Interpolation

Motivation:

- Zahl der Meßpunkte ist gering
- Aufwand zur Berechnung eines Punktes ist sehr hoch
- Numerische Methode verlangt ein bestimmtes Gitter

gesucht:

- Funktion $f(x)$ (eventuell geschlossener Ausdruck), die als sinnvolle Näherung an die tatsächliche Funktion betrachtet werden kann *und durch die Datenpunkte (Stützstellen) $f(x_i)$ geht*.

Unterscheidung:

Lineare Interpolationsprobleme:

$$f(x;a_0,a_1,\ldots,a_n) = a_0\phi_0 + a_1\phi_1 + \ldots + a_n\phi_n$$

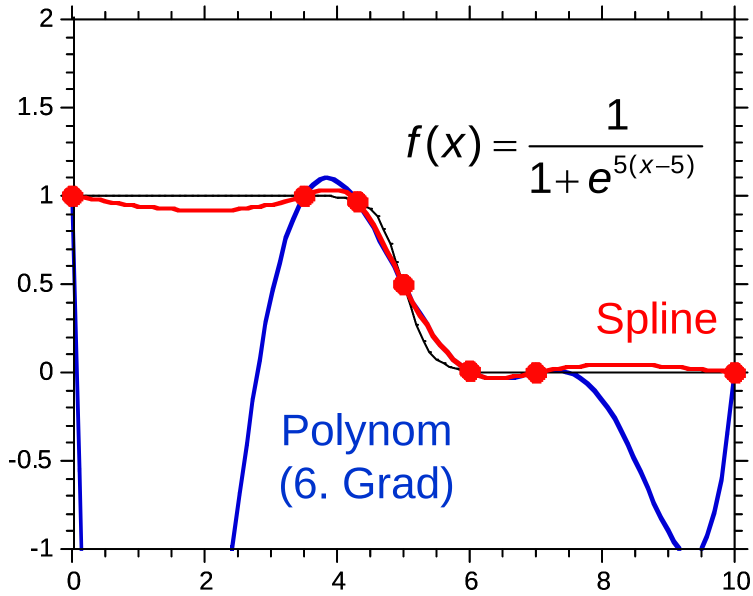
Polynom-Interpolation, Spline-Interpolation

Nichtlineare Interpolationsprobleme:

rationale Funktionen, Exponentialsummen

Interpolation I: Splines

Beispiel:



spline: Kurvenbrett, Kurvenlineal

Idee aus Statik:

Definition der Norm einer Spline-Funktion $s(x)$:

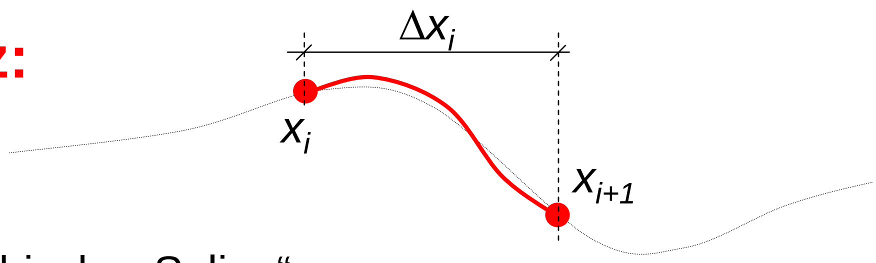
$$\|s(x)\|^2 \equiv \int_{x_0}^{x_N} |s''(x)|^2 \, dx$$

Es gilt die Minimum-Norm-Eigenschaft

$$\|f - s\|^2$$

falls 1) $s''(x_0) = s''(x_N) = 0$
2) $s(x)$, $f(x)$ periodisch sind
3) $f'(x_0) = s'(x_0)$ und $f'(x_N) = s'(x_N)$

Ansatz:



„kubischer Spline“

$$s_i(x) = a_i(x-x_i)^3 + b_i(x-x_i)^2 + c_i(x-x_i) + d_i$$

Grenzen des Intervalls:

$$s_i(x_i) = f(x_i) = d_i$$

$$s_i(x_{i+1}) = f(x_{i+1}) = a_i \Delta x_i^3 + b_i \Delta x_i^2 + c_i \Delta x_i + d_i$$

2. Ableitung an Intervallgrenzen:

$$s_i''(x_i) = M_i = 2b_i$$

$$s_i''(x_{i+1}) = M_{i+1} = 6a_i \Delta x_i + 2b_i$$

Bestimmung der M_i:

insgesamt N+1 Werte gesucht, N-1 Gleichungen aus der Stetigkeit an den inneren Intervallgrenzen;
verbleibende Gleichungen aus Zusatzbedingungen:

1) verschwindende zweite Ableitung an den Enden: $M_0 = M_N = 0$
   → Tridiagonal-Matrix mit Dimension *N-1*
2) Erste Ableitung an den Enden bekannt → 2 zusätzliche Gleichungen → Tridiagonal-Matrix mit Dimension *N+1*
3) Periodische Randbedingung:

$$\begin{pmatrix} 2 & \lambda_1 & 0 & \dots & 0 & 0 & \mu_1 \\ \mu_2 & 2 & \lambda_2 & \dots & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & & 0 \\ 0 & 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & & \mu_{N-1} & 2 & \lambda_{N-1} \\ \lambda_N & 0 & 0 & \dots & 0 & \mu_N & 2 \end{pmatrix} \begin{pmatrix} M_1 \\ M_2 \\ M_3 \\ \vdots \\ \vdots \\ M_{N-1} \\ M_N \end{pmatrix} = \begin{pmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \vdots \\ \vdots \\ \delta_{N-1} \\ \delta_N \end{pmatrix}$$

Bestimmungsgleichungen für M_i:

$$\mu_i M_{i-1} + 2M_i + \lambda_i M_{i+1} = \delta_i$$

$$\mu_i = \frac{\Delta x_{i-1}}{\Delta x_{i-1} + \Delta x_i} = 1 - \lambda_i \qquad \lambda_i = \frac{\Delta x_i}{\Delta x_{i-1} + \Delta x_i} = 1 - \mu_i$$

$$\delta_i = \frac{6}{\Delta x_{i-1} + \Delta x_i} \cdot \left(\frac{f(x_{i+1}) - f(x_i)}{\Delta x_i} - \frac{f(x_i) - f(x_{i-1})}{\Delta x_{i-1}} \right)$$

Rekonstruktion von s_i:

$$s_i(x) = a_i (x - x_i)^3 + b_i (x - x_i)^2 + c_i (x - x_i) + d_i$$

$$a_i = \frac{1}{6\Delta x_i}\left(M_{i+1} - M_i\right) \qquad c_i = \frac{f(x_{i+1}) - f(x_i)}{\Delta x_i} - \Delta x_i \cdot \frac{2M_i + M_{i+1}}{6}$$

$$b_i = \frac{1}{2} M_i \qquad d_i = f(x_i)$$

Interpolation II: Polynominterpolation

Interpolationsfunktion aus der Menge der reellen oder komplexen Polynome vom Grad N:

$\phi(x; a_0; a_1; : : : ; a_N) = p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_N x^N$

Das Interpolationsproblem $(x_i, f(x_i)),\ i = 0,1,\ldots,N$ ist mit einem Polynom N-ten Grades **eindeutig** lösbar und das Interpolationspolynom **existiert**.

Lagrange Interpolationsformel:

$$p(x) = \sum_{i=0}^{N} f(x_i) L_i(x) = \sum_{i=0}^{N} f(x_i) \prod_{\substack{k=0 \\ k \neq i}}^{N} \frac{x - x_k}{x_i - x_k}$$

Das Verfahren von Neville

Rekursionsvorschrift zur Ermittlung eines Interpolationswertes

	$k=0$	$k=1$	$k=2$	$k=3$
x_0	$f_0 = P_0(x)$			
		$P_{01}(x)$		
x_1	$f_1 = P_1(x)$		$P_{012}(x)$	
		$P_{12}(x)$		$P_{0123}(x)$
x_2	$f_2 = P_2(x)$		$P_{123}(x)$	
		$P_{23}(x)$		
x_3	$f_3 = P_3(x)$			

$$P_i(x) = f(x_i)$$

$$P_{i,i+1}(x) = \frac{(x-x_i)P_{i+1}(x) - (x-x_{i+1})P_i(x)}{(x_{i+1}-x_i)}$$

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

$$\vdots$$

$$p_N(x) = P_{0,\ldots,N}(x) = \frac{(x-x_0)P_{1,N}(x) - (x-x_N)P_{0,N-1}(x)}{(x_N-x_0)}$$

Newtonsche Dividierte Differenzen

falls Polynom oder viele Interpolationswerte benötigt werden

Rekursionsvorschrift:

Rekursionsanfang: $f[x_i] = f(x_i); \quad f[x_0] = a_0$

"dividierte Differenz": $f[x_i, x_{i+1}] = \dfrac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i} = f[x_{i+1}, x_i]$

$$\vdots$$

$$f[x_0,\ldots,x_i] = \frac{f[x_1,\ldots,x_i] - f[x_0,\ldots,x_{i-1}]}{x_i - x_0} = a_i$$

$$\boxed{\rightarrow \quad p_N(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \ldots + a_N(x-x_0)\ldots(x-x_{N-1})}$$

Beispiel:

x_i	f_i	$f[x_i, x_{i+1}]$	$f[x_i, \ldots, x_{i+2}]$	$f[x_i, \ldots, x_{i+3}]$	$f[x_i, \ldots, x_{i+4}]$
3.2	22.0				
		8.400			
2.7	17.8		2.856		
		2.118		-0.5280	
1.0	14.2		2.012		0.256
		6.342		0.0865	
4.8	38.3		2.263		
		16.750			
5.6	51.7				

$$\rightarrow \quad p_4(x) = 22 + 8.400(x - x_0) + 2.856(x - x_0)(x - x_1) - 0.528(x - x_0)(x - x_1)(x - x_2) + 0.256(x - x_0)(x - x_1)(x - x_2)(x - x_3)$$

Aufgaben:

1. Berechnen Sie die natürliche Spline-Funktion durch die Datenpunkte im file `/home/EDV2/Unterlagen/data` unter Verwendung einer Bibliotheksfunktion aus CERNLIB (FORTRAN) oder GSL (C).
2. Interpolieren Sie die selben Datenpunkte mithilfe der Methode der dividierten Differenzen.
3. Stellen Sie Ihre Ergebnisse in `gnuplot` zusammen mit der Ursprungsfunktion $f(x) = \sin(x)/x$ dar.