

2. Matrix-Zerlegungen

Lösen linearer Gleichungssysteme ist zentral!

Literatur: K. Nipp / D. Stoffer: Lineare Algebra

Software: Matlab "Matlab is a strongly typed language. It has **one** type: the complex matrix"

Grundaufgabe: $A \underline{x} = \underline{b}$
 $\underline{x}, \underline{b} \in \mathbb{R}^n, A \in \mathbb{R}^{n \times n}$

- Berechne **die** Lösung \underline{x} , falls A regulär
- Berechne **alle** Lösungen (ev. keine!), f. A singular

Lineare Algebra: $\underline{x} = A^{-1} \underline{b}$ falls A regulär

Matlab: $\underline{x} = A \setminus \underline{b}$ "Linksdivision"

Bewirkt Ausführung des genuesten, effizientesten heute bekannten Algorithmus zur Lösung des lin. Gl'systems $A \underline{x} = \underline{b}$

Viele gute Algorithmen dazu beruhen auf Faktorzerglegungen der Matrix A .

2.1. Die LR-Zerlegung (LU decomposition)

Wir suchen Faktorisierung

$$A = \left(\begin{array}{c|c} \begin{array}{ccc} 1 & & 0 \\ & \ddots & \\ & & 1 \end{array} & \begin{array}{c} 0 \\ \vdots \\ 0 \end{array} \\ \hline \begin{array}{ccc} L & & \\ & \ddots & \\ & & 1 \end{array} & \begin{array}{c} \\ \vdots \\ \\ \end{array} \end{array} \right) \left(\begin{array}{c} R \\ \hline 0 \end{array} \right)$$

Links-(lower)

Rechts-Dreiecksm. (upper)

Ist $A = L \cdot R$ bekannt, so ist die Auflösung einfach:

$$\underline{A} \underline{x} = \underline{b} \Rightarrow \underline{L} (\underline{R} \underline{x}) = \underline{b}$$

(i) Berechne \underline{y} aus $\underline{L} \underline{y} = \underline{b}$

durch "Vorwärtseinsetzen", d.h.

berechne der Reihe nach y^1, y^2, \dots, y^n

(die jeweilige Gleichung hat nur 1 Unbek.)

(ii) Berechne \underline{x} aus $\underline{R} \underline{x} = \underline{y}$

durch "Rückwärtseinsetzen", d.h.

berechne der Reihe nach x^n, x^{n-1}, \dots, x^1 .

Der wesentliche Schritt: Zerlegung $A = L \cdot R$

Sei $A = (a_{ij})$; Voraussetzung: $\boxed{a_{11} =: \alpha \neq 0}$
(Pivot)

Praktische Notation: Partitionierung $n = 1 + (n-1)$

$$A = \begin{array}{|c|c|} \hline \alpha & q^T \\ \hline p & A_1 \\ \hline \end{array} = \underbrace{\begin{array}{|c|c|} \hline 1 & 0 \\ \hline l & L_1 \\ \hline \end{array}}_L \cdot \underbrace{\begin{array}{|c|c|} \hline p & r^T \\ \hline 0 & R_1 \\ \hline \end{array}}_R = \begin{array}{|c|c|} \hline p & r^T \\ \hline lp & lr^T + L_1 R_1 \\ \hline \end{array}$$

Man setze

$$p = \alpha, \quad r^T = q^T, \quad l = \frac{1}{\alpha} \cdot p$$

und zerlege analog die Untermatrix

$$A_1 - l r^T = L_1 \cdot R_1$$

Notwendig für Fortgang des Algorithmus: $(A_1 - l r^T)_{11} \neq 0$.

Zeilenvertauschungen

Def: Die Matrix $P \in \mathbb{R}^{n \times n}$ heißt Permutationsmatrix, falls jede Zeile und jede Kolonne genau ein Element 1 enthält, sonst Nullen.

Eigenschaften:

- Sei $P = \begin{bmatrix} 0 & \dots & 1 & \dots & 0 \\ 0 & 1 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}$,

$$A = \begin{bmatrix} \underline{a}_1^T \\ \vdots \\ \underline{a}_n^T \end{bmatrix} \Rightarrow PA = \begin{bmatrix} \underline{a}_{i_1}^T \\ \underline{a}_{i_2}^T \\ \vdots \end{bmatrix}$$

\underline{a}_i^T :
Zeile
Nr. i
von A

- P ist orthogonal, $P^{-1} = P^T$

SATZ: Für jede reguläre Matrix $A \in \mathbb{R}^{n \times n}$ existieren Matrizen P (Perm'matrix), L (Linksdreiecksmatrix mit Diag. 1), R (Rechtsdr'matrix mit Diag. $\neq 0$), so dass

$$PA = LR$$

Matlab: $[L, R, P] = \text{lu}(A)$ liefert P, L, R

$[M, R] = \text{lu}(A)$ liefert

$$M = P^{-1}L, R$$

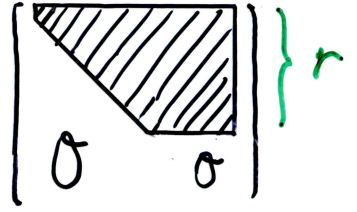
d.h. es gilt $A = MR$

M : "psychologically lower triangular"

Bemerkungen

(i) A singular, Rang $r < n$:

Es gilt $PA = LR$ mit



(ii) Determinante $\det(A) \in \mathbb{R}$

A regulär $\Leftrightarrow \det(A) \neq 0$

A singular $\Leftrightarrow \det(A) = 0$

Bem: $\det(A)$ ist das Volumen des von den Kolonnen von A aufgespannten "Parallelepiped"

$\det(A)$ kann aus der Zerlegung von A abgelesen werden:

$$\det(A) = (-1)^v \cdot \prod_{i=1}^n r_{ii}$$

v = Anzahl der Vertauschungen zur Herstellung der Permutation P

(iii) Vorteile der LR-Zerlegung:

- Speicherung der Faktoren L, R, P an Ort und Stelle von A, zusammen mit einem Vektor der Länge n für P.

- Zerlegungsaufwand $\mu_{LR} = \frac{n^3}{3} + O(n^2)$ Mult.
 $\mu_{Solve} = n^2 + O(n)$ Mult.

Man kann mehrere Systeme $A \underline{x}_j = \underline{b}_j$ mit derselben Matrix billig lösen

Zusammen:

$X = (\underline{x}_1, \dots, \underline{x}_L), B = (\underline{b}_1, \dots, \underline{b}_L) \Rightarrow A \cdot X = B$

Matlab:

$X = A \setminus B$

2.2. Kondition, Normen

$$A \underline{x} = \underline{b}, \quad \underline{b}, \underline{x} \in \mathbb{R}^n, \quad A \in \mathbb{R}^{n \times n}$$

Ist A fast singular, kann die berechnete (approximative) Lösung \tilde{x} ungenau sein, auch wenn das Residuum

$$\underline{r} := A \tilde{x} - \underline{b}$$

klein ist.

Beispiel: $n=2, \quad A = \begin{pmatrix} 137 & 100 \\ 100 & 73 \end{pmatrix} \Rightarrow A^{-1} = \begin{pmatrix} 73 & -100 \\ -100 & 137 \end{pmatrix}$

Mit $\underline{b} = \begin{pmatrix} 4.3 \\ 3.1 \end{pmatrix} \Rightarrow \underline{x} = \begin{pmatrix} 3.9 \\ -5.3 \end{pmatrix}$

Störung der rechten Seite:

Sei $\tilde{\underline{b}} = \begin{pmatrix} 4.31 \\ 3.10 \end{pmatrix} = \underline{b} + \underline{\Delta}, \quad \underline{\Delta} = \begin{pmatrix} .01 \\ 0 \end{pmatrix} \quad (2\%)$

\Rightarrow Lösung $\tilde{\underline{x}} = \underline{x} + A^{-1} \underline{\Delta} = \begin{pmatrix} 4.63 \\ -6.30 \end{pmatrix}$

Jemand behauptet, \tilde{x} löse $A \underline{x} = \underline{b}$

Kontrolle: Residuum $\underline{r} := A \tilde{x} - \underline{b} = \begin{pmatrix} .01 \\ 0 \end{pmatrix}$

scheint OK!

Trotzdem ist \tilde{x} um 20% falsch: $\tilde{\underline{x}} - \underline{x} = \begin{pmatrix} 4.63 \\ -6.30 \end{pmatrix} - \begin{pmatrix} 3.9 \\ -5.3 \end{pmatrix} = \begin{pmatrix} .73 \\ -1 \end{pmatrix}$

Zum Verständnis dieses Phänomens brauchen wir die Begriffe

- Vektornorm Nipp/stoffer S. 83 ff.
- Matrixnorm S. 133 ff.
- Kondition S. 166 ff.

Vektornormen

Sei $\underline{x} \in \mathbb{R}^n$.

Def: Die Zahl $\|\underline{x}\| \in \mathbb{R}$ heißt eine **Norm**, falls die Normaxiome (i), (ii), (iii) gelten:

$$(i) \quad \|\underline{x}\| \geq 0 \quad \forall \underline{x} \in \mathbb{R}^n; \quad \|\underline{x}\| = 0 \Rightarrow \underline{x} = 0$$

$$(ii) \quad \|\alpha \underline{x}\| = |\alpha| \cdot \|\underline{x}\| \quad \forall \alpha \in \mathbb{R}, \underline{x} \in \mathbb{R}^n$$

$$(iii) \quad \|\underline{x} + \underline{y}\| \leq \|\underline{x}\| + \|\underline{y}\| \quad \forall \underline{x}, \underline{y} \in \mathbb{R}^n$$

(Dreiecks-Ungleichung)

Bemerkung: Vektornorm liefert Abstandsbegriff:

$$\|\underline{y} - \underline{x}\| = \text{"Abstand" zwischen Pkten } \underline{x}, \underline{y}$$

Wichtiges konkretes Beispiel einer Norm:

Hölder'sche p -Norm

$$\underline{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

$$\text{Def: } \|\underline{x}\|_p := \left(\sum_{k=1}^n |x_k|^p \right)^{\frac{1}{p}}, \quad p \geq 1$$

Spezialfälle

$$p=2, \quad \|\underline{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$$

Euklidische Norm
(Länge)

$$p=1, \quad \|\underline{x}\|_1 = \sum_{k=1}^n |x_k|$$

Summen-Norm

$$p=\infty, \quad \|\underline{x}\|_\infty = \max_k |x_k|$$

Maximum-Norm

Die Hölder-Norm erfüllt für jedes $p \geq 1$ die Norm-Axiome.

Die (einer Vektornorm) zugeordnete Matrixnorm (2)

Sei $A \in \mathbb{R}^{n \times n}$

$$\text{Def: } \|A\| := \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \max_{\|x\|=1} \|Ax\|$$

$\|A\|$ ist die maximale Länge des Bildes eines Einheitsvektors

(1) Hauptsatz: $\|Ax\| \leq \|A\| \cdot \|x\|$
(folgt direkt aus der Definition)

Berechnung von $\|A\|$

SATZ. Es gilt mit $A = (a_{ij})$:

$$\|A\|_1 = \max_j \sum_i |a_{ij}| \quad \text{maximale Kolonnen-Betrags-Summe}$$

$$\|A\|_\infty = \max_i \sum_j |a_{ij}| \quad \text{maximale Zeilen-Betrags-Summe}$$

Im (wichtigsten) Fall $p=2$ ist die Matrixnorm schwieriger zu berechnen. Es gilt:

$$\|A\|_2 = |\lambda_{\max}|, \text{ falls } A = A^T$$

↑
grösster Eigenwert

allgemein: $\|A\|_2 = \sigma_{\max}$ (grösster Singulärwert, siehe später)

Matlab: $\text{norm}(x)$, $\text{norm}(A)$ liefert 2-Norm

allgemein: $\text{norm}(x, p)$, $\text{norm}(A, p)$
↑ z. B. $p = \infty$

Sei $A \in \mathbb{R}^{n \times n}$ regulär, $\underline{b} \neq 0$.

Betrachte lineares Gleichungssystem:

(2) ungestört $A \underline{x} = \underline{b}$

gestört $A(\underline{x} + \underline{\Delta x}) = \underline{b} + \underline{\Delta b}$

Subtraktion liefert $A \cdot \underline{\Delta x} = \underline{\Delta b}$

(3) $\Rightarrow \underline{\Delta x} = A^{-1} \cdot \underline{\Delta b}$

Wende Hauptsatz (1) auf (2) und (3) an:

$$\|\underline{b}\| \leq \|A\| \cdot \|\underline{x}\|$$

$$\|\underline{\Delta x}\| \leq \|A^{-1}\| \cdot \|\underline{\Delta b}\|$$

Bilde Produkt und dividiere durch $\|\underline{b}\| \cdot \|\underline{x}\|$:

$$\frac{\|\underline{\Delta x}\|}{\|\underline{x}\|} \leq \|A\| \cdot \|A^{-1}\| \cdot \frac{\|\underline{\Delta b}\|}{\|\underline{b}\|}$$

\uparrow relative Störung in \underline{x} \uparrow relative Störung in \underline{b}

Def: $\text{cond}(A) := \|A\| \cdot \|A^{-1}\|$
 (Kondition von A bezügl. der verwendeten Norm)

(4) SATZ: $\frac{\|\underline{\Delta x}\|}{\|\underline{x}\|} \leq \text{cond}(A) \cdot \frac{\|\underline{\Delta b}\|}{\|\underline{b}\|}$

Die Kondition von A ist der maximale Verstärkungsfaktor, mit dem sich die **relative Störung** $\|\underline{\Delta b}\|/\|\underline{b}\|$ von \underline{b} auf die Lösung \underline{x} von $A\underline{x} = \underline{b}$ überträgt.

Bemerkungen

$$(i) \quad \text{cond}(A) = \frac{\max_{\|x\|=1} \|Ax\|}{\min_{\|y\|=1} \|Ay\|}$$

$$\text{z. B. für } A = A^T: \quad \text{cond}_2(A) = \frac{\max |\lambda|}{\min |\lambda|}$$

$$(ii) \quad \text{Matlab: } \text{cond}(A, p)$$

$$\text{cond}(A) := \text{cond}(A, 2)$$

Es wird die exakte Kondition berechnet,
teuer für $p \neq 1, \infty$!

(iii) Man kann $\text{cond}(A)$ (z. B. mit $p=1$)
billig auf 10% Genauigkeit **Schätzen**,
mit einem Zusatzaufwand von nur
 $O(n^2)$ verglichen mit der LR-Zerlegung allein.

(iv) Beispiel von S. 19:

$$\text{cond}_2 \begin{pmatrix} 137 & 100 \\ 100 & 73 \end{pmatrix} = \lambda_1^2 \approx 2 \cdot 10^2 - 2 = 44098$$

Fehleranalyse

Rundungsfehler bei der LR-Zerlegung (d.h. beim Gauss-Algorithmus) wirken sich aus wie eine Störung $\underline{\Delta b} \neq 0$ der rechten Seite.

Sei \underline{x} die exakte Lösung von $A\underline{x} = \underline{b}$,
 $\tilde{\underline{x}} = \underline{x} + \underline{\Delta x}$ das Resultat der numerischen
 Rechnung (z. B. Matlab $\tilde{\underline{x}} = A \setminus \underline{b}$).

Residuum: $\underline{r} = A\tilde{\underline{x}} - \underline{b} = \underline{\Delta b}$

Relatives Residuum: $\frac{\|\underline{r}\|}{\|\underline{b}\|} = \frac{\|\underline{\Delta b}\|}{\|\underline{b}\|}$

Ein guter Algorithmus (z. B. Gauss mit guter Pivotstrategie, d.h. mit guten Zeilenvertauschungen, z. B. realisiert in Matlab) erreicht unabhängig von n , $\text{cond}(A)$

$$(5) \quad \frac{\|\underline{\Delta b}\|}{\|\underline{b}\|} \leq \rho \cdot \text{eps}$$

Dabei ist: $\rho \approx 1$ (meistens)
 ρ selten über 10, häufig $\rho \in (0, 1)$
 eps (eine Matlab-Zahl)
 $\text{eps} = \min \epsilon$ mit $1 + \epsilon > 1$ } "Des Maschinenepsilon"

Damit
 (4), (5) \Rightarrow $\frac{\|\underline{\Delta x}\|}{\|\underline{x}\|} \leq \text{cond}(A) \cdot \rho \cdot \text{eps}$

2.3. QR-Zerlegung und Ausgleichung nach kleinsten Quadraten

Vorbereitung: orthogonale Matrizen

$$Q \in \mathbb{R}^{m \times m} \text{ orthogonal} \iff Q^{-1} = Q^T \quad (QQ^T = I)$$

Eigenschaften:

- Normtreue: $\|Qx\|_2 = \|x\|_2 \quad \forall x \in \mathbb{R}^m$

Die Abbildung $x \mapsto Qx$ ist eine Drehung im \mathbb{R}^m

- $\det(Q) = \pm 1$ (eigentliche / uneigentl. Drehung)

- Euklidische Norm ist die natürliche

Matrixnorm: $\|Q\|_2 = 1$

$$\implies \text{cond}_2(Q) = 1$$

QR-Zerlegung

SATZ: Sei $A \in \mathbb{R}^{m \times n}$ eine beliebige Rechtecksmatrix. Dann existiert eine orthogonale Matrix $Q \in \mathbb{R}^{m \times m}$ und eine Rechtsdreiecksmatrix $R \in \mathbb{R}^{m \times n}$ so dass

$$A = Q \cdot R$$

Fälle:

$$\begin{array}{c}
 m \geq n: \\
 \begin{array}{c}
 \begin{array}{|c|} \hline A \\ \hline \end{array}
 \end{array}
 \end{array}
 =
 \begin{array}{|c|} \hline Q_1 \\ \hline \end{array}
 \begin{array}{|c|} \hline Q_2 \\ \hline \end{array}
 \cdot
 \begin{array}{|c|} \hline R \\ \hline \end{array}
 =
 \begin{array}{|c|} \hline Q_1 \\ \hline \end{array}
 \begin{array}{|c|} \hline R \\ \hline \end{array}$$

$A = Q \cdot R$

$$\begin{array}{c}
 m < n: \\
 \begin{array}{|c|} \hline A \\ \hline \end{array}
 =
 \begin{array}{|c|} \hline Q \\ \hline \end{array}
 \cdot
 \begin{array}{|c|} \hline R \\ \hline \end{array}$$

"economy size decomposition"
 Matlab:
 $[Q_1, R] = qr(A, 0)$

Beweis (konstruktiv, liefert ein Berechnungsverf.)

Idee: $R = Q^T A$

Schrittweiser Aufbau von Q
 als Matrixprodukt von "Elementar-drehungen".

Definition: Givens - Rotation,
 Drehung in der (k, l) -Ebene
 des \mathbb{R}^m um den Winkel φ :

$\underline{x} \in \mathbb{R}^m \mapsto Q_{kl}^T(\varphi) \underline{x}, \quad k \neq l$

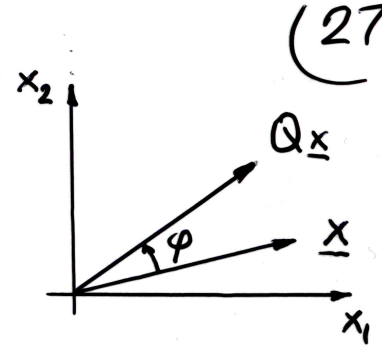
mit $Q_{kl}^T(\varphi) :=$

| | | | | | | |
|----------|------------------|------------------|--|--|--|-----------|
| 1 | | | | | | |
| | $1 \cos \varphi$ | $\sin \varphi$ | | | | zeile k |
| | $-\sin \varphi$ | $1 \cos \varphi$ | | | | zeile l |
| σ | | | | | | |

Kolonne k
Kolonne l

Bemerkung: Ebene Drehung

$\underline{x} \mapsto Q \underline{x}, Q = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix}$



Wirkung der Givens-Rotation auf A:

$A_1 := Q_{ke}^T(\varphi) \cdot A$

Sei $A = \begin{bmatrix} \underline{a}_1^T \\ \vdots \\ \underline{a}_m^T \end{bmatrix}$ gegeben durch Zeilenvektoren

Zeile l $A_1 = \begin{bmatrix} \vdots & \underline{a}_1^T & \vdots \\ \vdots & \underline{a}_k^T \cos\varphi + \underline{a}_l^T \sin\varphi & \vdots \\ \vdots & -\underline{a}_k^T \sin\varphi + \underline{a}_l^T \cos\varphi & \vdots \\ \vdots & \vdots & \underline{a}_m^T \end{bmatrix} = \begin{bmatrix} \vdots & * & \vdots \\ \vdots & * & \vdots \\ \vdots & -a_{kk} \sin\varphi + a_{lk} \cos\varphi & \vdots \\ \vdots & \vdots & * \end{bmatrix}$
Kolonne k

Ziel: Wähle φ so, dass

$(A_1)_{lk} = -a_{kk} \sin\varphi + a_{lk} \cos\varphi = 0$
 $\Rightarrow \varphi = \text{angle}(a_{kk} + i \cdot a_{lk} \sqrt{-1})$ Matlab

Kolonnenweise alle Elemente $\neq 0$ des unteren Dreiecks von A "anullieren":

$R = \underbrace{Q_{m-1,m}^T(\varphi_{m-1,m}) \cdots Q_{2,m}^T \cdots Q_{23}^T \cdot Q_{1m}^T \cdots Q_{13}^T Q_{12}^T(\varphi_{12})}_{Q^T} \cdot A$
 Q^T , nur aufbauen, wenn gewünscht

Anwendung: Ausgleichung nach kleinsten Quadraten
(Least-squares approximation)

Typisches Problem: Lineares Modell für eine Messreihe

Gegeben: ^{viele} m Zeitpunkte $t_k, k=1, 2, \dots, m$ (exakt)
 m Messwerte $f_k, k=1, \dots, m$ (fehlerbehaftet)
außerdem $n \leq m$ Basisfunktionen
^{wenige} $\Phi_\ell(t), \ell=1, 2, \dots, n$

Betrachte "lineares Modell"

$$\Phi(t) := \sum_{\ell=1}^n \Phi_\ell(t) \cdot c_\ell, \quad \text{linear in } c_\ell$$

das die Messreihe approximiert: $\Phi(t_k) \approx f_k \quad \forall k$

Gesucht: Koeffizientenvektor $c = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} \in \mathbb{R}^n$

Lösung: Betrachte Residuenvektor

$$r = \begin{bmatrix} r_1 \\ \vdots \\ r_m \end{bmatrix} \in \mathbb{R}^m, \quad r_k = \Phi(t_k) - f_k$$

Mit der "Modellmatrix"

$$A := \left(\Phi_\ell(t_k) \right) \Big|_{\substack{k=1, \dots, m \\ \ell=1, \dots, n}} = \begin{bmatrix} \Phi_1(t_1) & \dots & \Phi_n(t_1) \\ \vdots & & \vdots \\ \Phi_1(t_m) & \dots & \Phi_n(t_m) \end{bmatrix}$$

gilt

$$r = A c - f, \quad f = \begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix} \in \mathbb{R}^m$$

Somit

(*) Suche $\underline{c} \in \mathbb{R}^n$ so, dass $\|\underline{r}\| \stackrel{!}{=} \min$ in einer geeigneten Norm

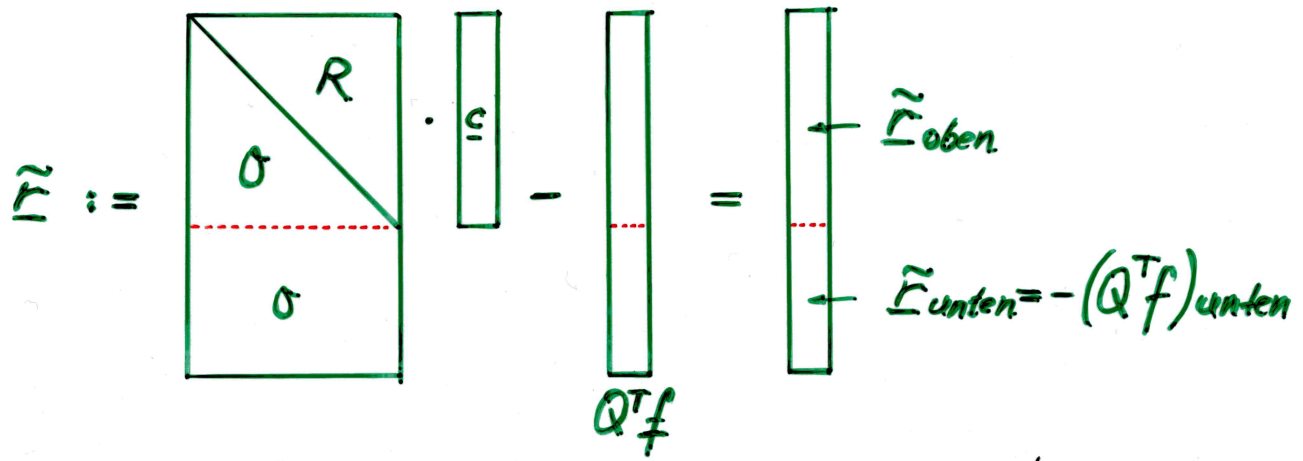
Gaussche Ausgleichung ("least squares")

Euklidische Norm, $\|\underline{r}\| = \|\underline{r}\|_2$

Damit ist (*) elegant lösbar durch QR-Zerleg.

$$\|\underline{r}\|^2 = \|\underline{A}\underline{c} - \underline{f}\|^2 = \|\underline{Q}\underline{R}\underline{c} - \underline{f}\|^2 = \|\underbrace{\underline{R}\underline{c} - \underline{Q}^T \underline{f}}\|^2 \stackrel{!}{=} \min$$

gedrehtes Residuum $\tilde{\underline{r}} := \underline{Q}^T \underline{r}$



Aus $\|\tilde{\underline{r}}\|^2 = \underbrace{\|\tilde{\underline{r}}_{\text{oben}}\|^2}_0 + \underbrace{\|\tilde{\underline{r}}_{\text{unten}}\|^2}_{\text{fest}} \stackrel{!}{=} \min$

folgt $\tilde{\underline{r}}_{\text{oben}} = 0$, d.h. \underline{c} folgt aus

$$\underline{R}\underline{c} = (\underline{Q}^T \underline{f})_{\text{oben}}$$

durch Rückwärts einsetzen.

Ein Service von Matlab:

Seien $\underline{A} \in \mathbb{R}^{m \times n}$, $\underline{f} \in \mathbb{R}^m$, $m \geq n$ Modellmatrix und (Kolonnen-)Vektor der Messwerte.

Der Befehl $\underline{c} = \underline{A} \setminus \underline{f}$ bewirkt Lösung des Ausgl'pr. $\|\underline{A}\underline{c} - \underline{f}\|_2 \stackrel{!}{=} \min$ dch. QR-Zerleg.