

Der k-te Schritt der GE kann als Matrixprodukt beschrieben werden in der Form

$$A_{k+1} = (I - L_k)A_k = A_k - L_k A_k \quad \text{mit Einheitsmatrix } I.$$

(Ziehe von den unteren Zeilen von A_k jeweils die entsprechend gewichtete k-te Zeile ab.)

$$L_k \cdot A_k = \begin{pmatrix} 0 & \dots & 0 \\ \cdot & 0 & \cdot \\ \cdot & l_{k+1,k} & \cdot \\ \vdots & \vdots & \vdots \\ 0 & l_{n,k} & 0 \end{pmatrix} * \begin{pmatrix} * \\ A_{k,.} \\ * \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ l_{k+1,k} * A_{k,.} \\ \vdots \\ l_{n,k} * A_{k,.} \end{pmatrix}$$

Dabei bezeichnet $A_{k,.}$ die k-te Zeile von A_k .

Insgesamt:

$$U = A_n = (I - L_{n-1})A_{n-1} = \dots = \overbrace{(I - L_{n-1}) \dots (I - L_1)}^{\tilde{L}} A_1 = \tilde{L}A$$

mit der Matrix

$$\tilde{L} := (I - L_{n-1}) \dots (I - L_2)(I - L_1)$$

Wie sieht die Inverse dieser Matrix aus? Es gilt

$$(I + L_j) \cdot (I - L_j) = I - L_j + L_j - L_j^2 = I$$

Dabei verwenden wir: $L_i L_j = \mathbf{0}$ für $i \leq j$;

$$\begin{array}{|c|} \hline \text{ } \\ \hline \end{array} \begin{array}{|c|} \hline \text{ } \\ \hline \end{array} = \mathbf{0}$$

i j

Also:

$$\begin{aligned}\tilde{L}^{-1} &= \left((I - L_{n-1}) \cdots (I - L_1) \right)^{-1} = \\ &= (I - L_1)^{-1} \cdots (I - L_{n-1})^{-1} = \\ &= (I + L_1) \cdots (I + L_{n-1})\end{aligned}$$

und wegen

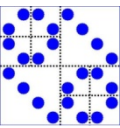
$$(I + L_1) \cdot (I + L_2) = I + L_1 + L_2 + L_1 L_2 = I + L_1 + L_2$$

dann auch

$$\tilde{L}^{-1} = (I + L_1) \cdots (I + L_{n-1}) = I + L_1 + L_2 + \cdots + L_{n-1} = L.$$

Damit ergibt sich:

$$U = \tilde{L}A \quad \Leftrightarrow \quad A = LU$$



Allgemeiner mit Pivotsuche muss noch die Permutation **P** mitberücksichtigt werden, die durch das Auswählen der Pivotelemente hervorgerufen wird. Dann gilt:

$$PA = LU$$

Allgemeines Vorgehen:

Sammele aus dem Algorithmus 3.3.4 der Gauss-Elimination die Gewichte $l_{j,k}$ in Matrix **L** und bezeichne mit **U** die obere Dreiecksmatrix, die sich als Endresultat der GE ergibt.

Sammele die Pivotvertauschungen in Permutationsmatrix **P**.

In dem betrachteten Beispiel:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ -0.3 & -0.04 & 1 \end{pmatrix} \cdot \begin{pmatrix} 10 & -7 & 0 \\ 0 & 2.5 & 5 \\ 0 & 0 & 6.2 \end{pmatrix}$$

Vorteil: Nach einmaliger Durchführung und Speicherung von L und U kann jedes weitere Gleichungssystem in A schnell gelöst werden:

$$Ax = P^T L(Ux) = b \quad \rightarrow \quad Ly = Pb \quad \text{und} \quad Ux = y$$

Bem.: Speichere ev. L im Array von A an Stelle der entstehenden Nullen.

3.5 Die Kondition eines linearen Gleichungssystems (einer Matrix)

Neu benötigt: Matrixnorm und ihre Eigenschaften.

$\| \cdot \|$ Matrixnorm: $\|A\| > 0$ für $A \neq 0$

$$\|a \cdot A\| = |a| \cdot \|A\| \text{ für } a \in \mathbb{R}$$

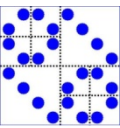
$$\|A+B\| \leq \|A\| + \|B\|$$

Besonders wichtig sind Matrixnormen, die zu einer Vektornorm „passen“ :

3.5.1. Eine Matrixnorm ist mit einer Vektornorm verträglich, wenn

$$\|Ax\| \leq \|A\| \cdot \|x\|$$

Vektor
Matrix
Vektornorm



3.5.2. Eine Matrixnorm heißt submultiplikativ, wenn stets gilt

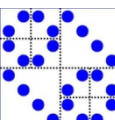
$$\|A \cdot B\| \leq \|A\| \cdot \|B\|$$

3.5.3. Eine submultiplikative Matrixnorm, verträglich mit einer vorgegebenen Vektornorm, kann leicht definiert werden durch

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}$$

die sog. Grenzen-Norm oder lub-Norm (least upper bound):

$$\frac{\|Ay\|}{\|y\|} \leq \sup_x \frac{\|Ax\|}{\|x\|} = \|A\| \Rightarrow \|Ay\| \leq \|A\| \cdot \|y\| \quad \forall y \neq 0$$



Für $Bx \neq 0$:

$$\begin{aligned} \|A \cdot B\| &= \sup \frac{\|ABx\|}{\|x\|} = \sup_x \left(\frac{\|ABx\|}{\|Bx\|} \cdot \frac{\|Bx\|}{\|x\|} \right) \leq \\ &\leq \sup_{y=Bx} \frac{\|Ay\|}{\|y\|} \cdot \sup_x \frac{\|Bx\|}{\|x\|} \leq \|A\| \cdot \|B\| \end{aligned}$$

So erhält man zu den Vektornormen $\|x\|_1 = \sum_{i=1}^n |x_i|$
 $\|x\|_\infty = \max_{i=1, \dots, n} |x_i|$

dazugehörige Matrix-Grenzen-Normen $\|\cdot\|_1$ und $\|\cdot\|_\infty$.

3.5.4. Das innere Produkt

$$(x, y) = x^T y = \sum x_i \cdot y_i$$

führt auf die

3.5.5. Euklid'sche Norm:

$$\|x\|_2^2 = (x, x) = x^T \cdot x = \sum x_i^2$$

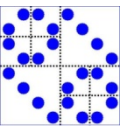
und die damit verträgliche **Matrixnorm**

3.5.6. (2-Norm): $\|A\|_2 = \sup_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2}$ als maximale Längenänderung

mit der Eigenschaft

$$\|Ax\|_2^2 = (Ax, Ax) = (Ax)^T (Ax) = x^T A^T Ax = x^T (A^T A)x$$

$$\|A\|_2^2 = \sup_{x \neq 0} \frac{\|Ax\|_2^2}{\|x\|_2^2} = \sup_{x \neq 0} \frac{x^T (A^T A)x}{x^T x} = \lambda_{\max}(A^T A)$$



3.5.7. Anmerkung:

Speziell wichtige Klasse von Matrizen mit Norm 1:

Q ist orthogonale Matrix \leftrightarrow $Q^{-1} = Q^T$
 oder $Q Q^T = I = Q^T Q$,

$$\|Qx\|_2^2 = x^T Q^T Qx = x^T x = \|x\|_2^2;$$

$$\|Qx\|_2 = \|x\|_2 \quad \rightarrow \quad \|Q\|_2 = \sup_{x \neq 0} \frac{\|Qx\|_2}{\|x\|_2} = 1$$

Basistransformation auf "ideales" Koordinatensystem:

$$\|A\|_2 = \|U^T \Lambda U\|_2 = \|\Lambda\|_2 = \max_{i=1,\dots,n} |\lambda_i|$$

Eigenwertzerlegung von $A=A^T$. U ist orthogonale Matrix

Sei A nun eine allgemeine Matrix \rightarrow SVD

Singuläre Werte Zerlegung $A=U^T \Sigma V$

mit Basis zu $A^T A$ und $A A^T \rightarrow$ zwei „ideale“ Koordinatensysteme

$A = U^T \Sigma V$ Wobei U die Eigenvektoren von $A A^T$ sind,
 V die Eigenvektoren von $A^T A$, und
 Σ , die sog. Singulären Werte von A
sind die Wurzeln aus den Eigenwerten
von $A^T A$ und auch von $A A^T$.

$$\|A\|_2 = \|U^T \Sigma V\|_2 = \|\Sigma\|_2 = \max_{i=1,\dots,n} \sqrt{|\lambda_i(A^T A)|} = \sigma_{\max}$$

Also $\|A\|_2 = \sqrt{\lambda_{\max}(A^T A)} = \sigma_{\max}(A)$

wobei $\lambda_{\max}(A^T A)$ der größte Eigenwert von $A^T A$ ist, und $\sigma_{\max}(A)$ der größte Singulärwert von A .

Diese Matrixnorm ist i.A. nicht einfach berechenbar!

3.5.8. Def. der Frobeniusnorm: $\|A\|_F := \sqrt{\sum |a_{j,k}|^2}$

Fasse die Matrix A als Vektor auf und benutze für diesen langen Vektor die euklid'sche Vektornorm $\rightarrow \|\cdot\|_F$

Die Frobeniusnorm ist mit der euklid'scher Vektornorm verträglich und submultiplikativ.

3.6 Gestörte Eingabedaten

Für ein lineares Gleichungssystem $\mathbf{A} \mathbf{x} = \mathbf{b}$ mit Matrix \mathbf{A} , Vektor der rechten Seite \mathbf{b} und gesuchtem Lösungsvektor \mathbf{x} , untersuchen wir wieder den Einfluss von Eingabefehlern bei sonst exakter Rechnung

→ Kondition der Matrix

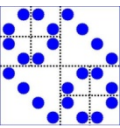
Eingabedaten mit Fehler: $b \rightarrow b + \Delta b = \tilde{b}$

Damit ergibt sich an Stelle der exakten Lösung x die Näherung

$$x \rightarrow x + \Delta x = \tilde{x}$$

Mit $\mathbf{A} \mathbf{x} = \mathbf{b}$ und $\mathbf{A} \tilde{\mathbf{x}} = \mathbf{A}(\mathbf{x} + \Delta \mathbf{x}) = \mathbf{b} + \Delta \mathbf{b} = \tilde{\mathbf{b}}$

gilt: $\mathbf{A} \Delta \mathbf{x} = \Delta \mathbf{b}$ **oder** $\Delta \mathbf{x} = \mathbf{A}^{-1} \Delta \mathbf{b}$



Die Matrix \mathbf{A} wird hier als exakt angenommen!
Damit erhält man die Ungleichungen

$$\|\Delta x\|_2 = \|\mathbf{A}^{-1} \Delta b\|_2 \leq \|\mathbf{A}^{-1}\|_2 \|\Delta b\|_2$$

$$\|b\|_2 = \|Ax\|_2 \leq \|A\|_2 \|x\|_2$$

wegen der Verträglichkeit der euklid'schen Vektor- und Matrixnorm.

Also auch $\frac{1}{\|x\|_2} \leq \frac{\|A\|_2}{\|b\|_2}$ und damit insgesamt:

3.6.1.

$$\frac{\|\Delta x\|_2}{\|x\|_2} \leq \left(\|\mathbf{A}^{-1}\|_2 \cdot \|\mathbf{A}\|_2 \right) \cdot \frac{\|\Delta b\|_2}{\|b\|_2}$$

rel. Fehler in x **Kondition von A** **rel. Fehler in b**

3.6.2 Definition: Die Kondition der Matrix A bzgl. der euklid'schen Norm ist gegeben durch

$$\text{cond}_2(A) = \|A^{-1}\|_2 \cdot \|A\|_2$$

(Genauso kann man Konditionszahlen bzgl. anderer verträglicher Normen definieren.)

Die Konditionszahl beschreibt also wieder, wie sich ein relativer Eingabefehler in Vektor b auf das Resultat, den Lösungsvektor x , auswirkt.

$\text{cond}(A)$ groß \rightarrow kleine Störungen in b bewirken große Fehler in x

$$\text{cond}_2(Q) = \|Q^{-1}\|_2 \|Q\|_2 = \|Q^T\|_2 \|Q\|_2 = 1$$

Außerdem gilt

$$\text{cond}_2(QA) = \text{cond}_2(A)$$

denn

$$\|QA\|_2 = \sup_{x \neq 0} \frac{\|QAx\|_2}{\|x\|_2} = \sup_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \|A\|_2$$

Orthogonale Matrizen sind selbst gut konditioniert und lassen bei Multiplikation mit einer Matrix die Kondition der Ausgangsmatrix unverändert!

Man beachte: Der relative Fehler des Gesamtvektors wird abgeschätzt, nicht der Fehler einzelner Komponenten!

Als Vektor betrachtet ist $\begin{pmatrix} 10^{-5} \\ 1 \end{pmatrix} \approx \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ 

Aber komponentenweise ist natürlich $10^{-5} \neq 0$



Beispiel: $A = \begin{pmatrix} 10^{-9} & 1 \\ 0 & 1 \end{pmatrix}$ und $A^{-1} = \begin{pmatrix} 10^9 & -10^9 \\ 0 & 1 \end{pmatrix}$ 

Normen: $\|A\|_2 \approx \sqrt{2}$, $\|A^{-1}\|_2 \approx \sqrt{2} \cdot 10^9$, $\text{cond}_2(A) = 2 \cdot 10^9$

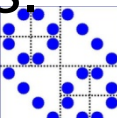
Wähle speziell: $b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $x = A^{-1}b = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $\Delta b = \begin{pmatrix} \Delta b_1 \\ 0 \end{pmatrix}$,

Dann folgt $|\Delta x_1| = 10^9 \cdot |\Delta b_1|$, $\Delta x_2 = 0$ $\|x\| = 1$, $\|b\| = \sqrt{2}$

Damit ergibt sich $\frac{\|\Delta x\|}{\|x\|} = \frac{10^9 \cdot |\Delta b_1|}{1} = \sqrt{2} \cdot 10^9 \cdot \frac{\|\Delta b\|}{\|b\|}$

Vgl.: $\text{cond}(A) = \|A\| \cdot \|A^{-1}\| = \sqrt{2} \cdot \sqrt{2} \cdot 10^9 = 2 \cdot 10^9$

Ein kleiner Fehler in der ersten Komponente von b wirkt sich verstärkt um Faktor 10^9 in der ersten Komponente von x aus.



3.7 Kosten der Gauss-Elimination

Zunächst Dreieckssystem:

$$x_n = b_n / a_{nn};$$

für $i = n-1, n-2, \dots, 1$:

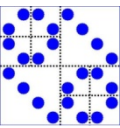
$$x_i = \frac{b_i - \sum_{j=i+1}^n a_{ij} x_j}{a_{ii}}$$

Programm:

```

FOR i = n, ..., 1 DO
  x(i) = b(i);
  FOR j=i+1, ..., n DO
    x(i) = x(i) - a(i,j)x(j);
  ENDFOR
  x(i) = x(i)/a(i,i);
ENDFOR

```



In jedem Schritt i fallen eine Division und jeweils $n-i$ Additionen und Multiplikationen an:

Also
$$\sum_{i=1}^{n-1} (n-i) = \sum_{j=1}^{n-1} j = \frac{n(n-1)}{2} = \frac{n^2}{2} - \frac{n}{2}$$

Additionen und genauso viele Multiplikationen.

Dazu kommen n Divisionen.

3.7.2. Definition: Unter flop verstehen wir eine elementare ‚floating point operation‘ (Gleitpunktoperation $+$, $-$, $*$, $/$). Die Kosten eines Algorithmus werden üblicherweise in flop angegeben. Dazu gibt man in erster Näherung nur den Term höchster Ordnung an.

In unserem Fall:
$$2 \cdot \left(\frac{n^2}{2} - \frac{n}{2} \right) + n = n^2 \rightarrow n^2 \text{ flop}$$

oder
$$O(n^2) \text{ flop}$$

3.7.3. Exkurs: Landau'sche Symbole:

1. Betrachte Funktion in n für $n \rightarrow \infty$:

$$f(n) = O(g(n)), \quad \text{falls} \quad \left| \frac{f(n)}{g(n)} \right| \leq M < \infty \quad \text{für} \quad n \geq N$$

Beispiel: $n^2 + n = O(n^2)$; denn n^2 ist der am stärksten wachsende Term $\rightarrow f(n) / g(n) = 1 + 1/n \leq M$

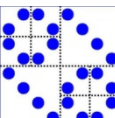
2. Betrachte Funktion in h für $h \rightarrow 0$:

$$f(h) = O(g(h)), \quad \text{falls} \quad \left| \frac{f(h)}{g(h)} \right| \leq c < \infty \quad \text{für} \quad |h| \leq \delta$$

Beispiel: $h^2 + h = O(h)$; denn h ist der am langsamsten schrumpfende Term $\rightarrow f(h) / g(h) = h + 1 \leq c$



$O(..)$ bezeichnet also jeweils den dominanten Term!

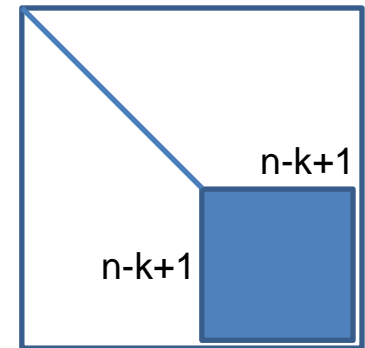


3.7.4. Kosten der Gauss-Elimination:

Im k -ten Teilschritt arbeitet man in einer $(n-k+1) \times (n-k+1)$ Untermatrix A_k

In dieser Matrix wird für $i=k+1, \dots, n$ neu berechnet :

$$a_{ij} = a_{ij} - l_{ik} a_{kj}$$



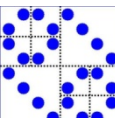
Das sind ca. $(n-k)^2$ Additionen und genauso viele Multiplikationen.

Insgesamt also

$$2 \sum_{k=1}^{n-1} (n-k)^2 = 2 \sum_{j=1}^{n-1} j^2 = \frac{2(n-1)(2n-1)n}{6} = \frac{2}{3}n^3 + O(n^2) \quad \text{flop}$$

Dazu kommen $O(n^2)$ flop für die Spaltenpivotsuche und $O(n^2)$ flop für das Auflösen des Dreiecksgleichungssystems.

Diese Kosten fallen aber praktisch nicht ins Gewicht gegenüber den obigen $\frac{2}{3}n^3$.



3.7.5 Beispiel zur Verdeutlichung der Kondition

$$A = \begin{pmatrix} 1 & 0 & \dots & 0 & 1 \\ -1 & 1 & \ddots & \vdots & 1 \\ -1 & -1 & \ddots & 0 & 1 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ -1 & -1 & \dots & -1 & 1 \end{pmatrix} \begin{matrix} \downarrow \\ \\ \\ \vdots \\ \downarrow \end{matrix} \begin{matrix} + \\ \\ \\ \downarrow \end{matrix}$$

Eliminiere die erste Spalte durch Addieren der ersten Zeile:

$$\begin{pmatrix} 1 & & & & 1 \\ 0 & 1 & & & 2 \\ 0 & -1 & 1 & & 2 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & -1 & \dots & -1 & 2 \end{pmatrix} \begin{matrix} \downarrow \\ \\ \\ \vdots \\ \downarrow \end{matrix} \begin{matrix} + \\ \\ \\ \downarrow \end{matrix}$$

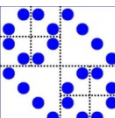
Im nächsten Schritt

$$\begin{pmatrix} 1 & & & & 1 \\ 0 & 1 & & & 2 \\ 0 & 0 & 1 & & 4 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & -1 & 4 \end{pmatrix}$$

und schließlich

$$U = \begin{pmatrix} 1 & & & & 1 \\ & 1 & & & 2 \\ & & 1 & & 4 \\ & & & \ddots & \vdots \\ & & & & 1 & 2^{n-2} \\ & & & & & 2^{n-1} \end{pmatrix}$$

In jedem Schritt verdoppelt sich der größte Eintrag in der Matrix!



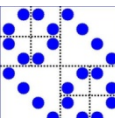
Kondition von A selbst ist $O(n)$, Kondition von U ist $O(2^{n-1})$!

Hilft Pivotsuche?

Im Verlauf der Gauss-Elimination *kann* die Kondition der Matrizen sehr stark anwachsen!

Aber: In der Praxis kommt das *so gut wie* nie vor!

Gauss-Elimination mit Pivotsuche *gilt als* numerisch stabil.



3.8 Methode der kleinsten Quadrate

(Least Squares, Normalgleichung)

Ausgangspunkt: Überbestimmtes System.

Mehr Gleichungen als Unbekannte

$$\boxed{\mathbf{A}} \mathbf{x} = \mathbf{b}$$

Sei A eine $m \times n$ – Matrix mit $m > n$ und maximal vollem Rang:

$\text{rang}(A) = n$, d.h. A bildet den \mathbf{R}^m in den ganzen \mathbf{R}^n ab.

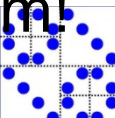
Das System $Ax = b$ ist dann i.A. nicht lösbar!

Was bedeutet es, wenn A nicht vollen Rang hat?

Versuche, das Problem so gut wie möglich zu lösen!

Kompromiss!

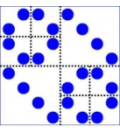
Minimiere dazu die Abweichung $Ax - b$ in passender Norm!

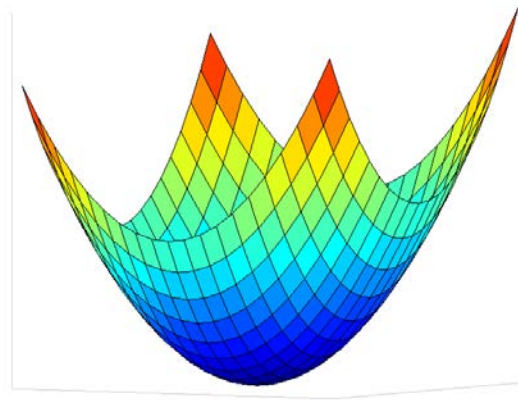


Am besten eignet sich dazu die euklid'sche Norm, da sie auf eine differenzierbare Funktion f führt:

$$3.8.1. : \min_x \|Ax - b\|_2^2$$

$$\begin{aligned} f(x_1, \dots, x_n) &:= \|Ax - b\|_2^2 = \\ &= (Ax - b)^T (Ax - b) = x^T A^T Ax - 2x^T A^T b + b^T b = \\ &= \left\| \begin{pmatrix} \left(\sum_{j=1}^n a_{1,j} x_j \right) - b_1 \\ \vdots \\ \left(\sum_{j=1}^n a_{m,j} x_j \right) - b_m \end{pmatrix} \right\|_2^2 = \sum_{k=1}^m \left(\sum_{j=1}^n a_{k,j} x_j - b_k \right)^2 \end{aligned}$$





Die Funktion f beschreibt einen Paraboloiden (n-dim. Parabel).

Das eindeutige Minimum dieser Funktion ist an der Stelle, an der die Ableitung gleich Null ist (waagrechte Tangente).

$$0 = \frac{df}{dx_i} = 2 \sum_{k=1}^m \left(\sum_{j=1}^n a_{k,j} x_j - b_k \right) a_{k,i} \quad \text{für } i=1, \dots, n$$

oder

$$\sum_{k=1}^m a_{k,i} \sum_{j=1}^n a_{k,j} x_j = \sum_{k=1}^m a_{k,i} b_k$$

In Matrixschreibweise: $(A^T Ax)_i = (A^T b)_i$, $i=1, \dots, n$

3.8.2. Normalgleichung zu $Ax=b$: $A^T Ax = A^T b$

$$A^T \cdot (Ax = b) \Rightarrow A^T Ax = A^T b$$

Die Matrix $A^T A$ ist eine $n \times n$ – Matrix von Rang n (da A Rang n hat) und beschreibt daher ein eindeutig lösbares, quadratisches lineares Gleichungssystem.

Allerdings ist die Kondition von $A^T A$ oft sehr viel schlechter als die von A , denn:

$$\begin{aligned}
 \text{cond}_2(A^T A) &= \|A^T A\|_2 \cdot \|(A^T A)^{-1}\|_2 = \\
 &= \sqrt{\lambda_{\max}(A^T A A^T A) * \lambda_{\max}((A^T A)^{-1} (A^T A)^{-1})} = \\
 &= \lambda_{\max}(A^T A) \cdot \lambda_{\max}(\text{inv}(A^T A)) = \\
 &= \lambda_{\max}(A^T A) / \lambda_{\min}(A^T A) = \sigma_{\max}^2(A) / \sigma_{\min}^2(A) = \\
 &= \|A\|_2^2 \cdot \|A^{-1}\|_2^2 = \text{cond}^2(A)
 \end{aligned}$$

Im folgenden Abschnitt werden wir daher ein besseres Verfahren zur Lösung dieses Problems kennen lernen.

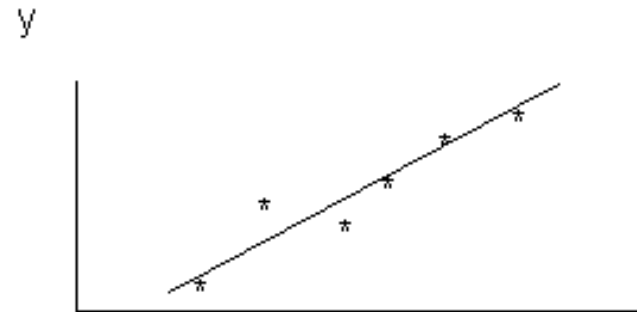
Dazu werden besser orthogonale Matrizen verwendet, um diese Konditionsverschlechterung zu vermeiden.

3.8.3. Lineares Ausgleichsproblem (Ausgleichsgerade)

Gegeben: Punktepaare in der Ebene , (x_i, y_i) , $i=1, \dots, n$;

Gesucht: beste Gerade, die möglichst nahe an den Punkten liegt.

$$y = g(x) = ax + b .$$



Es soll also gelten:

$$\begin{pmatrix} a + bx_1 \\ \vdots \\ a + bx_n \end{pmatrix} \approx \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

oder in Matrixschreibweise

$$A \begin{pmatrix} a \\ b \end{pmatrix} \approx y \quad \text{mit} \quad A = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}, \quad \text{und} \quad y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}.$$

Die Normalgleichung lautet also

$$A^T A \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} n & \sum_{j=1}^n x_j \\ \sum_{j=1}^n x_j & \sum_{j=1}^n x_j^2 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^n y_j \\ \sum_{j=1}^n x_j y_j \end{pmatrix}.$$

Die Lösung dieses 2×2 – Gleichungssystems liefert a und b , und damit die gesuchte Gerade $y = ax + b$.

Allgemeiner:

Ansatzfunktionen $g_1(x), \dots, g_m(x)$ z.B. $\cos(kx)$ und

Punkte $(x_1, y_1), \dots, (x_n, y_n)$, $n > m$

Gesucht : $f(x) = \sum_{k=1}^m a_k g_k(x)$ mit $f(x_j) \approx y_j, j = 1, \dots, n$

Mit $G = \begin{pmatrix} g_1(x_1) & \cdots & g_m(x_1) \\ \vdots & & \vdots \\ g_1(x_n) & \cdots & g_m(x_n) \end{pmatrix}$ ist dann $G^T G \begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix} = G^T \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$

zu lösen.

Ergebnis ist die ‚näheste‘ Funktion an den vorgegebenen Punkten, die aus den g_1, \dots, g_m linear zusammengesetzt ist.

Beispiel: Schwingungsanteile

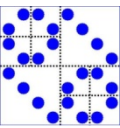
Gemessenes Audio-Signal $y = (y_1, \dots, y_n)$ an Stellen $x = (x_1, \dots, x_n)$

Gesucht: Frequenzen, Schwingungsanteile

Ansatzfunktionen: $g_k(x) = \cos(kx), k = 0, \dots, m$

$$f(x) = a_0 + a_1 \cos(x) + \dots + a_m \cos(mx) \quad \longrightarrow \quad f(x_j) = y_j, \quad j = 1, \dots, n$$

$$\begin{pmatrix} 1 & \cos(x_1) & \cdots & \cos((m-1)x_1) & \cos(mx_1) \\ 1 & \cos(x_2) & \cdots & \cos((m-1)x_2) & \cos(mx_2) \\ \vdots & \vdots & & \vdots & \vdots \\ 1 & \cos(x_{n-1}) & \cdots & \cos((m-1)x_{n-1}) & \cos(mx_{n-1}) \\ 1 & \cos(x_n) & \cdots & \cos((m-1)x_n) & \cos(mx_n) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{m-1} \\ a_m \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix}$$



3.9. Die QR-Zerlegung einer Matrix

Schon vorher haben wir bemerkt:

- $\text{cond}(U)$ in der Gauß-Elimination ev. groß, auch bei kleinem $\text{cond}(A)$;
- falls A schlecht konditioniert: was ist der Rang von A ? Welche Pivotelemente werden wir als 0?
- $\text{cond}(A^T A)$ oft sehr groß.

Andererseits: $\text{cond}(QA) = \text{cond}(A)$, falls Q orthogonal.

Also sind orthogonale Matrizen sehr gut für äquivalente Umformungen von A geeignet (vgl. LU-Zerlegung).

Außerdem gilt: $\mathbf{Q}^{-1} = \mathbf{Q}^T$.

Also sind Gleichungssysteme in \mathbf{Q} sehr leicht zu lösen.

Versuche daher, analog zur LU-Zerlegung $\mathbf{A}=\mathbf{LR}$,
eine Zerlegung der Form

$$\mathbf{A} = \mathbf{QR}$$

zu bestimmen mit

\mathbf{Q} orthogonal

\mathbf{R} obere Dreiecksmatrix

Vorteile:

- numerisch stabiler als LU
- ähnliche Kosten
- Systeme in \mathbf{Q} und \mathbf{R} leicht zu lösen

3.9.2 Elementare orthogonale Matrizen



Orthogonale 2 x 2 – Matrix :

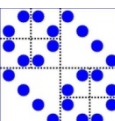
Frage: orthogonale 1 x 1 – Matrix?

$$G = \begin{pmatrix} \cos(\varphi) & \sin(\varphi) \\ \sin(\varphi) & -\cos(\varphi) \end{pmatrix} \quad \text{heißt **Givensreflexion**.} \quad \text{Denn}$$

$$G^T G = GG =$$

$$= \begin{pmatrix} \cos(\varphi) & \sin(\varphi) \\ \sin(\varphi) & -\cos(\varphi) \end{pmatrix} \begin{pmatrix} \cos(\varphi) & \sin(\varphi) \\ \sin(\varphi) & -\cos(\varphi) \end{pmatrix} =$$

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



$$\left\{ \text{Alternativ **Givensrotation:** } \begin{pmatrix} \cos(\varphi) & \sin(\varphi) \\ -\sin(\varphi) & \cos(\varphi) \end{pmatrix} \right\}$$

G ist eindeutig bestimmt durch den 'Winkel' φ .
Bestimme nun φ so, dass

$$\tilde{A} = GA = \begin{pmatrix} \tilde{a}_{11} & \tilde{a}_{12} \\ \tilde{a}_{21} & \tilde{a}_{22} \end{pmatrix} \quad \text{obere Dreiecksmatrix wird.}$$

Dazu muss gelten:

$$\tilde{a}_{21} = (\sin(\varphi) \quad -\cos(\varphi)) \begin{pmatrix} a_{11} \\ a_{21} \end{pmatrix} = \sin(\varphi)a_{11} - \cos(\varphi)a_{21} \stackrel{!}{=} 0$$

Lösung:

$$\cot(\varphi) = \frac{a_{11}}{a_{21}}; \quad \varphi = \operatorname{arcctg}\left(\frac{a_{11}}{a_{21}}\right) \quad \text{oder} \quad \varphi = \operatorname{arctg}\left(\frac{a_{21}}{a_{11}}\right)$$

Ist $a_{21} = 0$, so ist keine weitere Transformation nötig!

Numerisch stabilere Art der Berechnung :

(a_{21} oder a_{11} könnten fast 0 sein):

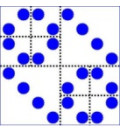
$$\rho = \operatorname{sign}(a_{11})\sqrt{a_{11}^2 + a_{21}^2}; \quad \cos(\varphi) = \frac{a_{11}}{\rho}; \quad \sin(\varphi) = \frac{a_{21}}{\rho};$$

3.9.3. Givens-Reflexion für den allgemeinen $n \times n$ – Fall:

n -dimensionale Givens-Reflexion ist im Wesentlichen wie die Einheitsmatrix, bis auf den gerade zu betrachtenden 2×2 – Block.

Dieser Block wird wieder - wie oben definiert - abhängig von φ bestimmt.

Man eliminiert wieder in der ersten Spalte $a_{2,1}, \dots, a_{m,1}$, und dann entsprechend in der zweiten Spalte die Unterdiagonale, usw. wie bei Gauss.



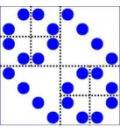
Zur Elimination eines Elementes a_{ij} der Matrix A multiplizieren wir $G_{ij} \cdot A$.

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.

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



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

verändert nur j-te und i-te Zeile;

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

speziell: $a_{ij} \rightarrow sa_{jj} - ca_{ij} = 0$

für $k=j, \dots, n$

! 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

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

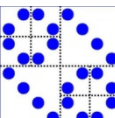
zur Bearbeitung der ersten Spalte,

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.



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:

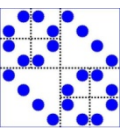
$$\text{cond}(A) = \text{cond}(QR) = \text{cond}(R)$$

Gut für schlecht konditionierte Systeme

Anwendbar auf rechteckige Systeme

Andere Orthogonalisierungsverfahren:

- Gram-Schmidt (orthonormalisiere Vektoren)
- Householder (erzeuge in einem Schritt eine ganze Nullspalte). $H = I - 2 u u^T$

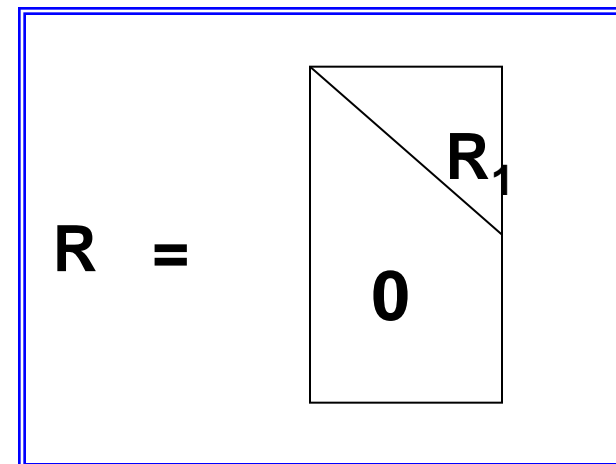


3.9.4. Anwendung bei Linearer Ausgleichsrechnung:

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

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 $\left\| \tilde{b}_2 \right\|_2^2$