

Vorlesungsskript

Num. Lin. Algebra

Inhaltsverzeichnis

1. Einleitung	2
2. Das Gauß-Verfahren I	4
2.1. Gaußsche Eliminationsverfahren und LR-Zerlegung	4
2.2. Pivot-Strategien	8

Definitionen

1. Einleitung

Wichtige Aufgabenklassen der linearen Algebra sind **lineare Gleichungssysteme**.

Gegeben: $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$

Gesucht: Ein/alle $x \in \mathbb{R}^n$ mit $Ax = b$

Herkunft:

- “direkt” aus der Anwendung, z.B. Beschreibung von Netzwerken, Tragwerk
- “indirekt” als Diskretisierung von stationären Prozessen, z.B. Belastung einer Membran
- “mittelbar” durch die Linearisierung nichtlinearer Modelle, z.B. Newton-Verfahren, Approximation von Lösungen gewöhnlicher DGL, notwendige Optimalitätsbedingungen

Klassifizierung:

- $m = n$: A quadratisch

Generische Situation: A regulär

$\implies \exists!$ Lösung

- $m < n$: “Unterbestimmtes System”

Generische Situation:

$$\begin{aligned} \operatorname{rg}(A) &= m \text{ (Vollrang)} \\ A &\hat{=} [A_1 A_2] \quad A_1 \in \mathbb{R}^{m \times m} \end{aligned}$$

Lösungsmenge:

$$\mathcal{L} = \{x \in \mathbb{R}^n \mid Ax = b\} = \{x = x^+ + h, h \in \ker(A)\}$$

= $(n - m)$ -dimensionale lineare Mannigfaltigkeit

Gesucht ist dann z.B. norm-minimale Lösung (Kap. 5)

- $m > n$: “Überbestimmtes System”

$$\text{lösbar} \iff b \in \text{im}(A) = \{y \in \mathbb{R}^m \mid \exists x : Ax = y\}$$

Generisch nicht lösbar!

Sinnvoll: Bestimme $\bar{x} \in \mathbb{R}^m$, so dass

$$\|A\bar{x} - b\| = \min_{x \in \mathbb{R}^m} \|Ax - b\|$$

$\|\cdot\| \hat{=}$ geeignete Norm, $\bar{x} \hat{=}$ Bestapproximierender für diese Norm.

Mögliche Ansätze:

- $\|\cdot\|_\infty : \|Ax - b\|_\infty = \max_{1 \leq i \leq m} |(Ax - b)_i|$

Ein nichtglattes Optimierungsproblem auch als lineares Optimierungsproblem formulierbar, schwierig zu lösen für m bzw. n groß.

- $\|\cdot\|_1 : \|Ax - b\|_1 = \sum_{i=1}^m |Ax - b|$

Wie bei $\|\cdot\|_\infty$ stückweise lineares Optimierungsproblem.

Aber stabil gegen Ausreißer.

- $\|\cdot\|_2 : \|Ax - b\|_2^2 = \sum_{i=1}^m ((Ax - b)_i)^2$

$\hat{=}$ lineares Quadratmittelpunktproblem, kleinste Quadrateproblem (Kap. 5)

Verfahren zur Lösung von LGS:

Direkte Verfahren:

- Transformation der Daten (A, b) in endlich viele in ein leichter zu lösendes LGS $\tilde{A}x = \tilde{b} \hat{=}$ CG-Verfahren
- Transformationen lassen sich oftmals als Faktorisierung von A interpretieren

$$A = L \cdot R \quad \text{bzw.} \quad A = Q \cdot R$$

- Dafür i.d.R. Zugriff auf Elemente von $A \implies$ limitiert die Größe der Matrix!

Kap. 2-5

Indirekte Verfahren:

- Ausgehend von einem Startvektor x^0 Iteration zur Berechnung von x^k mit $Ax^k \approx b$
Hierbei wird oftmals nur das Matrix-Vektor-Produkt Av benötigt! (Kap. 6)
- Eigenwertprobleme

Stabilitätsanalyse von Bauwerken. Verfahren dazu: numerische Optimierung

2. Das Gauß-Verfahren I

Jetzt: $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^n$, $x : Ax = b$?

Satz 2.1: Existenz und Eindeutigkeit einer Lösung

Sei $A \in \mathbb{R}^{m \times n}$ eine Matrix mit $\det(A) \neq 0$ und $b \in \mathbb{R}^n$. Dann existiert genau ein $x \in \mathbb{R}^n$ mit

$$Ax = b$$

Beweis: lineare Algebra

□

\Rightarrow Anwendung von Algorithmen zur Berechnung von x sinnvoll! Wie?

2.1. Gaußsche Eliminationsverfahren und LR-Zerlegung

$\hat{=}$ direktes Verfahren für quadratisches System

Erste Idee: Systeme spezieller Struktur, z.B.

$$Rx = c, \quad R = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ 0 & \ddots & \cdots \\ 0 & 0 & r_{nn} \end{pmatrix} \in \mathbb{R}^{n \times n}, \quad c = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} \in \mathbb{R}^n$$

$$Rx = c$$

$$r_{nn}x_n = c_n \Rightarrow x_n = \frac{c_n}{r_{nn}}, \quad r_{nn} \neq 0$$

$$r_{n-1n-1}x_{n-1} + r_{n-1n}x_n = c_{n-1}$$

$$x_{n-1} = \frac{c_{n-1} - r_{n-1n}x_n}{r_{n-1n-1}}, \quad r_{n-1n-1} \neq 0$$

Algorithmus 2.2: Rückwärtssubstitution

$$\begin{aligned}
x_n &= \frac{c_n}{r_{nn}} \quad \text{falls } r_{nn} \neq 0 \\
&\vdots \\
x_i &= \frac{c_i - \sum_{j=i+1}^n r_{ij} x_j}{r_{ii}} \quad \text{falls } r_{ii} \neq 0 \\
&\vdots \\
x_1 &= \frac{c_1 - \sum_{j=2}^n r_{1j} x_j}{r_{11}} \quad \text{falls } r_{11} \neq 0
\end{aligned}$$

Algo. 2.2 anwendbar, wenn $\det(R) \neq 0$ (vgl. Theo. 2.1)

Wichtiger Aspekt dieser Vorlesung: Aufwandsabschätzung

Aufwand: i -te Zeile je $n - i$ Additionen und Multiplikationen und 1 Division
insgesamt:

$$\sum_{i=1}^n (i-1) = \frac{n(n-1)}{2} = \mathcal{O}(n^2)$$

Addition und Multiplikationen und n Divisionen.

Landau-Symbol: $\mathcal{O}(\cdot)$

$$f(n) = \mathcal{O}(g(n)) \iff \exists c > 0 : |f(n)| \leq C|g(n)|$$

Für ein lineares Gleichungssystem der Form

$$Lx = z, \quad L = \begin{pmatrix} l_{11} & & 0 \\ \vdots & \ddots & \\ l_{n1} & \dots & l_{nn} \end{pmatrix} \in \mathbb{R}^{n \times n} \quad z \in \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix} \in \mathbb{R}^n$$

gibt es einen analogen Algorithmus:

$$\begin{aligned}
x_1 &= \frac{z_1}{l_{11}} \quad l_{11} \neq 0 \\
&\vdots \\
x_n &= \frac{z_n - \sum_{i=1}^{n-1} l_{ni} x_i}{l_{nn}} \quad l_{nn} \neq 0
\end{aligned}$$

\implies Vorwärtssubstitution mit gleichem Aufwand $\mathcal{O}(n^2)$

Lösungsidee für ein allgemeines Gleichungssystem:

Faktorisiere $A = L \cdot R$ und berechne die Lösung x von $Ax = b$ durch

$$Ax = L \underbrace{Rx}_{=:z} = b$$

$$Lz = b \implies z = L^{-1}b \quad \text{Vorwärtssubstitution}$$

$$Rx = z \implies x = R^{-1}z \quad \text{Rückwärtssubstitution}$$

Mit Aufwand: $\mathcal{O}(n^2)$

Frage: Wie berechnet man Zerlegung $A = L \cdot R$

Man generiert eine Folge von Matrizen:

$$A = A^{(1)} \longrightarrow A^{(2)} \longrightarrow \dots \longrightarrow A^n = R$$

von Matrizen der Gestalt

$$A^{(k)} = \begin{pmatrix} a_{11}^{(1)} & \dots & \dots & \dots & \dots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \dots & \dots & \dots & a_{2n}^{(2)} \\ & 0 & \ddots & & & \vdots \\ & & 0 & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ & & & \vdots & & \vdots \\ & & & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} \end{pmatrix}$$

Wie?

Sei $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$, $x_k \neq 0 \hat{=} k$ -Spalte

Definiere: $l_{jk} = \frac{x_j}{x_k}$

$$l_k = \left(\underbrace{0, \dots, 0}_{k \text{ mal}}, l_{k+1k}, \dots, l_{nk} \right)^T$$

$e_k = k$ -ter Einheitsvektor

$$L_k = I_n - l_k e_k^T \in \mathbb{R}^{n \times n}$$

Dann gilt

$$L_k x = \begin{pmatrix} 1 & & & & 0 \\ & \ddots & & & \\ & & 1 & & \\ & & -l_{k+1k} & \ddots & \\ & & \vdots & & \\ & & -l_{nk} & & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} x_1 \\ \vdots \\ x_k \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Jeder Eliminationsschritt $A^{(k)} \longrightarrow A^{(k+1)}$ lässt sich damit als Multiplikation mit einer Matrix $L_k \in \mathbb{R}^{n \times n}$ von links

$$A^{k+1} = L_k A^{(k)} = \begin{pmatrix} I_k & 0 \\ 0 & * & I_{n-k} \end{pmatrix} \begin{pmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ A_{21}^{(k)} & A_{22}^{(k)} \end{pmatrix} = \begin{pmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ 0 & A_{22}^{(k+1)} \end{pmatrix} \quad * \in \mathbb{R}^{n-k,1}$$

Eine Matrix der Gestalt L_k heißt Frobeniusmatrix \rightarrow weitere Eigenschaften siehe Übung.

Der Eliminationsschritt ist genau dann durchführbar wenn $a_{kk}^{(k)} \neq 0$ gilt. Angenommen, dies gilt, dann erhält man

$$L_n \cdots L_2 L_1 A = R$$

$$A = \underbrace{L_1^{-1} \cdots L_{n-2}^{-1} L_{n-1}^{-1}}_{=: L} R$$

Induktiv beweist man

$$L = L_1^{-1} \cdots L_{n-1}^{-1} = \begin{pmatrix} 1 & & & & 0 \\ l_{21} & \ddots & & & \\ \vdots & l_{32} & \ddots & & \\ \vdots & \vdots & \vdots & \ddots & \\ \vdots & \vdots & & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn-1} & 1 \end{pmatrix}$$

Durch diese Struktur kann der Speicherplatz für A zum Speichern von L und R genutzt werden!

Algorithmus 2.3: LR -Zerlegung

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

```

for  $i = 1, \dots, n-1$ 
  for  $j = i, \dots, n$ 
    for  $k = 1, \dots, i-1$ 
       $a_{ij} = a_{ij} - a_{ik} * a_{kj}$ 
    end
  end
  for  $j = i+1, \dots, n$ 
    for  $k = 1, \dots, i-1$ 
       $a_{ji} = a_{ji} - a_{jk} * a_{ki}$ 
    end
     $a_{ji} = \frac{a_{ji}}{a_{ii}}$ 
  end
end
end

```

Aufwand für die Dreieckszerlegung $A = L \cdot R$

#Operationen =

$$\begin{aligned} \sum_{i=1}^{n-1} \left((n-i)^2 + (n-i) \right) \frac{1}{3}n^3 - \frac{1}{2}n^2 + \frac{1}{6} + \frac{1}{2}n^3 - \frac{1}{2}n \\ = \frac{1}{3}n^3 + \mathcal{O}(n^2) = \mathcal{O}(n^3) \end{aligned}$$

\Rightarrow kubischer Aufwand! Nur akzeptabel für moderates n !

Algorithmus 2.4: Gaußsche Eliminationsverfahren

Gegeben: $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$

1) Berechne $A = L \cdot R \quad \mathcal{O}(n^3)$

2) Berechne z aus $Lz = b \quad \mathcal{O}(n^3)$

3) Berechne x aus $Rx = z \quad \mathcal{O}(n^3)$

\Rightarrow Gesamtaufwand (Operationen): $\mathcal{O}(n^3)$, (Speicherplatz): $n^2 + n$

Vorteil der Faktorisierung:

Zerlegung (teuer) kann für mehrere rechte Seiten nachgenutzt werden.

2.2. Pivot-Strategien

Beispiel 2.5: Algo 2.4 kann selbst für einfache Schritte scheitern:

$$Ax = b, \quad x = \begin{pmatrix} w \\ z \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \det(A) = -1, \quad b = \begin{pmatrix} c \\ e \end{pmatrix}$$

$$\Rightarrow a_{11} = 0 \quad \nmid$$

Bei der völlig äquivalenten Formulierung

$$\tilde{A}\tilde{x} = \tilde{b}, \quad \tilde{x} = \begin{pmatrix} w \\ z \end{pmatrix}, \quad \tilde{A} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \det(\tilde{A}) = 1, \quad \tilde{b} = \begin{pmatrix} e \\ c \end{pmatrix}$$

funktioniert Algo 2.4 mit

$$\begin{aligned} \tilde{A} &= I_2 = L \cdot R \\ L &= I_2 \quad R = I_2 \end{aligned}$$

\Rightarrow Zeilenvertauschung in A und der rechten Seite, **nicht** in x bzw. \tilde{x} !

Die LR -Zerlegung versagt nicht nur bei verschwindenden Diagonalelementen, sondern auch wenn diese betragsmäßig klein im Vergleich zu den restlichen Elementen sind.

\leadsto Praktikum, Fehlertheorie (Kap. III)