

3.10 Regularisierung

In vielen praktischen Anwendungen hat man zwar ein überbestimmtes lineares Gleichungssystem vorliegen, aber so, dass die Normalmatrix $\mathbf{A}^T\mathbf{A}$ auch noch (fast) singular ist!

Dadurch erhält man bei der Lösung dieses Problems einen Vektor \mathbf{x} , der extrem groß ist:

Ist in $\mathbf{B}\mathbf{x}=\mathbf{b}$ die Matrix \mathbf{B} (fast) singular \rightarrow

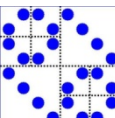
$\rightarrow \|\text{inv}(\mathbf{B})\|$ sehr groß \rightarrow

$\rightarrow \|\mathbf{x}\| = \|\text{inv}(\mathbf{B})\cdot\mathbf{b}\|$ sehr groß

Durch Mess/Rundungsfehler enthält aber die rechte Seite b viele kleine Störungen (noise, Rauschen), die in der berechneten Näherungslösung x dann sehr groß werden, so dass - selbst bei exakter Rechnung - x unbrauchbar ist.

$$\tilde{x} = B^{-1}(b + \Delta b) = B^{-1}b + B^{-1}\Delta b = x + \underbrace{B^{-1}\Delta b}$$

Störanteil
Viel größer als x



Ausweg:

Suche ‚vernünftige‘ Least Squares Lösung durch Minimierung mit Nebenbedingung:
 ‚ x soll nicht zu groß werden‘.

$$\min_x \left(\|Ax - b\|_2^2 + \gamma^2 \|x\|_2^2 \right)$$

Minimierung \leftrightarrow Nullstelle der Ableitung führt auf das sog. regularisierte Gleichungssystem

$$\left(A^T A + \gamma^2 I \right) x = A^T b$$

Idee: Verschiebe $A^T A$ durch Aufaddieren von $\gamma^2 I$, so dass die neue Matrix besser konditioniert ist.

Dann ist $\|inv(A^T A + \gamma^2 I)\|_2 \ll \|inv(A^T A)\|_2$

Daher führen in dem neuen Gleichungssystem die Rauschkomponenten in b nicht mehr zu einem extremen Anwachsen der Lösung x .

Man weiß, dass die gesuchte Lösung x nicht zu groß sein kann, und dies wird durch die Regularisierung gewährleistet.

γ heißt Regularisierungsparameter und die hier beschriebene Methode heißt
Tikhonov-Regularisierung.

Regularisierung muss häufig angewendet werden bei Problemen der Bildverarbeitung

(z.B. bei verrauschten, unscharfen Bildern)

- Gauss-Elimination,
- Normalengleichung und QR-Zerlegung,
- Regularisierung

sind die wichtigsten Werkzeuge zur direkten Lösung von $Ax=b$.

IV. Interpolation und Quadratur

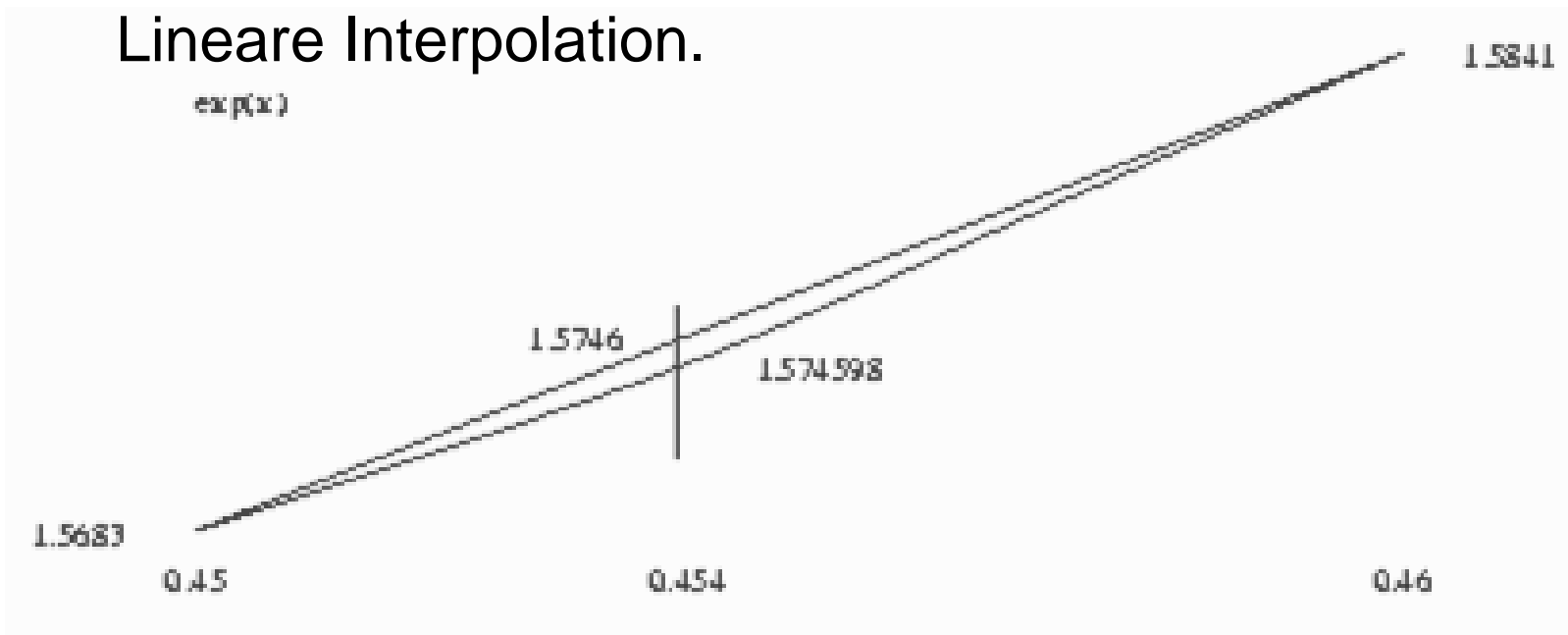
4.1. Interpolation

4.1.1. Beispiel: Interpolation mit Tafelwerken für exp, sin, cos, log

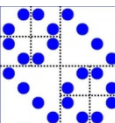
Gesucht: $\exp(0.454)$;

Tabelliert: $\exp(0.45)$ und $\exp(0.46)$

Lineare Interpolation.



x:	0.42	0.43	0.44	0.45	0.46
exp(x):	1.5683	1.5841



4.1.2. Allgemeine Problemstellung:

Gegeben:

Punktepaare (x_j, y_j) , $j=0,1,\dots,n$, paarweise verschieden
und linear unabhängige Funktionen $g_k(x)$, $k=0,1,\dots,n$

Gesucht: Koeffizienten c_k , $k=0,1,\dots,n$ mit

$$G(x_j) = \sum_{k=0}^n c_k g_k(x_j) = y_j \quad \text{für } j=0,1,\dots,n$$

$$\begin{pmatrix} g_0(x_0) & \cdots & g_n(x_0) \\ \vdots & & \vdots \\ g_0(x_n) & \cdots & g_n(x_n) \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ \vdots \\ y_n \end{pmatrix}$$

$(n+1) \times (n+1)$ lineares Gleichungssystem

Interpolation führt auf quadratisches Gleichungssystem

Man unterscheide Interpolation und Approximation!

Beispiel zu Approximation:

Ausgleichsgerade führt auf überbestimmtes Gleichungssystem

(Normalgleichung, QR)

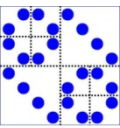
4.1.3. Spezialfall Polynom-Interpolation:

Ansatzfunktionen $g_k(x)$ sind Polynome x^k
 Gesucht: Koeffizienten c_k , $k=0,1,\dots,n$ mit

$$p(x_j) = \sum_{k=0}^n c_k x_j^k = y_j \quad \text{für } j=0,1,\dots,n$$

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \cdots & x_n^n \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ \vdots \\ y_n \end{pmatrix}$$

$n+1$ Gleichungen für $n+1$ Unbekannte: Teuer! $O(n^3)$



Definiere geschickt Basis-Polynome:

$$L_j(x) := \prod_{i=0, i \neq j}^n \frac{x - x_i}{x_j - x_i} = \frac{(x - x_0) \dots (x - x_{j-1})(x - x_{j+1}) \dots (x - x_n)}{(x_j - x_0) \dots (x_j - x_{j-1})(x_j - x_{j+1}) \dots (x_j - x_n)}$$

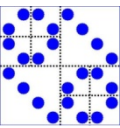
$n+1$ Polynome vom Grad n , besser als $1, x, x^2, \dots, x^n$

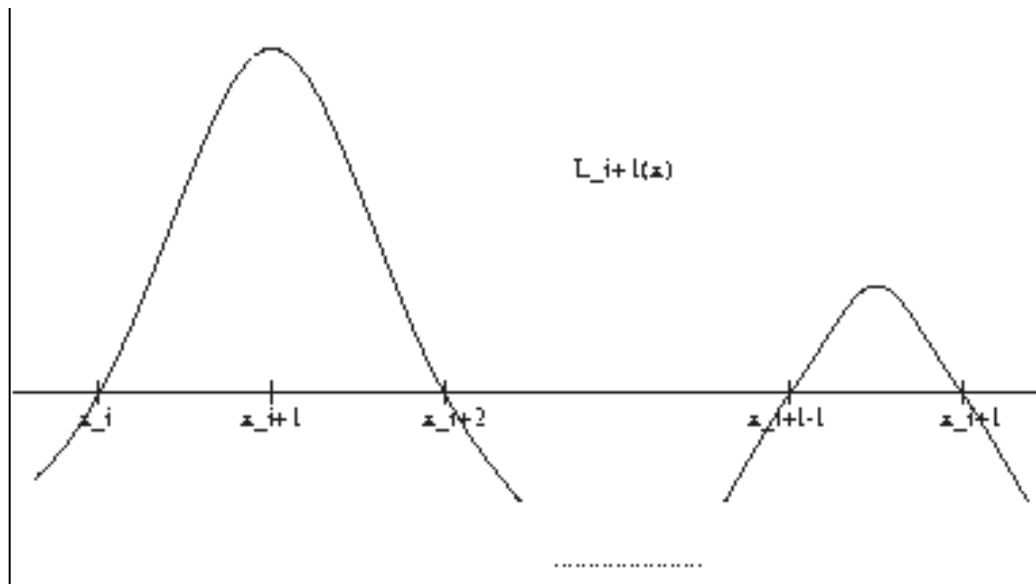
Eigenschaften der Lagrange-Polynome: Grad n mit

$$L_j(x_i) = \begin{cases} 1 & \text{falls } i = j \\ 0 & \text{falls } i \neq j \end{cases}$$

Gesucht $p(x) = \sum_{j=0}^n c_j L_j(x)$

das die Interpolationsbedingungen erfüllt.





Aus diesen Eigenschaften ergibt sich zur Lösung des Interpolationsproblems ein lineares Gleichungssystem für die Koeffizienten:

$$\begin{pmatrix} L_0(x_0) & \cdots & L_n(x_0) \\ \vdots & & \vdots \\ L_0(x_n) & \cdots & L_n(x_n) \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ \vdots \\ y_n \end{pmatrix}$$

Die Lösung ist $c_j = y_j$; daher ist das Interpolationspolynom:

$$p(x) = \sum_{j=0}^n y_j L_j(x)$$

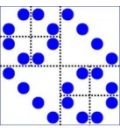
denn es ist $p(x_i) = \sum_{j=0}^n y_j L_j(x_i) = y_i L_i(x_i) = y_i$

Damit ist die Existenz eines interpolierenden Polynoms gezeigt!
Eindeutigkeit?

Hauptsatz der Algebra:

Jedes Polynom $p(x)$ vom Grad n kann als Produkt von n linearen Faktoren (den ev. komplexen Nullstellen z_k) geschrieben werden in der Form

$$p(x) = \alpha(x - z_1)(x - z_2) \dots (x - z_n)$$



Annahme: Es gibt zwei Polynome p und q vom Grad $\leq n$,
die beide die Interpolationsbedingungen erfüllen.

Definiere neues Polynom $h(x) := p(x) - q(x)$. Dann gilt

$$h(x) = \alpha(x - z_1)(x - z_2)\dots(x - z_n)$$

und $h(x_j) := p(x_j) - q(x_j) = 0$ für $j=0,1,\dots,n$

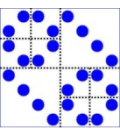
Daher hat das Polynom $h(x)$ den Grad n und $n+1$ Nullstellen.

Aus dem Hauptsatz der Algebra folgt daher, dass

$\alpha = 0$ sein muss, und daher ist $h(x) \equiv 0$, oder
 $p(x) \equiv q(x)$.

Also es existiert genau ein Interpolationspolynom!

Lagrange zur Lösung der Interpolation nicht geeignet,
da numerisch problematisch und teuer.

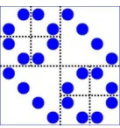


Löse nicht das lineare Gleichungssystem, da dies zu teuer ist!
 Außerdem wird oft nur der Wert des Polynoms an einer einzigen Stelle gesucht!

Idee: Berechne induktiv interpolierende Polynome für immer mehr Stützstellen.

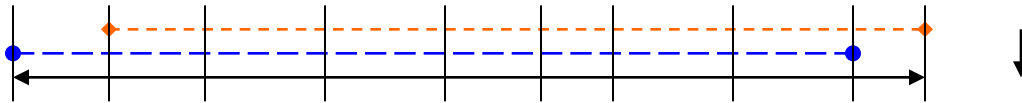
Setze dazu $p_{i,\dots,i+l}(x)$ als das interpolierende Polynom vom Grade l , das genau an den Stellen $x_i, x_{i+1}, \dots, x_{i+l}$ die Interpolations-Bedingungen erfüllt.

Zur Bestimmung von $p_{i,\dots,i+l}(x)$ verwende die interpolierenden Polynome vom Grade $l-1$ $p_{i+1,\dots,i+l}(x)$ und $p_{i,\dots,i+l-1}(x)$, die zu den Stützstellen x_{i+1}, \dots, x_{i+l} , bzw. x_i, \dots, x_{i+l-1} , gehören.



Wesentliche Formel :

$$p_{i,\dots,i+l}(x) = \frac{(x - x_i)p_{i+1,\dots,i+l}(x) - (x - x_{i+l})p_{i,\dots,i+l-1}(x)}{x_{i+l} - x_i} \quad (*)$$



Beweis: Nachprüfen der Interpolationsbedingung.

$$p_{i,\dots,i+l}(x_i) = \frac{0 - (x_i - x_{i+l})y_i}{x_{i+l} - x_i} = y_i$$

$$p_{i,\dots,i+l}(x_{i+l}) = \frac{(x_{i+l} - x_i)y_{i+l} - 0}{x_{i+l} - x_i} = y_{i+l}$$

und für alle anderen j mit $i < j < i+l$:

$$p_{i,\dots,i+l}(x_j) = \frac{(x_j - x_i)y_j - (x_j - x_{i+l})y_j}{x_{i+l} - x_i} = y_j$$

Wegen der Eindeutigkeit des interpolierenden Polynoms ist jedes der so definierten Polynome genau die eindeutige Lösung des jeweiligen Interpolationsproblems!

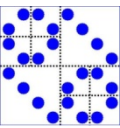
Daher ist $p_{i,\dots,i+l}(x)$ die gesuchte Lösung!

Anwendung der Formel (*) zur punktweisen Auswertung des Interpolationspolynoms an einer Stelle \bar{x} : $p(\bar{x}) = ?$

Eingabe: Stützwerte (x_j, y_j) , $j=0,\dots,n$ und Stelle \bar{x} ;

Ausgabe: $p_{i,\dots,i+l}(\bar{x})$

Tableau-artige Berechnung der Interpolationspolynome aufsteigenden Grades, aber nur an der Stelle \bar{x} .



Neville-Tableau:

Grad	0	1	2	3
x_0	$y_0 = p_0(x)$			
		$p_{01}(x)$		
x_1	$y_1 = p_1(x)$		$p_{012}(x)$	
		$p_{12}(x)$		$p_{0123}(x)$
x_2	$y_2 = p_2(x)$		$p_{123}(x)$	
		$p_{23}(x)$		
x_3	$y_3 = p_3(x)$			

Erste Spalte sind konstante interpolierende Polynome, also genau die jeweils vorgegebenen Werte y_i .

Zweite Spalte sind interpolierende lineare Polynome zu jeweils zwei benachbarten Stützstellen.

Letzte Spalte enthält das interpolierende Polynom zu allen vorgegebenen Stützstellen.

Neue Stützstelle x_4 mit Wert y_4 kann in das Tableau eingefügt werden und führt zu einer neuen ‚Zeile‘ und einer neuen Endspalte $p_{01234}(x)$.

Auswertung des Tableaus jeweils nur an einer festen Stelle x möglich.

Beispiel: $x_0 = 0, y_0 = 1$,
 $x_1 = 1, y_1 = 3$, $x_2 = 3, y_2 = 2$

Auswertung des interpolierenden Polynoms an der Stelle $x=2$ mit Lagrange, bzw. Neville-Tableau:

Lagrange:

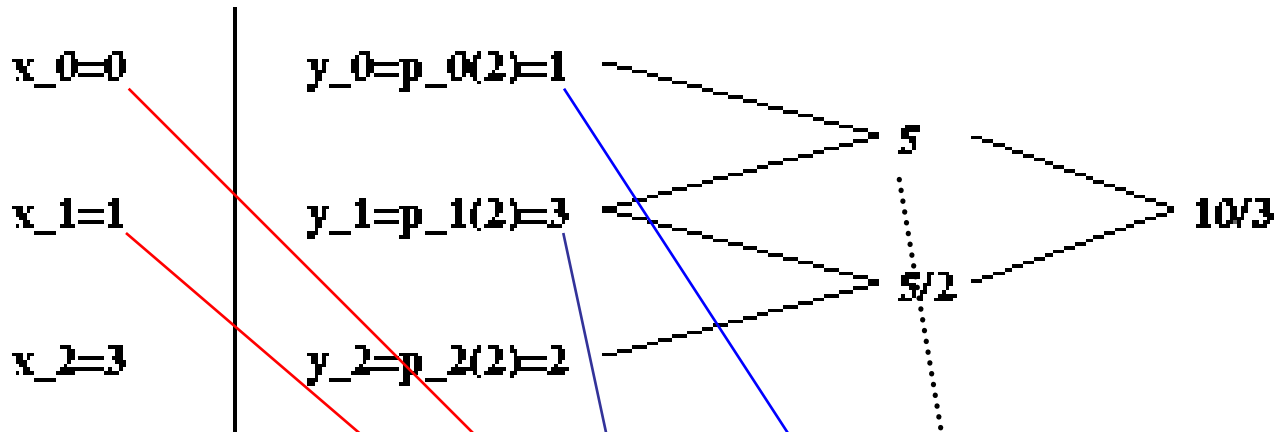
$$L_0(x) = \frac{(x-1)(x-3)}{(0-1)(0-3)} \quad , \quad L_1(x) = \frac{(x-0)(x-3)}{(1-0)(1-3)} \quad \text{und}$$

$$L_2(x) = \frac{(x-0)(x-1)}{(3-0)(3-1)}$$

→

$$p_{012}(2) = 1 \cdot L_0(2) + 3 \cdot L_1(2) + 2 \cdot L_2(2) = -\frac{1}{3} + 3 + 2 \cdot \frac{1}{3} = \frac{10}{3}$$

Neville-Tableau:



Da nach (*)

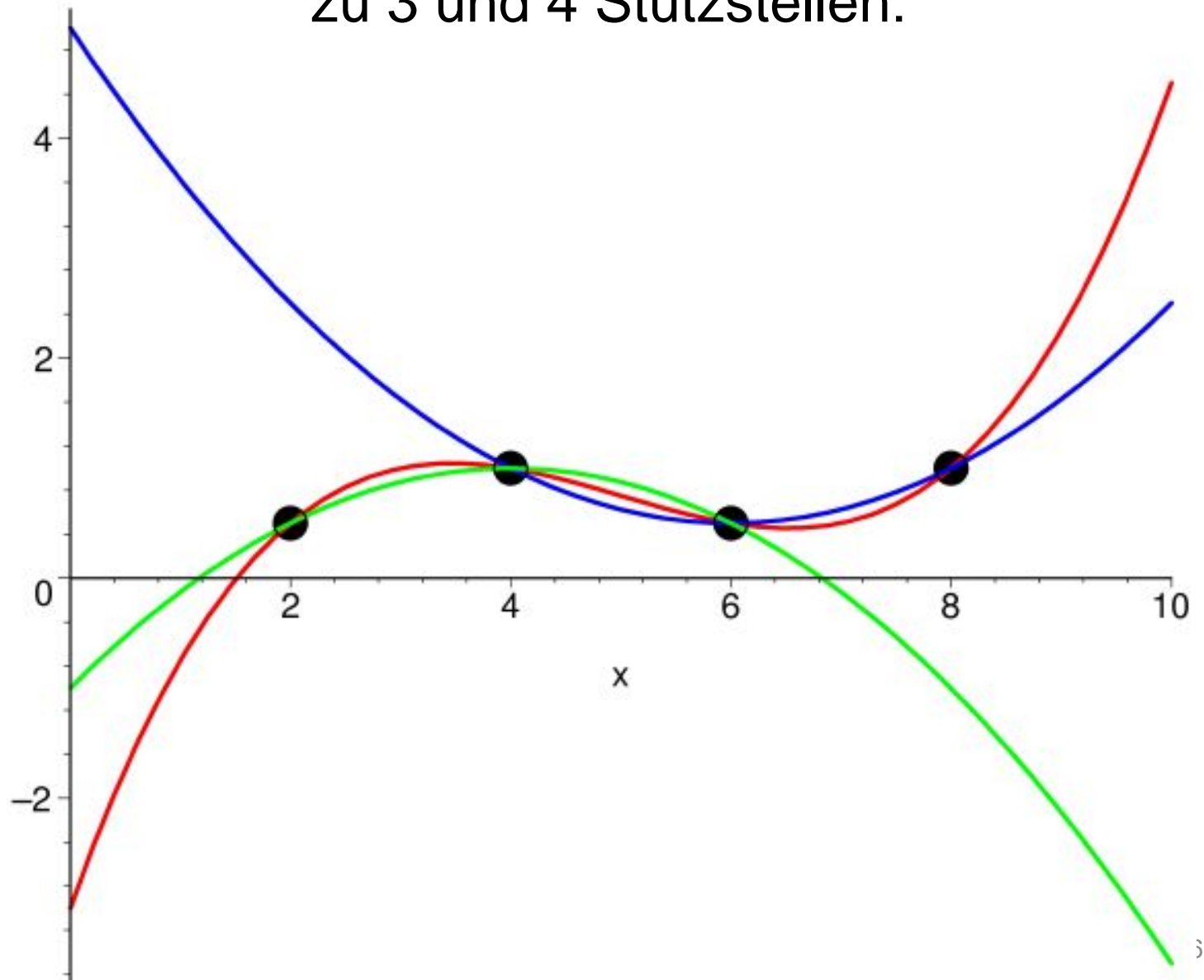
$$p_{01}(2) = \frac{(2-0) \cdot 3 - (2-1) \cdot 1}{1-0} = 5$$

$$p_{01}(2) = \frac{(2-x_0)p_1(2) - (2-x_1)p_0(2)}{x_1 - x_0}$$

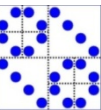
$$p_{12}(2) = \frac{(2-1) \cdot 2 - (2-3) \cdot 3}{3-1} = \frac{5}{2}$$

und daher auch
$$p_{012}(2) = \frac{(2-0) \cdot \frac{5}{2} - (2-3) \cdot 5}{3-0} = \frac{10}{3}$$

Folge von Interpolationspolynomen zu 3 und 4 Stützstellen:



interpol.m



4.1.6. Fehler bei der Polynominterpolation

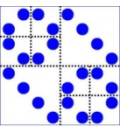
Satz: Gegeben Stützstellen $x_0 < \dots < x_n$ und genügend oft diff'bare Funktion $f(x)$. $p(x)$ sei das interpolierende Polynom vom Grad n mit $p(x_i) = f(x_i)$, $i = 0, \dots, n$.
An einer beliebigen Stelle gilt dann für die Abweichung zwischen p und f

$$f(\bar{x}) - p(\bar{x}) = \frac{f^{(n+1)}(\chi)}{(n+1)!} \cdot (\bar{x} - x_0) \cdots (\bar{x} - x_n)$$

Dabei ist $f^{(n+1)}(\chi)$ die $(n+1)$ -te Ableitung von f an einer Zwischenstelle χ aus dem Intervall

$$I := [\min\{x_0, x_n, \bar{x}\}, \max\{x_0, x_n, \bar{x}\}]$$

Frage: Wie gut wird f durch p dargestellt?



Beweis: Definiere Hilfsfunktion

$$g(x) = f(x) - p(x) - K \cdot (x - x_0) \cdots (x - x_n)$$

mit $p(x)$ = das interpolierende Polynom, und K Konstante.

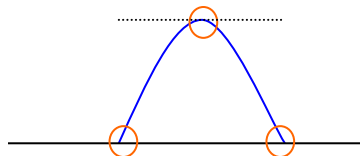
Nullstellen von g : x_0, \dots, x_n

Durch die Festlegung von K

$$K := \frac{f(\bar{x}) - p(\bar{x})}{(\bar{x} - x_0) \cdots (\bar{x} - x_n)} \quad \text{wird auch} \quad g(\bar{x}) = 0$$

Also hat die Funktion $g(x)$ $n+2$ Nullstellen!

Mit Mittelwertsatz besagt der Satz von Rolle, dass zwischen zwei Nullstellen einer stetig diff'baren Funktion f stets mindestens eine Nullstelle der Ableitung f' liegen muss (relatives Extremum mit waagrechter Tangente)



Also hat die erste Ableitung g' mindestens noch $n+1$ Nullstellen im Intervall I .

Die zweite Ableitung g'' noch n Nullstellen

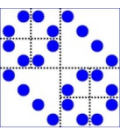
...

die $n+1$ -te Ableitung noch eine Nullstelle in I

$$\begin{aligned}
 0 &= g^{(n+1)}(\chi) = \\
 &= f^{(n+1)}(\chi) - p^{(n+1)}(\chi) - K \cdot \frac{d^{n+1}}{dx^{n+1}} \left((x-x_0) \cdots (x-x_n) \right) \Big|_{\chi} = \\
 &= f^{(n+1)}(\chi) - K \cdot (n+1)!
 \end{aligned}$$

Daher folgt:
$$K = \frac{f^{(n+1)}(\chi)}{(n+1)!}$$

Denn die $(n+1)$ -te Ableitung von $(x-x_0) \cdots (x-x_n)$ ist gleich der $(n+1)$ -ten Ableitung von x^{n+1} allein.



Die $(n+1)$ -te Ableitung ist auf dem Intervall I beschränkt, wenn f $(n+1)$ -mal stetig diff'bar ist. Dann existiert M mit

$$|K| \leq \frac{M}{(n+1)!}, \text{ z.B. mit } M = \max_{x \in I} |f^{(n+1)}(x)| < \infty$$

Insgesamt ergibt sich daher

$$\begin{aligned} |f(\bar{x}) - p(\bar{x})| &= |g(\bar{x}) + K(\bar{x} - x_0) \cdots (\bar{x} - x_n)| = \\ &= \left| \frac{f^{(n+1)}(\chi)}{(n+1)!} (\bar{x} - x_0) \cdots (\bar{x} - x_n) \right| \leq \\ &\leq \frac{M}{(n+1)!} |(\bar{x} - x_0) \cdots (\bar{x} - x_n)| \end{aligned}$$

□

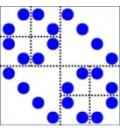
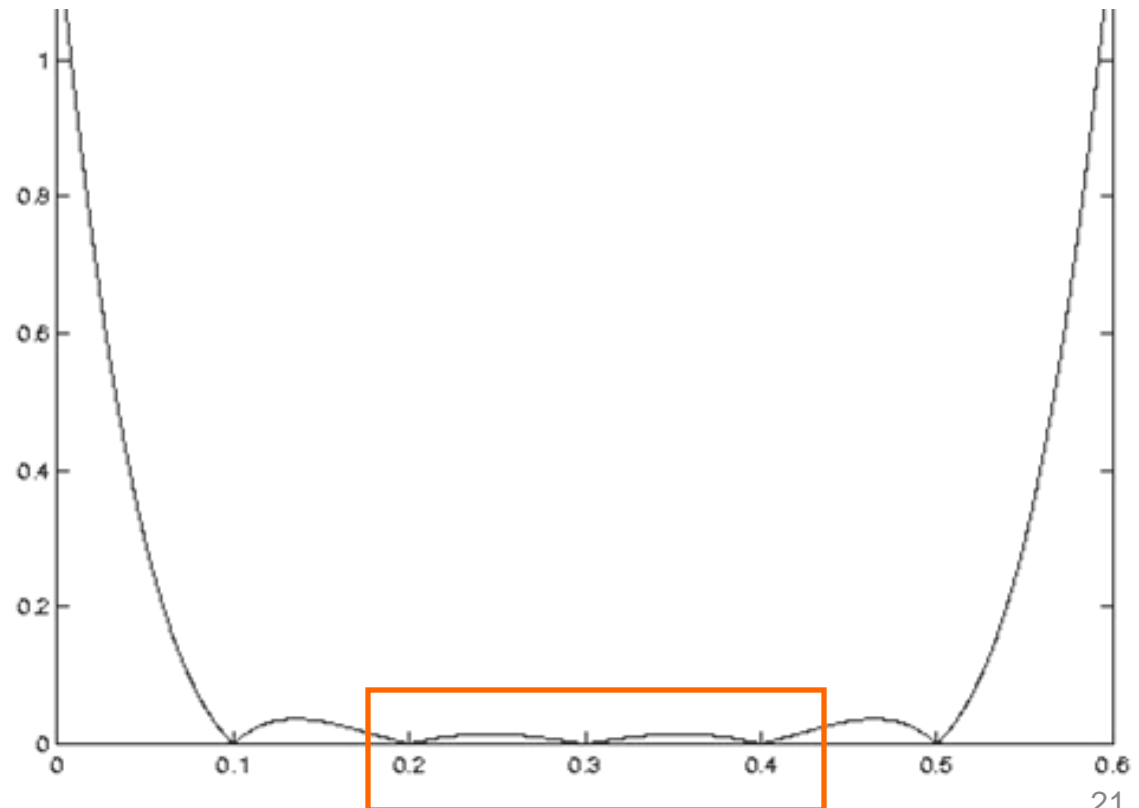
Frage: Wie gut stellt $p(x)$ die gegebene Funktion $f(x)$ dar?

Entscheidend dafür sind

- die Größe der $n+1$ -ten Ableitung von f auf dem Intervall I
- die Größe der Funktion $w(x) := (x - x_0) \cdots (x - x_n)$ an der Stelle \bar{x} abhängig von Wahl der Stützstellen x_i

Verlauf der Funktion

$$w(x) = |x-0.1| * |x-0.2| * |x-0.3| * |x-0.4|$$

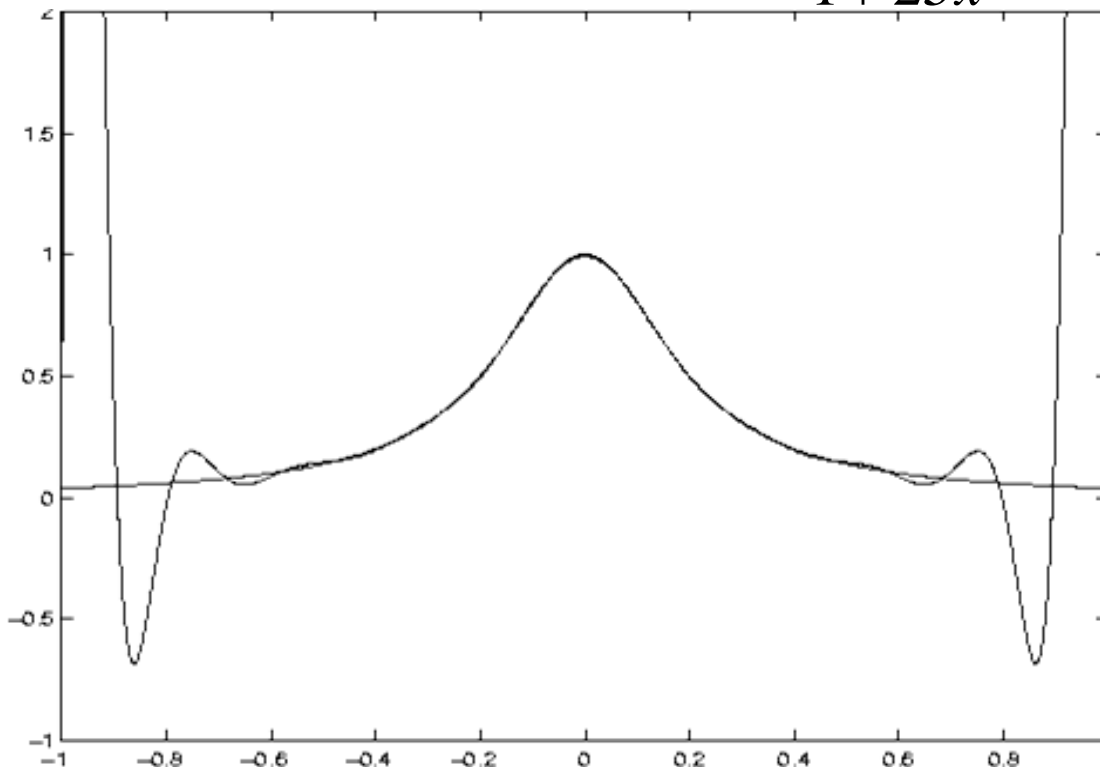


Folgerung: Näherung am Rand schlechter!

Der Abstand zwischen Funktion und interpolierendem Polynom ist klein in der Mitte der Stützstelle.

Am Rand und außerhalb kann der Fehler schnell groß werden.

Beispiel: Runge-Funktion $R(x) = \frac{1}{1 + 25x^2}$, $x \in [-1, 1]$



$R(x)$ und $p(x)$
vom Grad 20,
äquidistante
Stützstellen.

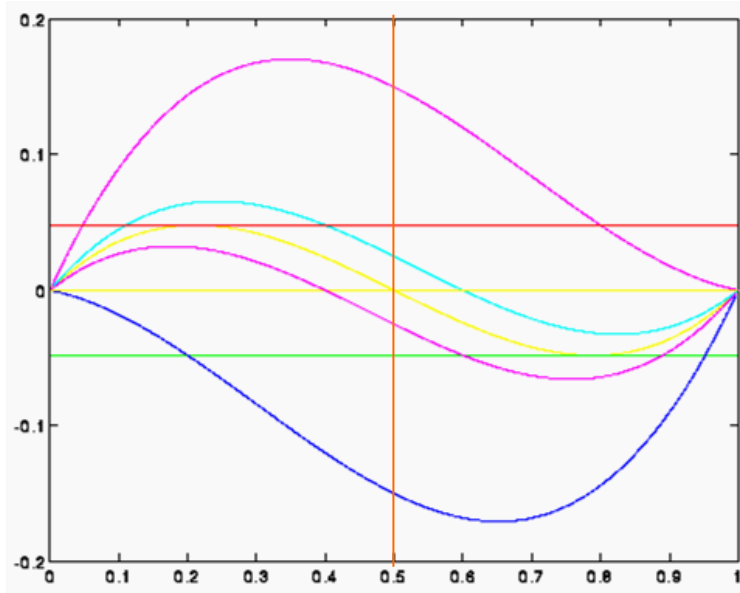
Bessere Wahl der Stützstellen, um den Fehler gleichmäßig auf dem ganzen Intervall klein zu halten:

Wähle die Stützstellen so, dass $w(x)=x^{n+1}+\dots = (x-x_0)\dots$ als Polynom vom Grad $(n+1)$ auf dem Intervall I möglichst **gleichmäßig klein** wird.

Spezialfall $n=2$, $x_0=0$ und $x_2=1$;
Bestimme x_1 so, dass $w(x)$ möglichst klein:

$$\begin{aligned} \min_{x_1} : \max_x |w(x)| &= \max_x |(x-0)(x-x_1)(x-1)| = \\ &= \max_x |x(x-1)(x-x_1)| \quad \text{in } [0,1] \end{aligned}$$

Lösung: $x_1=0.5$, gelbe Kurve



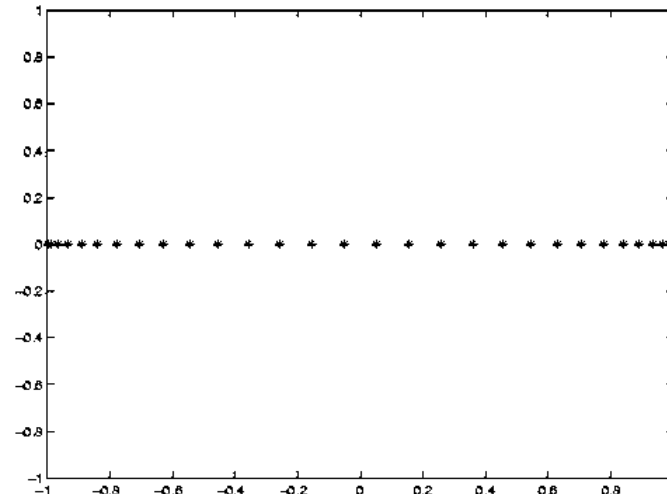
$w(x)$ für verschiedene x_1 .
Optimal, wenn alle Werte
in schmalem Band liegen.

Beispiel: stuetz.m

Allgemeine Lösung im Intervall $[-1, 1]$: Tchebycheff-Polynom

Nullstellen:

$$x_j = \cos\left(\frac{(2j+1)\pi}{2n+2}\right), \quad j = 0, 1, \dots, n$$



Also verteile Stützstellen besser so, dass am Rand mehr Punkte sind, um den ev. großen Fehler dort auszugleichen.

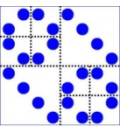
Allgemein gilt:

Polynominterpolation mit Polynomen hohen Grades neigt zu Oszillationen (siehe Runge-Funktion) und wird kaum verwendet.

An Stelle von äquidistanten Stützstellen

$$x_j = a + \frac{j}{n}(b - a), \quad j = 0, 1, \dots, n$$

verwende man besser eine Verteilung, bei der am Rande mehr Stützstellen sind, z.B. Tchebycheff-knoten (s.o.)



Allgemeines Stützstellenproblem im Intervall $[-1, 1]$:

$$\min_{a_0, \dots, a_n} \left\{ \max_{-1 \leq x \leq 1} \left| x^{n+1} + a_n x^n + \dots + a_1 x + a_0 \right| \right\} =$$

$$\min_{x_0, \dots, x_n} \left\{ \max_{-1 \leq x \leq 1} \left| (x - x_0) \cdots (x - x_n) \right| \right\}$$

MATLAB: `cheb.m`

4.1.7 Newtonform des Interpolationspolynoms

Manchmal ist man auch an einer expliziten Darstellung des Interpolationspolynoms selbst interessiert.

Allerdings eignet sich dazu nicht die übliche Form

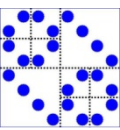
$$p(x) = a_n x^n + \cdots + a_1 x + a_0$$

Effiziente Auswertung dieser Form mittels Hornerschema:

$$(\cdots((a_n x + a_{n-1})x + a_{n-2})x + \cdots + a_1)x + a_0$$

durch das Programm:

```
y=a(n);
FOR j=n-1,...,0:
    y=y*x+a(j);
ENDFOR
```



An Stelle der Standardform verwendet man $p(x)$ in einer Darstellung, in der die Stützstellen eingehen:

$$\begin{aligned}
 p(x) &= f[x_0] + f[x_0, x_1](x - x_0) + \\
 &+ f[x_0, x_1, x_2](x - x_0)(x - x_1) + \cdots + \\
 &+ f[x_0, \dots, x_n](x - x_0) \cdots (x - x_{n-1})
 \end{aligned}$$

Die Koeffizienten $f[x_0, \dots, x_j]$ erhält man wieder aus der wesentlichen Formel (*) aus 4.1.5:

$$\begin{aligned}
 p_{i, \dots, i+k}(x) &= \frac{(x - x_i)p_{i+1, \dots, i+k}(x) - (x - x_{i+k})p_{i, \dots, i+k-1}(x)}{x_{i+k} - x_i} = \\
 &f[x_i] + f[x_i, x_{i+1}](x - x_i) + \dots + f[x_i, \dots, x_{i+k}](x - x_i) \cdots (x - x_{i+k-1})
 \end{aligned}$$

Sie lassen sich wieder aus einem Tableau (Tableau der ‚Dividierten Differenzen‘) der Reihe nach berechnen mittels

$$f[x_i, \dots, x_{i+k}] = \frac{f[x_{i+1}, \dots, x_{i+k}] - f[x_i, \dots, x_{i+k-1}]}{x_{i+k} - x_i}$$

Dann kann durch Horner-artiges Schema $p(x)$ an der Stelle x ausgewertet werden.

4.1.8. Weitere Interpolationsansätze

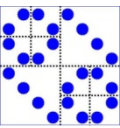
(a) Hermite-Interpolation:

Um Oszillationen zu vermeiden will man auch die Ableitungen an den Stützstellen kontrollieren, im einfachsten Fall genau die ersten Ableitungen:

Vorgegeben sind (x_j, y_j, y'_j) , $j=0, \dots, n$

Gesucht ein Polynom vom Grad $2n+1$ mit

$$p(x_j) = y_j \quad \text{und} \quad p'(x_j) = y'_j, \quad j=0, 1, \dots, n$$



- (1) Lösung durch Aufstellen des dadurch gegebenen linearen $2n+2 \times 2n+2$ Gleichungssystems für die gesuchten *Koeffizienten des Polynoms*:

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^{2n+1} \\ 0 & 1 & \cdots & (2n+1)x_0^{2n} \\ 1 & x_1 & \cdots & x_1^{2n+1} \\ \vdots & \vdots & & \vdots \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{2n+1} \end{pmatrix} = \begin{pmatrix} y_0 \\ y'_0 \\ y_1 \\ \vdots \end{pmatrix}$$

- (2) Lösung durch ‚Lagrange‘-Polynome, die genau an einer Stützstelle z.B. Wert 0 und Ableitung 1 haben, sonst an allen anderen Stützstellen Wert und Ableitung 0.

Setze dazu

$$L_j(x) := (x - x_0)^2 \cdots (x - x_{j-1})^2 (x - x_{j+1})^2 \cdots (x - x_n)^2$$

1. Aufgabe:

Finde Polynom vom Grad $2n+1$, das

an der Stelle x_j den Wert 1 und Ableitung 0 hat, und
an allen anderen Stützstellen Wert und Ableitung 0:

$$\text{Ansatz: } G_j(x) = L_j(x) \cdot (\alpha + \beta x)$$

α und β sind daher so zu wählen, dass

$$L_j(x_j)(\alpha + \beta x_j) = 1 \quad \text{gilt} \quad \text{und} \quad L_j'(x_j)(\alpha + \beta x_j) + L_j(x_j)\beta = 0$$

Dies liefert zwei Bedingungen für α und $\beta \rightarrow G_j(x)$

2. Aufgabe:

Finde Polynom $H_j(x)$, das an der Stelle x_j den Wert 0 und Ableitung 1 hat, an allen anderen Stützstellen Wert und Ableitung 0.

Dann sind α und β so zu wählen, dass

$$L_j(x_j)(\alpha + \beta x_j) = 0 \quad \text{gilt} \quad \text{und} \quad L_j'(x_j)(\alpha + \beta x_j) + L_j(x_j)\beta = 1$$

Lösung liefert α und $\beta \rightarrow H_j(x)$

Damit erhält man als Lösung der Hermite-Interpolation:

$$p(x) = \sum_{j=0}^n \left(y_j G_j(x) + y'_j H_j(x) \right)$$