

Numerik MA0008: Zusammenfassung

Jonas Treplin

February 14, 2023

1 Grundlagen

Theorem 1 (Satz von Gerschgorin) Sei $(a_{ij}) = A \in \mathbb{R}^{n \times n}$ dann sind die Eigenwerte von A enthalten in $\bigcup_{i=1}^n S_i \subset \mathbb{C}$, dabei sind die $S_i := K(a_{ii}, \sum_{j=1, j \neq i}^n |a_{ij}|)$. Wobei mindestens ein Eigenwert jeder Zusammenhangskomponente zugeordnet ist.

2 Matrixfaktorisierung

Theorem 2 Die Permutationsmatrizen, die unitären Matrizen, die invertierbaren Matrizen und die unteren/oberen Dreiecksmatrizen bilden jeweils unter Matrixmultiplikation eine Gruppe. Insbesondere sind ihre Inverse von der selben Klasse von Matrizen.

Gleichungssysteme für die unitären Matrizen (und damit auch Permutationsmatrizen) lassen sich einfach durch adjungieren lösen. Für untere und obere Dreiecksmatrizen existieren Vorwärts- und Rückwärtssubstitution. Diese sind aus dem Endschrift des Lösen von Gleichungssystemen mit dem Gauß-Algorithmus bekannt. Glücklicherweise kann jede invertierbare Matrix (fast eindeutig)

Algorithm 1 Vorwärtssubstitution (Lösen einer unteren Dreiecksmatrix)

Require: $(l_{ij}) = L \in \mathbb{R}^{n \times n}$ Untere Dreiecksmatrix, $b \in \mathbb{R}^n$.

for $i \in 1 : n$ **do**

$$x_i \leftarrow \frac{1}{l_{ii}} (b_i - \sum_{j=1}^{i-1} l_{ij} * x_j)$$

end for

Algorithm 2 Rückwärtssubstitution (Lösen einer oberen Dreiecksmatrix)

Require: $(u_{ij}) = U \in \mathbb{R}^{n \times n}$ Obere Dreiecksmatrix, $b \in \mathbb{R}^n$.

for $i \in n : 1$ **do**

$$x_i \leftarrow \frac{1}{u_{ii}} (b_i - \sum_{j=i+1}^n u_{ij} * x_j)$$

end for

in solche Matrizen zerlegt werden. Dies geschieht Wahlweise durch eine LU-Zerlegung oder eine QR-Zerlegung. Mit Pivots erreicht man, dass jede invertierbare Matrix $A \in GL(n)$ zerlegt werden kann, sodass $PA = LU$. Dazu wählt man in jedem Schritt i die Zeile $j = \arg \max_{j \geq i} |a_{ji}^i|$ und vertauscht diese mit der i -ten Zeile.

Algorithm 3 LU-Zerlegung ohne Pivots

Require: $(a_{ij}) = A \in GL(n)$

```
for  $i \in 1 : n$  do
   $l_{ii} \leftarrow 1$ 
  for  $j \in i + 1 : n$  do
     $l_{ji} \leftarrow -\frac{a_{ji}}{a_{ii}}$ 
    for  $k \in i : n$  do
       $u_{jk} \leftarrow u_{jk} - a_{ji}a_{ki}$ 
    end for
  end for
end for
end for
```

Theorem 3 (LU-Zerlegung mit Pivots) Sei $A \in GL(n)$. Dann existieren eine eindeutige Permutationsmatrix P , sowie untere (obere) Dreiecksmatrix L (U), sodass $PA = LU$. Dabei ist L normiert also $l_{ii} = 1$.

Theorem 4 (Cholesky-Zerlegung) Sei A symmetrisch positiv definit dann lässt sich eine nicht normierte untere Dreiecksmatrix \tilde{L} finden, sodass $A = \tilde{L}\tilde{L}^T$.

Algorithm 4 Berechnung der Cholesky Zerlegung

Require: A s.p.d.

```
 $L, U \leftarrow \text{LU\_Zerlegung}(A)$   
 $D = (u_{ii})$  Diagonalmatrix.  
 $\tilde{L} = \sqrt{D}L$ .
```

Eine weitere Möglichkeit ist die der QR Zerlegung.

Definition 1 (Givensrotation) Für ein $a \in \mathbb{R}^2$ sei $Q = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$. Wobei $c = \frac{1}{\sqrt{1+\tau^2}}$ und $s = c\tau$ mit $\tau = \frac{v_2}{v_1}$ wenn $|v_1| \geq |v_2|$ und $s = \frac{1}{\sqrt{1+\tau^2}}$ und $c = s\tau$ mit $\tau = \frac{v_1}{v_2}$ wenn $|v_1| < |v_2|$.

Diese Fallunterscheidung ist so gewählt, dass $\|Q\| \leq 1$, damit sich Rundungsfehler nicht akkumulieren.

Theorem 5 (Givens-Rotation) Es gilt $Qa = \xi e_1$.

Algorithm 5 QR-Zerlegung mit Givens-Rotationen

Require: $A \in \mathbb{R}^{n \times m}$ $Q \leftarrow I_n$ **for** $i \in [n]$ **do****for** $j \in (n, \dots, i+1)$ **do**

$$G := \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & c & s & \\ & & -s & c & \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix} \leftarrow \text{Givens}\left(\begin{Bmatrix} A_{ji-1} \\ A_{ji} \end{Bmatrix}\right)$$

 $Q \leftarrow GQ$ $A \leftarrow GA$ **end for****end for** $Q \leftarrow Q^*$

Definition 2 (Householder Spiegelung) Die Householder Spiegelung für einen Vektor $a \in \mathbb{R}^n$ ist:

$$Q := Id - \frac{2}{v^T v} v v^T$$

Wobei $v := a + \text{sign}(a_1) \|a\| e_1$ Sie erfüllt ebenfalls $Qa = \alpha e_1$

Algorithm 6 QR-Zerlegung mit Householder Rotationen

Require:**Require:** $A \in \mathbb{R}^{n \times m}$ $Q \leftarrow I_n$ **for** $i \in [n]$ **do**

$$H \leftarrow \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix} \text{Householder}((a_i, \dots, a_n))$$

 $Q \leftarrow HQ$ $A \leftarrow HA$ **end for** $Q \leftarrow Q^*$

Die QR-Zerlegung ist der LU-Zerlegung hinsichtlich numerischer Stabilität überlegen, besonders bei Betrachtung der Wilkinson-Matrix:

Definition 3 (Wilkinson-Matrix) Die Wilinon-Matrix ist definiert als:

$$W_n := \begin{bmatrix} 1 & 0 & \dots & 0 & 1 \\ -1 & \ddots & & \vdots & \vdots \\ \vdots & & & \ddots & \vdots \\ -1 & \dots & \dots & \dots & 1 \end{bmatrix}$$

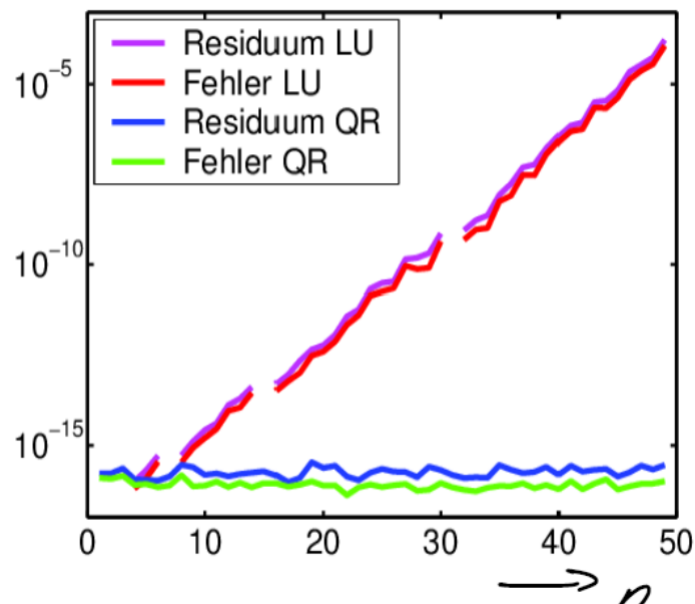


Figure 1: Fehler beim Lösen von $W_n x = b_n$

Ein besonders instabiler Lösungsvektor ist $b_n = \begin{bmatrix} 0 \\ \frac{1}{n} \\ \vdots \\ \frac{n-2}{n} \\ 1 \end{bmatrix}$

3 Fehlerrechnung

Definition 4 (Fehlermaße) Wir definieren für eine Tupel $T = (A_1, A_2, \dots, A_n)$ mit Störung $\tilde{T} = (A_1 + E_1, \dots, A_n + E_n)$:

- das absolute Fehlermaß:

$$[[E]]_{abs} := \max ||E_i||$$

- das relative Fehlermaß:

$$[[E]]_{rel} := \max \frac{||E_i||}{||A_i||}$$

Definition 5 (Maschinenepsilon) Das **Maschinen- ϵ** ist der relative Fehler, der bei Addition und Multiplikation von Skalaren auftritt. Er liegt für IEEE double-precision bei ca. 10^{-16}

Definition 6 (Kondition) Die Kondition einer Abbildung f im Punkt x ist definiert als:

$$\kappa(f, x) = \limsup_{y \rightarrow x} \frac{\|f(y) - f(x)\|}{\|y - x\|}$$

Man unterscheidet zwischen:

- **gut konditionierten Problemen:** $\kappa(f, x) = O(1)$
- **schlecht konditionierten Problemen:** $\kappa(f, x) \gg 1$
- **schlecht gestellten Problemen:** $\kappa(f, x) = \infty$

Theorem 6 Die Kondition einer linearen Gleichung $Ax = b$ hängt nur von A ab und ist:

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

4 Ausgleichsrechnung

Definition 7 (Lineares Ausgleichsproblem) Sei $A \in \mathbb{R}^{n \times m}$, $b \in \mathbb{R}^n$ mit $m \leq n$. Gesucht ist $x \in \mathbb{R}^m$. Sodass

$$\|Ax - b\|_2$$

minimiert wird. Es wird im Allgemeinen angenommen, dass $\text{Rang}(A) = m$.

Definition 8 (Normalengleichung) Gegeben ein Ausgleichsproblem A, b , nennt man:

$$A^T A x = A^T b$$

die Normalengleichung.

Theorem 7 (Lösung des Linearen Ausgleichsproblems mittels Normalengleichung)

Die Lösung der Normalengleichung ist das eindeutige gesuchte Minimum des Ausgleichsproblems.

Algorithm 7 Lösen des Ausgleichsproblems mittels Normalengleichung

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

$L \leftarrow \text{Cholesky}(A^T A)$

Löse $LL^T x = A^T b$ durch Vorwärts und Rückwärtssubstitution.

Die Stabilität des Algorithmus hängt von der Stabilität der Matrixmultiplikation $A^T A$ ab. Falls $\kappa(A^T A)$ groß ist und $\|Ax - b\|$ klein treten hier Stabilitätsprobleme auf. Um diesen entgegen zu wirken, kann man die Orthogonalisierungsmethode verwenden.

Theorem 8 (Aufwand der Lösungsmethoden) Der Aufwand beträgt:

1. Normalengleichung: $nm^2 + \frac{m^3}{3}$.

2. QR-Methode: $2nm^2 + 2\frac{m^3}{3}$

Algorithm 8 Lösen des Ausgleichproblems mittels QR-Methode

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

Zerlege $A = Q\hat{R}$ mit $\hat{R} = \begin{bmatrix} R \\ 0 \end{bmatrix}$ wobei $R \in \mathbb{R}^{m \times m}$ obere Dreiecksmatrix sei.

Löse $Rx = (Q^T b)_1$ wobei $(\cdot)_1$ die ersten m Elemente seien.

Für $n \gg m$ ist also die QR-Methode doppelt so teuer wie der Ansatz der Normalengleichung.

Theorem 9 (Kondition der Normalengleichung) Sei $A \in \mathbb{R}^{n \times m}$ und R wie in der Zerlegung für die QR-Methode. Es gilt:

$$\kappa(A^T A) = \kappa(R)^2$$

5 Eigenwertapproximation

Eine Einfache Idee um einen einzelnen Eigenwert mit Eigenwert einer Matrix $A \in \mathbb{R}^{n \times n}$ zu bestimmen, ist die Vektoriteration. Diese beruht darauf, dass Eigenräume hoffentlich anziehende Fixpunkte sind.

Algorithm 9 Vektoriteration

Require: $A \in \mathbb{R}^{n \times n}$ und Startvektor $x^{(0)} \in \mathbb{R}^n$.

for $i = 1, 2, 3, \dots$ **do**

$y^{(i)} \leftarrow Ax^{(i-1)}$

$\lambda^{(i-1)} \leftarrow (x^{(i-1)})^T y^{(i)}$

$x^{(i)} \leftarrow \frac{y^{(i)}}{\|y^{(i)}\|}$

end for

Theorem 10 (Konvergenz der Vektoriteration) Sei $A \in \mathbb{R}^{n \times n}$ mit EW $|\lambda_n| \leq \dots \leq |\lambda_2| < |\lambda_1|$ und EV $(v_i)_{i \in [n]}$. Dann gilt:

$$|\lambda^{(i)} - \lambda_1| \leq C_1(x^{(0)}) \left| \frac{\lambda_2}{\lambda_1} \right|^{2i}$$

$$\| \text{sign}(\lambda_1)^i x^{(i)} - \text{sign}(\beta_1) v_1 \|_2 \leq C_2(x^{(0)}) \left| \frac{\lambda_2}{\lambda_1} \right|^i$$

Wobei $\beta_1 := v_1^T x^{(0)}$ und C_1, C_2 Konstanten sind die vom Startwert abhängen

Durch diese Methode lässt sich nur ein einzelner Eigenvektor und auch nur der zum Betragsmäßig größten Eigenwert bestimmen. Um andere Eigenvektor zu berechnen, nutzen wir, dass der Betragsmäßig größte Eigenwert von $(A - \mu I)^{-1}$ der Kehrwert des nächsten Eigenwerts an μ ist.