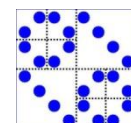


$$G_{ij} = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & \ddots & & & \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \\ & & & & & & & \ddots & & & \\ & & & & & & & & & 1 & \\ & & & & & & & & & & \ddots & \\ & & & & & & & & & & & & 1 \end{pmatrix}$$

The matrix is partitioned into four quadrants by rows j and i and columns j and i . The elements in the (j, j) to (j, i) and (i, j) to (i, i) regions are highlighted in blue. The element at (i, j) is highlighted in yellow.



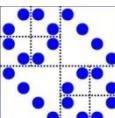
Zur Elimination eines Elementes a_{ij} der Matrix A multiplizieren wir $G_{ij} \cdot A$.

Dieses Produkt verändert nur die i -te und die j -te Zeile von A . Es genügt, vom Gesamt-System nur diesen 2×2 – Teil zu betrachten. Also muss wieder

$$\varphi = \text{arcctg} \left(\frac{a_{jj}}{a_{ij}} \right) \text{ gesetzt sein wie oben. } (1 \rightarrow j \text{ und } 2 \rightarrow i)$$

Mit einer solchen Matrix G_{21} wird dann im ersten Schritt a_{21} zu Null gemacht.

$$G_{21} = \begin{pmatrix} G & 0 \\ 0 & I \end{pmatrix}, \quad G = \begin{pmatrix} \cos(\varphi) & \sin(\varphi) \\ \sin(\varphi) & -\cos(\varphi) \end{pmatrix} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}$$





Genauere Analyse eines allgemeinen Eliminationsschritts:



$$\begin{pmatrix} \ddots & & & & & \\ & \ddots & & & & \\ j & \boxed{c} & & \boxed{s} & & \\ & & \ddots & & & \\ i & \boxed{s} & & \boxed{-c} & & \\ & & & & \ddots & \end{pmatrix} \cdot \begin{pmatrix} \ddots & & & & & \\ & 0 & \boxed{a_{jj}} & & \boxed{a_{ji}} & \\ & 0 & 0 & \ddots & & \\ & 0 & \boxed{a_{ij}} & & \boxed{a_{ii}} & \\ \vdots & \vdots & & \uparrow & & \ddots \\ & & & k & & \end{pmatrix} = G_{i,j} \cdot A$$

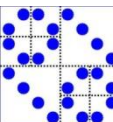
verändert nur j-te und i-te Zeile;

i-te Zeile: $a_{ik} \rightarrow sa_{jk} - ca_{ik}$ für $k=j, \dots, n$

speziell: $a_{ij} \rightarrow sa_{jj} - ca_{ij} = 0$! soll Null werden
 Legt daher φ fest.

j-te Zeile: $a_{jk} \rightarrow ca_{jk} + sa_{ik}$ für $k=j, \dots, n$
 mit c und s zu obigen φ

Keine extra Pivotszeile sondern Kombination zweier Zeilen!



Verwende der Reihe nach $G_{21}, G_{31}, \dots, G_{n1}$

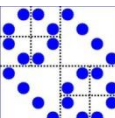
zur Bearbeitung der ersten Spalte,

also um $a_{21}, a_{31}, \dots, a_{n1}$ zu Null zu machen,

und danach $G_{32}, G_{42}, \dots, G_{n2}$, \dots , G_{n-1n-2}, G_{nn-2} , und G_{nn-1}

um $a_{32}, a_{42}, \dots, a_{n2}$, \dots , a_{n-1n-2}, a_{nn-2} , und a_{nn-1}

zu Null zu machen.



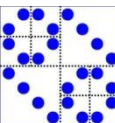
Die Reihenfolge, in der die $a_{j,i}$ zu Null gemacht werden, ist gegeben durch:

$$\begin{array}{ccccccc}
 & \boxed{\cdot} & & & & & \cdot \\
 1 & & \boxed{\cdot} & & & & \\
 2 & & n & & \boxed{\cdot} & & \\
 \vdots & & \vdots & & & \boxed{\cdot} & \\
 n-1 & 2n-3 & \dots & n(n-1)/2 & \boxed{\cdot} & &
 \end{array}$$

Jeweils nötig ist eine Multiplikation mit Givens-Reflexion

$$\mathbf{G}_{i,j}, \quad i=1, \dots, n-1 \quad \text{und} \quad j=i+1, \dots, n.$$

Also benötigt man insgesamt $n(n-1)/2$ Givensreflexionen um eine quadratische $n \times n$ –Matrix auf Dreiecksgestalt zu transformieren.



Man benutze also immer das Diagonalelement a_{jj} und eine Kombination von i -ter/ j -ter Zeile, um a_{ij} zu Null zu machen.

$$Q^T := G_{n,n-1} G_{n,n-2} \cdots G_{n-1,n-2} \cdots G_{n1} \cdots G_{21}$$

$$Q^T A = G_{n,n-1} G_{n,n-2} \cdots G_{n-1,n-2} \cdots G_{n1} \cdots G_{21} A = R$$

mit einer oberen Dreiecksmatrix R .

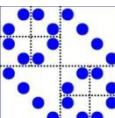
Daher ist

$$Q = G_{21} \cdots G_{n1} \cdots G_{n,n-1}, \text{ da } G_{ij}^T = G_{ij} \text{ und } A=Q^*R.$$

Q ist gegeben durch die einzelnen G_{ij} ;

jedes G_{ij} ist eindeutig gegeben durch das φ_{ij} , das nötig war, um genau ein a_{ij} zu eliminieren.

Q wird meist nicht explizit berechnet. Speichere nur die G_s .





Genauso kann man für eine $m \times n$ Matrix A ($m > n$) mit $\text{rank}(A)=n$ eine QR-Zerlegung berechnen

$$\boxed{A} = \boxed{Q} \cdot \boxed{R}$$

Wie bei der Gauss-Elimination eliminiert man also mit den Diagonalelementen der Reihe nach sämtliche Unterdiagonalelemente.

Der Vorteil der QR-Zerlegung:

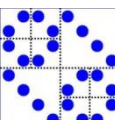
$$\text{cond}(A) = \text{cond}(QR) = \text{cond}(R)$$

Gut für schlecht konditionierte Systeme

Anwendbar auf rechteckige Systeme

Andere Orthogonalisierungsverfahren:

- Gram-Schmidt (orthonormalisiere Vektoren), geo. Proj.
- Householder (erzeuge in einem Schritt eine ganze Nullspalte). $H = I - 2 u u^T$, $\|u\|_2 = 1$.

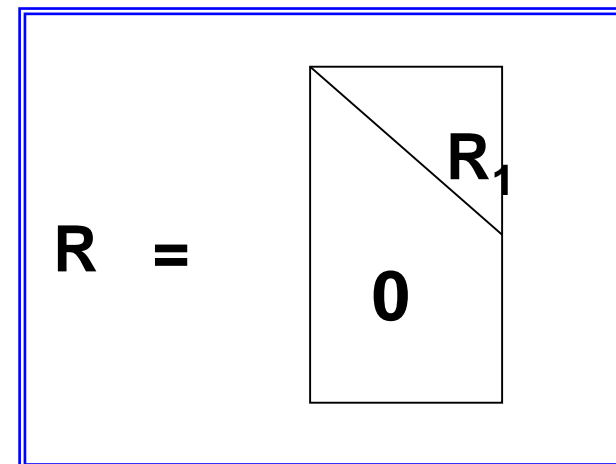


3.9.4. Anwendung bei Linearer Ausgleichsrechnung:

$$\begin{aligned} \min_x \|Ax - b\|_2 &= \min_x \|QRx - b\|_2 = \\ &= \min_x \|Q^T(QRx - b)\|_2 = \min_x \|Rx - Q^T b\|_2 \end{aligned}$$

da Q orthogonal und euklid'sche Norm.

R ist obere Dreiecksmatrix der Dimension $m \times n$ und vollen Ranges n .



Das obige Minimum erhält man wegen

$$\min_x \left\| \begin{pmatrix} R_1 \\ 0 \end{pmatrix} x - Q^T b \right\|_2^2 = \min_x \left\| \begin{pmatrix} R_1 x \\ 0 \end{pmatrix} - \begin{pmatrix} \tilde{b}_1 \\ \tilde{b}_2 \end{pmatrix} \right\|_2^2 = \min_x \left\| R_1 x - \tilde{b}_1 \right\|_2^2 + \left\| \tilde{b}_2 \right\|_2^2$$

aus der Lösung des Dreieckssystems

$$R_1 x = \tilde{b}_1$$

Der Wert des Minimums ist gegeben durch $\left\| \tilde{b}_2 \right\|_2^2$

$$\min_x \left\| \begin{pmatrix} 1 & 1 \\ \mathbf{1} & 1 \\ 0 & 1 \end{pmatrix} x - \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\|_2 = \min_x \|Ax - b\|_2$$

Erster Schritt: $\mathbf{a}_{21} \rightarrow 0$:

$$\begin{pmatrix} c & s & 0 \\ s & -c & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{1} & 1 \\ \mathbf{1} & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} c+s & c+s \\ s-c & s-c \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \sqrt{2} & \sqrt{2} \\ 0 & 0 \\ 0 & \mathbf{1} \end{pmatrix}$$

mit

$$c = s = 1/\sqrt{2}, \quad \varphi = \pi/4$$

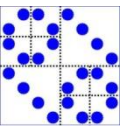


$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & c & s \\ 0 & s & -c \end{pmatrix} \cdot \begin{pmatrix} \sqrt{2} & \sqrt{2} \\ 0 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \sqrt{2} & \sqrt{2} \\ 0 & s \\ 0 & -c \end{pmatrix} = \begin{pmatrix} \sqrt{2} & \sqrt{2} \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

mit $c = 0, s = 1, \varphi = \pi/2$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \sqrt{2} & \sqrt{2} \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

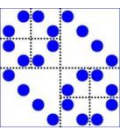
$Q^T \cdot A = R$



Also $Q = \begin{pmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 1/\sqrt{2} & 0 & -1/\sqrt{2} \\ 0 & 1 & 0 \end{pmatrix}$

Anwendung auf Minimierungsproblem :

$$\begin{aligned} \min_x \left\| Ax - \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\|_2 &= \min_x \left\| \begin{pmatrix} \sqrt{2} & \sqrt{2} \\ 0 & 1 \\ 0 & 0 \end{pmatrix} x - \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\|_2 = \\ &= \min_x \left\| \underbrace{\begin{pmatrix} \sqrt{2} & \sqrt{2} \\ 0 & 1 \end{pmatrix} \cdot x - \begin{pmatrix} 0 \\ 1 \end{pmatrix}}_0 \right\|_2 = \min_x \left\| \begin{pmatrix} \sqrt{2} & \sqrt{2} \\ 0 & 1 \end{pmatrix} \cdot x - \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\|_2 \end{aligned}$$

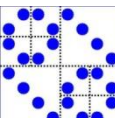


Lösung x als Lösung des Dreiecksgleichungssystems:

$$x = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

In diesem Fall liefert x sogar eine genaue Lösung von $Ax=b$, da der Fehlerterm $\|b_2\|$ gleich Null ist.

QR-Zerlegung ist in dieser Form anwendbar für beliebige rechteckige Matrix A , so lange A vollen Rang besitzt.



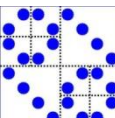
Kosten des QR-Verfahrens mit Givens für $n \times n$ – Matrix:
 $2n^3 + O(n^2)$ (also teurer als Gauss-Elimination mit $2n^3/3$)

Ein Eliminationsschritt bei Spalte k :

$$(2 \text{ mult} + 1 \text{ add})2k = 6k \text{ flop}$$

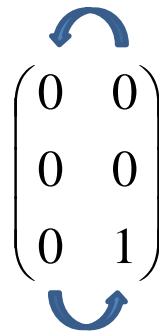
Insgesamt:
$$\sum_{k=n-1}^2 (k-1) \cdot 6k = 2n^3 + O(n^2)$$

Bei $m \times n$ – Matrix mit $m > n$ und $\text{Rang}(A)=n$: $n^2 (3m-n)$



- schlecht konditioniertem Gleichungssystem
- überbestimmtem Gleichungssystem mit vollem Rang (an Stelle der Normalgleichung), wie oben beschrieben
- allgemeinem nichtquadratischen System in der Form $QAP = R$ mit Permutation P zum Vertauschen von Spalten. (P ist nötig, um einen Block vollen Ranges nach vorne/oben zu transportieren)

Beispiel:

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$$


- Entdeckung fast linear abhängiger (eigentlich überflüssiger) Gleichungen oder Unbekannter (numerische Bestimmung des Rangs von A)
- Reduktion der Matrix auf den wesentlichen Teil (Noise-reduction)

$$A = (Q_1 \quad Q_2) \cdot \begin{pmatrix} R & S \\ 0 & \varepsilon \end{pmatrix} \approx Q_1 \cdot (R \quad S) = (Q_1 R \quad Q_1 S)$$

3.10 Regularisierung

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

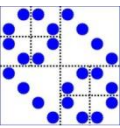
Dadurch erhält man bei der Lösung dieses Problems einen Vektor x , der extrem groß ist:

Ist in $Ax=b$ die Matrix A (fast) singulär \rightarrow

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

\rightarrow $\| x \| = \| \text{inv}(A) * b \|$ kann sehr groß sein.

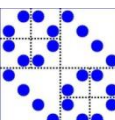
$\text{inv}(A)$ gefährlich, da schlecht konditioniert, und daher kleinste Rundungsfehler in b große Fehler in x erzeugen!

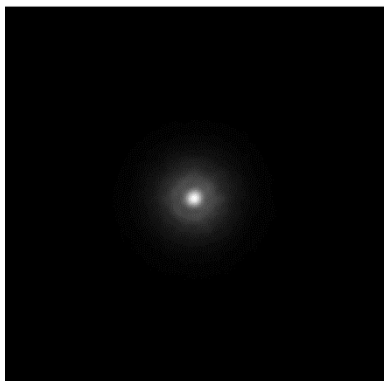


Durch Mess/Rundungsfehler enthält aber die rechte Seite b viele kleine Störungen (noise, Rauschen), die in der berechneten Näherungslösung x dann sehr groß werden, so dass - selbst bei exakter Rechnung - die so berechnete Lösung x unbrauchbar ist.

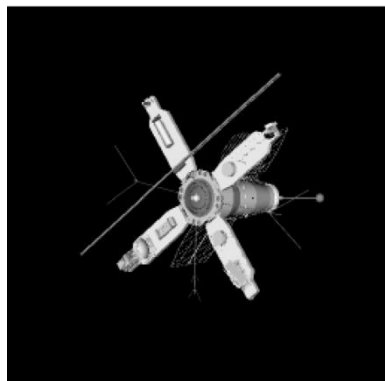
$$\tilde{x} = A^{-1}(b + \Delta b) = A^{-1}b + A^{-1}\Delta b = x + \underbrace{A^{-1}\Delta b}$$

Störanteil
Viel größer als x

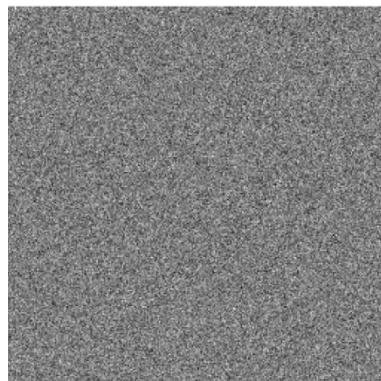




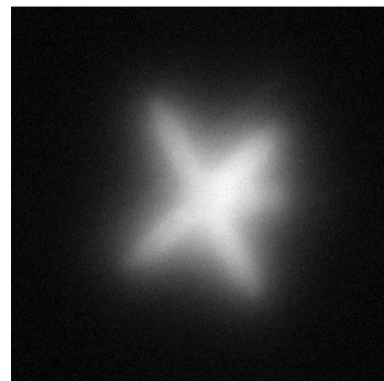
*



+



=



$$A * x + n = b$$

Ausweg:

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

Groß ist „unphysikalisch“ und kommt vom Rauschen!

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

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 und spd 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)

Regularisierende Zusatzbedingungen:

- Beschränktheit der Lösung $\|x\|$ in einer passenden Norm
- Ev. Dünnbesetztheit der Lösung $(x_1, 0, \dots, 0, x_k, 0, \dots, 0)$
- Ev. Glattheit der Lösung $\Delta x_i \approx 0$
- Nähe zu schon bekannter guter Näherung $\|x - x_{\text{approx}}\|$

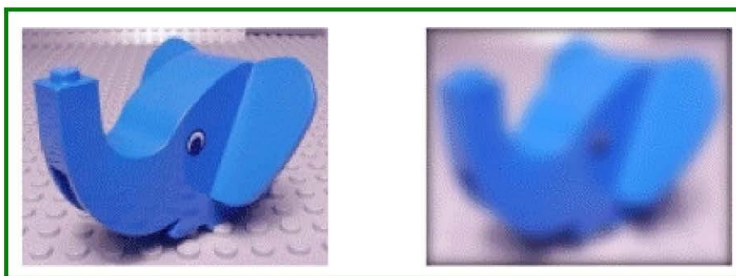


The Difficult Task of Image Deblurring

The underlying linear model:

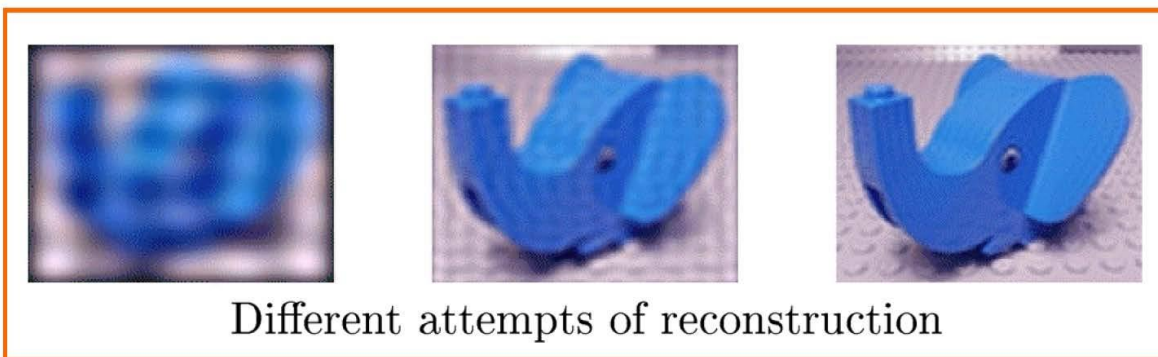
$$Ax = b \quad \rightarrow$$

$$\text{cond}(A) = \infty$$

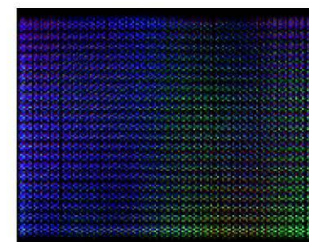


The “naive” reconstruction:

$$x_{\text{naive}} = A^{-1}b$$



Different attempts of reconstruction



Überblick über Lösungsmethoden in Abhängigkeit vom Charakter des Problems:

- Gauss-Elimination, LR, ($n \times n$, gut konditioniert)
- QR-Zerlegung, ($m \times n$, voller Rang, schlecht kond.)
- QR-Zerlegung mit Permutation $AP=QR$, falls nicht voller Rang
- Regularisierung (extrem schlecht, singular, Rauschen)

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

IV. Interpolation und Quadratur

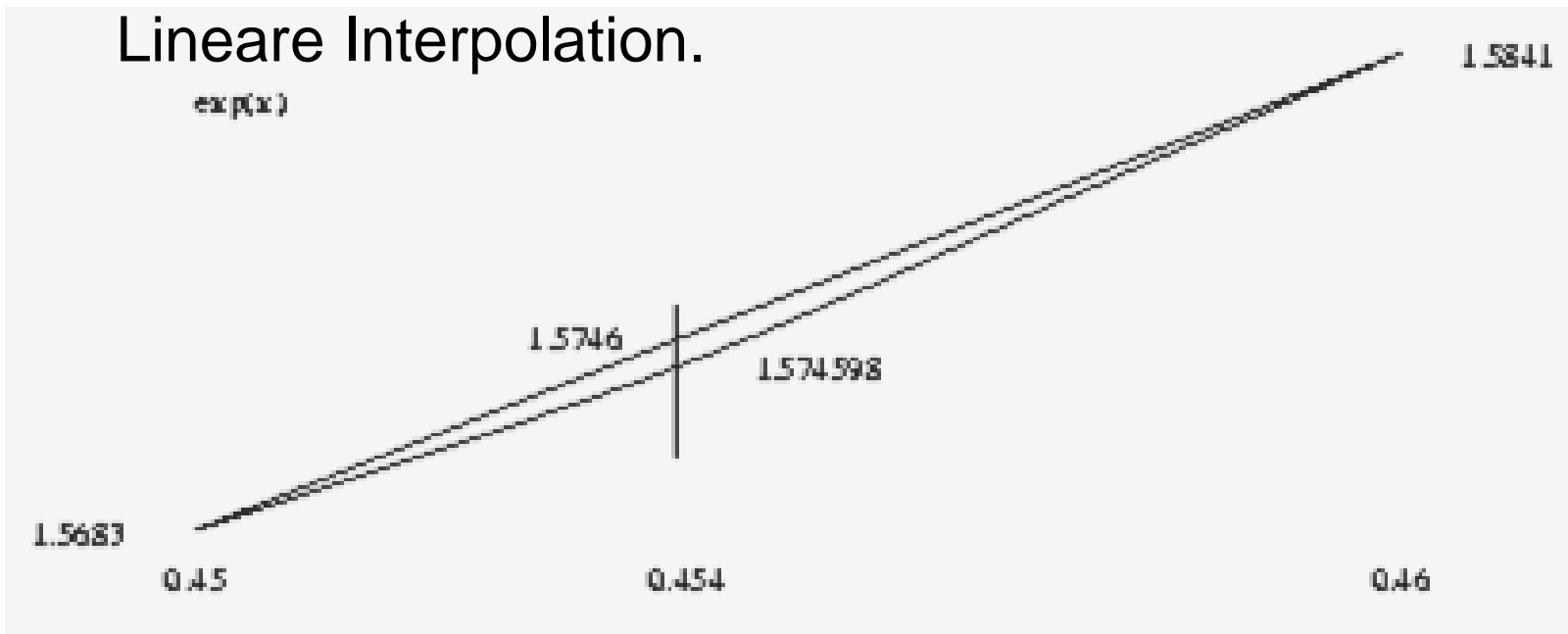
4.1. Interpolation

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

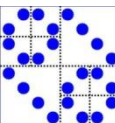
Gesucht: $\exp(0.454)$;

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

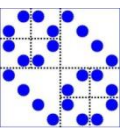


Verwandle diskrete Punkte in kontinuierliche
Kurve oder Fläche:

- Audiosignal
- Gitterpunkte in Fläche - CAD

quasi-kontinuierlich \rightarrow diskret \rightarrow quasi-kontinuierlich

Funktion \rightarrow Maschinenzahl \rightarrow Graphik



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!
Genauso viele Bedingungen wie Freiheitsgrade!

Man unterscheide Interpolation und Approximation!

Bei Approximation mehr Bedingungen (Punkte) als
Freiheitsgrade (Unbekannte)!

Beispiel zu Approximation:

*Ausgleichsgerade führt auf überbestimmtes Gleichungssystem
(Normalgleichung, QR)*

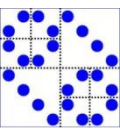
4.1.3. Spezialfall Polynom-Interpolation:

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! $O(n^3)$



4.1.4. Lösung mit Lagrange-Polynomen

Definiere geschickt Basis-Polynome:

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

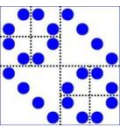
$n+1$ Polynome vom Grad n , besser als $1, x, x^2, \dots, x^n$

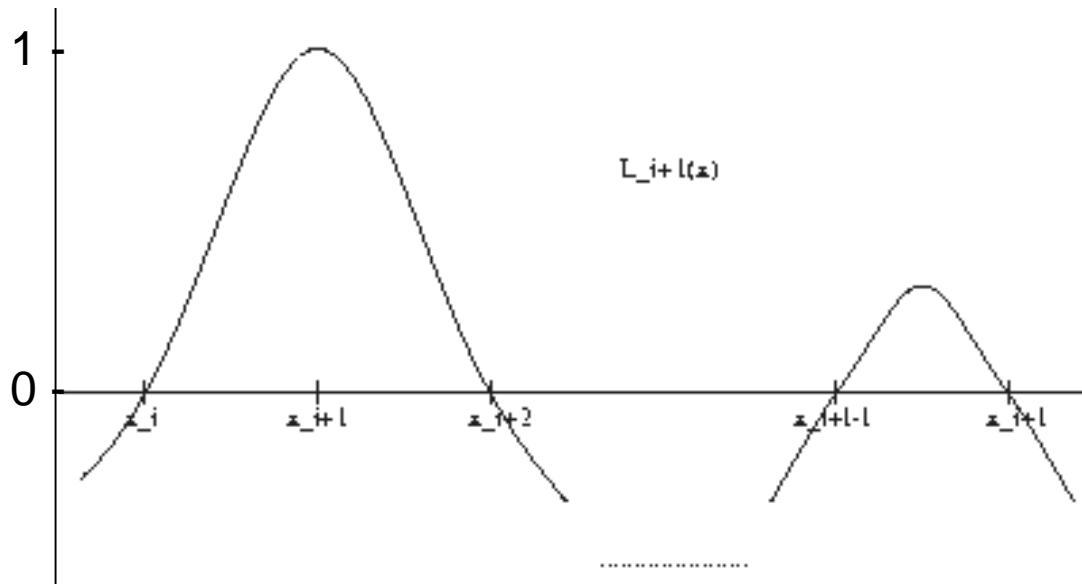
Eigenschaften der Lagrange-Polynome: Grad n mit

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

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:

$$\begin{pmatrix} L_0(x_0) & \cdots & L_n(x_0) \\ \vdots & & \vdots \\ L_0(x_n) & \cdots & L_n(x_n) \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix} = \begin{matrix} \boxed{\phantom{\text{matrix}}} \\ \phantom{\text{matrix}} \end{matrix} \begin{pmatrix} y_0 \\ \vdots \\ y_n \end{pmatrix}$$

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

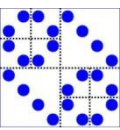
Damit ist die Existenz eines interpolierenden Polynoms gezeigt!

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.

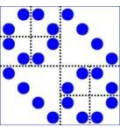
Aus dem Hauptsatz der Algebra folgt daher, dass

$\alpha = 0$ 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 gut geeignet, da numerisch problematisch und teuer!

Aber wichtig für theoretische Resultate!

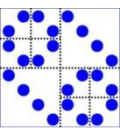


Löse nicht das lineare Gleichungssystem, da dies zu teuer ist!
 Außerdem wird oft nur der Wert des Polynoms an einer einzigen Stelle gesucht!

Idee: Berechne induktiv interpolierende Polynome für immer mehr Stützstellen.

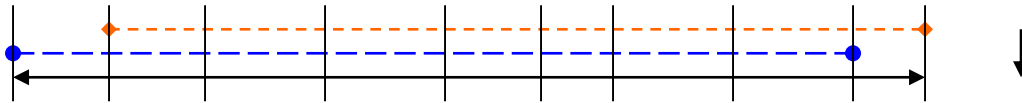
Setze dazu $p_{i,\dots,i+l}(x)$ als das interpolierende Polynom vom Grade l , das genau an den Stellen $x_i, x_{i+1}, \dots, x_{i+l}$ die Interpolations-Bedingungen erfüllt.

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!

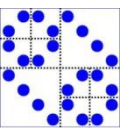
Anwendung der Formel (*) zur punktweisen Auswertung des Interpolationspolynoms an einer Stelle \bar{x} : $p(\bar{x}) = ?$

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} .

Wir sind nicht an theoretischer Kenntnis des Polynoms Interessiert sondern nur an Auswertung!



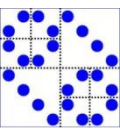
Neville-Tableau:

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.

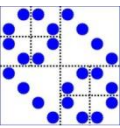


Neue Stützstelle x_4 mit Wert y_4 kann in das Tableau eingefügt werden und führt zu einer neuen ‚Zeile‘ und einer neuen Endspalte $p_{01234}(x)$.

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: Stützstellen $x = 0, 1, 3$

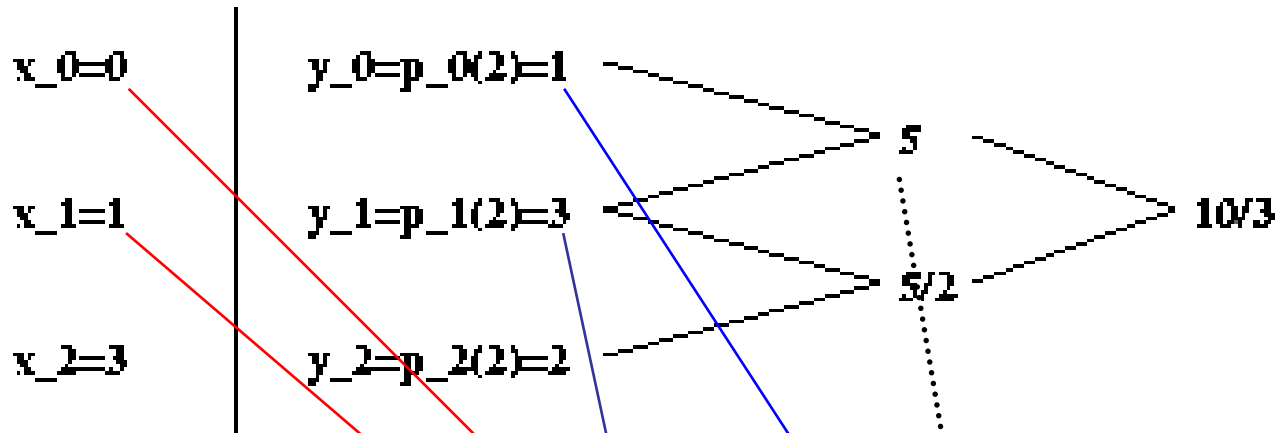
$$L_0(x) = \frac{(x-1)(x-3)}{(0-1)(0-3)} \quad , \quad L_1(x) = \frac{(x-0)(x-3)}{(1-0)(1-3)} \quad \text{und}$$

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

→

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

Neville-Tableau:



Da nach (*)

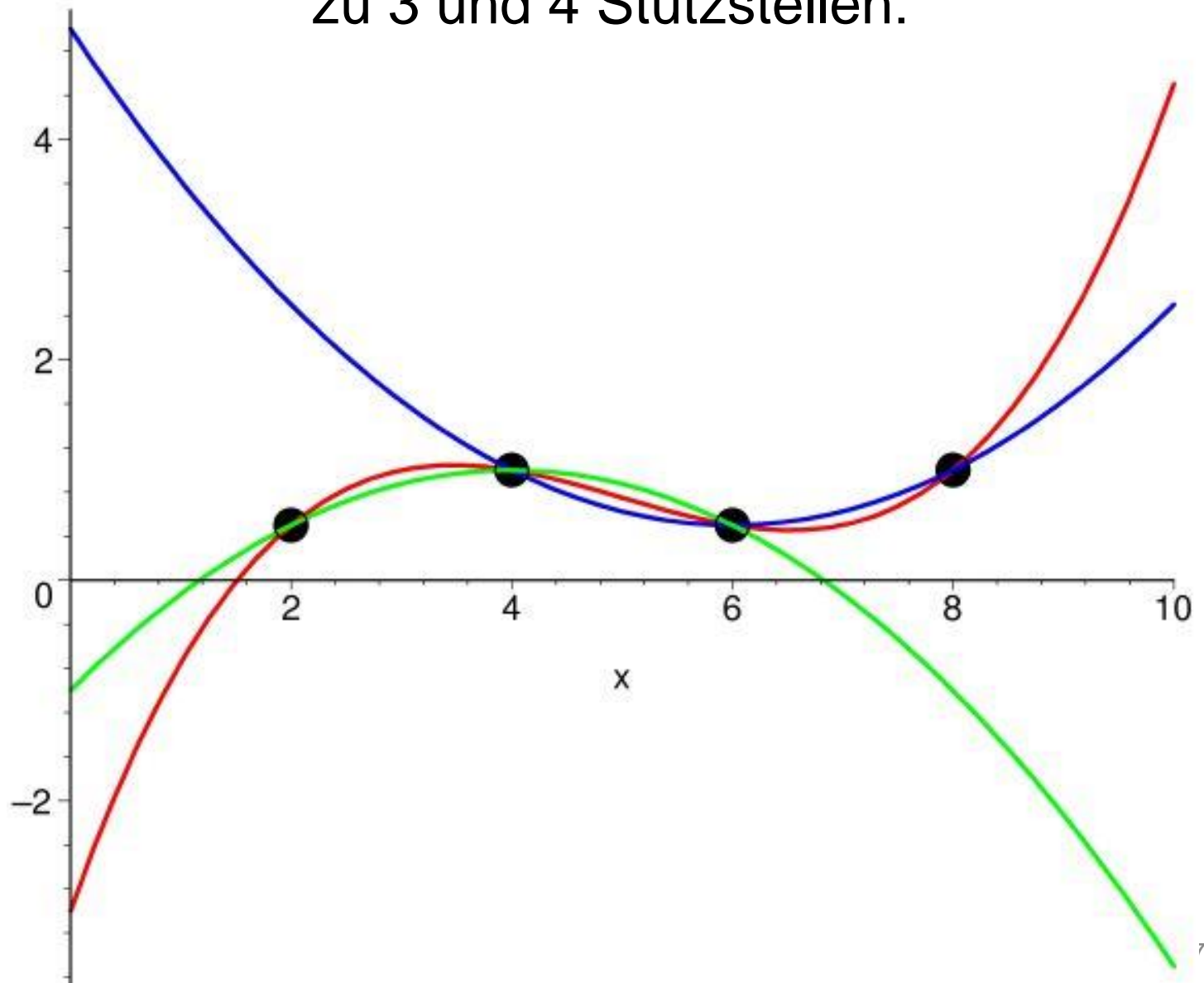
$$p_{01}(2) = \frac{(2-0) \cdot 3 - (2-1) \cdot 1}{1-0} = 5$$

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

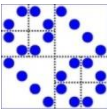
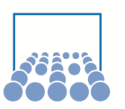
$$p_{12}(2) = \frac{(2-1) \cdot 2 - (2-3) \cdot 3}{3-1} = \frac{5}{2}$$

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

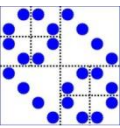
Satz: Gegeben Stützstellen $x_0 < \dots < x_n$ und genügend oft diff'bare Funktion $f(x)$. $p(x)$ sei das interpolierende Polynom vom Grad n mit $p(x_i) = f(x_i)$, $i = 0, \dots, n$.
An einer beliebigen Stelle gilt dann für die Abweichung zwischen p und f

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

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

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

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.

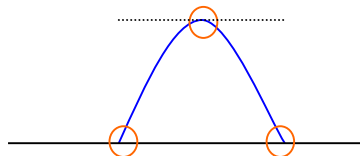
Nullstellen von g : x_0, \dots, x_n

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

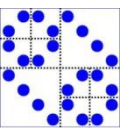
...

die $n+1$ -te Ableitung noch **eine** Nullstelle χ 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

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

□