

Formelsammlung—Numerische Methoden

Tim Hilt

Emil Slomka

25. Juni 2020

1 Lineare Gleichungssysteme

Die unten beschriebenen Verfahren suchen Lösungen für die x -Werte.

1.1 Jacobi-Iteration

1.1.1 Jacobi-Iteration in Matrix-Vektor-Notation

L: Unterer Teil der Matrix

D: Diagonale der Matrix

U: Oberer Teil der Matrix

$$\mathbf{x}^{(k+1)} = -\mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{x}^{(k)} + \mathbf{D}^{-1}\mathbf{b}$$

Achtung: Wenn bei der Jacobi-Iteration alle Startwerte = 0 sind muss **Nur der zweite Term $\mathbf{D}^{-1}\mathbf{b}$** betrachtet werden!!!

1.1.2 Vorgehen

1. Stelle einzelne Gleichungen auf
2. Auflösen nach den Variablen der jeweiligen Zeile
3. Links steht jetzt die Variable der nächsten Iteration, rechts stehen die vorhergehenden Werte.
4. Gleichungen ausrechnen

1.2 Gauss-Seidel-Iteration

Die Gauss-Seidel-Iteration funktioniert genauso wie die Jacobi-Iteration, mit dem Unterschied, dass neu berechnete Werte direkt weiterverwendet werden.

1.3 Konvergenz

Die untenstehenden Kriterien stellen das Konvergenzkriterium für beide Iterationsverfahren (Jacobi genauso wie Gauss-Seidel) dar. Es gilt sowohl für das Jacobi- als auch für das Gauss-Seidel-Verfahren und für beliebige Startwerte.

1.3.1 Diagonaldominanz

Eine Matrix ist dann diagonaldominant, wenn in allen Zeilen der Betrag des Diagonalelements der Matrix größer ist als die Summe des Betrages der restlichen Elemente.

Wenn die Matrix A diagonaldominant ist ist eine Konvergenz der Iterationsverfahren garantiert. Ist sie nicht diagonaldominant, so ist die Konvergenz nicht garantiert (jedoch trotzdem nicht ausgeschlossen).

$$\sum_{j \neq k} |a_{ij}| < |a_{kk}| \text{ für } k = 1, \dots, n.$$

1.3.2 Spektralradius

$$\rho(\mathbf{A}) = \max_{j=1, \dots, n} |\lambda_j| = \max(|-\mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})|)$$

\Rightarrow Die Iteration konvergiert, wenn $|\rho(\mathbf{A})| < 1$

\Rightarrow Je kleiner der Spektralradius, desto schneller die Konvergenz

2 Nicht-lineare Gleichungssysteme

2.1 Fixpunktiteration

Gegeben ist ein nichtlineares Gleichungssystem (NGS), dessen Nullstellen es zu bestimmen gilt. Das gegebene NGS hat in unserem Fall nur eine einzelne Variable. Zur Berechnung der Nullstellen der Funktion f müssen wir die Funktion:

1. $= 0$ setzen
2. Nach x auflösen (auf der anderen Seite steht dann die Iterationsfunktion $g(x)$)
3. Gegeben ist ein Startwert $x^{(0)}$, der in $g(x)$ eingesetzt wird um $x^{(k+1)}$ zu berechnen

Konvergiert die Fixpunktfunktion g gegen einen Fixpunkt x^* , dann ist dieser Fixpunkt eine Nullstelle von f .

2.2 Eigenschaften von Iterationsfunktionen

2.3 Konvergenz

2.3.1 Alternierende Konvergenz

Wenn bei einer konvergenten Zahlenfolge der nachfolgende Wert $x^{(k+1)}$ zwischen den beiden Vorgängerwerten $x^{(k-1)}$ und $x^{(k)}$ liegt, d.h.

$$x^{(k+1)} \leq x^{(k+1)} \leq x^{(k)}, \quad k = 1, 2, \dots$$

oder

$$x^{(k)} \leq x^{(k+1)} \leq x^{(k-1)}, \quad k = 1, 2, \dots$$

dann bezeichnet man die Konvergenz als **alternierend**.

2.3.2 Monotone Konvergenz

Bei einer konvergenten Zahlenfolge spricht man von **monoton fallender Konvergenz**, wenn die nachfolgenden Werte stets kleiner sind als die Vorgängerwerte

$$x^{(k+1)} \leq x^{(k)}, \quad k = 1, 2, \dots$$

und von **monoton wachsender Konvergenz**, wenn die nachfolgenden Werte stets größer sind als die Vorgängerwerte

$$x^{(k+1)} \geq x^{(k)}, \quad k = 1, 2, \dots$$

2.4 Konvergenz der Fixpunktiteration

2.5 Lipschitz

2.6 Fehlerabschätzung

2.6.1 A priori

Bsp.: „Wie viele Iterationen benötigt man, um die Genauigkeit x zu erreichen?“

„Welche Genauigkeit erhält man nach k Iterationen?“

2.6.2 A posteriori

3 Interpolation und Approximation

3.1 Newton-Tableau

3.2 Hermite-Tableau

Das Hermite-Tableau stellt eine Generalisierung der Newton-Tableaus dar, mithilfe derer auch Eigenschaften der Ableitungen ins Interpolationspolynom aufgenommen werden können.

4 Numerische Integration

4.1 Trapezregel

$$h = \frac{b-a}{n}$$

$$T(h) = h \left(\frac{1}{2}f(a) + f(a+h) + f(a+2h) + \dots + f(a+(n-1)h) + \frac{1}{2}f(b) \right)$$

$$T(h) = h \cdot \left(\frac{1}{2} \cdot (f(a) + f(b)) + \sum_{k=1}^{n-1} f(a+k \cdot h) \right)$$

4.2 Romberg-Verfahren

$$\begin{aligned} T(4)_1 & \\ T(2)_1 & \rightarrow T(2)_2 = \frac{4}{3} \cdot T(2) - \frac{1}{3} \cdot T(4) \\ T(1)_1 & \rightarrow T(1)_2 = \frac{4}{3} \cdot T(1) - \frac{1}{3} \cdot T(2) \qquad \rightarrow T(1)_3 = \frac{16}{15} \cdot T(1)_2 - \frac{1}{15} \cdot T(2)_2 \end{aligned}$$

5 Optimierung

5.1 Gradientenverfahren

$$\mathbf{x}^{(k+1)} = \mathbf{x}^k - t^{(k)} \cdot \nabla f(\mathbf{x}^{(k)})$$

6 Gewöhnliche Differenzialgleichungen

6.1 Eulerverfahren

$$x'(t) = f(t, x(t)), \quad x(t_0) = x_0, \quad t \in [t_0, t_n]$$

$$\tilde{x}_{k+1} = \tilde{x}_k + h \cdot f(t_k, \tilde{x}_k)$$

$$t_{k+1} = t_k + h$$

6.1.1 Fehlerabschätzung / Toleranz

Lokaler Fehler Benötigt:

- $\tilde{x}^{(k+1)}$: Näherungswert nach einem Iterationsschritt mit Schrittweite h
- $\tilde{y}^{(k+1)}$: Näherungswert nach **zwei** Iterationsschritten mit Schrittweite $h/2$

$$e_{\text{lokal}} = |\tilde{x}^{(k+1)} - \tilde{y}^{(k+1)}|$$

Globaler Fehler

$$e_{\text{global}} = \frac{|\tilde{x}^{(k+1)} - \tilde{y}^{(k+1)}|}{h}$$

6.1.2 Optimale Schrittweite

$$h_{\text{opt}} = \frac{\text{tol} \cdot h}{|\tilde{x}^{(k+1)} - \tilde{y}^{(k+1)}|} \cdot h$$

6.1.3 Schrittweitensteuerung

Falls $e_{\text{global}} > \text{tol}$ setze $h = h/2$

Ansonsten berechne h_{opt} und setze $h = h_{\text{opt}}$