

Numerisches Programmieren (IN0019)

Frank R. Schmidt

Winter Semester 2016/2017

4. QR-Zerlegung	2
Gauß-Elimination	3
Kondition des LGS (Wdh.)	4
Interpretation der Matrix-Kondition	5
Orthogonale Matrizen (Wdh.)	6
Laufzeit der Gauß-Elimination	7
Landausche Symbole (Wdh.)	8
Gauß-Elimination (Laufzeit)	9
Stabilität der LU-Zerlegung.	10
Stabilität der LU-Zerlegung.	11
Lineares Ausgleichproblem	12
Lineares Ausgleichproblem	13
Methode der kleinsten Quadrate	14
Methode der kleinsten Quadrate	15
Kondition der Normalgleichung	16
Lineare Regression.	17
Lineare Regression als Lineares Ausgleichproblem	18

QR-Zerlegung	19
QR-Zerlegung	20
Elementare orthogonale Matrizen	21
QR-Zerlegung ($\mathbf{n} = 2$)	22
Givensreflexion ($\mathbf{n} = 2$)	23
Givensreflexion	24
Givensreflexion	25
Elimination mittels Givens-Reflexionen	26
QR-Zerlegung mittels Givens-Reflexionen	27
Allgemeine QR-Zerlegung	28
Anwendung von QR-Zerlegung (Ausgleichproblem)	29
Ausgleichproblem (Beispiel)	30
Ausgleichproblem (Beispiel)	31
Ausgleichproblem (Beispiel)	32
Tikhonov-Regularisierung	33
Anwendung der QR-Zerlegung	34
Regularisierung	35
Tikhonov Regularisierung	36
Zusammenfassung	37

Kondition des LGS (Wdh.)

Wenn wir ein **lineares Gleichungssystem (LGS)** $A \cdot x = b$ lösen wollen, wird ein **relativer Eingabefehler** in b zu einem **relativen Ausgabefehler** in x führen. Das Verhältnis dieser beiden Fehler ist durch die **Kondition**

$$\kappa(A^{-1}) = \|A\| \cdot \|A^{-1}\|,$$

nach oben beschränkt.

Messen wir den relativen Fehler in der **Euklidischen Norm** $\|x\|_2 := \langle x, x \rangle^{\frac{1}{2}}$, so wird bei der Berechnung der Kondition die **Spektralnorm** $\|A\|_2 := \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2}$ benutzt.

Betrachten wir die **Singulärwertzerlegung** von $A = U\Sigma V^T$, wobei σ_{\max} den größten und σ_{\min} den kleinsten Singulärwert von A beschreibt, so gilt

$$\kappa(A) = \kappa(A^{-1}) = \frac{\sigma_{\max}}{\sigma_{\min}}$$

Interpretation der Matrix-Kondition

Bei dem relativen Fehler von Vektoren $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ betrachten wir den Vektor **als Ganzes**, d.h. ein kleiner relativer Fehler von x kann einem großen relativen Fehler einer **einzelnen Komponente** x_i entsprechen.

Betrachten wir

$$x = (10^{-5}, 1)$$

$$\tilde{x} = (0, 1),$$

so ist

$$\frac{\|\tilde{x} - x\|}{\|x\|} < 10^{-5}$$

$$\frac{|\tilde{x}_1 - x_1|}{|x_1|} = 1$$

Ob wir eine Funktion komponentenweise betrachten oder als eine mehrdimensionale Funktion interpretieren, beeinflusst die Konditionszahl. Diese Wahl hängt davon ab, wie Fehler in unseren Eingabedaten entstehen.

Orthogonale Matrizen (Wdh.)

Wenn $Q \in \mathbb{R}^{n \times n}$ eine **orthogonale Matrix** ist ($Q^\top Q = I$), dann gilt

$$\kappa(Q) = 1$$

Haben wir nun $B = Q \cdot A$ mit $A, B, Q \in \mathbb{R}^{n \times n}$, wobei Q orthogonal ist, so gilt

$$\|Bx\|^2 = x^\top A^\top Q^\top Q A x = x^\top A^\top A x = \|Ax\|^2$$

In anderen Worten

$$\kappa(B) = \kappa(QA) = \kappa(A)$$

Also sind orthogonale Matrizen Q mehr als nur gut konditioniert.

Sie verändern die Kondition einer Matrix A nicht, wenn wir sie mit Q multiplizieren.

Laufzeit der Gauß-Elimination

Gegeben sei die LU-Zerlegung $A = L \cdot U$ der Matrix $A \in \mathbb{R}^{n \times n}$. Dann müssen wir zwei Gleichungssysteme lösen, die Matrizen in Dreiecksform benutzen. Um $Ux = y$ zu lösen benutzen wir folgenden Algorithmus:

```
1 // Solve Ux=y
2 for (i=n; i>0; i--) {
3     x[i] = b[i];
4     for (j=i+1; j<=n; j++) {
5         x[i] -= U[i][j]*x[j];
6     }
7     x[i] /= U[i][i];
8 }
```

Die Anzahl der **Flops** (Fließkomma-Operationen) ist

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

Landausche Symbole (Wdh.)

Um die **Laufzeit-Komplexität** eines Algorithmus anzugeben, geben wir üblicherweise das Monom mit dem höchsten Exponenten an, d.h. n^2 anstelle von $n^2 + 2n$. Formal wird das mit der sogenannten \mathcal{O} -Schreibweise gemacht.

Für jede Funktion $f: \mathbb{N} \rightarrow \mathbb{N}$ ist $\mathcal{O}(f)$ die Menge aller Funktionen, die schlimmstenfalls so "stark wachsen" wie f . Die mathematische Definition ist

$$\mathcal{O}(f) := \left\{ g: \mathbb{N} \rightarrow \mathbb{N} \mid \limsup_{\substack{n \rightarrow \infty \\ f(n) > 0}} \frac{g(n)}{f(n)} < \infty \right\}$$

Für $g \in \mathcal{O}(f)$ hat sich die Schreibweise $g = \mathcal{O}(f)$ durchgesetzt.

Es gilt $n^{d-1} = \mathcal{O}(n^d)$ und $g_1, g_2 = \mathcal{O}(f) \Rightarrow g_1 + g_2 = \mathcal{O}(f)$. Damit gilt

$$\sum_{i=0}^d a_i n^i = \mathcal{O}(n^d).$$

Gauß-Elimination (Laufzeit)

Wir haben gesehen, dass man innerhalb von $\mathcal{O}(n^2)$ ein Gleichungssystem $Ax = b$ lösen können, wenn A in der LU-Zerlegung vorliegt. Nun wollen wir sehen wie schnell diese LU-Zerlegung gefunden werden kann.

Für jede der n Spalten ($i = 1, \dots, n$) werden die gleichen Operationen auf der $i' \times i'$ -Untermatrix von A durchgeführt ($i' = n + 1 - i$):

- Finde ein Pivot-Element ($\mathcal{O}(i')$ bzw. $\mathcal{O}(i'^2)$ für Totalpivotsuche)
- Austausch von Zeilen und Spalten bei Totalpivotsuche ($\mathcal{O}(n)$)
- Subtraktion von Zeilen ($\mathcal{O}(i'^2)$)

Insgesamt ist die Anzahl der Flops also

$$\sum_{i'=n}^1 n + i'^2 = n^2 + \frac{n(n+1)(2n+1)}{6} = \frac{1}{3}n^3 + \mathcal{O}(n^2)$$

Insgesamt können wir also $Ax = b$ mit $\mathcal{O}(n^3)$ Flops lösen.

Stabilität der LU-Zerlegung

Der erste Schritt der Gauß-Elimination produziert Folgendes

$$A_0 = \begin{pmatrix} 1 & 0 & \cdots & 0 & 2^0 \\ -1 & 1 & \ddots & \vdots & 2^0 \\ -1 & -1 & \ddots & 0 & 2^0 \\ \vdots & & \ddots & 1 & \vdots \\ -1 & \cdots & -1 & -1 & 2^0 \end{pmatrix} \rightsquigarrow A_1 = \begin{pmatrix} 1 & 0 & \cdots & 0 & 2^0 \\ 0 & 1 & \ddots & \vdots & 2^1 \\ 0 & -1 & \ddots & 0 & 2^1 \\ \vdots & \vdots & \ddots & 1 & \vdots \\ 0 & -1 & \cdots & -1 & 2^1 \end{pmatrix}$$

und nach dem $(n - 1)$ -ten Schritt

$$A_{n-1} = \begin{pmatrix} 1 & 0 & \cdots & 0 & 2^0 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & 1 & 0 & \vdots \\ 0 & \cdots & 0 & 1 & \vdots \\ 0 & \cdots & 0 & 0 & 2^{n-1} \end{pmatrix}$$

Stabilität der LU-Zerlegung

Für dieses Beispiel ist $\kappa(U) = \kappa(A_{n-1}) = 2^{n-1}$.

Weiter kann man zeigen, dass $\kappa(A) = \kappa(A_0) = \mathcal{O}(n)$.

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

Allerdings würde eine Spaltenpivotsuche den Algorithmus nicht verändern.

In der Praxis kommen diese Matrizen selten vor.

Wenn man sich numerischer Stabilität sicher sein möchte, sollte man das LU-Verfahren meiden.

Lineares Ausgleichproblem

Viele Probleme wie z.B. Polynombestimmung sind nicht immer numerisch stabil. Um weniger anfällig für Fehler zu sein, kann man z.B. mehr Daten sammeln als man tatsächlich benötigt, um ein Problem zu lösen.

Dies führt dann zu einem Gleichungsproblem $Ax = b$ mit $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $x \in \mathbb{R}^n$, wobei $m > n$. Man spricht hierbei von einem **überbestimmten System**, da wir mehr Gleichungen (Zeilen) als Variablen (Spalten) haben.

Weiter gehen wir davon aus, dass A **vollen Rang** hat, d.h. $\text{rank}(A) = n$. Man könnte nun eine beliebige $n \times n$ -Untermatrix von A wählen und das Problem lösen.

Aber: Wir würden dann einige Beobachtungen ignorieren. Besser ist es, den Fehler $Ax - b$ bzgl. einer zuvor gewählten Norm zu minimieren. Hiermit wählt man einen Kompromiss, der durch die m verschiedenen Gleichungen gegeben ist. Man spricht von einem **Linearen Ausgleichproblem**.

Methode der kleinsten Quadrate

Wenn wir die Euklidische Norm benutzen, redet man von der **Methode der kleinsten Quadrate**.

Wir möchten also die folgende Funktion minimieren

$$f: \mathbb{R}^n \rightarrow \mathbb{R} \\ x \mapsto \|Ax - b\|^2$$

Für das Minimum gilt

$$0 = \frac{d}{dx} [\langle Ax, Ax \rangle - \langle Ax, b \rangle - \langle b, Ax \rangle + \langle b, b \rangle] \\ = \frac{d}{dx} [x^\top A^\top Ax - 2x^\top A^\top b] = 2A^\top Ax - 2A^\top b$$

Methode der kleinsten Quadrate

Das Gleichungssystem

$$A^T Ax = A^T b$$

entsteht aus dem Ausgangsproblem $Ax = b$ wenn man es von links mit A^T multipliziert.

Die Gleichung heißt auch **Normalgleichung zu $Ax = b$** .

Die Matrix $A^T A \in \mathbb{R}^{n \times n}$ hat wie A vollen Rang. Da es eine $n \times n$ -Matrix ist, ist sie im Gegensatz zu A invertierbar und die Normalgleichung besitzt eine eindeutige Lösung.

Die Kondition von $A^T A$ ist aber immer schlechter als die Kondition von A .

Kondition der Normalgleichung

Zu $A \in \mathbb{R}^{n \times n}$ betrachten wir die Singulärwertzerlegung $A = U\Sigma V^T$. Dann gilt

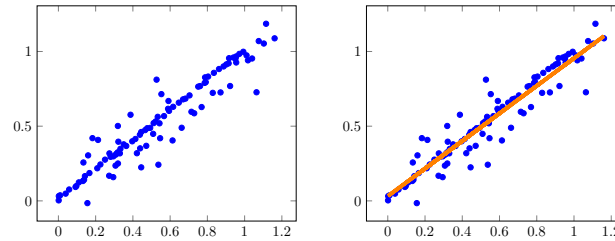
$$A^T A = V\Sigma^2 V^T.$$

Für die Konditionen gilt also

$$\begin{aligned}\kappa(A) &= \frac{\sigma_{\max}}{\sigma_{\min}} \\ \kappa(A^T A) &= \frac{\sigma_{\max}^2}{\sigma_{\min}^2} = \kappa(A)^2.\end{aligned}$$

Da die Konditionszahl von Matrizen mindestens 1 ist, ist die Kondition der Normalgleichung üblicherweise sehr viel schlechter als das ursprüngliche Ausgleichsproblem.

Lineare Regression



Gegeben: $n \in \mathbb{N}$ verschiedene Punktepaare $(x_i, y_i) \in \mathbb{R}^2$ für $i = 1, \dots, n$.

Gesucht: $m, b \in \mathbb{R}$ so dass

$$y_i \approx m \cdot x_i + b$$

Die Gerade $\ell: x \mapsto m \cdot x + b$ nennt man die **Ausgleichsgerade**.

Lineare Regression als Lineares Ausgleichproblem

Wir wollen also folgendes Problem lösen:

$$\begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} \cdot \begin{pmatrix} b \\ m \end{pmatrix} \approx \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Die Normalgleichung (Multiplikation mit A^\top) hierzu ist

$$\begin{pmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{pmatrix} \cdot \begin{pmatrix} b \\ m \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{pmatrix}$$

Wenn man mit $\mathbf{1} := (1, \dots, 1)$ den Einsvektor beschreibt lässt sich das wie folgt umschreiben

$$\begin{pmatrix} \langle \mathbf{1}, \mathbf{1} \rangle & \langle \mathbf{1}, x \rangle \\ \langle \mathbf{1}, x \rangle & \langle x, x \rangle \end{pmatrix} \cdot \begin{pmatrix} b \\ m \end{pmatrix} = \begin{pmatrix} \langle \mathbf{1}, y \rangle \\ \langle x, y \rangle \end{pmatrix}$$

QR-Zerlegung

Die **QR-Zerlegung** beschreibt eine Matrix $A \in \mathbb{R}^{m \times n}$ als Produkt von zwei Matrizen $Q \in \mathbb{R}^{m \times m}$ und $R \in \mathbb{R}^{m \times n}$, also $A = Q \cdot R$.

Genau wie in der LU-Zerlegung soll es einfach sein, die Gleichungssysteme $Qy = b$ und $Rx = y$ zu lösen. Dies wird erreicht, indem R eine obere Dreiecksmatrix ist und Q orthogonal ist ($Q^T Q = I$). Warum QR?

Wir haben Folgendes beobachtet:

- Bei der LU-Zerlegung kann $\text{cond}(U)$ sehr groß sein (bei kleiner $\text{cond}(A)$).
- Die Kondition von Q ist gerade 1. Daher ist

$$\text{cond}(R) = \text{cond}(Q^{-1}QR) = \text{cond}(Q^T A) = \text{cond}(A)$$

Eine exponentielle Explosion der Konditionszahl kann also bei der QR-Zerlegung ausgeschlossen werden.

Elementare orthogonale Matrizen

Die einzigen orthogonalen 1×1 -Matrizen sind

$$Q = 1 \quad \vee \quad Q = -1,$$

d.h. es ist sinnvoll, sich die orthogonalen 2×2 -Matrizen anzusehen.

Die orthogonalen 2×2 -Matrizen haben die folgende Struktur

$$G_1 = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \quad \vee \quad G_2 = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ \sin(\alpha) & -\cos(\alpha) \end{pmatrix}$$

Beide Matrizentypen G_1 und G_2 hängen alleine von einem Parameter α ab. Matrizen der Art G_1 heißen **Givensrotationen** und Matrizen der Art G_2 heißen **Givensreflexionen**.

Wir werden im Folgenden die Givensreflexionen benutzen.

QR-Zerlegung (n = 2)

Wir suchen eine Givensreflexion G , so dass $GA = R$ obere Dreiecksmatrix ist:

$$\begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ \sin(\alpha) & -\cos(\alpha) \end{pmatrix} \cdot \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{pmatrix}$$

Dies liefert uns eine Nebenbedingung

$$\sin(\alpha)a_{11} - \cos(\alpha)a_{21} = 0$$

Damit ist α eindeutig bestimmt durch

$$\tan(\alpha) = \frac{a_{21}}{a_{11}}$$

Givensreflexion ($n = 2$)

Wir benötigen für die Givensreflexion die beiden Werte

$$\cos(\alpha) \quad \sin(\alpha) \quad , \text{ wobei } \alpha = \tan^{-1} \left(\frac{a_{21}}{a_{11}} \right)$$

Für $a_{21} = 0$ benötigen wir keine Transformationen, aber wir haben

$$\cos(\alpha) = 1 \quad \sin(\alpha) = 0 \quad G = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Im Allgemeinen bietet sich folgende (numerisch stabilere) Berechnung an:

$$\cos(\alpha) = \frac{a_{11}}{\rho} \quad \sin(\alpha) = \frac{a_{21}}{\rho} \quad \rho = \sqrt{a_{11}^2 + a_{21}^2}$$

Givensreflexion

Für $n = 2$ können wir ähnlich zur LU-Zerlegung mit Hilfe einer einzigen Matrizenmultiplikation eine Null in unserer Matrix $A \in \mathbb{R}^{2 \times 2}$ erzeugen. Das besondere der Givensreflexion ist ihre numerische Stabilität ($\text{cond}(G) = 1$).

Bei allgemeinen Matrizen $A \in \mathbb{R}^{n \times n}$ wählt man zwei Dimensionen i und j und wendet in diesem Teilraum die obige Reflexion an. Die anderen Dimensionen werden nicht verändert:

$$G_{ij} = \begin{pmatrix} I & 0 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & 0 & \sin(\alpha) & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & \sin(\alpha) & 0 & -\cos(\alpha) & 0 \\ 0 & 0 & 0 & 0 & I \end{pmatrix}$$

Givensreflexion

Berechnet man $\tilde{A} := G_{ij} \cdot A$, so ändern sich nur die i -te und j -te Zeile von A . Bezeichnet man mit a_k die k -te Zeile von A , so gilt

$$\begin{aligned} \tilde{a}_k &= a_k && \text{für } k \neq i, j \\ \tilde{a}_i &= \cos(\alpha)a_i + \sin(\alpha)a_j \\ \tilde{a}_j &= \sin(\alpha)a_i - \cos(\alpha)a_j \end{aligned}$$

Analog zum Fall von 2×2 -Matrizen können wir den Winkel α bestimmen, indem wir uns die Einträge a_{ii} und a_{ji} ansehen:

$$\tan(\alpha) = \frac{a_{ji}}{a_{ii}}$$

Mit einer solchen Matrix G_{12} wird im ersten Schritt a_{21} auf Null gesetzt. Danach wird mit G_{13} der Eintrag a_{31} auf Null gesetzt usw.

Elimination mittels Givens-Reflexionen

Damit erhalten wir einen Eliminations-Algorithmus, der sehr ähnlich zur LU-Zerlegung ist

```
1 for (i=1; i<=n; i++) {
2   for (j=i+1; j<=n; j++) {
3     alpha = atan(A[j][i]/A[i][i]);
4     G = Gij bzgl. alpha;
5     A = G*A;
6   }
7 }
```

Wie bei der LU-Zerlegung werden $\mathcal{O}(n^2)$ Matrixmultiplikationen benötigt.

Da jede Multiplikation mit einer Givensreflexion lediglich zwei Zeilen verändert, ist die Anzahl der Flops gerade $n^3 + \mathcal{O}(n^2)$.

QR-Zerlegung mittels Givens-Reflexionen

Nach der Multiplikation mit $\mathcal{O}(n^2)$ Givensreflexionen erhalten wir eine Matrix $R \in \mathbb{R}^{n \times n}$ in oberer Dreiecksform.

Definieren wir nun

$$Q^{-1} = G_{n-1,n} \cdot G_{n-2,n} G_{n-2,n-1} \cdots G_{1,n} G_{1,n-1} \cdots G_{1,2}$$

so ist

$$Q^{-1}A = G_{n-1,n} \cdot G_{n-2,n} G_{n-2,n-1} \cdots G_{1,n} G_{1,n-1} \cdots G_{1,2}A = R$$
$$A = QR$$

Wegen $G_{ij}^{-1} = G_{ij}^\top = G_{ij}$ erhalten wir

$$Q = G_{1,2} \cdots G_{1,n-1} G_{1,n} \cdots G_{n-2,n-1} G_{n-2,n} \cdot G_{n-1,n}$$

Allgemeine QR-Zerlegung

Das beschriebene Verfahren kann auch angewandt werden, um eine QR-Zerlegung von $A \in \mathbb{R}^{m \times n}$ mit $m > n$ zu finden.

Wie zuvor eliminiert man also in jedem Schritt ein Element, das sich unterhalb der Diagonalen befindet und erhält

$$A = Q \cdot R$$

mit orthogonalem $Q \in \mathbb{R}^{m \times m}$ und $R \in \mathbb{R}^{m \times n}$ in oberer Dreiecksform.

Es gibt weitere QR-Verfahren, die aber numerisch weniger stabil sind

Householder-Verfahren. Im i -ten Schritt werden $n - i$ Einträge auf Null gesetzt. Die erreicht man mit Matrizen der Art $H = I - 2 \frac{u \otimes u}{\langle u, u \rangle}$.

Gram-Schmidt. Die Spaltenvektoren von A werden schrittweise orthonormalisiert. Es können Auslöschungen entstehen.

Anwendung von QR-Zerlegung (Ausgleichproblem)

Wir wollen nun die QR-Zerlegung auf das lineare Ausgleichproblem anwenden.

Sei also $A \in \mathbb{R}^{m \times n}$ mit $m > n$, $b \in \mathbb{R}^m$. Dann gilt

$$\|Ax - b\|_2^2 = \|QRx - b\|_2^2 = \|Q^\top(QRx - b)\|_2^2 = \|Rx - Q^\top b\|_2^2$$

Schreiben wir $Rx = \begin{pmatrix} \tilde{y}_1 \\ 0 \end{pmatrix}$ und $Q^\top b = \begin{pmatrix} \tilde{b}_1 \\ \tilde{b}_2 \end{pmatrix}$ mit $\tilde{y}_1, \tilde{b}_1 \in \mathbb{R}^n$, $\tilde{b}_2 \in \mathbb{R}^{m-n}$, so ist

$$\|Ax - b\|_2^2 = \|Rx - Q^\top b\|_2^2 = \|\tilde{y}_1 - \tilde{b}_1\|_2^2 + \|\tilde{b}_2\|_2^2 \geq \|\tilde{b}_2\|_2^2$$

Um das Ausgleichsproblem zu lösen, **reicht es, die Gleichung $Rx = \begin{pmatrix} \tilde{b}_1 \\ 0 \end{pmatrix}$ zu lösen**, was wegen der Dreiecksstruktur von R eindeutig machbar ist.

Es gilt für diese Lösung x^* gerade $\|Ax^* - b\|_2 = \min_{x \in \mathbb{R}^n} \|Ax - b\|_2$.

Ausgleichproblem (Beispiel)

Betrachten wir das lineare Ausgleichproblem

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

Dann gilt für den Winkel φ bzgl. G_{12} gerade $\cos(\varphi) = \sin(\varphi) = \frac{1}{\sqrt{2}}$

$$G_{12} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad G_{12} \cdot A = \begin{pmatrix} \sqrt{2} & \sqrt{2} \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$$

Weiter gilt für den Winkel φ bzgl. G_{23} gerade $\cos(\varphi) = 0$, $\sin(\varphi) = 1$.

$$G_{23} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad G_{23}G_{12} \cdot A = \begin{pmatrix} \sqrt{2} & \sqrt{2} \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

Ausgleichproblem (Beispiel)

Wir haben also

$$Q = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \end{pmatrix} \quad R = \begin{pmatrix} \sqrt{2} & \sqrt{2} \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

Weiter ist

$$\tilde{b} = Q^T b = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 1 \\ \frac{1}{\sqrt{2}} \end{pmatrix} \quad \tilde{b}_1 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 1 \end{pmatrix} \quad \tilde{b}_2 = \frac{1}{\sqrt{2}}$$

und somit

$$x^* = \begin{pmatrix} \sqrt{2} & \sqrt{2} \\ 0 & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 1 \end{pmatrix} = \begin{pmatrix} -\frac{1}{2} \\ 1 \end{pmatrix}$$

Ausgleichproblem (Beispiel)

Das ursprüngliche Ausgleichproblem

$$\underbrace{\begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}}_A \cdot x \approx \underbrace{\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}}_b$$

wird also durch $x^* = \begin{pmatrix} -\frac{1}{2} \\ 1 \end{pmatrix}$ gelöst.

Es gilt

$$\|Ax^* - b\|_2 = \left\| \begin{pmatrix} \frac{1}{2} & - & 1 \\ \frac{1}{2} & - & 0 \\ 1 & - & 1 \end{pmatrix} \right\|_2 = \frac{1}{\sqrt{2}} = \tilde{b}_2$$

Anwendung der QR-Zerlegung

Die QR-Zerlegung wird üblicherweise benutzt bei

- dem Lösen eines schlecht konditioniertem Gleichungssystems
- Linearen Ausgleichproblemen (wie oben beschrieben)
- dem Lösen von nicht-quadratischen Gleichungssystem von vollem Rang.

Darüber hinaus können die Diagonaleinträge von R benutzt werden, um eine Matrix A auf ihren "wesentlichen" Teil zu reduzieren:

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

Ein solcher Ansatz ist oft sinnvoll bei praktischen Problemen, wenn eine Matrix A verrauscht ist und man sie "entrauschen" möchte.

Allerdings ist die Wahl des geeigneten ε oft sehr schwierig. Ein anderer Ansatz besteht in der sogenannten **Regularisierung**.

Regularisierung

In der Praxis hat man oft ein überbestimmtes lineares Gleichungssystem $Ax = b$ mit $A \in \mathbb{R}^{m \times n}$, so dass $A^\top A$ auch noch (fast) singularär ist. Damit ist die Kondition von $A^\top A$ sehr groß und kleine Fehler in b führen zu großen Fehlern in x .

Man kann das umgehen, indem man die Größe von x zusätzlich bestraft. Man löst also statt

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2$$

folgendes Problem

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2 + \gamma \|x\|_2^2,$$

wobei der Parameter $\gamma \geq 0$ angibt, wie stark die Größe von x bestraft wird.

Tikhonov Regularisierung

Das Regularisierungsproblem

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2 + \gamma \|x\|_2^2,$$

kann gelöst werden, indem man die Nullstelle der Ableitung betrachtet:

$$(A^\top A + \gamma I) x = A^\top b,$$

d.h. die Eigenwerte von $A^\top A$ werden alle um γ erhöht, was die Kondition des Problems sinken lässt.

Diese Form der Regularisierung nennt man **Tikhonov-Regularisierung** und man bezeichnet γ als den **Regularisierungsparameter**.

Zusammenfassung

- Die LU-Zerlegung (Gauß-Elimination) kann numerisch instabil sein, da eine große Kondition erzeugt werden kann.
- Die QR-Zerlegung beschreibt die Matrix A als Produkt einer orthogonalen Matrix Q mit einer Dreiecksmatrix R .
- Dies ist numerisch stabiler, da $\text{cond}(Q) = 1$.
- Die Matrix Q wird durch viele einzelne Reflexionen bestimmt.
- Die QR-Zerlegung kann auch für Ausgleichsprobleme angewendet werden (mehr Gleichungen als Variablen)
- Weitere Stabilität kann durch Regularisieren erreicht werden.