

Iterative Lösung auf Basis des Jacobi bzw. Gauss-Seidel Verfahrens

Für allgemeinere Randwertprobleme (auch nicht-lineare) werden oft iterative Verfahren genutzt.

Ausgehend von einem Schätzwert für das Ergebnis wird durch Nutzen der DGL iterativ eine (hoffentlich) verbesserte Lösung gefunden.

Das wird auch später bei partiellen DGL eine Rolle spielen.

Einfachster Fall: **Jacobi-Verfahren**

Dazu betrachten wir die DGL $u''(x) = f(x, u(x))$ mit Randbedingungen

$$u(x_a) = u_a \quad \text{und} \quad u(x_b) = u_b$$

mit der einfachen Diskretisierung (äquidistant)

$$u''(x_i) = \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1}))}{h^2} + O(h^2) = f(x_i, u(x_i))$$

gibt es einen Zusammenhang zwischen alter (k -ter) Schätzung und neuer Funktion ($k+1$ -ter)

$$u_i^{(k+1)} = \frac{1}{2} \left(u_{i+1}^{(k)} + u_{i-1}^{(k)} - h^2 f(x_i, u_i^{(k)}) \right) \quad (\text{Iterationsvorschrift})$$

Frage: Konvergiert man nach dieser Vorschrift gegen eine Lösung?

Dazu konstruieren wir das **Funktional** $\varepsilon[u(x)] = \int_{x_a}^{x_b} dx g(x, u, u')$

$$\text{mit } g(x, u, u') = \frac{1}{2}(u'(x))^2 + F(x, u(x)) \quad \text{und} \quad f(x, u) = \frac{\partial F}{\partial u}(x, u)$$

—> Variation des Funktionals mit $\delta\varepsilon \stackrel{!}{=} 0$ (minimales Funktional) führt auf DGL zurück:

$$\delta\varepsilon \stackrel{!}{=} 0 \Leftrightarrow \frac{\partial g}{\partial u} - \frac{d}{dx} \frac{\partial g}{\partial u'} = 0 \quad (\text{Vgl. Prinzip der kleinsten Wirkung, Euler-Lagrange-Gl.})$$

$$\text{Hier: } f(x, u(x)) - \frac{d}{dx} u'(x) = 0 \quad \Rightarrow \quad u''(x) = f(x, u(x))$$

Zu zeigen: **Iterationsvorschrift** verkleinert das Funktional in jedem Schritt
(d.h. mit jedem Iterationsschritt nähern wir uns der echten Lag der DGL an, für die das Funktional minimal wird.)

Zur Verbesserung der Konvergenz betrachten wir gleich eine Verallgemeinerung:
das **Gauss-Seidel-Verfahren**

$$\begin{aligned} u_i^{(k+1)} &= \frac{1}{2} \left(u_{i+1}^{(k)} + u_{i-1}^{(k)} - h^2 f(x_i, u_i^{(k+1)}) \right) \\ &\approx \frac{1}{2} \left[u_{i+1}^{(k)} + u_{i-1}^{(k)} - h^2 \left(f(x_i, u_i^{(k)}) + (\partial_u f)(x_i, u_i^{(k)}) \left(u_i^{(k+1)} - u_i^{(k)} \right) \right) \right] \end{aligned}$$

Hier haben wir die DGL implizit genutzt. Durch Linearisierung können wir nach der neuen Lösung auflösen.

Damit ergibt sich

$$\tilde{u}_i^{(k+1)} = \frac{u_{i+1}^{(k)} + u_{i-1}^{(k)} - h^2 f(x_i, u_i^{(k)}) + h^2 (\partial_u f)(x_i, u_i^{(k)}) u_i^{(k)}}{2 + h^2 (\partial_u f)(x_i, u_i^{(k)})}$$

Dies wird nun modifiziert mit einem **Relaxationsparameter** R und der Annahme, dass schon bekannte verbesserte Lösungen möglichst genutzt werden.

$$\begin{aligned} u_i^{(k+1)} &= (1 - R)u_i^{(k)} + R \frac{u_{i+1}^{(k)} + u_{i-1}^{(k+1)} - h^2 f(x_i, u_i^{(k)}) + h^2 (\partial_u f)(x_i, u_i^{(k)}) u_i^{(k)}}{2 + h^2 (\partial_u f)(x_i, u_i^{(k)})} \\ &= u_i^{(k)} + R \frac{u_{i+1}^{(k)} + u_{i-1}^{(k+1)} - 2u_i^{(k)} - h^2 f(x_i, u_i^{(k)})}{2 + h^2 (\partial_u f)(x_i, u_i^{(k)})} \equiv u_i^{(k)} + R \Delta_i \end{aligned}$$

R ist ein willkürlicher Parameter. Die Frage ist nun, wie verändert sich das Funktional durch eine Iteration?

Diskretisieren ergibt für das Funktional

$$\mathcal{E}[u] \approx E[u] = \frac{1}{2h} \sum_{i=1}^n (u_i - u_{i-1})^2 + h \sum_{i=1}^n F(x_i, u_i)$$

Mit dieser Approximation findet man

$$\begin{aligned}
 \Delta E_i &= E(u_1^{k+1}, \dots, u_{i-1}^{k+1}, u_i^{k+1}, u_{i+1}^k, \dots, u_n^k) - E(u_1^{k+1}, \dots, u_{i-1}^{k+1}, u_i^k, u_{i+1}^k, \dots, u_n^k) \\
 &= \frac{1}{2h} [(u_i^{k+1} - u_{i-1}^{k+1})^2 + (u_{i+1}^k - u_i^{k+1})^2] + h F(x_i, u_i^{k+1}) \\
 &\quad - \frac{1}{2h} [(u_i^k - u_{i-1}^{k+1})^2 + (u_{i+1}^k - u_i^k)^2] - h F(x_i, u_i^k) \\
 &= \frac{1}{2h} [(u_i^k + R\Delta_i - u_{i-1}^{k+1})^2 + (u_{i+1}^k - u_i^k - R\Delta_i)^2] \\
 &\quad - \frac{1}{2h} [(u_i^k - u_{i-1}^{k+1})^2 + (u_{i+1}^k - u_i^k)^2] + h f(x_i, u_i^k) R \Delta_i + \frac{h}{2} (\partial_u f)(x_i, u_i^k) R^2 \Delta_i^2 \\
 &= - \frac{2R\Delta_i}{2h} \underbrace{[u_{i-1}^{k+1} + u_{i+1}^k - 2u_i^k - h^2 f(x_i, u_i^k)]}_{=(2+h^2 (\partial_u f)(x_i, u_i^k)) \Delta_i} + \frac{R^2 \Delta_i^2}{h} + \frac{h}{2} (\partial_u f)(x_i, u_i^k) R^2 \Delta_i^2 \\
 &= - \frac{R(2-R)}{h} \Delta_i^2 \left(1 + \frac{h^2}{2} (\partial_u f)(x_i, u_i^k) \right)
 \end{aligned}$$

D.h. die Änderung ist negativ für $0 < R < 2$.

Die **Konvergenzgeschwindigkeit** des Verfahrens hängt von R ab.

Das werden Sie in den Übungen an einem Beispiel untersuchen.

Direkte oder iterative Lösung mit Hilfe eines linearen Gleichungssystems

Die DGL kann durch Diskretisierung auch direkt in ein lineares Gleichungssystem überführt werden.

Die Form des Gleichungssystems erlaubt dabei eine besonders effiziente Lösung. Wir gehen wieder von der DGL

$$u''(x) = f(x, u(x))$$

mit Randbedingungen $u(x_a) = u_a$ und $u(x_b) = u_b$ aus.

Durch Diskretisierung der Ableitung erhalten wir dann (wie oben)

$$u''(x_i) = \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1}))}{h^2} + O(h^2) = f(x_i, u(x_i))$$

wobei wir hier $n + 1$ Stützstellen verwenden

$$x_i = x_a + i h \quad \text{mit} \quad h = \frac{x_b - x_a}{n} \quad \text{und} \quad i = 0, \dots, n$$

Kürzer kann man bis einschließlich zur Ordnung h^3 schreiben

$$u_{i+1} - 2u_i + u_{i-1} - h^2 f_i = 0$$

Wir nehmen wieder an, dass wir eine Näherung an die Lösung kennen, weil uns das erlaubt die Gleichung zu linearisieren.

(bei linearen Gleichungen ist das natürlich nicht nötig)

Die Näherung sei $\mathbf{u}^{(1)} = (u_0^{(1)}, \dots, u_n^{(1)})$ und erfülle die Randbedingungen $u_0^{(1)} = u_a$ und $u_n^{(1)} = u_b$

Durch Einsetzen in die DGL erhalten wir ein Maß für den *lokalen Fehler*:

$$e_i^{(1)} = e_i^{(1)}(u_{i+1}^{(1)}, u_i^{(1)}, u_{i-1}^{(1)}) = u_{i+1}^{(1)} - 2u_i^{(1)} + u_{i-1}^{(1)} - h^2 f(x_i, u_i^{(1)})$$

der nur von den benachbarten Funktionswerten abhängt. Wir müssen diese erste Näherung so ändern, dass der Fehler verschwindet.

D.h. mit $\mathbf{u}^{(2)} = \mathbf{u}^{(1)} + \eta$ muss gelten

$$e_i^{(1)}(u_{i+1}^{(1)} + \eta_{i+1}, u_i^{(1)} + \eta_i, u_{i-1}^{(1)} + \eta_{i-1}) = 0$$

Mit der Taylor-Entwicklung (oder im Falle linearer Gleichungen exakt)

$$e_i^{(1)}(u_{i+1}^{(1)}, u_i^{(1)}, u_{i-1}^{(1)}) + \frac{\partial e_i^{(1)}}{\partial u_{i+1}} \eta_{i+1} + \frac{\partial e_i^{(1)}}{\partial u_i} \eta_i + \frac{\partial e_i^{(1)}}{\partial u_{i-1}} \eta_{i-1} \equiv e_i^{(1)} + \alpha_i \eta_{i-1} + \beta_i \eta_i + \gamma_i \eta_{i+1} = 0$$

wobei $\alpha_i = \frac{\partial e_i^{(1)}}{\partial u_{i-1}} = 1$ und $\beta_i = \frac{\partial e_i^{(1)}}{\partial u_i} = -2 - h^2 \frac{\partial f}{\partial u}(x_i, u_i^{(1)})$ und $\gamma_i = \frac{\partial e_i^{(1)}}{\partial u_{i+1}} = 1$

In **Matrixform** entspricht das

$$\begin{pmatrix} \beta_0 & \gamma_0 & 0 & 0 & \cdots & 0 \\ \alpha_1 & \beta_1 & \gamma_1 & 0 & \cdots & 0 \\ 0 & \alpha_2 & \beta_2 & \gamma_2 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \alpha_{n-1} & \beta_{n-1} & \gamma_{n-1} \\ 0 & \cdots & \cdots & 0 & \alpha_n & \beta_n \end{pmatrix} \cdot \begin{pmatrix} \eta_0 \\ \eta_1 \\ \eta_2 \\ \vdots \\ \eta_{n-1} \\ \eta_n \end{pmatrix} = - \begin{pmatrix} e_0^{(1)} \\ e_1^{(1)} \\ e_2^{(1)} \\ \vdots \\ e_{n-1}^{(1)} \\ e_n^{(1)} \end{pmatrix} \quad (*)$$

η_0 und η_n sind bei uns keine Unbekannten sondern werden je nach Randbedingung besonders behandelt (siehe Beispiel später).

Die Lösung des Gleichungssystems ergibt die notwendige Änderung der approximierten Lösung und damit die verbesserte (oder exakte) Lösung **u⁽²⁾**

Beachte: Das Gleichungssystem ist *tridiagonal*.

Im Folgenden betrachten wir eine sehr schnelle, i.e. effiziente, Möglichkeit dieses System mit **Bandstruktur** zu lösen.

Dazu möchten wir in zwei Schritten jeweils **rekursiv** vorgehen.
Für die allgemeine Lösung gehen wir erstmal davon aus, dass alle η_i Unbekannte sind.

Mit dem Ansatz $\eta_{i+1} = p_i \eta_i + q_i$ erhält man für eine Zeile des Gleichungssystems

$$\alpha_i \eta_{i-1} + \beta_i \eta_i + \gamma_i \eta_{i+1} = \alpha_i \eta_{i-1} + (\beta_i + \gamma_i p_i) \eta_i + \gamma_i q_i = -e_i^{(1)}$$

Damit findet man die Form von oben für $i \rightarrow i - 1$


$$\eta_i = -\frac{\alpha_i}{\beta_i + \gamma_i p_i} \eta_{i-1} - \frac{e_i^{(1)} + \gamma_i q_i}{\beta_i + \gamma_i p_i}$$

und kann eine Rückwärtsrekursion für p_i und q_i ablesen:

$$p_{i-1} = -\frac{\alpha_i}{\beta_i + \gamma_i p_i} \quad q_{i-1} = -\frac{e_i^{(1)} + \gamma_i q_i}{\beta_i + \gamma_i p_i} \quad (*)$$

Die Startwerte bestimmt man daraus, dass die letzte Gleichung des Systems erfüllt ist für eine beliebige Wahl von η_{i-1} .

$$\alpha_n \eta_{n-1} + \beta_n (p_{n-1} \eta_{n-1} + q_{n-1}) = -e_n^{(1)}$$


$$p_{n-1} = -\frac{\alpha_n}{\beta_n} \quad q_{n-1} = -\frac{e_n^{(1)}}{\beta_n}$$

Mit diesen Startwerten und der Rekursionsformel (*) werden alle p_i und q_i festgelegt.

Sobald alle p und q bestimmt sind, kann man durch die erste Gleichung des Systems

$$\beta_0 \eta_0 + \gamma_0(p_0 \eta_0 + q_0) = -e_0^{(1)}$$

den Startwert $\eta_0 = -\frac{\gamma_0 q_0 + e_0^{(1)}}{\beta_0 + \gamma_0 p_0}$ finden

und durch **Vorwärtsrekursion** die Lösung $\eta_{i+1} = p_i \eta_i + q_i$ bestimmen.

Im Code-Beispiel unten ist dieses Verfahren allgemein implementiert.

$$\begin{pmatrix} b_0 & c_0 & 0 & \dots & 0 \\ a_1 & b_1 & c_1 & 0 & \dots \\ 0 & \ddots & \ddots & \ddots & \\ 0 & & & & \\ 0 & & & a_{n-1} & b_{n-1} \end{pmatrix} \cdot \begin{pmatrix} d_0 \\ \vdots \\ d_{n-1} \end{pmatrix} = \begin{pmatrix} e_0 \\ \vdots \\ e_{n-1} \end{pmatrix}$$

Wir werden später noch diskutieren, dass die Lösung eines Gleichungssystems üblicherweise $\propto n^3$ Schritte benötigt.

Die Anzahl der notwendigen Schritte ist bei diesem Verfahren nur $\propto n$!