Partielle Ableitung

$$\frac{\partial f}{\partial x} f\left(g\left(x\right)\right)=f'\left(g\left(x\right)\right)\cdot g'\left(x\right)$$

$$\frac{\partial f}{\partial x} f\left(x\right)\cdot g\left(x\right)=f'\left(x\right)\cdot g\left(x\right)+f\left(x\right)\cdot g'\left(x\right)$$

$$\frac{\partial f}{\partial x}\frac{f\left(x\right)}{g\left(x\right)}=\frac{f'\left(x\right)\cdot g\left(x\right)-f\left(x\right)\cdot g'\left(x\right)}{g(x)^2}$$

$$\log\left(x\right)'=\frac{1}{x}$$

$$\left(e^{5x}\right)'=5\cdot e^{5x}$$

Inverse

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1}=\frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

$$\left(\begin{array}{ccc|ccc} a & b & c & 1 & 0 & 0 \\ d & e & f & 0 & 1 & 0 \\ g & h & i & 0 & 0 & 1 \end{array}\right)$$

Linearisierung

Gegeben: $y=\begin{pmatrix} f_1\left(x\right) \\ \dots \\ f_n\left(x\right) \end{pmatrix}$ und $x=\begin{pmatrix} x_1 \\ \dots \\ x_n \end{pmatrix}$

Jakobi Matrix: $Df=\begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \dots & \dots & \dots \\ \frac{\partial f_m}{\partial x_1} & \dots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$

Linearisierung: $g\left(x\right)=f\left(\vec{x}_0\right)+Df\left(\vec{x}_0\right)\cdot\left(\vec{x}-\vec{x}_0\right)$

Newton Verfahren für nichtlineare Gleichungssysteme

(Es gilt: f(x) = vec 0)

Basic: x1 = x0 - f(x) / f'(x)

Ungedämpft:

1. $Df\left(\vec{x}^{(k)}\right)\cdot\vec{\delta}^{(n)}=-\vec{f}\left(\vec{x}^{(k)}\right)$ nach $\vec{\delta}$ auflösen.
2. $\vec{x}^{(k+1)}=\vec{x}^{(k)}+\vec{\delta}^{(k)}$

Vereinfacht (konvergiert linear statt quadratisch):

1. $Df\left(\vec{x}^{(0)}\right)\cdot\vec{\delta}^{(k)}=-\vec{f}\left(\vec{x}^{(k)}\right)$ nach $\vec{\delta}$ auflösen.
2. $\vec{x}^{(k+1)}=\vec{x}^{(k)}+\vec{\delta}^{(k)}$

Abbruchsbedinungen:

- $\left\|x^{(n+1)}-x^{(n)}\right\|\leqslant\varepsilon$
- $\left\|f\left(x^{(n+1)}\right)\right\|\leqslant\varepsilon$

Gedämpft:

1. $Df\left(\vec{x}^{(k)}\right) \cdot \vec{\delta}^{(n)} = -\vec{f}\left(\vec{x}^{(k)}\right)$ nach $\vec{\delta}$ auflösen.
2. Finde minimales $k \in \{0, 1, \ldots, k_{\max}\}$ für welches gilt: $\left\| f\left(x^{(n)} + \frac{\delta^{(n)}}{2^k}\right) \right\|_2 < \left\| f\left(x^{(n)}\right) \right\|_2$
3. Sonst $k = 0$
4. $x^{(n+1)} := x^{(n)} + \frac{\delta^{(n)}}{2^k}$

Auflösen:

$\vec{\delta} = -Df^{-1}\left(\vec{x}^{(k)}\right) f\left(\vec{x}^{(k)}\right)$

Polynominterpolation mit Vandermonde

$$\begin{pmatrix} a_0 \\ \ldots \\ a_n \end{pmatrix} = \begin{pmatrix} 1 & x_0 & \ldots & x_0^n \\ \ldots & \ldots & \ldots & \ldots \\ 1 & x_n & \ldots & x_n^n \end{pmatrix}^{-1} \begin{pmatrix} y_0 \\ \ldots \\ y_n \end{pmatrix}$$

$P_n(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$

Grad n bei n+1 Stützpunken.

Überstimmt = Mehr Daten als Grad des Polynoms

Lagrange Polynominterpolationsformel bei n+2 Stützpunkten

$P_{n+1}(x) = \Sigma_{i=0}^{n} l_i(x) \cdot y_i$

l_i sind die n+1 Lagrangepolynome:

$l_i(x) = \Pi_{j=0}^{n} \frac{x - x_j}{x_i - x_j}$ wobei $i \neq j$

Fehlerabschätzung:

$|f(x) - P_n(x)| \leqslant \frac{|(x - x_0) \cdot \ldots \cdot (x - x_n)|}{(n+1)!} \cdot \max_{z \epsilon [x_0, x_n]} \left| f^{(n+1)}(z) \right|$

Spline-Interpolation

Interpolation der Stützpunkte:

$$\begin{aligned} S_0(x_0) &= y_0 \\ &\ldots \\ S_n(x_n) &= y_n \end{aligned}$$

Stetiger Übergang):

$$\begin{aligned} S_0(x_1) &= S_1(x_1) \\ &\ldots \\ S_{n-1}(x_{n-1}) &= S_n(x_{n-1}) \end{aligned}$$

Erste Ableitung an der Übergängen:

$$\begin{aligned} S_0'(x_1) &= S_1'(x_1) \\ &\ldots \\ S_{n-1}'(x_n) &= S_n'(x_n) \end{aligned}$$

Zweite Ableitung:

$$\begin{aligned} S_0''(x_1) &= S_1''(x_1) \\ &\ldots \\ S_{n-1}''(x_n) &= S_n''(x_n) \end{aligned}$$

Natürliche Splines:

$S_0''(x_0) = 0$

$S_2''(x_3) = 0$

Periodische Splines:

$S_0'(x_0) = S_2'(x_3)$

$S_0''(x_0) = S_2''(x_3)$

not-a-knot Splines:

$S_0'''(x_1) = S_1'''(x_1)$

$S_1'''(x_2) = S_2'''(x_2)$

Lösen des Gleichungssystems:

$(a_0, \ldots, a_n, b_0, \ldots, b_n, \ldots)^T = A^{-1}\vec{y}$

$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 \ldots$

Mit h: $S_i(x_i) = y_i$ ... $S_n'(x) = b_n + 2c_n(x - x_n) + 3d_n(x - x_n)^2$ $S_n''(x) = 2c_n + 6d_n(x - x_n)$

Algorithmus: Natürliche kubische Splinefunktion

n = Anzahl Splines. i bis n-1. n = 3 → A ist 2×2

$a_i = y_i$, $h_i = x_{i+1} - x_i$

$c_0 = 0$, $c_n = 0$

für $c_1 - c_{n-1}$!!! lösen von $Ac = z$

$$A = \begin{pmatrix} 2(h_0 + h_1) & h_1 & 0 & 0 & 0 \\ h_1 & 2(h_1 + h_2) & h_2 & 0 & 0 \\ 0 & h_2 & 2(h_2 + h_3) & h_3 & 0 \\ 0 & \ldots & \ldots & \ldots & 0 \\ 0 & 0 & h_{n-3} & 2(h_{n-3} + h_{n-2}) & h_{n-2} \\ 0 & 0 & 0 & h_{n-2} & 2(h_{n-2} + h_{n-1}) \end{pmatrix}$$

$$c = \begin{pmatrix} c_1 \\ \ldots \\ c_{n-1} \end{pmatrix}$$

$$z = \begin{pmatrix} 3\frac{y_2 - y_1}{h_1} - 3\frac{y_1 - y_0}{h_0} \\ 3\frac{y_3 - y_2}{h_2} - 3\frac{y_2 - y_1}{h_1} \\ \ldots \\ 3\frac{y_n - y_{n-1}}{h_{n-1}} - 3\frac{y_{n-1} - y_{n-2}}{h_{n-2}} \end{pmatrix}$$

$$b_i = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{3}(c_{i+1} + 2c_i)$$

$$d_i = \frac{1}{3h_i}(c_{i+1} - c_i)$$

$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 \ldots$

Lineares Ausgleichssystem, Normalengleichungssystem

$A^T A\vec{\lambda} = A^T\vec{y}$

$f(x) = \lambda_0 \cdot f_0(x) + \ldots + \lambda_m + f_m(x)$

$$A = \begin{pmatrix} f_0(x_0) & \ldots & f_m(x_0) \\ \ldots & \ldots & \ldots \\ f_0(x_n) & \ldots & f_m(x_n) \end{pmatrix}$$

Fehlerfunktional:

$\|y - f(x)\|_2^2$ bzw $\left\|y - A\vec{\lambda}\right\|_2^2$

Allgemeines Ausgleichsproblem, Nichtlinear, Quadratmittelproblem

Von Hand:

$$g(\lambda) = \begin{pmatrix} y_0 - f(x_0) \\ \ldots \\ y_n - f(x_n) \end{pmatrix}$$

$E(f) = g(\lambda)^2$

$\|E(f)\|_2 \to \min \implies \dfrac{\partial E(f)}{\partial \lambda_i} = 0$

Mit Jakobi-Matrix / Newton Verfahren:

$$g(\lambda) = \begin{pmatrix} y_0 - f(x_0) \\ \ldots \\ y_n - f(x_n) \end{pmatrix}$$

$$Dg\left(\lambda^{(0)}\right) = \begin{pmatrix} \frac{\partial g_0}{\partial \lambda_0} & \ldots & \frac{\partial g_0}{\partial \lambda_n} \\ \ldots & \ldots & \ldots \\ \frac{\partial g_m}{\partial \lambda_0} & \ldots & \frac{\partial g_m}{\partial \lambda_n} \end{pmatrix}$$

$Dg\left(\lambda^{(n)}\right)^T Dg\left(\lambda^{(n)}\right)\vec{\delta} = \left(-Dg\left(\lambda^{(n)}\right)\right)^T g\left(\lambda^{(n)}\right)$

$\lambda^{(n+1)} = \lambda^{(n)} + \vec{\delta}$

QR: $R\delta = -Q^T g\left(\lambda^{(0)}\right)$

Numerisches Integrieren Quadraturformeln

Rechteck:

$Rf = (b-a) \cdot f\left(\dfrac{b+a}{2}\right)$

$Rf(h) = h \cdot \Sigma_{i=0}^{n-1} f\left(x_i + \dfrac{h}{2}\right)$

Trapez:

$T(f) = (b-a) \cdot \dfrac{f(a) + f(b)}{2}$

$Tf(h) = h \cdot \left(\dfrac{f(a) + f(b)}{2} + \Sigma_{i=1}^{n-1} f(x_i)\right)$

Simpson:

$Sf = \dfrac{b-a}{6}\left(f(a) + 4f\left(\dfrac{a+b}{2}\right) + f(b)\right)$

$Sf(h) = \dfrac{h}{3}\left(\dfrac{1}{2} f(a) + \Sigma_{i=1}^{n-1} f(x_i) + 2\Sigma_{i=1}^{n} f\left(\dfrac{x_{i-1} + x_i}{2}\right) + \dfrac{1}{2} f(b)\right)$

Summiert: Der Bereich wird in n gleich grosse Abschnitte der Länge $h = \frac{b-a}{n}$ unterteilt. $x_i = a + i \cdot h$

Fehlerabschätzung Quadraturformeln

$$\left|\int_a^b f(x)\mathrm{d}x - Rf(h)\right| \leqslant \frac{h^2}{24}(b-a) \cdot \max|f''(x)|$$

$$\left|\int_a^b f(x)\mathrm{d}x - Tf(h)\right| \leqslant \frac{h^2}{12}(b-a) \cdot \max|f''(x)|$$

$$\left|\int_a^b f(x)\mathrm{d}x - Sf(h)\right| \leqslant \frac{h^4}{2880}(b-a) \cdot \max\left|f^{(4)}(x)\right|$$

Bei R&T ist dies exakt für Polynom vom Grad 1, S für Polynom vom Grad 3. /24 macht R doppelt so gut wie /12 T

Gauss Formeln

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

$$n = 2 : \frac{b-a}{2}\left[f\left(-\frac{1}{\sqrt{3}} \cdot \frac{b-a}{2} + \frac{b+a}{2}\right) + f\left(\frac{1}{\sqrt{3}} \cdot \frac{b-1}{2} + \frac{b+a}{2}\right)\right]$$

$$n = 3 : \frac{b-1}{2}\left[\frac{5}{9} \cdot f\left(-\sqrt{0.6} \cdot \frac{b-a}{2} + \frac{b+a}{2}\right) + \frac{9}{8} \cdot f\left(\frac{b+a}{2}\right)\right] + \frac{b-a}{2} \cdot \left[\frac{5}{9} \cdot f\left(\sqrt{0.6} \cdot \frac{b-a}{2} + \frac{b+a}{2}\right)\right]$$

Romberg-Extrapolation

$$\int_a^b$$

$$h_j = \frac{b-a}{2^j}, n = 2^j, k = 0 \ldots$$

$$T_{j,0} = \frac{b-a}{2^j} \cdot \left(\frac{f(a) + f(b)}{2} + \Sigma_{i=1}^{n-1} f\left(a + i \cdot \frac{b-a}{2^j}\right)\right)$$

$$T_{j,k} = \frac{4^k \cdot T_{j+1,k-1} - T_{j,k-1}}{4^k - 1}$$

DGL Separierung der Variablen

$$\frac{\mathrm{d}y}{\mathrm{d}x} = x^2 y$$

$$\mathrm{d}x \cdot x^2 = \frac{\mathrm{d}y}{y}$$

$$\int x^2 \mathrm{d}x = \int \frac{1}{y}\mathrm{d}y$$

Klassisches Euler-Verfahren

$\frac{\mathrm{d}y}{\mathrm{d}x} = f(x, y(x))$ mit $y(x_0) = y_0$

$$x_{i+1} = x_i + h$$

$$y_{i+1} = y_i + h \cdot f(x_i, y_i)$$

Mittelpunktverfahren

$$k_1 = f(x_k, y_k)$$

$k_2 = f\left(x_k + \frac{h}{2}, y_k + \frac{h}{2}k_1\right)$

$x_{k+1} = x_k + h$

$y_{k+1} = y_k + h \cdot k_2$

Modifiziertes Euler-Verfahren

$k_1 = f(x_k, y_k)$

$k_2 = f(x_k + h, y_k + h \cdot k_1)$

$x_{k+1} = x_i + h$

$y_{k+1} = y_i + \frac{h}{2}(k_1 + k_2)$

Fehlerordnung

kleines h: gösserer Rundungsfehler, kleinerer Diskretisierungsfehler

Lokaler Fehler der Ordnung p = $|y(x_{i+1}) - y_{i+1}| \leqslant C \cdot h^{p+1}$

Globaler Fehler der Ordnung p = $|y(x_{x_n}) - y_n| \leqslant C \cdot h^p$

Euler-Verfahren: Lokal = $\frac{h^2}{2}y''(z)$ Global = $\frac{h}{2}\max|y''(x)| \cdot \tilde{C}$ → P = 1

Mittelpunkt und Modifiziert: P = 2

Klassisches vierstufiges Runge-Kutta-Verfahren

$k_1 = f(x_k, y_k)$

$k_2 = f\left(x_k + \frac{h}{2}, y_k + \frac{h}{2}k_1\right)$

$k_3 = f\left(x_k + \frac{h}{2}, y_k + \frac{h}{2}k_2\right)$

$k_4 = f(x_k + h, y_k + hk_3)$

$x_{k+1} = x_i + h$

$y_{k+1} = y_k + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$

Allgemeines s-stufiges Runge-Kutta-Verfahren

$k_1 = f(x_k, y_k)$

$k_2 = f(x_k + c_2 \cdot h, y_k + h \cdot (a_{21} \cdot k_1))$

$k_3 = f(x_k + c_3 \cdot h, y_k + h \cdot (a_{31} \cdot k_1 + a_{32} \cdot k_2))$

$\ldots$

$y_{k+1} = y_k + h \cdot (b_1 k_1 + b_2 k_2 \ldots)$

Butcher Schema

Linke Seite: $c_1 - c_s$ = Skalar von h von k1 bis ks

Unten: $b_1 - b_s$ = Kombination von allen k für die Berechnung der Steigung

Mitte: $a_{2,1} - a_{s,s-1}$ = Kombination von vorherigen k in nächstem k

Startet mit 0 | 0 oder bei a_{21}

DGL n-ter Ordnung zu 1. Ordnung

$x''' = 2x'' + 3x'$

$x''' = 2z_3 + 3z_2 + 5$

$$\begin{pmatrix} z_1' \\ z_2' \\ z_3' \end{pmatrix} = \begin{pmatrix} z_2 \\ z_3 \\ 2z_3 + 3z_2 + 5 \end{pmatrix}$$

$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 3 & 2 \end{pmatrix}$, $b = \begin{pmatrix} 0 \\ 0 \\ 5 \end{pmatrix}$

$z' = Az + b$

Stabilität

Gegeben: f' ist monoton sinkend

Verfahren (hier Euler): $y_{i+1} = y + h \cdot f'(y_i)$

Da monoton sinkend, gilt: $y_i > y_i + h \cdot f'(y_i)$

$(*)$ Zwischenergebnis: $\ldots < 1$ (y_i wurde auf die gleiche Seite getan)

Ergebnis: $h < \ldots$

Stabilitätsintervall:

Wenn $y' = -\alpha y$ als $y_{i+1} = g(h\alpha) \cdot y_i$ geschrieben werden kann, dann ist das Verfahren für alle Alpha stabil, wo $|g(x)| < 1$ gilt. Diese Stabilitätsfunktion berechnet man bei $(*)$

Eulerverfahren: 1 + x

s-stufiges Runge Kutta: Polynom von Grad s