

Iterative Lösung linearer Gleichungssysteme

Ergänzung zum Kapitel 2

Dies ist ein Ergänzungsartikel zum Buch *Lineare Algebra: Eine anwendungsorientierte Einführung*, von Andreas Müller. Erschienen im Verlag Springer-Vieweg, ISBN 978-3-662-67865-7 und ISBN 978-3-662-67866-4 (eBook).

<https://doi.org/10.1007/978-3-662-67866-4>

Die Rechte an den Bildern gehören den in der Bildunterschrift angegebenen Bildquellen, wenn keine Quelle angegeben ist dem Autor. Die Teile des Autors werden unter der Lizenz CC BY-SA 4.0 zur Verfügung gestellt, Details: <https://creativecommons.org/licenses/by-sa/4.0/>.

Website zum Buch: <https://linalg.ch>

Springer-Link: <https://link.springer.com/book/10.1007/978-3-662-67866-4>

Iterative Lösung linearer Gleichungssysteme

Andreas Müller

Zusammenfassung

In Kapitel 2 wurde der Gauß-Algorithmus als universelles Verfahren zur Lösung linearer Gleichungssysteme dargestellt. Allerdings wächst der Rechenaufwand für die Lösung eines $n \times n$ -Gleichungssystems mit $O(n^3)$. In diesem Artikel werden iterative Methoden zur Lösung vorgestellt, die versprechen, mit geringerem Rechenaufwand approximative Lösungen zu liefern. Damit wird es möglich, Genauigkeit gegen Laufzeit einzutauschen.

1 Jacobi-Verfahren

Das lineare Gleichungssystem $Ax = b$ mit $A \in M_n(\mathbb{R})$ und $b \in \mathbb{R}^n$ kann mit dem Gauß-Algorithmus von Kapitel 2 gelöst werden. Die Gleichung Nummer i des Gleichungssystems ist

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{i,i-1}x_{i-1} + a_{ii}x_i + a_{i,i+1}x_{i+1} + \dots + a_{i,n-1}x_{n-1} + a_{in}x_n = b_i. \quad (1)$$

Falls die x_k keine Lösung des Gleichungssystems sind, kann man auf jeden Fall die Gleichung (1) nach x_i auflösen und so die Werte modifizieren, so dass wenigstens die Gleichung Nummer i erfüllt ist. Das x_i , das dazu verwendet werden muss, ist

$$\begin{aligned} x_i &= \frac{1}{a_{ii}}(b_i - a_{i1}x_1 - a_{i2}x_2 - \dots - a_{i,i-1}x_{i-1} - a_{i,i+1}x_{i+1} - \dots - a_{i,n-1}x_{n-1} - a_{in}x_n) \\ &= \frac{1}{a_{ii}}\left(b_i - \sum_{\substack{k=1 \\ k \neq i}}^n a_{ik}x_k\right). \end{aligned} \quad (2)$$

Man kann diese Formel auch als Versuch interpretieren, für die Unbekannten neue Werte zu berechnen, die das Gleichungssystem möglicherweise "besser" erfüllen.

Beispiel 3.1. Ist A eine Diagonalmatrix und x der Nullvektor, dann liefert (2) die Werte

$$x_i = \frac{1}{a_{ii}}b_i,$$

dies ist die Lösung des Gleichungssystems. ○

Beispiel 3.2. Das Gleichungssystem

$$A_0x = \begin{pmatrix} 10 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 15 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 12 \\ 91 \\ 47 \end{pmatrix}$$

hat die Lösung

$$x_0 = A_0^{-1}b = \begin{pmatrix} \frac{10}{9} \\ \frac{97}{20} \\ \frac{47}{15} \end{pmatrix}.$$

Die Matrix A_0 wird jetzt zufällig gestört, so dass sich die Matrix

$$A = \begin{pmatrix} 9.9538 & 0.2053 & 0.4608 \\ -0.2655 & 19.8005 & 0.2701 \\ -0.1795 & 0.4551 & 15.4141 \end{pmatrix}$$

ergibt. Der Vektor x_0 ist nur noch eine approximative Lösung der Gleichung $Ax = b$. Durch wiederholte Anwendung der Gleichung (2) wird die Lösung jedoch immer besser:

i	x_1	x_2	x_3
0	1.200000000000	4.550000000000	3.133333333333
1	0.966670517792	4.569192023770	2.928792144854
2	0.975743681526	4.568853527141	2.925508337686
3	0.975902683284	4.569019981588	2.925623990413
4	0.975893896100	4.569020535976	2.925620927464
5	0.97590268461	4.569020459933	2.925620808768
6	0.975894033524	4.569020463300	2.925620812531
7	0.975894033281	4.569020463344	2.925620812514
8	0.975894033281	4.569020463341	2.925620812510
∞	0.975894033281	4.569020463341	2.925620812510

Die korrekte Lösung x_∞ ist nach acht Iterationen mit einer Genauigkeit von 12 Stellen erreicht. ○

Das Beispiel illustriert, dass für Gleichungssysteme, die außerhalb der Diagonalen nur kleine Einträge haben, durch die Formel (2) eine Verbesserung der Lösung erreicht werden kann. Dieses Verfahren ist als *Jacobi-Verfahren* bekannt.

2 Splitting-Verfahren

Im Abschnitt 1 wurde das Jacobi-Verfahren vorgestellt und in Beispielen gezeigt, wie es in gewissen Fällen ermöglichen kann, eine Lösung eines linearen Gleichungssystems zu finden. Die Übersicht darüber, für welche Gleichungssysteme dies funktionieren könnte, war etwas erschwert durch die schwerfällig Notation für die Iterationsformel (2). In diesem Abschnitt soll daher eine Matrixschreibweise gefunden werden, mit der sich später auch komplexere Verfahren analysieren lassen.

2.1 Aufspaltung der Matrix A

In der Formel (2) haben spielen die Diagonalelemente eine von allen anderen Elementen der Matrix verschiedene Rolle. Wir teilen die Matrix A daher in die drei Matrizen

$$L = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 \\ a_{21} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-1,1} & a_{n-1,2} & \dots & 0 & 0 \\ a_{n1} & a_{n2} & \dots & a_{n,n-1} & 0 \end{pmatrix},$$

$$D = \begin{pmatrix} a_{11} & 0 & \dots & 0 & 0 \\ 0 & a_{22} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & a_{n-1,n-1} & 0 \\ 0 & 0 & \dots & 0 & a_{nn} \end{pmatrix},$$

$$U = \begin{pmatrix} 0 & a_{12} & \dots & a_{1,n-1} & a_{1n} \\ 0 & 0 & \dots & a_{2,n-1} & a_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & a_{n-1,n} \\ 0 & 0 & \dots & 0 & 0 \end{pmatrix}.$$

Die Matrix L enthält also die Elemente von A unterhalb der Diagonalen, U enthält die Elemente von A oberhalb der Diagonalen und D ist die Diagonale von A . Die Matrix A kann jetzt als Summe $A = L + D + U$ geschrieben werden. Man beachte, dass die Matrizen L und U nichts zu tun haben mit der in Kapitel 12 besprochenen LU-Zerlegung einer Matrix.

2.2 Jacobi-Verfahren in Matrixform

Die Formel (2) ist entstanden dadurch, dass Gleichungssystem $Ax = b$ aufgelöst worden ist nach den diagonalen Termen aufgelöst worden ist. In Matrixschreibweise wird dies besonders einfach:

$$Ax = (L + D + U)x = b \quad \Rightarrow \quad Dx = b - Lx - Ux \quad \Rightarrow \quad x = D^{-1}(b - Lx - Ux). \quad (3)$$

Dies ist die Matrixform der Formel (2). Sie verdeutlicht, dass das Verfahren nur anwendbar ist, wenn die Matrix D invertierbar ist, wenn also alle Diagonalelemente von A von 0 verschieden sind.

2.3 Konvergenz des Jacobi-Verfahrens

Das Jacobi-Verfahren versucht die Lösung x^* des Gleichungssystems $Ax = b$ dadurch zu finden, dass aus einem Startvektor x_0 die Folge

$$x_{n+1} = D^{-1}(b - Lx_n - Ux_n)$$

berechnet wird. Wir versuchen zu bestimmen, unter welchen Voraussetzungen diese Folge konvergiert. Dazu schreiben wir die Differenz $\bar{x}_n = x_n - x^*$ und verwenden die Tatsache, dass

$$Lx_n + Dx_{n+1} + Ux_n = b \quad \text{und} \quad Ax^* = Lx^* + Dx^* + Ux^* = b.$$

Es folgt, dass

$$(L + U)(x_n - x^*) + D(x_{n+1} - x^*) = 0 \quad \Rightarrow \quad \bar{x}_{n+1} = D^{-1}(L + U)\bar{x}_n.$$

Konvergenz wird also dann eintreten, wenn die Matrix $D^{-1}(L + U)$ die Länge eines Vektors reduziert. Die in Definition 14.30 eingeführte Norm einer Matrix hat die Eigenschaft, dass für jeden Vektor x die Ungleichung $|Ax| \leq |x|$ gilt. Daraus ergibt sich der folgende Satz.

Satz 3.3. *Wenn $\|D^{-1}(L + U)\| < 1$ ist, dann ist das Jacobi-Verfahren konvergent.*

Die Konvergenz des Jacobi-Verfahrens ist linear: der Fehler wird in jedem Schritt mit dem gleichen Faktor $\|D^{-1}(L + U)\|$ multipliziert.

Noch etwas genauere Aussagen sind mit dem Spektralradius der Matrix $D^{-1}(L + U)$ möglich, der in Definition 14.32 eingeführt worden ist. Er lässt die folgende Aussage zu:

Satz 3.4. *Das Jacobi-Verfahren ist genau dann konvergent, wenn der Spektralradius*

$$\rho(D^{-1}(L + U)) < 1$$

ist.

2.4 Rechenaufwand

Der Gauß-Algorithmus benötigt etwa $O(n^3)$ Operationen zur Bestimmung einer Lösung. Ist die Matrix $D^{-1}(L + U)$ einmal bestimmt, was in $O(n^2)$ Operationen benötigt, ist jede weitere Iteration nur eine Multiplikation einer $n \times n$ -Matrix mit einem n -dimensionalen Vektor, was in $O(n^2)$ Operationen möglich ist. Wenn man also mit weniger als $O(n)$ Iterationen auskommt, kann ein Iterationsverfahren schneller sein.

Eine weitere Einsparung ergibt sich, wenn die Matrix $L + U$ dünn besetzt ist, wenn also nur wenige Einträge in jeder Zeile von 0 verschieden sind. Sind nur maximal k Einträge in einer Zeile von 0 verschieden, dann braucht ein Iterationsschritt sogar nur $O(nk)$ Operationen.

3 Gauß-Seidel-Verfahren

Das Jacobi-Verfahren berechnet im Idealfall bessere Approximationen der Variablen x_i . Allerdings wird bei der Berechnung des neuen Wertes von x_{i+1} immer noch der alte, "schlechte" Wert von x_i verwendet. Es liegt daher nahe zu versuchen, bei der Berechnung von x_{i+1} die bereits modifizierten Variablen zu verwenden. Die nächste Iteration der x -Vektoren wird also aus der Gleichung

$$Lx_{n+1} + Dx_{n+1} + Ux_n = b$$

bestimmt, es ist daher

$$x_{n+1} = (L + D)^{-1}(b - Ux_n). \quad (4)$$

Auch die Matrix $L + D$ ist als untere Dreiecksmatrix genau dann invertierbar, wenn die Diagonale D invertierbar ist oder wenn keines der Diagonalelemente von A verschwindet. Die Iterationsformel (4) heißt das *Gauß-Seidel-Verfahren*.

Für die Konvergenz des Gauß-Seidel-Verfahrens betrachten wir wieder die Fehler $\bar{x}_n = x_n - x^*$. Wie für das Jacobi-Verfahren folgt, dass für den Fehler die Iterationsformel

$$\bar{x}_{n+1} = (L + D)^{-1}U\bar{x}_n$$

gilt, woraus der folgende Satz folgt.

Satz 3.5 (Gauß-Seidel-Verfahren). *Das Gauß-Seidel-Verfahren zur Lösung des Gleichungssystems $Ax = b$ ist genau dann konvergent, wenn der Spektralradius*

$$\rho((L + D)^{-1}U) < 1$$

ist.

4 SOR-Verfahren

Die Verfahren von Jacobi und Gauß-Seidel versuchen die “Spannungen” in der Gleichung $Ax = b$ dadurch aufzulösen, dass die Werte der Variablen durch bessere Werte ersetzt werden, die Spannungen “entspannen” oder “relaxieren”. Beide Verfahren sind für gewisse Matrizen erfolgreich, müssen aber mehrfach wiederholt werden, um gute Resultate zu erhalten. Der Genauigkeitsgewinn in jeder Iteration kann aber eher klein sein. Oft wird der nächste Schritt den aktuellen Lösungsvektor nochmals in ungefähr der gleichen Richtung. Es kann daher eine erfolgversprechende Strategie sein, die Korrektur durch das Gauß-Seidel-Verfahren von Anfang an zu überhöhen und damit möglicherweise schnellere Konvergenz zu erreichen. Sie heißt sukzessive Überrelaxation oder SOR für *successive overrelaxation*.

Die Iterationsformel für die Lösung im Gauß-Seidel-Verfahren ist

$$x_{n+1} = b - (L + D)^{-1}Ux_n.$$

Die konvexe Kombination

$$\omega x_{n+1} + (1 - \omega)x_n$$

liefert x_n für $\omega = 0$ und x_{n+1} für $\omega = 1$. Durch Wahl eines Wertes $\omega > 1$ kann man den nächsten Wert weiter in die Richtung verschieben, in die die Korrektur zwischen x_n und x_{n+1} ohnehin erfolgt ist. Wir erwarten daher, dass das Verfahren mit der Iterationsformel

$$x_{n+1} = (1 - \omega)x_n + \omega(b - (L + D)^{-1}Ux_n) \quad (5)$$

schneller konvergiert. Für $\omega = 1$ ist die Formel (5) das bisherige Gauß-Seidel-Verfahren.

Die Iterationsformel kann auch als Gleichungssystem geschrieben werden, es ist

$$Dx_{n+1} + \omega Lx_{n+1} = (1 - \omega)Dx_n + \omega b - \omega Ux_n$$

oder zusammengefasst

$$(D + \omega L)x_{n+1} = \omega b + ((1 - \omega)D - \omega U)x_n. \quad (6)$$

Für die Lösung x^* gilt

$$\omega(L + D + U)x^* = \omega b$$

oder analog zu (6) aufgeteilt

$$\omega Lx^* = \omega b - \omega Ux^* - \omega Dx^*.$$

Durch Addition von Dx^* auf beiden Seiten entsteht

$$(D + \omega L)x^* = \omega b - \omega Ux^* + (1 - \omega)Dx^*,$$

was mit (6) übereinstimmt. Die Differenz liefert uns daher eine Formel für die Entwicklung des Fehlers:

$$(D + \omega L)\bar{x}_{n+1} = ((1 - \omega)D - \omega U)\bar{x}_n$$

oder aufgelöst nach \bar{x}_{n+1}

$$\bar{x}_{n+1} = (D + \omega L)^{-1}((1 - \omega)D - \omega U)\bar{x}_n.$$

Die Konvergenzgeschwindigkeit hängt vom Spektralradius der Matrix $M_\omega = (D + \omega L)^{-1}((1 - \omega)D - \omega U)$ ab.

Satz 3.6. *Das Verfahren der sukzessiven Überrelaxation (SOR) ist genau dann konvergent, wenn der Spektralradius*

$$\varrho((D + \omega L)^{-1}((1 - \omega)D - \omega U)) < 1$$

ist.

Je kleiner der Spektralradius ist, desto schneller wird auch die Konvergenz. Man kann zeigen, dass die optimale Wahl des Parameters ω durch

$$\omega_{\text{opt}} = \frac{2}{1 + \sqrt{1 - \varrho^2}}$$

gegeben ist, wobei ϱ der Spektralradius der Matrix $(L + D)^{-1}U$ ist. Für Werte von ω außerhalb des Intervalls $(0, 2)$ kann man nicht mehr mit Konvergenz rechnen, wie Kahan gezeigt hat.