Schranke $\kappa_2(A') \geq 1$. Die Matrix B soll eine gute Approximation von A sein, denn dann folgt $B^{-1}A \approx I$ mit der Einheitsmatrix, wodurch die Konditionszahl von A' nahe eins wäre. In der Praxis möchte man das Aufstellen der Matrix $B^{-\frac{1}{2}}$ vermeiden, weil dazu die Eigenwerte und Eigenvektoren von B bestimmt werden müssten. Stattdessen sind dann lineare Gleichungssysteme mit der Matrix B zu lösen. Diese Gleichungssysteme sollen daher leicht direkt lösbar sein.

Konstruktionsmöglichkeiten für die Matrix B sind:

1. Aus stationären Iterationsverfahren (Zerlegung $A = D + L + L^\top$):
   symmetrisches Gauß-Seidel-Verfahren: $B = (D + L)D^{-1}(D + L)^\top$
   symmetrisches SOR-Verfahren: $B(\omega) = \frac{\omega}{2-\omega}(\frac{1}{\omega}D + L)D^{-1}(\frac{1}{\omega}D + L)^\top$
2. Unvollständige Cholesky-Zerlegung:
   $B = \tilde{L}\tilde{L}^\top$, wobei $\tilde{L}$ eine Approximation des exakten Faktors $\hat{L}$ aus der Zerlegung $A = \hat{L}\hat{L}^\top$ ist. Diese Approximation kann erhalten werden, indem der Cholesky-Algorithmus nur teilweise durchgeführt wird.

Eine Näherung $x^{k'}$ aus dem CG-Verfahren für das System $A'x' = b'$ muss auf eine Näherung x^k für das System $Ax = b$ rücktransformiert werden

über $x^{k'} = B^{\frac{1}{2}}x^k$ bzw. $x^k = B^{-\frac{1}{2}}x^{k'}$. Die Suchrichtungen p_k werden zur Änderung der Näherungen eingesetzt und transformieren sich daher auch über $p'_k = B^{\frac{1}{2}}p_k$. Für die Residuen gilt

$$r'_k = b' - A'x^{k'} = B^{-\frac{1}{2}}b - (B^{-\frac{1}{2}}AB^{-\frac{1}{2}})(B^{\frac{1}{2}}x^k) = B^{-\frac{1}{2}}(b - Ax^k) = B^{-\frac{1}{2}}r_k.$$

Das CG-Verfahren wird jetzt auf das Gleichungssystem $A'x' = b'$ angewendet. Wir schreiben einfach die Formeln gemäß Algorithmus 1 auf

$$\alpha'_k = \frac{{r'_k}^\top r'_k}{{p'_k}^\top A'p'_k}, \qquad x^{k+1'} = x^{k'} + \alpha'_k p'_k, \qquad r'_{k+1} = r'_k - \alpha'_k A'p'_k,$$

$$\beta'_k = \frac{{r'_{k+1}}^\top r'_{k+1}}{{r'_k}^\top r'_k}, \qquad p'_{k+1} = r'_{k+1} + \beta'_k p'_k.$$

Es gelten die Äquivalenzen

$$\begin{aligned} x^{k+1'} &= x^{k'} + \alpha'_k p'_k \quad \Leftrightarrow \quad x^{k+1} &= x^k + \alpha'_k p_k, \\ r'_{k+1} &= r'_k - \alpha'_k A'p'_k \quad \Leftrightarrow \quad r_{k+1} &= r_k - \alpha'_k Ap_k. \end{aligned}$$

Für die Euklidischen Normen der Residuen folgt

$${r'_k}^\top r'_k = (B^{-\frac{1}{2}}r_k)^\top (B^{-\frac{1}{2}}r_k) = r_k^\top (B^{-\frac{1}{2}})^\top B^{-\frac{1}{2}}r_k = r_k^\top B^{-\frac{1}{2}}B^{-\frac{1}{2}}r_k = r_k^\top B^{-1}r_k.$$

Desweiteren ist

$${p'_k}^\top A'p'_k = (B^{\frac{1}{2}}p_k)^\top (B^{-\frac{1}{2}}AB^{-\frac{1}{2}})(B^{\frac{1}{2}}p_k) = p_k^\top Ap_k$$

sowie

$$p_{k+1} = B^{-\frac{1}{2}}p'_{k+1} = B^{-\frac{1}{2}}(r'_{k+1} + \beta'_k p'_k) = B^{-1}r_{k+1} + \beta'_k p_k.$$

Die obigen Formeln zum CG-Algorithmus schreiben wir damit um zu

$$\alpha'_k = \frac{r_k^\top B^{-1}r_k}{p_k^\top Ap_k}, \qquad x^{k+1} = x^k + \alpha'_k p_k, \qquad r_{k+1} = r_k - \alpha'_k Ap_k,$$

$$\beta'_k = \frac{r_{k+1}^\top B^{-1}r_{k+1}}{r_k^\top B^{-1}r_k}, \qquad p_{k+1} = B^{-1}r_{k+1} + \beta'_k p_k.$$

Bezüglich der Matrix B werden somit nur die Vektoren $q_k := B^{-1}r_k$ in der Iteration benötigt. Diese erhält man aus der Lösung des linearen Gleichungssystems $Bq_k = r_k$.

Mit einem Notationswechsel $\alpha_k := \alpha'_k$ und $\beta_k := \beta'_k$ lautet der Algorithmus des Verfahrens der konjugierten Gradienten mit Vorkonditionierungstechnik (engl. preconditioned conjugate gradient (PCG) method) wie folgt.

Algorithmus 2 (PCG-Verfahren)

Wähle symmetrische, positiv definite Matrix $B \in \mathbb{R}^{n \times n}$.
Wähle $x^0 \in \mathbb{R}^n$.
Setze $p_0 := r_0 := b - Ax^0$.
Löse LGS $Bq_0 = r_0$.

Für $k = 0, 1, 2, \ldots$

1.) Falls $p_k = 0$: ENDE
x^k ist Lösung von $Ax = b$.

2.) Berechne

$$\begin{aligned} \alpha_k &= \frac{{r_k}^\top q_k}{{p_k}^\top A p_k} \\ x^{k+1} &= x^k + \alpha_k p_k \\ r_{k+1} &= r_k - \alpha_k A p_k \\ B q_{k+1} &= r_{k+1} \\ \beta_k &= \frac{{r_{k+1}}^\top q_{k+1}}{{r_k}^\top q_k} \\ p_{k+1} &= q_{k+1} + \beta_k p_k. \end{aligned}$$

Der Rechenaufwand pro Iterationsschritt des PCG-Verfahrens kennzeichnet sich durch:

1. Eine Matrix-Vektor-Multiplikation: Ap_k,
2. Ein lineares Gleichungssystem: $Bq_{k+1} = r_{k+1}$
   ($q_k = B^{-1} r_k$ liegt aus vorhergehenden Schritt vor),
3. Skalarprodukte und Vektoradditionen wie im CG-Verfahren.

Wird für B beispielsweise das symmetrische Gauß-Seidel-Verfahren eingesetzt, so sind für die Lösung des Gleichungssystems im wesentlichen eine Vorwärts- und eine Rückwärtssustitution erforderlich. Der Aufwand dazu entspricht einer Matrix-Vektor-Multiplikation mit der Matrix A. Daher ist der Rechenaufwand in einem Iterationsschritt beim PCG-Verfahren etwa doppelt so hoch wie beim gewöhnlichen CG-Verfahren.

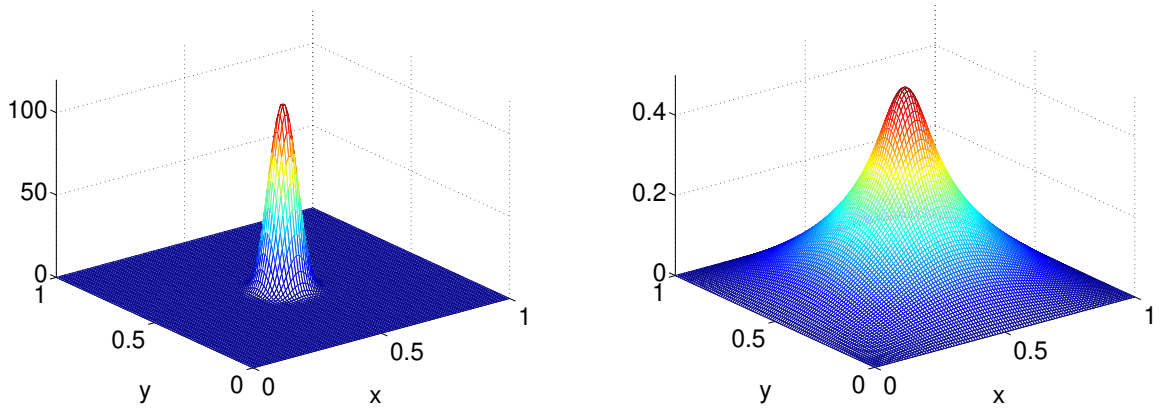


Abbildung 4: Rechte Seite f in Poisson Gleichung (links) und zugehörige numerische Lösung auf Gitter (rechts).

Vergleich der Verfahren für Beispiel

Wir greifen das Anwendungsbeispiel aus Abschnitt 4 auf, d.h. die Poisson-Gleichung. Homogene Dirichlet-Randbedingungen werden verwendet. Abbildung 4 (links) zeigt die gewählte rechte Seite f. Eine numerische Lösung wurde auf einem Gitter mit $M = 100$ iterativ berechnet, siehe Abbildung 4 (rechts). Dabei erfolgten 158 Schritte des SOR-Verfahrens mit optimalem Relaxationsparameter zu Startwert $x^0 = 0$.

Nun Vergleichen wir die eingeführten Verfahren bezüglich ihrer Effizienz, d.h. der Fehler der Näherungen gegenüber dem Rechenaufwand. Wir verwenden die folgenden Verfahren:

1. Jacobi-Verfahren,
2. Gauß-Seidel-Verfahren,
3. SOR-Verfahren mit optimalem Relaxationsparameter,
4. CG-Verfahren,
5. PCG-Verfahren mit symm. Gauß-Seidel-Vorkonditionierung,
6. PCG-Verfahren mit symm. SOR-Vorkonditionierung
   bei optimalem Relaxationsparameter.

Als Startwert wird immer $x^0 = 0$ eingesetzt.

Wir berechnen zwei Fälle:

	M	$n = M^2$	$\kappa_2(A)$
1. Fall:	100	10 000	$6.0 \cdot 10^3$
2. Fall:	1 000	1 000 000	$5.9 \cdot 10^5$

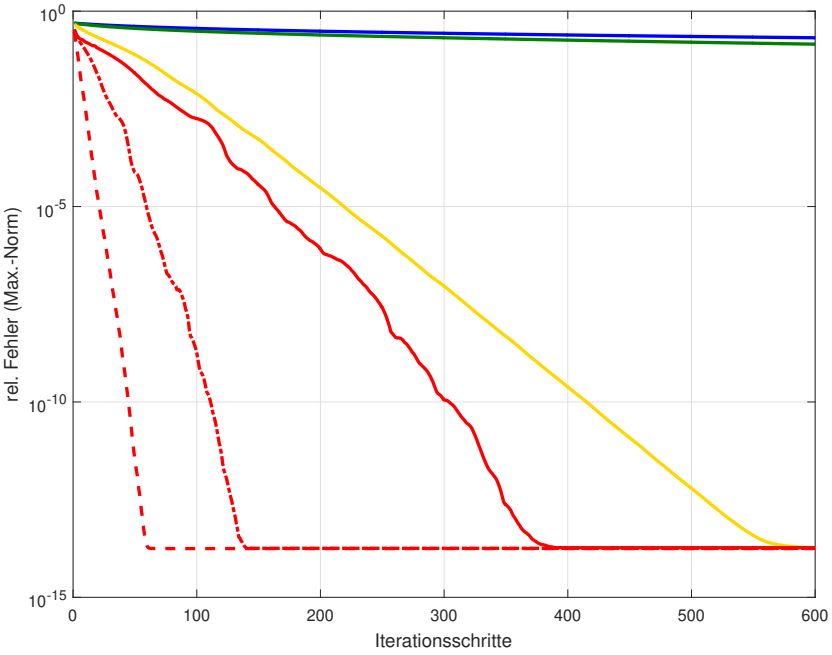
Der relative Fehler

$$\frac{\|x^k - \hat{x}\|_\infty}{\|\hat{x}\|_\infty}$$

der Näherungen x^k wird in der Maximum-Norm berechnet. Als Referenzlösung $\hat{x} \approx A^{-1}b$ dient dabei die direkte Lösung des linearen Gleichungssystems mit dem Cholesky-Verfahren. Man beachte, dass diese Referenzlösung nicht ganz der exakten Lösung entspricht wegen Rundungsfehlern. Daher Konvergiert der Fehler nicht gegen null sondern nimmt nach einigen Schritten einen konstanten Wert an. Jedoch konvergieren die Verfahren auf dem Rechner auch nicht gegen die exakte Lösung wegen der Rundungsfehler. Abbildung 5 stellt diese bestimmten relativen Fehler für die sechs Methoden bzw. Varianten dar.

Beim Effizienzvergleich ist zu beachten, dass ein Iterationsschritt der PCG-Verfahren etwa doppelt soviel an Rechenaufwand kostet wie ein Iterationsschritt des CG-Verfahren. Trotzdem stellen wir einen deutlichen Gewinn durch die Vorkonditionierungstechnik fest.

$M = 100$



$M = 1\,000$

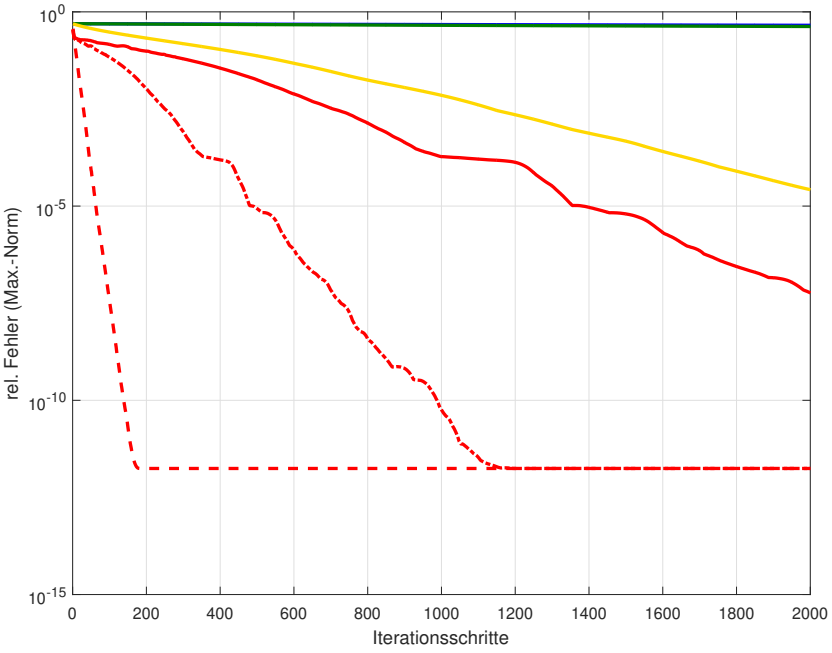


Abbildung 5: Maximaler relativer Fehler der Näherungen in verschiedenen Iterationsverfahren bezüglich der Schittanzahl: Jacobi-V. (blau), Gauß-Seidel-V. (grün), SOR-V. (gelb), CG-V. (rot,—), PCG-V. mit Gauß-Seidel (rot,-·-), PCG-V. mit SOR (rot,- - -).

6 GMRES-Verfahren

Wir betrachten jetzt ein lineares Gleichungssystem $Ax = b$ mit einer beliebigen regulären Matrix $A \in \mathbb{R}^{n\times n}$. Das Residuum einer Näherung $z \in \mathbb{R}^n$ der Lösung $x = A^{-1}b$ lautet $r(z) = b - Az$. Wir definieren die reellwertige Funktion $F : \mathbb{R}^n \to \mathbb{R}$ durch

$$F(z) = \tfrac{1}{2}\|b - Az\|_2^2 = \tfrac{1}{2}(b - Az)^\top (b - Az). \tag{27}$$

Offensichtlich gilt $F(z) \geq 0$ für alle z. Es ist $F(z) = 0$ genau dann, wenn $z = A^{-1}b$. Also ist die gesuchte Lösung x das eindeutige globale Minimum der Funktion (27). Die Idee des GMRES-Verfahrens (generalized minimal residual) besteht darin, ein Iterationsverfahren zur Minimierung der Funktion (27) einzusetzen. Man beachte jedoch, dass ein kleines Residuum nicht notwendigerweise eine gute Näherung der Lösung impliziert.

Def.: Zu einem Vektor $v \in \mathbb{R}^n$ und einer Matrix $A \in \mathbb{R}^{n\times n}$ lautet der k-te Krylov-Raum

$$\mathcal{K}_k(v, A) = \operatorname{span}\{v, Av, A^2v, A^3v, \ldots, A^{k-2}v, A^{k-1}v\}$$

für $k \geq 1$. Zudem sei $\mathcal{K}_0(v, A) = \{0\}$.

Die Krylov-Räume bilden eine Kette von Untervektorräumen in $\mathbb{R}^n$, d.h.

$$\mathcal{K}_0(v, A) \subseteq \mathcal{K}_1(v, A) \subseteq \mathcal{K}_2(v, A) \subseteq \cdots \subseteq \mathcal{K}_n(v, A).$$

Bezüglich der Lösung des linearen Gleichungssystems besitzen die Krylov-Räume eine günstige Eigenschaft.

Lemma 2 *Sei $A \in \mathbb{R}^{n\times n}$ regulär und $b, x^0 \in \mathbb{R}^n$. Sei $r_0 = b - Ax^0$. Ist k die kleinste ganze Zahl mit $\mathcal{K}_k(r_0, A) = \mathcal{K}_{k+1}(r_0, A)$, dann gilt $x = A^{-1}b \in x^0 + \mathcal{K}_k$.*

Beweis: siehe Ch. Kanzow: Numerik linearer Gleichungssysteme, Lemma 6.3.

Nimmt also die Dimension der Krylov-Räume nicht mehr weiter zu, dann befindet sich die exakte Lösung bereits in dem affinen Raum $x^0 + \mathcal{K}_k$.

Algorithmus 3 (GMRES)
Wähle Startwert $x^0 \in \mathbb{R}^n$.
Setze $r_0 := b - Ax^0$.

Für $k = 1, 2, 3, \ldots$

1.) Falls $r_{k-1} = 0$: ENDE
x^{k-1} ist Lösung von $Ax = b$.

2.) Bestimme $x^k \in \mathbb{R}^n$ aus dem Minimierungsproblem

$$\min \left\{ \|b - Az\|_2 \ : \ z \in x^0 + \mathcal{K}_k(r_0, A) \right\}.$$

Wegen Lemma 2 liefert die Iteration nach spätestens n Schritten die exakte Lösung. Unser Wunsch ist wieder, dass nach nur relativ wenigen Iterationsschritten eine gute Näherung erhalten wird.

Für eine effiziente Lösung der Minimierungsprobleme aus Algorithmus 3 ist eine sukzessive Berechnung erforderlich.

1. Konstruktion einer Orthonormalbasis $v_1, v_2, \ldots, v_k$ der Krylov-Räume $\mathcal{K}_k(r_0, A) = \text{span}\{r_0, Ar_0, \ldots, A^{k-1}r_0\}$ über den Algorithmus von Arnoldi (Gram-Schmidt-Orthogonalisierung):

   $\beta = \|r_0\|_2, \quad v_1 = \frac{1}{\beta} r_0$
   für $k = 1, 2, 3, \ldots$
   $u = Av_k$
   für $i = 1, 2, \ldots, k$
   $h_{ik} = v_i^\top u$
   $$w_k = u - \sum_{i=1}^{k} h_{ik} v_i, \qquad h_{k+1,k} = \|w_k\|_2, \qquad v_{k+1} = \frac{1}{h_{k+1,k}} w_k$$

   Im k-ten Schritt beträgt der Rechenaufwand: eine Matrix-Vektor-Multiplikation mit A, k Skalarprodukte und k Vektoradditionen. Werden insgesamt m Schritte ausgeführt, dann lautet der Gesamtaufwand für $k = 1, \ldots, m$: m Matrix-Vektor-Multiplikationen (Aufwand bei dünnbesetzten Matrizen oft proprotional mn) sowie $\frac{m(m+1)}{2}$ Skalarprodukte und Vektor-Additionen (damit ca. $m^2 n$ Operationen). Die Anzahl der

Rechenoperationen steigt proportional zu m^2, d.h. für hohe m entsteht extrem viel Aufwand.

Der Arnoldi-Algorithmus liefert eine Hessenberg-Matrix $\bar{H}_k \in \mathbb{R}^{(k+1)\times k}$ der Gestalt

$$\bar{H}_k = \begin{pmatrix} h_{11} & h_{12} & \cdots & h_{1,k-1} & h_{1k} \\ h_{21} & h_{22} & \cdots & h_{2,k-1} & h_{2k} \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & h_{k-1,k-1} & h_{k-1,k} \\ \vdots & & \ddots & h_{k,k-1} & h_{kk} \\ 0 & \cdots & \cdots & 0 & h_{k+1,k} \end{pmatrix}.$$

2. Minimierung im Krylov-Raum:

   Ist $V_k = (v_1, v_2, \ldots, v_k) \in \mathbb{R}^{n\times k}$, dann gilt $\mathcal{K}_k(r_0, A) = \{V_k y \ : \ y \in \mathbb{R}^k\}$. Jedes $x \in x^0 \in \mathcal{K}_k(r_0, A)$ hat die Darstellung $x = x^0 + V_k y$ mit einem geeigneten $y \in \mathbb{R}^k$. Es gilt

   $$AV_k = V_{k+1}\bar{H}_k$$

   wegen $Av_j = u = h_{j+1,j}v_{j+1} - h_{1j}v_1 - \cdots - h_{jj}v_j$. Für $x = x^0 + V_k y$ folgt

   $$\begin{aligned} \|b - Ax\|_2 &= \|b - Ax^0 - AV_k y\|_2 \\ &= \|r_0 - V_{k+1}\bar{H}_k y\|_2 \\ &= \|V_{k+1}(\beta e_1 - \bar{H}_k y)\|_2 \\ &= \|\beta e_1 - \bar{H}_k y\|_2 \end{aligned}$$

   mit $e_1 = (1, 0, \ldots, 0)^\top$, weil $V_{k+1}^\top V_{k+1} = I$ durch die Orthogonalität. Nun wird y^k als Lösung des linearen Ausgleichsproblems

   $$\min_{y\in\mathbb{R}^k} \|\beta e_1 - \bar{H}_k y\|_2$$

   bestimmt. Dazu wird $\bar{H}_k$ mit orthogonalen Transformationen in eine obere Dreiecksmatrix überführt. Givens-Rotationen sind bei dieser Ge-

stalt der Matrix einzusetzen. Die Matrix besitzt die Aufteilung

$$\bar{H}'_k = \left(\begin{array}{ccc|c} & \bar{H}'_{k-1} & & * \\ & & & \vdots \\ & & & * \\ \hline 0 & \cdots & 0 & * \end{array}\right),$$

wobei $\bar{H}'_{k-1}$ bereits in den vorhergehenden Schritten transformiert wurde. Die letzte Spalte von $\bar{H}'_k$ muss noch transformiert werden durch: (i) $(k-1)$ vorhergehende Rotationen auf die Zeilen $2, 3, \ldots, k$ anwenden, (ii) eine neue Rotation für das Element in der letzten Zeile. Insgesamt sind also k Rotationen für die Spalte erforderlich, wodurch der Rechenaufwand für $k = 1, \ldots, m$ Schritte proportional zu m^2 ist (jedoch unabhängig von n). Analog müssen die Givens-Rotationen auch auf den Vektor βe_1 angewendet werden. Es folgt das transformierte Ausgleichsproblem

$$\min_{y \in \mathbb{R}^k} \left\| \begin{pmatrix} g_k \\ \gamma_k \end{pmatrix} - \begin{pmatrix} R_k \\ 0 \end{pmatrix} y \right\|_2$$

mit $g_k \in \mathbb{R}^k$, $\gamma_k \in \mathbb{R}$ und $R_k \in \mathbb{R}^{k \times k}$. Zudem gilt $\gamma_k = \|b - Ax^k\|_2$. Als Residuum-basiertes Abbruchkriterium kann daher $\frac{\gamma_k}{\|b\|_2} \leq \varepsilon$ mit einer Genauigkeitsforderung $\varepsilon > 0$ verwendet werden. Die Berechnung von y^k erfolgt aus dem linearen Gleichungssystem $R_k y^k = g_k$ mit Rückwärtssubstitution erst, wenn das Abbruchkriterium erfüllt ist. Der Aufwand ist dann einmalig proportional zu k^2 bzw. m^2 (unabhängig von n). Schließlich geben wir $x^k = x^0 + V_k y^k$ aus.

Konvergenzaussagen

Zu einer beliebigen Matrix $A \in \mathbb{R}^{n \times n}$ bezeichne $\sigma(A) = \{\lambda_1, \ldots, \lambda_n\} \subset \mathbb{C}$ das Spektrum von A. Ist A regulär, dann gilt $0 \notin \sigma(A)$. Desweiteren sei $\mathcal{P}_k$ die Menge aller Polynome über $\mathbb{C}$ vom Grad kleinergleich k

Satz 7 *Sei $A \in \mathbb{R}^{n\times n}$ regulär und diagonalisierbar, d.h. $A = SDS^{-1}$ mit $S \in \mathbb{C}^{n\times n}$ und Diagonalmatrix $D \in \mathbb{C}^{n\times n}$ mit Eigenwerten $\lambda_1, \ldots, \lambda_n \in \mathbb{C}$. Ist $p \in \mathcal{P}_k$ mit $p(0) = 1$, dann gilt für das Residuum im k-ten Schritt des GMRES-Verfahrens*

$$\frac{\|r_k\|_2}{\|r_0\|_2} \leq \kappa_2(S) \max_{z\in\sigma(A)} |p(z)| \tag{28}$$

mit der Konditionszahl κ_2 bezüglich der Euklidischen Norm.

Beweis:

Im folgenden bezeichnet $\|\cdot\|$ die Euklidische Vektornorm oder die Spektral-(Matrix-)Norm.

i) Wir setzen $q(z) = \frac{1-p(z)}{z}$. Dann gilt $q \in \mathcal{P}_{k-1}$. Somit ist $\bar{x} := q(A)r_0 \in \mathcal{K}_k(r_0, A)$. Wegen $p(z) = 1 - zq(z)$ folgt

$$p(A)r_0 = (I - Aq(A))r_0 = r_0 - Aq(A)r_0 = b - Ax^0 - A\bar{x} = b - A(x^0 + \bar{x}).$$

Da die k-te Näherung das Residuum im affinen Raum $x^0 + \mathcal{K}_k(r_0, A)$ minimiert, können wir abschätzen

$$\|r_k\| = \|b - Ax^k\| \leq \|b - A(x^0 + \bar{x})\| = \|p(A)r_0\| \leq \|p(A)\|\,\|r_0\|.$$

ii) Mit $A = SDS^{-1}$ berechnen wir

$$p(A) = \sum_{j=0}^{k} \alpha_j A^j = \sum_{j=0}^{k} \alpha_j (SDS^{-1})^j = \sum_{j=0}^{k} \alpha_j SD^jS^{-1} = S\Big(\sum_{j=0}^{k} \alpha_j D^j\Big)S^{-1} = S\,p(D)\,S^{-1}.$$

Es ist $p(D)$ eine Diagonalmatrix und insbesondere symmetrisch. Dadurch ist die Spektralnorm identisch dem Spektralradius, d.h. $\|p(D)\| = \rho(p(D))$. Wir erhalten

$$\|p(A)\| \leq \|S\|\,\|p(D)\|\,\|S^{-1}\| = \kappa_2(S)\,\rho(p(D)) = \kappa_2(S) \max_{j=1,\ldots,n} |p(\lambda_j)|,$$

da die Operationen in $p(D)$ im jedem Diagonaleintrag separat erfolgen. □

Da Satz 7 für jedes Polynom mit den geforderten Eigenschaften gilt, folgt

$$\|r_k\|_2 \leq \kappa_2(S) \left[\inf_{p\in\mathcal{P}_k, p(0)=1} \max_{z\in\sigma(A)} |p(z)|\right] \|r_0\|_2.$$

Jedoch ist dieses Infimum in der Praxis nicht berechenbar.

Satz 7 liefert auch die Aussage $r_n = 0$ (d.h. $Ax^n = b$), welche aus Lemma 2 folgt. Wir können nämlich $p(z) = \eta(z - \lambda_1)(z - \lambda_2) \cdots (z - \lambda_n)$ setzen. Der Vorfaktor $\eta \in \mathbb{C}$ wird dann so gewählt, dass $p(0) = 1$ gilt. Es folgt $p(z) = 0$ für alle $z \in \sigma(A)$. Also zeigt (28) $\|r_n\|_2 \leq 0$, somit $\|r_n\|_2 = 0$.

Nun sei $\sigma(A) \subset B_r(\bar{z})$ mit $B_r(\bar{z}) = \{z \in \mathbb{C} : |z - \bar{z}| \leq r\}$ mit $\bar{z} \in \mathbb{C}\backslash\{0\}$ sowie $0 < r < |\bar{z}|$. Wir setzen $p(z) = \left(\frac{\bar{z}-z}{\bar{z}}\right)^k$, wodurch $p \in \mathcal{P}_k$ und $p(0) = 1$ gilt. Mit einem Eigenwert λ_j von A folgt

$$|p(\lambda_j)| = \left|\frac{\bar{z} - \lambda_j}{\bar{z}}\right|^k \leq \left(\frac{r}{|\bar{z}|}\right)^k < 1$$

für $j = 1, \ldots, n$ und $k \geq 1$. Satz 7 zeigt dann

$$\|r_k\|_2 \leq \kappa_2(S) \left(\frac{r}{|\bar{z}|}\right)^k \|r_0\|_2 \tag{29}$$

für $k = 1, 2, \ldots$. Wir erkennen den Dämpfungsfaktor $\frac{r}{|\bar{z}|}$. Günstig ist somit $r \ll |\bar{z}|$, d.h. das Spektrum von A ist ein Cluster nahe eines Punkts $\bar{z}$, der wiederum relativ weit vom Nullpunkt in $\mathbb{C}$ entfernt ist.

Vorkonditionierung

Eine Vorkonditionierung des linearen Gleichungssystems $Ax = b$ erbringt häufig eine Konvergenzbeschleunigung im GMRES-Verfahren. Dazu wird das äquivalente Gleichungssystem

$$MAN(N^{-1}x) = Mb$$

angesetzt mit regulären Matrizen $M, N \in \mathbb{R}^{n \times n}$. Es folgt das Gleichungssystem $A'x' = b'$ mit $A' = MAN$, $x' = N^{-1}x$, $b' = Mb$. Lineare Gleichungssysteme mit Koeffizientenmatrizen M, N müssen dann leicht direkt lösbar sein.

Ziel ist $\kappa_2(A') \ll \kappa_2(A)$. Optimal wäre $A' \approx I$ mit der Einheitsmatrix I. Ist A' diagonalisierbar, so wäre $A' = S'D'S'^{-1} \approx I$. Die Eigenwerte bilden dann ein Cluster um $\bar{z} = 1$, was durch die Abschätzung (29) eine schnelle Abnahme der Größe der Residuen garantiert.

Vorteile von GMRES:

- Anwendbar bei jedem linearen Gleichungssystem mit regulärer Matrix.

Nachteile von GMRES:

- Rechenaufwand im k-ten Schritt ist proportional zu k. Für m Schritte dominiert der Term $m^2 n$ falls m hoch.
- Speicherbedarf im k-ten Schritt ist kn für die Orthonormalbasis.
- Es existiert keine Abschätzung für den Fehler der Näherungen. Eine Abschätzung liegt für das Residuum vor, welche aber in der Praxis nicht auswertbar ist.

GMRES mit Neustarts

Im Rechenaufwand des GMRES-Verfahrens mit insgesamt m Schritten dominiert der Term $m^2 n$. Hier kann der Aufwand reduziert werden, indem die Iteration nach je m Schritten mit einem kleinen $m \ll n$ neu gestartet wird. Es entsteht das GMRES(m)-Verfahren.

Algorithmus 4 (GMRES(m))
Wähle Startwert $x^0 \in \mathbb{R}^n$.

Für $\ell = 1, 2, 3, \ldots$

Führe m Schritte des GMRES-Verfahrens mit Startwert $x^{(\ell-1)m}$ aus.

In dieser Iteration entsteht die Folge

$$x^0, \underbrace{x^1, \ldots, x^m}_{\ell=1}, \underbrace{x^{m+1}, \ldots, x^{2m}}_{\ell=2}, \underbrace{x^{2m+1}, \ldots, x^{3m}}_{\ell=3}, \ldots$$

der Näherungen. Der Rechenaufwand bei $\ell_{\max}$ Schritten des GMRES(m)-Verfahrens ist nun dominiert durch den Term $\ell_{\max} m^2 n$ mit kleinem m und hohem n.

Jetzt gilt im Allgemeinen $x^k \neq A^{-1}b$ für alle k, selbst wenn $k \geq n$ vorliegt. Gewünscht ist wieder die Konvergenz

$$\lim_{k\to\infty} x^k = A^{-1}b.$$

Leider existiert keine Konvergenzaussage für allgemeine reguläre Matrix A. Konvergenzsätze können nur für Klassen von Matrizen mit bestimmten Eigenschaften erhalten werden:

- A unsymmetrisch, positiv definit (d.h. $x^\top Ax > 0$ für alle $x \neq 0$):
  GMRES(m) konvergiert für alle $m \geq 1$.
- A symmetrisch, indefinit:
  GMRES(m) konvergiert für alle $m \geq 2$.

Für symmetrische, positiv definite Matrizen wird man natürlich besser das CG-Verfahren aus Kapitel 5 verwenden.