

Nichtlineare - Gleichungssysteme:

Lösungsvorgehen:

- $g(x) = h(x) \rightarrow g(x) - h(x) = f(x) = 0$
- Gleichung in ein Nullstellenfindungsproblem umschreiben
- Variable einführen: $x = \sqrt{2} \rightarrow 24 - x^2 = 0$
- Nullstellensatz von Bolzano:

$f(x)$: stetig in $[a, b]$, $f(a) \cdot f(b) < 0$
 $\exists x^* \in (a, b)$ mit $f(x^*) = 0$ (unterschiedl. Vorzeichen)

Kondition (conditioning)

- beschreibt die Abhängigkeit der Lösung eines Problems von der Störung der Eingangsdaten
- well-conditioned:

$f(x)$ kleine Änderung des Inputs bewirkt kleine Änderung des Outputs

→ ill-conditioned: kleine Änderung des Inputs bewirkt große Änderung des Outputs

→ $f'(x^*) = 0$ (kleine $|f'(x^*)|$)

→ Nullstellenvielfachheit > 1

Konditionszahl (condition number): $K = \frac{|f(x)|}{|f'(x)|} = \frac{|f(x+b)| - f(x)|}{|f'(x)| \cdot b}$

$|f(x)| \ll 1 \rightarrow$ Taylor-Reihen-Approximation: $K \approx \frac{|f(x) + f'(x)b - f(x)|}{|f'(x)| \cdot b} = \frac{|f'(x)b|}{|f'(x)| \cdot b} = 1$

→ Nullstellenfindungsproblem: $K = \frac{2}{|f'(x^*)|} = \frac{2}{|f'(x^*)|} = 1$

Konvergenzgrad: $\lim_{k \rightarrow \infty} \frac{|e_{k+1}|}{|e_k|^r} = C$

→ Fehler: $e_k = x_k - x^*$

$r = 1: 0 < C < 1 \Leftrightarrow$ linear
 $C = 0 \Leftrightarrow$ superlinear
 $r = 2$: quadratisch

numerische Berechnung: $|e_{k+1}| \approx C|e_k|^r$

$C \approx \frac{|e_{k+1}|}{|e_k|^r} \Rightarrow \frac{|e_{k+1}|}{|e_k|} \approx C|e_k|^{r-1}$

$r \approx \frac{\log \frac{|e_{k+1}|}{|e_k|}}{\log \frac{|e_k|}{|e_{k-1}|}}$

Bsp.: Konvergenzrate $x_k = \frac{1}{k!}, x^* = 0$

$\lim_{k \rightarrow \infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^r} = \lim_{k \rightarrow \infty} \frac{(k+1)^r}{(k+1)!} = 0$

Bisektion:

→ Pseudocode:

Input: a, b (Anfangsintervall), ϵ_{tol} (minimale Toleranz), k_{max} (max # Durchführungen)

Steps: $k \leftarrow 1$, while $(b-a) > \epsilon_{tol}$ and $k \leq k_{max}$ do

$x_k \leftarrow (a+b)/2$, if $\text{sign}(f(a)) = \text{sign}(f(x_k))$ then $a \leftarrow x_k$

else $b \leftarrow x_k$, end if, $k \leftarrow k+1$, end while

Output: x_k (approximative Lösung nach k Iterationen)

Bisektionsverfahren:

→ Konvergenzgrad: $|e_k| = \frac{1}{2}|e_{k-1}| \leq 2^{-k}(b-a)$

lim $\frac{|e_{k+1}|}{|e_k|^r} = \lim_{k \rightarrow \infty} \frac{(2^{k+1})^r}{2^{k+1}} = \begin{cases} \frac{1}{2}, r=1 \\ \infty, r>1 \end{cases}$

minimale Iterationsanzahl: $|e_k| = \epsilon_{tol} \rightarrow (b-a)2^{-k} = \epsilon_{tol} \Rightarrow k = \log_2 \frac{(b-a)}{\epsilon_{tol}}$

Eigenschaften: ① stabil (robust), ② nur Vorzeichen, keine Werte & Ableitungen

Newton-Verfahren: $f(x) \approx f(x_k) + f'(x_k)(x - x_k)$

→ Herleitung (Taylor): $f(x) \approx f(x_k) + f'(x_k)(x - x_k)$

→ Konvergenzgrad: $\lim_{k \rightarrow \infty} \frac{|e_{k+1}|}{|e_k|^r} = \lim_{k \rightarrow \infty} \frac{|f''(x_k)|}{2|f'(x_k)|} = \frac{1}{2} \frac{|f''(x^*)|}{|f'(x^*)|}$

Bedingung $f'(x_k) \neq 0 \forall k$, $m=1 \Leftrightarrow r=2$

→ Spezialfälle: $x_{k+1} = \alpha x_k \Rightarrow$ divergiert, $|\alpha| > 1$

$x_{k+1} = \alpha x_k \Rightarrow$ konvergiert, $|\alpha| < 1$

$x_{k+1} = \alpha x_k \Rightarrow$ periodisch, $|\alpha| = 1$ (Konvergenz nicht)

Eigenschaften: ① schnelle Konvergenz, ② nicht robuste ($f'(x_k) = 0$)

sekanten-Methode: $x_{k+1} = x_k - f(x_k) \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}$

Konvergenzgrad: $r = \frac{1}{2}$ (quadratischer Schritt)

Eigenschaften: ① Ableitung wird approximiert, ② nur 1 Funktionswert berechnen

Gleichungssysteme: $F(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$

Jakobi-Matrix: $J(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$

Mehrdimensionales Newton-Verfahren: $x_{k+1} = x_k - J(x_k)^{-1} F(x_k)$

Herleitung: Taylor für Vektorfunktionen: $F(x) \approx F(x_k) + J(x_k)(x - x_k)$

$0 = F(x_k) + J(x_k)(x_{k+1} - x_k) \Rightarrow x_{k+1} = x_k + z$

$Az = b, A = J(x_k), b = -F(x_k)$

Newton-Raphson ($N=M$): $x_{k+1} = x_k - J(x_k)^{-1} F(x_k)$

Konvergenz: quadratisch, außer singular

Kosten: Bilden von $J: O(N^2)$, LGS lösen: $O(N^3)$

pseudo-Newton ($M \neq N$): $J^+ = (J^T J)^{-1} J^T$

$M > N: J^+ = (J^T J)^{-1} J^T$, stationäre

$M < N: J^+ = J^T (J J^T)^{-1}$, Punkte

Algorithmus konvergiert zu $\|F(x)\|_2$

Modifiziertes Newton-Verfahren: $J_0 = J(x_0) = \text{konst.}$

$J_0 z = -F(x_k)$, erfolgreich, falls J sich kaum ändert

Quasi-Newton: $J_{k+1} = J_k + \frac{\Delta F \Delta x^T (\Delta x)^T \Delta x}{(\Delta x)^T \Delta x}$

Nicht-lineare Optimierung: Gradient Descent: $z = -\nabla E(x_k)$

Methode der kleinsten Fehlerquadrate:

Linear Least Squares:

Datensatz: $\{(x_i, y_i)\}_{i=1}^N$

Funktion: $F(x, w) = \sum_{i=1}^N w_i \phi_i(x)$

Matrix-Darstellung: $\begin{pmatrix} \phi_1(x_1) & \phi_2(x_1) & \dots & \phi_N(x_1) \\ \phi_1(x_2) & \phi_2(x_2) & \dots & \phi_N(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(x_N) & \phi_2(x_N) & \dots & \phi_N(x_N) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}$

Lösungsmengen: $M = N$: eindeutig, $M > N$: unendliche # Lösungen

$M < N$: überbestimmt $\rightarrow w^* = \min \|E(w)\|$

Lineare Funktion: $w_1 + w_2 x_i \approx y_i$

orthonormale Alternativen: $w_1 = \frac{b \cdot c - a \cdot d}{N \cdot d - a \cdot c}$

$w_2 = \frac{N \cdot d - a \cdot c}{N \cdot b - a \cdot d}$

Orthonomaler Fall: $A^T A = I, w = A^T y$

Fehler: $\|e\|_2 = \sqrt{\sum e_i^2} = \sqrt{\sum (y_i - f(x_i))^2}$

P-2-Norm: $\|e\|_1 = \sum |e_i| = \sum |y_i - f(x_i)|$

P-1-Norm: $\|e\|_1 = \sum |e_i| = \sum |y_i - f(x_i)|$

P-∞-Norm: $\|e\|_\infty = \max |e_i| = \max |y_i - f(x_i)|$

P-Norm: $\|e\|_p = (\sum |e_i|^p)^{1/p} = (\sum |y_i - f(x_i)|^p)^{1/p}$

Anwendung bei LLS: $w^* = \min \|E(w)\|$

Normalgleichung: $A^T A w = A^T y$

ATA ist symmetrisch und positiv definit

invertierbar: $w^* = (A^T A)^{-1} A^T y$

Geometrische Interpretation: $\|e\|_2 = \|y - A w\|_2$

$\|e\|_2 = \text{minimal} \Leftrightarrow e \perp \text{span}\{a^{(1)}, \dots, a^{(N)}\}$

$A^T e = 0$, $A^T (y - A w) = 0 \Leftrightarrow A^T A w = A^T y$

Projektionsmatrix: $P = A(A^T A)^{-1} A^T$

$P + M = I, P M = 0, P A = 0, M A = 0$

Eigenschaften einer Projektionsmatrix: symmetrisch, idempotent ($P^2 = P$)

Numerische Lösung: $K(x) = \|A(x) - y\|_2$

Konditionszahl: $K(x) = \frac{\|A(x)\|_2}{\sigma_{\min}(A(x))}$

Normalgleichung ist ill-conditioned

Maß für wie singular eine Matrix ist

Orthogonale Matrizen: $K = 1$ (bestmöglich)

QR-Zerlegung: $A = QR$

Q: Orthogonal, R: obere Dreiecksmatrix

Lagrange-Interpolation:

Lagrange Polynom:

$$L_k(x) = \frac{(x-x_1)(x-x_2)\dots(x-x_{k-1})(x-x_{k+1})\dots(x-x_n)}{(x_k-x_1)(x_k-x_2)\dots(x_k-x_{k-1})(x_k-x_{k+1})\dots(x_k-x_n)}$$

$$L_k(x) = \prod_{i=1, i \neq k}^n \frac{x-x_i}{x_k-x_i} \quad L(x_j) = \delta_{jk} \rightarrow \text{Polynom vom Grad } n-1$$

Lagrange Interpolator:

→ Ziel: A soll diagonal sein ($\Delta \vec{u} = \vec{y}$)

→ Herleitung:

$$F(x) = y_k = \sum_{i=1}^n y_i L_i(x) \rightarrow y_k = y_k$$

$$\rightarrow \text{Interpolator: } F(x) = \sum_{i=1}^n y_i L_i(x)$$

Approximationsfehler:

$$|g(x) - f(x)| = \left| \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{k=1}^n (x-x_k) \right|$$

Wird minimiert, falls $x_k = -\cos\left(\frac{2k-1}{2n}\pi\right)$ als Wurzel des Chebyschev Polynom gewählt wird

Eigenschaften:

- geht durch alle Datenpunkte
- sehr empfindlich gegenüber Rauschen (kleine)
- nicht für Extrapolation geeignet
- LLS mit einem Polynom vom Grad $n-1$ liefert das gleiche Resultat
- Lagrange-Polynom ändert sich nicht durch Rauschen, da $L_k(x)$ nur eine Fkt. von x ist
- Lagrange-Interpolator ändert sich, er beginnt zu oszillieren
- Falls eine Fkt. von Grad P angenähert wird, spielen die Terme mit höherer Ordnung als P im Interpolator keine Rolle, streichen sich
- wird eine exakte Fkt. von Grad P mit einem Lagrange-Polynom vom Grad P angenähert, dann ist der Fehler 0
- das ganze Polynom ändert sich, falls x_k geändert werden
- Runge's Phänomen: Oszillation am Rand

→ **Runge's Phänomen:** Oszillation am Rand

→ **Runge's Phänomen:** Oszillation am Rand

Spline:

Cubic splines:

- Idee: Intervall in Subdomänen aufteilen, zwischen diesen Datenpunkten fitbt man ein Polynom 3. Ordnung
- Polynom geht durch alle Datenpunkte
- 1. D2. Ableitung sind stetig zwischen 2 Fkt.
- Verhindert Oszillierung wie bei Lagrange

Randbedingungen:

Natural spline: $F''(x_1) = F''(x_n) = 0$

Clamping ends: $F'(x_1) = F'(x_n) = 0$

Parabolic runout: $F''(x_1) = F''(x_n) = 0$

→ 2 Ableitung ist eine lineare Fkt.

$$F''(x) = F''_i \frac{x_{i+1}-x}{\Delta_i} + F''_{i+1} \frac{x-x_i}{\Delta_i} + K_i$$

$$F(x) = F''_i \frac{(x_{i+1}-x)^3}{6\Delta_i} + F''_{i+1} \frac{(x-x_i)^3}{6\Delta_i} + K_i(x-x_i) + D_i$$

→ Randbedingungen: → Funktionswerte:

$$y_i = f(x_i) = F''_i \frac{\Delta_i^2}{6} + K_i \Delta_i + L_i$$

$$y_{i+1} = f(x_{i+1}) = F''_{i+1} \frac{\Delta_i^2}{6} + K_i \Delta_i + L_i$$

→ Stetigkeit der 1. Ableitung:

$$F'(x_i) = F'(x_i)^b \quad a: x_i = x_{i-1} + \Delta_{i-1} \quad b: x_i = x_i + \Delta_i$$

$$F'(x_i) = F''_i \frac{\Delta_i}{2} + K_i \Delta_i + L_i$$

$$F'(x_i) = F''_{i+1} \frac{\Delta_i}{2} + K_i \Delta_i + L_i$$

$$A_i F''_{i-1} + B_i F''_i + C_i F''_{i+1} = D_i$$

→ Koeffizienten: (nicht am Rand)

$$A_i = \frac{x_i - x_{i-1}}{\Delta_i} \quad B_i = \frac{x_{i+1} - x_i}{\Delta_i} \quad C_i = \frac{x_{i+1} - x_{i-1}}{\Delta_i}$$

$$D_i = \frac{y_{i+1} - y_i}{\Delta_i} - \frac{y_i - y_{i-1}}{\Delta_{i-1}}$$

→ Koeffizienten: (am Rand)

$$\rightarrow \text{Natural: } B_1 = B_n = 1, C_1 = D_1 = A_n = D_n = 0$$

$$\rightarrow \text{Clamping: } B_1 = \Delta_1/3, C_1 = \Delta_1/6, D_1 = \frac{y_2 - y_1}{\Delta_1}$$

$$A_n = \Delta_n/6, B_n = \Delta_n/3, D_n = \frac{y_n - y_{n-1}}{\Delta_n}$$

$$\rightarrow \text{parabolic: } B_1 = A_n = 1, C_1 = B_n = -1, D_1 = D_n = 0$$

→ Periodizität:

$$\Delta_i = \text{konst} \Rightarrow A_i, B_i, C_i, D_i \text{ sind gleich } \forall i$$

→ Lösungen für den Rand sind nicht eindeutig

Tridiagonalmatrix:

→ Kosten: $O(n)$ zum lösen

$$\begin{bmatrix} B_1 & C_1 & 0 & \dots & 0 \\ A_1 & B_2 & C_2 & \dots & 0 \\ 0 & A_2 & B_3 & C_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & A_n & B_n \end{bmatrix} \begin{bmatrix} F''_1 \\ F''_2 \\ F''_3 \\ \vdots \\ F''_n \end{bmatrix} = \begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ \vdots \\ D_n \end{bmatrix}$$

→ Lösungsvorgehen: untere Nebendiagonale zu 0 machen

→ Lösung:

$$\begin{bmatrix} 1 & w_1 & 0 & \dots & 0 \\ 0 & 1 & w_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & w_n \\ 0 & \dots & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F''_1 \\ F''_2 \\ F''_3 \\ \vdots \\ F''_n \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ \vdots \\ g_n \end{bmatrix}$$

$$F''_i = g_i - w_i F''_{i+1}$$

$$x_n = g_n, x_i = g_i - w_i F''_{i+1}$$

→ B-splines: (Basis spline)

→ Allgemeine Form von Splines

→ Kontrollmüllgrad kann frei gewählt werden

→ Jede Spline ist eine Linearkombination von B-splines

→ Aufbau:

→ stückweise stetige Polynomfunktion, die sich in Knoten treffen

→ Knoten treffen → Knotenvektor $E_j(1, m, d, d)$

→ Grad d → stetige Ableitungen bis $d-1$

→ Menge von B-splines: $B_{i,0}, B_{i,1}, \dots, B_{i,M}$

→ rekursive Definition:

$$B_{i,0}, B_{i,1} = \begin{cases} 1 & \text{falls } x_i \leq x \leq x_{i+1} \\ 0 & \text{sonst} \end{cases}$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

$$B_{i,k}(x) = \frac{x-x_i}{x_{i+1}-x_i} B_{i,k-1}(x) + \frac{x-x_{i+1}}{x_{i+2}-x_{i+1}} B_{i+1,k-1}(x)$$

Numerische Integration:

Anwendung:

- Integral ist analytisch nicht lösbar
- Funktion ist nur für einen Satz diskreter Punkte bekannt (z.B. Messungen)

Grundidee:

→ Integrationsdomäne in N -Intervalle aufspalten

$$I = \int_a^b f(x) dx = \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} f(x) dx \approx \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} p_i(x) dx$$

Rectangle Rule:

$$p_i(x) = f(x_i) \Delta_i \quad E_i = O(\Delta_i^2)$$

$$I \approx \Delta x \sum_{i=0}^{N-1} f(x_i)$$

Midpoint Rule:

$$p_i(x) = f\left(\frac{x_i + x_{i+1}}{2}\right) \Delta_i$$

$$I \approx \Delta x \sum_{i=0}^{N-1} f\left(\frac{x_i + x_{i+1}}{2}\right)$$

$$E_i = \frac{1}{24} f''(\xi_i) \Delta_i^3 + O(\Delta_i^5)$$

Trapezoidal Rule:

$$p_i(x) = \frac{1}{2} (f(x_i) + f(x_{i+1})) \Delta_i$$

$$I \approx \frac{\Delta x}{2} [f(x_0) + 2 \sum_{i=1}^{N-1} f(x_i) + f(x_N)]$$

$$E_i = \frac{1}{12} f''(\xi_i) \Delta_i^3 + O(\Delta_i^5)$$

Simpson's Rule:

$$p_i(x) = \frac{1}{6} (f(x_i) + 4f\left(\frac{x_i + x_{i+1}}{2}\right) + f(x_{i+1})) \Delta_i$$

$$I \approx \frac{\Delta x}{3} [f(x_0) + 4 \sum_{i=1, \text{odd}}^{N-1} f(x_i) + 2 \sum_{i=2, \text{even}}^{N-2} f(x_i) + f(x_N)]$$

$$E_i = O(\Delta_i^5)$$

→ Formel für I (alle Regeln) gilt nur falls $\Delta_i = \Delta x \neq i$ ist

Konvergenz:

→ über die ganze Domäne ist die Konvergenz eine Ordnung kleiner (3. Grad → 2. Grad)

→ Fehler um Faktor k reduzieren

$$E \sim \frac{1}{N^4}$$

→ man muss die # Intervalle um Faktor $\sqrt[4]{k}$ erhöhen

Newton-Cotes-Formeln:

→ Allgemeiner Weg um Quadratur-Regeln aufzustellen → $p_i(x)$ wird mit Hilfe des Lagrange Polynom gebildet

→ Vorgehensweise:

1. L_k berechnen

→ im Subintervall $x_i = x_i + k \cdot h$ ($k=0,1,\dots,N$)

2. Koeffizienten berechnen

$$C_k^N = \frac{1}{x_{i+1}-x_i} \int_{x_i}^{x_{i+1}} L_k(x) dx$$

$$\sum_{k=0}^N C_k^N = 1$$

$$C_k^N = C_{N-k}^N$$

3. Integral berechnen:

$$I \approx (x_{i+1}-x_i) \sum_{k=0}^N C_k^N f(x_k)$$

$N=1$: Trapezoidal

$N=2$: Simpson

$N \geq 3$: höher dimensionale Regeln

Fehleranalyse:

$$E_{\text{rule},i} = I - I_{\text{rule},i}$$

→ Taylorreihenentwicklung: (Midpoint)

$$f(x) = f(x_{i+1/2}) + (x-x_{i+1/2}) f'(x_{i+1/2}) + \frac{1}{2} (x-x_{i+1/2})^2 f''(x_{i+1/2}) + \dots$$

$$\rightarrow I_i = \int_{x_i}^{x_{i+1}} f(x) dx = f(x_{i+1/2}) \int_{x_i}^{x_{i+1}} dx + \frac{1}{2} f''(x_{i+1/2}) \int_{x_i}^{x_{i+1}} (x-x_{i+1/2})^2 dx + \dots$$

$$= f(x_{i+1/2}) \Delta_i + O(\frac{1}{24} f''(\xi_i) \Delta_i^3) + O(\Delta_i^5)$$

$$= f(x_{i+1/2}) \Delta_i + O(\frac{1}{24} f''(\xi_i) \Delta_i^3) + O(\Delta_i^5)$$

$$= f(x_{i+1/2}) \Delta_i + O(\frac{1}{24} f''(\xi_i) \Delta_i^3) + O(\Delta_i^5)$$

$$= f(x_{i+1/2}) \Delta_i + O(\frac{1}{24} f''(\xi_i) \Delta_i^3) + O(\Delta_i^5)$$

$$= f(x_{i+1/2}) \Delta_i + O(\frac{1}{24} f''(\xi_i) \Delta_i^3) + O(\Delta_i^5)$$

$$= f(x_{i+1/2}) \Delta_i + O(\frac{1}{24} f''(\xi_i) \Delta_i^3) + O(\Delta_i^5)$$

$$= f(x_{i+1/2}) \Delta_i + O(\frac{1}{24} f''(\xi_i) \Delta_i^3) + O(\Delta_i^5)$$

$$= f(x_{i+1/2}) \Delta_i + O(\frac{1}{24} f''(\xi_i) \Delta_i^3) + O(\Delta_i^5)$$

$$= f(x_{i+1/2}) \Delta_i + O(\frac{1}{24} f''(\xi_i) \Delta_i^3) + O(\Delta_i^5)$$

$$= f(x_{i+1/2}) \Delta_i + O(\frac{1}{24} f''(\xi_i) \Delta_i^3) + O(\Delta_i^5)$$

$$= f(x_{i+1/2}) \Delta_i + O(\frac{1}{24} f''(\xi_i) \Delta_i^3) + O(\Delta_i^5)$$

$$= f(x_{i+1/2}) \Delta_i + O(\frac{1}{24} f''(\xi_i) \Delta_i^3) + O(\Delta_i^5)$$

$$= f(x_{i+1/2}) \Delta_i + O(\frac{1}{24} f''(\xi_i) \Delta_i^3) + O(\Delta_i^5)$$

$$= f(x_{i+1/2}) \Delta_i + O(\frac{1}{24} f''(\xi_i) \Delta_i^3) + O(\Delta_i^5)$$

$$= f(x_{i+1/2}) \Delta_i + O(\frac{1}{24} f''(\xi_i) \Delta_i^3) + O(\Delta_i^5)$$

$$= f(x_{i+1/2}) \Delta_i + O(\frac{1}{24} f''(\xi_i) \Delta_i^3) + O(\Delta_i^5)$$

Richardson Extrapolation:

→ Verfahren, bei dem man auf Grund von unterschiedlichen Diskretisierungen die Qualität einer Näherung verbessern kann

Herleitung:

$G \approx G(h)$ → die gesuchte Größe G ist eine Funktion der Diskretisierung (Abstand)

→ Taylorreihenentwicklung:

$$G(h) = G(0) + c_1 h + c_2 h^2 + \dots$$

$$G(h/2) = G(0) + \frac{1}{2} c_1 h + \frac{1}{4} c_2 h^2 + \dots$$

$$G_1(h) = 2G(h/2) - G(h) = G(0) + \frac{1}{4} c_2 h^2 + \dots$$

$$G_2(h) = 2G_1(h/2) - G_1(h) = G(0) + \frac{1}{16} c_2 h^2 + \dots$$

$$G_3(h) = 2G_2(h/2) - G_2(h) = G(0) + \frac{1}{64} c_2 h^2 + \dots$$

$$G_4(h) = 2G_3(h/2) - G_3(h) = G(0) + \frac{1}{4096} c_2 h^2 + \dots$$

$$G_5(h) = 2G_4(h/2) - G_4(h) = G(0) + \frac{1}{262144} c_2 h^2 + \dots$$

$$G_6(h) = 2G_5(h/2) - G_5(h) = G(0) + \frac{1}{16777216} c_2 h^2 + \dots$$

$$G_7(h) = 2G_6(h/2) - G_6(h) = G(0) + \frac{1}{1073741824} c_2 h^2 + \dots$$

$$G_8(h) = 2G_7(h/2) - G_7(h) = G(0) + \frac{1}{68412761600} c_2 h^2 + \dots$$

$$G_9(h) = 2G_8(h/2) - G_8(h) = G(0) + \frac{1}{4304673177600} c_2 h^2 + \dots$$

$$G_{10}(h) = 2G_9(h/2) - G_9(h) = G(0) + \frac{1}{272882556160000} c_2 h^2 + \dots$$

$$G_{11}(h) = 2G_{10}(h/2) - G_{10}(h) = G(0) + \frac{1}{17180159040000000} c_2 h^2 + \dots$$

$$G_{12}(h) = 2G_{11}(h/2) - G_{11}(h) = G(0) + \frac{1}{1073741824000000000} c_2 h^2 + \dots$$

$$G_{13}(h) = 2G_{12}(h/2) - G_{12}(h) = G(0) + \frac{1}{67108864000000000000} c_2 h^2 + \dots$$

$$G_{14}(h) = 2G_{13}(h/2) - G_{13}(h) = G(0) + \frac{1}{4194304000000000000000} c_2 h^2 + \dots$$

$$G_{15}(h) = 2G_{14}(h/2) - G_{14}(h) = G(0) + \frac{1}{262144000000000000000000} c_2 h^2 + \dots$$

$$G_{16}(h) = 2G_{15}(h/2) - G_{15}(h) = G(0) + \frac{1}{16383040000000000000000000} c_2 h^2 + \dots$$

$$G_{17}(h) = 2G_{16}(h/2) - G_{16}(h) = G(0) + \frac{1}{1023940000000000000000000000} c_2 h^2 + \dots$$

$$G_{18}(h) = 2G_{17}(h/2) - G_{17}(h) = G(0) + \frac{1}{64000000000000000000000000000} c_2 h^2 + \dots$$

$$G_{19}(h) = 2G_{18}(h/2) - G_{18}(h) = G(0) + \frac{1}{4000000000000000000000000000000} c_2 h^2 + \dots$$

$$G_{20}(h) = 2G_{19}(h/2) - G_{19}(h) = G(0) + \frac{1}{250000000000000000000000000000000} c_2 h^2 + \dots$$

$$G_{21}(h) = 2G_{20}(h/2) - G_{20}(h) = G(0) + \frac{1}{15625000000000000000000000000000000} c_2 h^2 + \dots$$

$$G_{22}(h) = 2G_{21}(h/2) - G_{21}(h) = G(0) + \frac{1}{976562500000000000000000000000000000} c_2 h^2 + \dots$$

$$G_{23}(h) = 2G_{22}(h/2) - G_{22}(h) = G(0) + \frac{1}{6103515625000000000000000000000000000} c_2 h^2 + \dots$$

$$G_{24}(h) = 2G_{23}(h/2) - G_{23}(h) = G(0) + \frac{1}{381469796875000000000000000000000000000} c_2 h^2 + \dots$$

$$G_{25}(h) = 2G_{24}(h/2) - G_{24}(h) = G(0) + \frac{1}{23841862304687500000000000000000000000000} c_2 h^2 + \dots$$

$$G_{26}(h) = 2G_{25}(h/2) - G_{25}(h) = G(0) + \frac{1}{1490116394043046875000000000000000000000000} c_2 h^2 + \dots$$

$$G_{27}(h) = 2G_{26}(h/2) - G_{26}(h) = G(0) + \frac{1}{93132274627690430468750000000000000000000000} c_2 h^2 + \dots$$

$$G_{28}(h) = 2G_{27}(h/2) - G_{27}(h) = G(0) + \frac{1}{5820767164230652031250000000000000000000000000} c_2 h^2 + \dots$$

$$G_{29}(h) = 2G_{28}(h/2) - G_{28}(h) = G(0) + \frac{1}{363822947764415750000000000000000000000000000000} c_2 h^2 + \dots$$

$$G_{30}(h) = 2G_{29}(h/2) - G_{29}(h) = G(0) + \frac{1}{22738934235275984375000000000000000000000000000000} c_2 h^2 + \dots$$

$$G_{31}(h) = 2G_{30}(h/2) - G_{30}(h) = G(0) + \frac{1}{1421183402204749062500000000000000000000000000000000} c_2 h^2 + \dots$$

$$G_{32}(h) = 2G_{31}(h/2) - G_{31}(h) = G(0) + \frac{1}{8882396263779681250000000000000000000000000000000000} c_2 h^2 + \dots$$

$$G_{33}(h) = 2G_{32}(h/2) - G_{32}(h) = G(0) + \frac{1}{55514976648623006250000000000000000000000000000000000} c_2 h^2 + \dots$$

$$G_{34}(h) = 2G_{33}(h/2) - G_{33}(h) = G(0) + \frac{1}{346968604053893781250000000000000000000000000000000000} c_2 h^2 + \dots$$

$$G_{35}(h) = 2G_{34}(h/2) - G_{34}(h) = G(0) + \frac{1}{2168553775336836125000000000000000000000000000000000000} c_2 h^2 + \dots$$

$$G_{36}(h) = 2G_{35}(h/2) - G_{35}(h) = G(0) + \frac{1}{13553461095855225625000000000000000000000000000000000000} c_2 h^2 + \dots$$

$$G_{37}(h) = 2G_{36}(h/2) - G_{36}(h) = G(0) + \frac{1}{84709131847845156250000000000000000000000000000000000000} c_2 h^2 + \dots$$

$$G_{38}(h) = 2G_{37}(h/2) - G_{37}(h) = G(0) + \frac{1}{5294320727990322500} c_2 h^2 + \dots$$

$$G_{39}(h) = 2G_{38}(h/2) - G_{38}(h) = G(0) + \frac{1}{3308950453743951562500000000000000000000000000000000000000} c_2 h^2 + \dots$$

$$G_{40}(h) = 2G_{39}(h/2) - G_{39}(h) = G(0) + \frac{1}{206809403359000} c_2 h^2 + \dots$$

$$G_{41}(h) = 2G_{40}(h/2) - G_{40}(h) = G(0) + \frac{1}{129255877099375000} c_2 h^2 + \dots$$

$$G_{42}(h) = 2G_{41}(h/2) - G_{41}(h) = G(0) + \frac{1}{80784923187109375000} c_2 h^2 + \dots$$

$$G_{43}(h) = 2G_{42}(h/2) - G_{42}(h) = G(0) + \frac{1}{5049057449194375000} c_2 h^2 + \dots$$

$$G_{44}(h) = 2G_{43}(h/2) - G_{43}(h) = G(0) + \frac{1}{31556609057464062500} c_2 h^2 + \dots$$

$$G_{45}(h) = 2G_{44}(h/2) - G_{44}(h) = G(0) + \frac{1}{197228806609150312500} c_2 h^2 + \dots$$

$$G_{46}(h) = 2G_{45}(h/2) - G_{45}(h) = G(0) + \frac{1}{12326800413071875000} c_2 h^2 + \dots$$

$$G_{47}(h) = 2G_{46}(h/2) - G_{46}(h) = G(0) + \frac{1}{77042502581699062500} c_2 h^2 + \dots$$

$$G_{48}(h) = 2G_{47}(h/2) - G_{47}(h) = G(0) + \frac{1}{4815156411356187500} c_2 h^2 + \dots$$

$$G_{49}(h) = 2G_{48}(h/2) - G_{48}(h) = G(0) + \frac{1}{30094727570726187500} c_2 h^2 + \dots$$

$$G_{50}(h) = 2G_{49}(h/2) - G_{49}(h) = G(0) + \frac{1}{188092047317039062500} c_2 h^2 + \dots$$

$$G_{51}(h) = 2G_{50}(h/2) - G_{50}(h) = G(0) + \frac{1}{1175575295731494375000} c_2 h^2 + \dots$$

$$G_{52}(h) = 2G_{51}(h/2) - G_{51}(h) = G(0) + \frac{1}{73473455980468437500} c_2 h^2 + \dots$$

$$G_{53}(h) = 2G_{52}(h/2) - G_{52}(h) = G(0) + \frac{1}{459209101127928125000} c_2 h^2 + \dots$$

$$G_{54}(h) = 2G_{53}(h/2) - G_{53}(h) = G(0) + \frac{1}{28700569070495312500} c_2 h^2 + \dots$$

$$G_{55}(h) = 2G_{54}(h/2) - G_{54}(h) = G(0) + \frac{1}{1793785566905937500} c_2 h^2 + \dots$$

$$G_{56}(h) = 2G_{55}(h/2) - G_{55}(h) = G(0) + \frac{1}{11211159793162187500} c_2 h^2 + \dots$$

$$G_{57}(h) = 2G_{56}(h/2) - G_{56}(h) = G(0) + \frac{1}{70070723707263437500} c_2 h^2 + \dots$$

$$G_{58}(h) = 2G_{57}(h/2) - G_{57}(h) = G(0) + \frac{1}{437942023170390625000} c_2 h^2 + \dots$$

$$G_{59}(h) = 2G_{58}(h/2) - G_{58}(h) = G(0) + \frac{1}{27371376448149437500} c_2 h^2 + \dots$$

$$G_{60}(h) = 2G_{59}(h/2) - G_{59}(h) = G(0) + \frac{1}{171071102800934375000} c_2 h^2 + \dots$$

$$G_{61}(h) = 2G_{60}(h/2) - G_{60}(h) = G(0) + \frac{1}{10691944175058187500} c_2 h^2 + \dots$$

$$G_{62}(h) = 2G_{61}(h/2) - G_{61}(h) = G(0) + \frac{1}{66824651094113687500} c_2 h^2 + \dots$$

Adaptive Integration:

Idee:

Die Diskretisierung wird nur lokal verfeinert, wo sich die Funktion schnell ändert
→ man braucht weniger Funktionsevaluationen

Verfahren: (rekursiv)

Funktion AdaptiveSimpson(a,b) $\rightarrow I_S[a,b]$

→ Simpson-Regel im Intervall [a,b] anwenden

→ Intervall in Subintervall [a,m] & [m,b] anwenden, $m = (a+b)/2$

→ Simpson im Intervall [a,m] & [m,b] anwenden

→ Fehler im Intervall [a,b] mit Richardson abschätzen $E(h/2) \approx G(h/2) - G(h) \leq \epsilon$

if Genauigkeit kleiner als gewünscht then return AdaptiveSimpson(a,m) + AdaptiveSimpson(m,b)

else return $I_S[a,m] + I_S[m,b]$ (aktuelle Werte)

end if end function

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen Rechenaufwand

→ Fehler konnte um einen Grad reduziert werden bei minimalen

