

Übung zu
Algorithmen und Datenstrukturen (für ET/IT)
Sommersemester 2018

Mai Bui

Computer Aided Medical Procedures
Technische Universität München



Lineare Gleichungen

Linear Gleichung

Eine **lineare Gleichung** hat die Form

$$a_1x_1 + a_2x_2 + \dots + a_nx_n = b,$$

wobei $x_1, \dots, x_n \in \mathbb{R}$ die **Unbekannten** sind, und die $a_1, \dots, a_n, b \in \mathbb{R}$ Konstanten.

- Schreibweise mit **Skalarprodukt**:

$$\langle a, x \rangle = b$$

oder auch

$$a^T x = b$$

mit $a = (a_i) \in \mathbb{R}^n$, $x = (x_i) \in \mathbb{R}^n$, $b \in \mathbb{R}$.

Lineares System

Lineares System

Ein **lineares System** bzw. **lineares Gleichungssystem** ist eine endliche Menge von linearen Gleichungen mit Unbekannten $x_1, \dots, x_n \in \mathbb{R}$:

$$a_{11}x_1 + \dots + a_{1n}x_n = b_1 \quad (1)$$

$$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad (2)$$

$$a_{m1}x_1 + \dots + a_{mn}x_n = b_m \quad (3)$$

mit Konstanten $(a_{ji}) \in \mathbb{R}^{m \times n}$, $(b_i) \in \mathbb{R}^m$.

- wir betrachten hier meist den **quadratischen** Fall mit $m = n$ Gleichungen und n Unbekannten

Lineares System

- ▶ Matrix-Schreibweise:

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$$

- ▶ Kurz-Notation mit Matrix:

$$Ax = b$$

mit $A = (a_{ji}) \in \mathbb{R}^{m \times n}$, $b = (b_i) \in \mathbb{R}^m$ und Unbekannten $x = (x_i) \in \mathbb{R}^n$.

Lineares System: Eigenschaften

Sei $Ax = b$ lineares System mit $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ und $x \in \mathbb{R}^n$.

- ▶ Ein lineares System hat immer entweder **keine Lösung**, **eine Lösung** oder **unendlich viele Lösungen**.
- ▶ Ist $m < n$, so heißt das lineare System **unterbestimmt**.
- ▶ Ist $m > n$, so heißt das lineare System **überbestimmt**.
- ▶ Wir betrachten im weiteren den Fall $m = n$ (A quadratisch).

Lösen von linearen Systemen

Inverse Matrix

Sei $A \in \mathbb{R}^{n \times n}$ quadratische Matrix. Falls ein $B \in \mathbb{R}^{n \times n}$ existiert mit

$$AB = BA = I_n$$

(wobei I_n die $n \times n$ Einheitsmatrix ist), dann heißt A **invertierbar** und wir nennen $B =: A^{-1}$ das **Inverse** von A .

Existiert A^{-1} , so heißt A **invertierbar**.

- ▶ Ist $Ax = b$ lineares System mit $A \in \mathbb{R}^{n \times n}$ invertierbar, dann ist

$$x = A^{-1}b$$

die **Lösung** des linearen Systems. Diese Lösung ist **eindeutig**.

Invertierbarkeit von Matrizen

Problem 1: wann ist eine Matrix $A \in \mathbb{R}^{n \times n}$ invertierbar?

- ▶ man findet explizit ein Inverses A^{-1}
 - ▶ für kleine Matrizen geeignet (z.B. 2×2)
- ▶ $\det(A) \neq 0$
- ▶ $\ker(A) = \{0\}$
- ▶ $\text{rank}(A) = n$
- ▶ $Ax = 0$ hat nur die triviale Lösung $x = 0$

Invertierbarkeit von Matrizen

Problem 1: wann ist eine Matrix $A \in \mathbb{R}^{n \times n}$ invertierbar?

- ▶ man findet explizit ein Inverses A^{-1}
 - ▶ für kleine Matrizen geeignet (z.B. 2×2)
- ▶ $\det(A) \neq 0$
- ▶ $\ker(A) = \{0\}$
- ▶ $\text{rank}(A) = n$
- ▶ $Ax = 0$ hat nur die triviale Lösung $x = 0$

Problem 2: falls A invertierbar, wie findet man das Inverse A^{-1} ?

- ▶ Berechnung mit Gauss-Jordan Elimination
- ▶ Umweg über Zerlegungen von A (z.B. LU-Zerlegung)
- ▶ **generell:** Berechnung der Inversen meist numerisch nicht stabil

Ansätze zum Lösen linearer Systeme

Gegeben sei lineares System $Ax = b$ mit $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$ und $x \in \mathbb{R}^n$.

- ▶ direktes Lösen über **Inverse** A^{-1}
 - ▶ numerisch oft nicht stabil
 - ▶ sehr effizient (besonders falls für mehrere b gelöst wird)
- ▶ Lösen über **Zerlegung** von A (z.B. $A = LU$)
 - ▶ numerisch stabil
 - ▶ sehr effizient (besonders falls für mehrere b gelöst wird)
- ▶ **iterative Algorithmen**, schrittweise Annäherung an Lösung
 - ▶ besonders geeignet für sehr grosse n
 - ▶ auch geeignet falls keine Lösung existiert, berechnet dann z.B. Approximation

$$\min \|Ax - b\|_2^2$$

Rückwärts-Substitution

Lineares System in oberer Dreiecksform:

$$\begin{aligned} a_{11}x_1 + a_{21}x_2 + \dots + a_{1,n-1}x_{n-1} + a_{1n}x_n &= b_1 \\ a_{22}x_2 + \dots + a_{2,n-1}x_{n-1} + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n &= b_{n-1} \\ a_{n,n}x_n &= b_n \end{aligned}$$

Rückwärts nacheinander nach x_n, x_{n-1}, \dots, x_1 auflösen:

$$x_i = \left(b_i - \sum_{j=i+1}^n a_{ij}x_j \right) / a_{ii}$$

Vorwärts-Substitution

Lineares System in unterer Dreiecksform:

$$a_{11}x_1 = b_1$$

$$a_{21}x_1 + a_{22}x_2 = b_2$$

\vdots

$$a_{n-1,1}x_1 + a_{n-1,2}x_2 + \dots + a_{n-1,n-1}x_{n-1} = b_{n-1}$$

$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{n,n-1}x_{n-1} + a_{nn}x_n = b_n$$

Vorwärts nacheinander nach x_1, x_2, \dots, x_n auflösen:

$$x_i = \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j \right) / a_{ii}$$

Zerlegungen von Matrizen: QR

QR Zerlegung:

- ▶ Sei $A \in \mathbb{R}^{n \times n}$ invertierbar.
- ▶ Dann gibt es eine orthogonale Matrix $Q \in \mathbb{R}^{n \times n}$ ($Q^{-1} = Q^T$) und eine obere Dreiecksmatrix $R \in \mathbb{R}^{n \times n}$ mit positiven Diagonaleinträgen, so daß

$$A = QR.$$

- ▶ Lösung von $Ax = b$: löse $Rx = Q^T b$ mit Rückwärts-Substitution nach x auf

Zerlegungen von Matrizen: LUP

LUP Zerlegung:

- ▶ Sei $A \in \mathbb{R}^{n \times n}$ invertierbar.
- ▶ Dann gibt es eine untere Einheitsdreiecksmatrix $L \in \mathbb{R}^{n \times n}$, eine obere Dreiecksmatrix $U \in \mathbb{R}^{n \times n}$ sowie eine Permutationsmatrix $P \in \mathbb{R}^{n \times n}$, so daß

$$PA = LU.$$

- ▶ Lösung von $Ax = b$: betrachte $PAx = Pb \Leftrightarrow LUx = Pb$,
 - ▶ löse $Lz = Pb$ mit Vorwärts-Substitution nach z auf
 - ▶ löse $Ux = z$ mit Rückwärts-Substitution nach x auf

Berechnung der LU Zerlegung

- ▶ Faktorisierung von A :

$$\begin{aligned} A &= \begin{pmatrix} a_{11} & w^T \\ v & A' \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ v/a_{11} & I_{n-1} \end{pmatrix} \begin{pmatrix} a_{11} & w^T \\ 0 & A' - vw^T/a_{11} \end{pmatrix}, \end{aligned}$$

wobei I_{n-1} die $n-1 \times n-1$ Einheitsmatrix ist.

- ▶ $A' - vw^T/a_{11}$ heißt **Schur-Komplement** von A bezüglich a_{11}
- ▶ Falls A eine LU Zerlegung hat, hat auch das Schur-Komplement eine LU Zerlegung
- ▶ **Annahme:**

$$A' - vw^T/a_{11} = L'U'$$

mit $L' \in \mathbb{R}^{n-1 \times n-1}$ untere Einheitsdreiecksmatrix,
 $U' \in \mathbb{R}^{n-1 \times n-1}$ obere Dreiecksmatrix.

Algorithmus LU Zerlegung

Input: Matrix $A \in \mathbb{R}^{n \times n}$

Output: Matrizen $L, U \in \mathbb{R}^{n \times n}$

LU-Zerlegung(A):

$U = 0; L = I_n;$

for $k = 1$ **to** n {

$u_{kk} = a_{kk};$

for $i = k + 1$ **to** n {

$l_{ik} = a_{ik} / u_{kk};$

$u_{ki} = a_{ki};$

}

for $i = k + 1$ **to** n {

for $j = k + 1$ **to** n {

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

}

}

}

► Aufwand: $\Theta(n^3)$

► Optimierung: falls A nicht weiter benötigt wird, können L, U direkt in A abgespeichert werden (**in place**)

Algorithmus LUP Zerlegung

Input: Matrix $A \in \mathbb{R}^{n \times n}$

Output: Matrizen L, U direkt in A , Feld π

LUP-Zerlegung(A):

π = Feld Länge n ;

for $i = 1$ **to** n { $\pi[i] = i$; }

for $k = 1$ **to** n {

$p = 0$; // **Pivot-Element**

for $i = k$ **to** n {

if ($|a_{ik}| > p$)

$p = |a_{ik}|$; $k' = i$;

}

if ($p == 0$) **error**("Matrix singular");

vertausche $\pi[k]$ mit $\pi[k']$;

for $i = 1$ **to** n

vertausche a_{ki} mit $a_{k'i}$;

for $i = k + 1$ **to** n {

$a_{ik} = a_{ik}/a_{kk}$;

for $j = k + 1$ **to** n

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

}

}

- ▶ Variante mit **in place** Optimierung
- ▶ Aufwand: $\Theta(n^3)$
 - ▶ extra Aufwand Pivoting nur quadratisch

 Code

Lineare Systeme: Anwendungen

- ▶ Kalibrierung von Kameras
 - ▶ z.B. für Augmented Reality
- ▶ Registrierung von Bilddaten
- ▶ Segmentierung von Bilddaten
- ▶ Objekt-Detektierung und -Tracking
- ▶ Modelle für Finanzmärkte, Populationsbiologie
- ▶ Tomographische Rekonstruktion

Lineare Systeme: Anwendungen

- ▶ Kalibrierung von Kameras
 - ▶ z.B. für Augmented Reality
- ▶ Registrierung von Bilddaten
- ▶ Segmentierung von Bilddaten
- ▶ Objekt-Detektierung und -Tracking
- ▶ Modelle für Finanzmärkte, Populationsbiologie
- ▶ **Tomographische Rekonstruktion**

 Code

Least-squares Lösung

Least-squares Lösung

Sei $A \in \mathbb{R}^{m \times n}$, $y \in \mathbb{R}^m$ mit $m \geq n$. Eine Lösung $c \in \mathbb{R}^n$ des Minimierungs-Problems

$$\min_c \|Ac - y\| \quad \text{or} \quad \min_c \|Ac - y\|^2$$

heißt **Least-squares Lösung**.

Anwendungs-Beispiele:

- ▶ Tracking von Objekten mit Kameras
- ▶ Kalibrierung von Kameras, Robotern, etc.
- ▶ Iterative Rekonstruktion für Tomographie

Least-squares Lösung

Least-squares Lösung

Sei $A \in \mathbb{R}^{m \times n}$, $y \in \mathbb{R}^m$ mit $m \geq n$. Eine Lösung $c \in \mathbb{R}^n$ des Minimierungs-Problems

$$\min_c \|Ac - y\| \quad \text{or} \quad \min_c \|Ac - y\|^2$$

heißt **Least-squares Lösung**.

Anwendungs-Beispiele:

- ▶ **Tracking von Objekten mit Kameras**
- ▶ Kalibrierung von Kameras, Robotern, etc.
- ▶ Iterative Rekonstruktion für Tomographie

 Code