

# 1 Vorlesung 01

## 1.1 Rechnerarithmetik

Wann wird approximativ gelöst:

- nicht möglich exakt zu lösen
- zu kompliziert/ zu aufwändig
- nicht nötig

### 1.1.1 Maschinenzahlen

Maschinenzahlen = Art wie ein Rechner Zahlen abspeichert, kann nicht unendlich viele abspeichern  
 beim Speichern von reellen Zahlen entsteht daher Rundungsfehler

Fixpunktzahlen    Gleitpunktzahlen    **normierte Gleitpunktzahlen**

123.45    1.2345 · 10<sup>2</sup>    **angleich 0**  
 (ganze Teil) (Zeil)    oder    **0.12345 · 10<sup>3</sup> ← Exponent**  
 ...    oder    **Mantisse Basis**

- Jede reelle Zahl  $x$  ausser 0 hat zu jeder Basis  $B$  eine **normierte Gleitpunkt Darstellung**  

$$x = (\pm 0.m_1 m_2 m_3 \dots)_B \cdot B^{(\pm e_1 e_2 e_3 \dots)_B}$$
 mit  $m_1, m_2, m_3, \dots$  und  $e_1, e_2, e_3, \dots$  aus  $0, \dots, B-1$  und  $m_1 \neq 0$ .
- Die gebrochene Zahl  

$$m = (\pm 0.m_1 m_2 m_3 \dots)_B$$
 heisst **Mantisse**, die ganze Zahl  

$$e = (\pm e_1 e_2 e_3 \dots)_B$$
 heisst **Exponent**.

Abbildung 1: Übersicht Gleitpunktzahlen

Je länger die Mantisse desto genauer kann Zahl abgespeichert werden  
 Je länger der Exponent desto grössere/kleinere Zahlen können abgespeichert werden  
 -> ansonsten gibt es positiven/negativen Overflow oder bei Abrundung auf 0 Underflow

# 2 Vorlesung 02

## 2.1 Float64

Standart für Numerische Berechnungen in Python  
 Befolgt IEEE 754 (legt weltweit einheitlich fest, wie Gleitpunktzahlen in Rechnern abgespeichert werden)

- Eine normale Zahl  $x$  im **Float64** hat folgende Gestalt:  

$$x = (-1)^S \cdot (1+m) \cdot 2^{e-b}$$
 mit  
 $S$  aus  $\{0,1\}$   
 $m = (0.m_1 m_2 \dots m_{52})_2$  mit  $m_1, m_2, \dots, m_{52}$  aus  $\{0,1\}$   
 $e = (e_1 e_2 \dots e_{11})_2$  mit  $e_1, e_2, \dots, e_{11}$  aus  $\{0,1\}$ ,  
 wobei nicht alle 0 und nicht alle 1 sind  
 $b = (11\ 1111\ 1111)_2 = 1023$
- $x$  wird wie folgt abgespeichert:  

S	e <sub>1</sub>	e <sub>2</sub>	...	e <sub>11</sub>	m <sub>1</sub>	m <sub>2</sub>	...	m <sub>52</sub>
↑	Exponentenbits				Mantissebits			
Signbit	64 Bits							

Abbildung 2: Aufbau Float64

- Normierung nicht durch  $m_1 \neq 0$  sondern durch Addierung von 1 zu  $m$
- Exponent hat kein Vorzeichen, dies wird durch Abzug von  $b$  (=Bias = 1023) gelöst
- es gibt noch nicht normale Zahlen: ZBsp: 0, +/- inf

## 2.2 Rundungsfehler und Maschinengenauigkeit

a)  $e_{\max} = (111)_2 = 4 + 2 + 1 = 7$

b)

$$x_{\min} = B^{e_{\min}-1} = 2^{-7-1} = 2^{-8} = 0.00390625 = 0.3906 \cdot 10^{-2}$$

$$x_{\max} = (1 - B^{-n})B^{e_{\max}} = (1 - 2^{-5})2^7 = 2^7 - 2^2 = 124 = 0.1240 \cdot 10^3$$

c)

$$eps_1 = \frac{B}{2} B^{-n} = 2^{-5} = 0.03125$$

$$eps_2 = \frac{B}{2} B^{-n} = 8 \cdot 16^{-2} = 0.03125$$

Abbildung 3: Aufgaben Maschinengenauigkeit

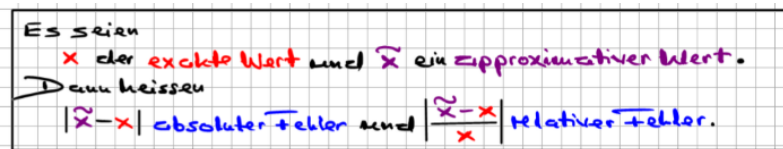


Abbildung 4: Absoluter vs. Relativer Fehler:

Meist ist relativer Fehler aussagekräftiger als absoluter

### 2.2.1 Rundungsprinzipien

Reelle Zahlen werden auf nächstgelegene Maschinenzahl gerundet  
Liegt  $x$  genau in der Mitte wird auf betragsmässig grössere Zahl gerundet  
(Rounding ties away from 0)

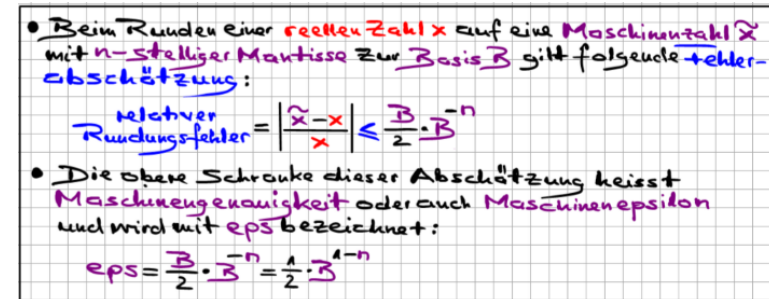


Abbildung 5: Rundungsfehler, Maschinengenauigkeit, Maschinenepsilon

## 2.3 Fehler bei Grundrechenoperationen und Auslöschung

### 2.3.1 Fehler bei Grundrechenoperationen

Passieren bei allen Operationen, auch bei eher kleinen Zahlen, durch; Normieren, Runden, Exponenten Angleichen, Operation durchführen, Runden, Denormieren bis zu etwa  $\epsilon$

### 2.3.2 Auslöschung

Bei Subtraktion von fast gleich grossen Zahlen kann ein grosser Relativer Fehler entstehen = Auslöschung:

Dabei löschen sich jeweils die Vorderen Signifikanten Stellen aus (da sie gleich sind) und es bleiben nur wenige Signifikante Stellen übrig

## 3 Vorlesung 03

### 3.1 Fehler bei Funktionsauswertung (Zsmhang Input/Output Fehler)

- Absolut:  $|f(\tilde{x}) - f(x)| \approx |f'(x)| * |\tilde{x} - x|$
- Relativ:  $|\frac{f(\tilde{x}) - f(x)}{f(x)}| \approx |\frac{\tilde{x} - x}{x}| * |\frac{f'(x) * x}{f(x)}|$
- Konditionszahl:  $K(x) = |\frac{f'(x) * x}{f(x)}|$

- Daraus folgt:  $\left| \frac{f(x) - f(x_0)}{f(x_0)} \right| = K(x_0) * \left| \frac{x - x_0}{x_0} \right| \Rightarrow |x - x_0| \leq p/K(x_0) * x_0$

### 3.1.1 Konditionierung (unabhängig von Verfahren mit dem Funktion ausgewertet wird)

$K(x) \leq 10 \Rightarrow f(x)$  gut konditioniert

$K(x) > 10 \Rightarrow f(x)$  schlecht konditioniert

### 3.2 Numerische Bestimmung von Nullstellen

Nullstellen: = 0 Auflösen

Fixpunkte: nach x Auflösen, =  $F(x) = \text{Fixpunkt}$

- Intervallhalbierungsverfahren:
  1. Wähle Intervall  $[x_L, x_R]$  mit  $[y_L = g(x_L) < 0$  und  $y_R = g(x_R) > 0$
  2. Bilde Intervallmitte:  $x_M = (x_L + x_R)/2$  und  $y_M = g(x_M)$
  3. Prozess:
    - Wenn:  $y_M > 0$  ersetze  $x_L$  durch  $x_M$  -> neues Intervall
    - Wenn:  $y_M < 0$  ersetze  $x_R$  durch  $x_M$  -> neues Intervall
- Fixpunktiteration (Vorlesung 04)
- Newton-Verfahren

## 4 Vorlesung 04

### 4.1 Definitionen Fixpunktiteration

- Zahl  $\bar{x}$  heisst Fixpunkt von Funktion, wenn gilt:  $F(\bar{x}) = \bar{x}$
- jede Gleichung in x kann umgeformt werden auf Fixpunktform ( $x = F(x)$  = ...)
- Fixpunktiteration:  $x_0, x_1 = F(x_0), x_2 = F(x_1), \dots$
- Wenn F mit  $x_0$  konvergiert, dann immer gegen einen Fixpunkt  $F(\bar{x})$  -> numerisches Verfahren z. Bestimmung von Fixpunkten

- Kann auch divergieren: dann kann Fixpunkt nicht bestimmt werden/ existiert nicht

- Damit F mit  $x_0$  konvergiert muss (= Anziehender Fixpunkt):  $|F'(\bar{x})| < 1$  und  $x_0$  nahe  $\bar{x}$ , sonst divergiert (= Abstossender Fixpunkt)

- Banachsche Fixpunktsatz: Kriterien f. Eindeutigkeit und Existenz v Fixpunkt, Konvergenz Fixpunktiteration, Fehlerabschätzung

- Wenn  $F: I[a, b] \rightarrow R$  eine Funktion bei der I stetig differenzierbar
- $\lambda = \max |F'(x)|$  mit  $[a \leq x \leq b]$  (betragsmässig grösster Wert der Ableitung)

- wenn  $\lambda < 1$ :

- \* F hat genau einen Fixpunkt  $\bar{x}$  in I
- \* die Fixpunktiteration konvergiert für jden Startwert  $x_0$  gegen  $\bar{x}$
- \* Es betragen  $|x_n - \bar{x}| \leq \frac{\lambda^n}{1 - \lambda} * |x_1 - x_0|$  [a-priori Fehlerabschätzung]
- $|x_n - \bar{x}| \leq \frac{\lambda^n}{1 - \lambda} * |x_n - x_{n-1}|$  [a-posteriori Fehlerabschätzung]

## 5 Vorlesung 05

### 5.1 Newtonverfahren

Vorteile: konvergiert fast immer und meist sehr schnell (Konvergenzordnung: 2)

1. Gleichung auf 0 auflösen
2. Startwert nahe Nullstelle  $\bar{x}$  wählen
3. Tangente an Stelle  $x_0$  an den Graphen legen, Schnittstelle mit x Achse =  $x_1$
4. Wiederholen von 3 um  $x_2$  zu erhalten

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Konvergiert sicher (kann auch sonst) wenn: Nullstelle, Startwert und alle Iterierten Werte im Intervall  $[a,b]$  liegen sodass: [für alle  $x$  aus  $[a,b]$ ]

$$\left| \frac{f(x_n) * f''(x_n)}{f'(x_n)^2} \right| < 1$$

### 5.1.1 Vereinfachtes Newton-Verfahren

Konvergiert i.d. Regel auch aber langsamer (Konvergenzordnung: 1)

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_0)}$$

### 5.2 Sekanten-Verfahren

Mit 2 Startwerten eine Sekante an Graph und schneiden mit x Achse (Konvergenzordnung: 1.6)

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} * f(x_n)$$

Konvergiert langsamer als Newton, schneller als Vereinfacht Newton, Kommt ohne Ableitung aus

### 5.3 Konvergenzordnung

Mass für Konvergenzgeschwindigkeit eines iterativen Verfahrens

Mit  $[q \text{ Zahl} \geq 1, x_n \text{ Zahlenfolge Resultate v. iterativem Verfahren}]$  Dann hat Verfahren die Konvergenzordnung  $q$  wenn  $c > 0$  existiert.

$$|x_{n+1} - \bar{x}| \leq c * |x_n - \bar{x}|^q$$

Wenn  $q = 1$  muss  $c < 1$  und ist lineare Konvergenz,  $q = 2$ : quadratische Konvergenz

Um aus Resultatwerten Konvergenzordnung zu erhalten: Prüfen  $f$   $q$  damit  $c$  konstant

$$c = \frac{|x_{n+1} - \bar{x}|}{|x_n - \bar{x}|^q}$$

### 5.4 Fehlerabschätzungskriterium

Feststellung ob absolute Fehler  $|x_n - \bar{x}|$  kleiner als Vorgegebene Schranke  $\epsilon > 0$

Wenn  $f(x_n - \epsilon) * f(x_n + \epsilon) < 0$  gilt  $|x_n - \bar{x}| < \epsilon$

## 6 Vorlesung 06

Lineare Gleichungssysteme Gauss Algorithmus (idee, elementare operationen, Stadndartisiert, Pivotisierung) LGS Anwendung: Bsp PageRank (Google, wichtigkeit d Websites)

### 6.1 LGS Bezeichnungen

- System von  $m$  linearen Gleichungen mit  $n$  Unbekannten =  $m * n$  LGS
- Zahlenwerte = Koeffizienten, Zahlenvariablen = Unbekannte
- kann dargestellt werden als  $A * x = b$  mit  $[A \text{ (Grossbuchstabe)} = \text{Matrix}]$
- wenn Anz Unbekannte = Anz Gleichungen: Koeffizientenmatrix  $A$  ist quadratisch
- LGS ist **regulär** wenn Determinante  $\neq 0$

### 6.2 Gauss-Algorithmus

Vorwärtselementation und Rückwärtseinsetzen

#### Elementare Zeilenoperationen (z Vorwärtselementation)

- Zeilensubtraktion: v. einer Zeile wird ein Vielfaches einer darüberliegenden Zeile subtrahiert
- Zeilenvertauschung: Zwei Zeilen werden vertauscht
- Pivots müssen in Numerik nicht 1 sein, einfach nicht 0

### 6.3 Pivotisierung (numerische Fehlerproblematik)

Wenn grösse d Zahl gegeben: (bsp 2 Mantissestellen) entsteht Fehler durch runden und bei Division s. grosser Fehler  
 Daher muss immer die Betragsmässig grösste Zahl als Pivot hochgetauscht werden

## 7 Vorlesung 07

### 7.1 Lineare Gleichungssysteme: LR-Zerlegung

Gleichungssystem:  $A \cdot x = b$  [mit  $A$  = Matrix,  $x, b$  = Vektoren]

1. Matrix  $A$  wird mit Gauss-Algorithmus auf rechts-obere Dreiecksform  $R$  gebracht (alles unter Diagonale ist 0, Diagonale egal)
2. Falls Gauss Algorithmus ohne Zeilenvertauschungen durchgeführt werden kann ergibt sich  $L$  (links untere Dreiecksmatrix, alles 0 oberhalb der Diagonale, Diagonale ist 1)
3. Damit ist  $A = L \cdot R$
4.  $L \cdot y = b$  (nach  $y$ )
5.  $R \cdot x = y$  (nach  $x$ ) [mit  $x$  ist tatsächliche Lösung des LGS]



Abbildung 6: Linksuntere Dreiecksmatrix

### 7.2 Lineare Gleichungssysteme: PLR Zerlegung

Falls Zeilenvertauschung benötigt (zBsp wegen Pivotisierung) muss LR zu PLR erweitert werden.

$P \cdot A = L \cdot R$  [mit  $P$  = Permutationsmatrix: quadratische Matrix wie Einheitsmatrix aber Zeilen in anderer Reihenfolge]

1.  $R$  entsteht wie bei LR
2.  $L$  entsteht wie LR aber wenn Zeilen vertauscht werden müssen auch Einträge in  $L$  vertauscht werden
3.  $P$  entsteht aus Einheitsmatrix die dieselben Vertauschungen erhält

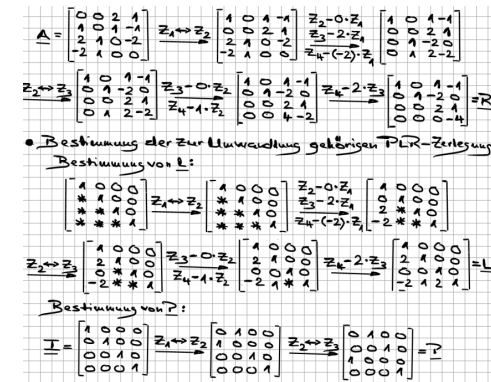


Abbildung 7: PLR Zerlegung

## 8 Vorlesung 08

### 8.1 Orthogonale Matrizen

Eine  $n \cdot n$  Matrix  $Q$  heisst orthogonal wenn:  $Q^T \cdot Q = I$  mit  $I$  = Einheitsmatrix]  
 Eigenschaften Orthogonale Matrizen:

- Spaltenvektoren stehen paarweise senkrecht und haben Länge 1
- Orthogonale Matrizen sind **regulär**:  $Q^{-1} = Q^T$
- Produkt von 2 orthogonalen Matrizen ist wieder orthogonale Matrix
- Orthogonale Matrizen sind **gut konditioniert**: falls  $x$  und  $\tilde{x}$  nahe beieinander sind auch  $Q \cdot x$  und  $Q \cdot \tilde{x}$  nahe beieinander

## 8.2 QR - Zerlegung

$A = Q \cdot R$  mit  $[A = n \cdot n$  Matrix,  $R =$  rechteckige Dreiecksmatrix,  $Q =$  Orthogonale Matrix]

QR Zerlegung existiert zu jeder  $n \cdot n$  Matrix und ist eindeutig bis auf Vorzeichen

## 8.3 Lösen von LGS via QR Zerlegung

1.  $A \cdot x = b$  [ein LGS]
2.  $A = Q \cdot R$  [eine QR Zerlegung von A]
3.  $Q \cdot R \cdot x = b$
4.  $R \cdot x = Q^T \cdot b$  [ $Q^{-1} = Q^T$ ]: Zu Lösen durch Rückwärtseinsetzen

## 8.4 Householder Matrizen

Matrix  $H$  ( $n \cdot n$ ) der Form:  $H = I - 2 \cdot \mu \cdot \mu^T$  mit  $[I =$  Einheitsmatrix,  $\mu$  Vektor d Länge 1]

Eigenschaften:

- Householder Matrizen sind orthogonal
- Geometrisch: Spiegelung:
  - $n=2$ : Spiegelung an Geraden die Senkrecht zu  $\mu$
  - $n=3$ : Spiegelung an Ebene Senkrecht zu  $\mu$
  - $n>3$ : (Hyperraum) => Hyperebene die Senkrecht zu  $\mu$

### 8.4.1 Berechnung QR Zerlegung mit Householder-Verfahren

Braucht  $n-1$  Iterationen

1. Initialisierung:  $R = A$  und  $Q = I_n$
2.  $i$ -te Iteration:
  - $A_i = R$

- $a_i$  (=  $i$ -te Spalte)
- $a_{i,1}$  (oben links Zahl)
- $\|a_i\|$  (Länge Vektor)
- $sign(a_{i,1})$  (1 für  $a_{i,1} \geq 1$  sonst -1)
- $e_1$  (Einheitsvektor mit erster Zahl =  $a_{i,1}$ )
- $v_i = a_i + sign(a_{i,1}) \cdot \|a_i\| \cdot e_1$
- $\mu_i = \frac{1}{\|v_i\|} \cdot v_i$
- $H_i = I_{n-i+1} - 2 \cdot \mu \cdot \mu^T$
- $Q_i = \begin{pmatrix} I_{i-1} & 0 \\ 0 & H_i \end{pmatrix}$
- $R_{neu} = Q_i \cdot R_{alt}$  und  $Q_{neu} = Q_{alt} \cdot Q_i^T$

## 9 Vorlesung 09

### 9.1 Eigenwert

Beispiel

$$A = \begin{pmatrix} 2 & 1 & 2 \\ 1 & 2 & 2 \\ 1 & 1 & 3 \end{pmatrix} \quad (A - \lambda \cdot E_n) = \begin{pmatrix} 2-\lambda & 1 & 2 \\ 1 & 2-\lambda & 2 \\ 1 & 1 & 3-\lambda \end{pmatrix}$$

$$\chi_\lambda(\lambda) = \det(A - \lambda \cdot E_n)$$

$$\chi_\lambda(\lambda) = -\lambda^3 + 7\lambda^2 - 11\lambda + 5$$

Nullstellen  
= gesuchte Eigenwerte

Nullstellen  
 $\lambda_1 = \lambda_2 = 1 \quad \lambda_3 = 5$

Abbildung 8: Eigenwerte Berechnung

### 9.2 Vektor (x) und Matrizen (A) Normen

- 1-Norm / Summennorm:  $\|x\|_1 = |x_1| + \dots + |x_n|$
- 2-Norm / Euklidische Norm:  $\|x\|_2 = \sqrt{x_1^2 + \dots + x_n^2}$
- $\infty$  Norm / Maximumnorm:  $\|x\|_\infty = \max(|x_1|, \dots, |x_n|)$

- 1-Norm / Spaltensummennorm:  $\|A\|_1 = \max(|x_{11}| + \dots + |x_{n1}|, \dots, |x_{1n}| + \dots + |x_{nn}|)$
- 2-Norm / Spektralnorm:  $\|x\|_2 = \max(\sqrt{\lambda})$  mit  $\lambda$  ist Eigenwert von  $A^T * A$
- $\infty$  Norm / Zeilensummennorm:  $\|A\|_\infty = \max(|x_{11}| + \dots + |x_{1n}|, \dots, |x_{n1}| + \dots + |x_{nn}|)$

### Eigenschaften Normen

- $\|x\| \geq 0$  und  $\|A\| \geq 0$
- $\|x\| = 0 \Leftrightarrow x = 0$  und  $\|A\| = 0 \Leftrightarrow A = 0$
- $\|\lambda * x\| = |\lambda| * \|x\|$  und  $\|\lambda * A\| = |\lambda| * \|A\|$
- $\|x + y\| \leq \|x\| + \|y\|$  und  $\|A + B\| \leq \|A\| + \|B\|$
- für jeweils jd entsprechenden Normen (1 und 1 Norm) gilt:  
 $\|A * x\| \leq \|A\| * \|x\|$

### 9.3 Lineare Gleichungssysteme: Fehlerrechnung

Wie stark ändert sich die Lösung eines linearen Gleichungssystems wenn deren Koeffizienten sich ändern? Können durch Normen quantifiziert werden.  
 mit  $[A * x = b$  und  $\tilde{A} * \tilde{x} = \tilde{b}]$ :

- $\|\tilde{x} - x\| \leq \|A^{-1}\| * \|\tilde{b} - b\|$
- $\frac{\|\tilde{x} - x\|}{\|x\|} \leq \|A\| * \|A^{-1}\| * \frac{\|\tilde{b} - b\|}{\|b\|}$
- $cond(A) = \|A\| * \|A^{-1}\|$  : gross = schlecht konditioniert

mit  $[A * x = b$  und  $\tilde{A} * \tilde{x} = \tilde{b}]$  und  $[cond(A) * \frac{\|\tilde{A} - A\|}{\|A\|} < 1]$ :

- $\frac{\|\tilde{x} - x\|}{\|x\|} \leq \frac{cond(A)}{1 - cond(A) * \frac{\|\tilde{A} - A\|}{\|A\|}} * (\frac{\|\tilde{A} - A\|}{\|A\|} + \frac{\|\tilde{b} - b\|}{\|b\|})$

### 9.4 Aufwandabschätzung

Für d Lösen eines LGS mit n Gleichungen und n Unbekannten im Gauss-Algorithmus müssen (für grosse n):  $\frac{2}{3} * n^3$  Grundrechenoperationen durchgeführt werden

Bsp: Rechner kann  $10^9$  Floats/Sekunde ausführen, für  $n = 10^3 \Rightarrow 2/3$  Sekunden

## 10 Vorlesung 10

### 10.1 Iterative Lösungsverfahren

Iterative Lösungsverfahren haben deutlich kleineren Rechenaufwand als bsp Gauss-Algorithmus (nur  $n^2$  statt  $n^3$  für n\*n LGS), Liefern aber nur Näherungslösungen und haben manchmal keine Konvergenz

#### Grundprinzip

1. Lineares Gleichungssystem:  $A * x = b$
2. Fixpunktform:  $x = F(x)$
3. Fixpunktiteration:  $x_1 = F(x_0) \dots$

#### 10.1.1 Jacobi Verfahren (= Gesamtschrittverfahren)

$A * x = b$  wird mit Jacobi Verfahren ausgehend von einem Startvektor  $x_0$  durch die Iteration: Mit  $[D, L$  und  $R = \text{Diagonal, Links und Rechtsteil von } A]$

$$x^{(k+1)} = F(x^{(k)}) = B * x^{(k)} + c$$

$$B = -D^{-1} * (L + R) \quad \text{und} \quad c = D^{-1} * b$$

Letztes Beispiel zum Ersten (aus Skript)

$$A \cdot x = b \text{ mit } A = \begin{bmatrix} 4 & -1 & 1 \\ -2 & 5 & 1 \\ 1 & -2 & 5 \end{bmatrix}, b = \begin{bmatrix} 5 \\ 11 \\ 12 \end{bmatrix} \text{ und Lösung } x^* = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$$A \cdot x = b \rightarrow \begin{cases} 4x_1 - x_2 + x_3 = 5 \\ -2x_1 + 5x_2 + x_3 = 11 \\ x_1 - 2x_2 + 5x_3 = 12 \end{cases} \rightarrow \begin{cases} 4x_1 = x_2 - x_3 + 5 \\ 5x_2 = 2x_1 - x_3 + 11 \\ 5x_3 = x_1 + 2x_2 + 12 \end{cases}$$

$$\rightarrow \begin{cases} x_1 = 0.25x_2 - 0.25x_3 + 1.25 \\ x_2 = 0.4x_1 - 0.2x_3 + 2.2 \\ x_3 = -0.2x_1 + 0.4x_2 + 2.4 \end{cases}$$

$$\rightarrow \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 & 0.25 & -0.25 \\ 0.4 & 0 & -0.2 \\ -0.2 & 0.4 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1.25 \\ 2.2 \\ 2.4 \end{bmatrix} \rightarrow x = \underbrace{B \cdot x + c}_{F(x)}$$

Fixpunktiteration mit Startvektor  $x^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ :

$$x^{(1)} = F(x^{(0)}) = B \cdot x^{(0)} + c = \begin{bmatrix} 0 & 0.25 & -0.25 \\ 0.4 & 0 & -0.2 \\ -0.2 & 0.4 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1.25 \\ 2.2 \\ 2.4 \end{bmatrix} = \begin{bmatrix} 1.25 \\ 2.2 \\ 2.4 \end{bmatrix}$$

$$x^{(2)} = F(x^{(1)}) = B \cdot x^{(1)} + c = \begin{bmatrix} 0 & 0.25 & -0.25 \\ 0.4 & 0 & -0.2 \\ -0.2 & 0.4 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1.25 \\ 2.2 \\ 2.4 \end{bmatrix} + \begin{bmatrix} 1.25 \\ 2.2 \\ 2.4 \end{bmatrix} = \begin{bmatrix} 1.2 \\ 2.22 \\ 3.03 \end{bmatrix}$$

usw.  
konvergiert gegen  $x^* = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$

Abbildung 9: Jacobi Verfahren

$$\rightarrow D \cdot x^{(k+1)} = -L \cdot x^{(k)} - R \cdot x^{(k)} + b$$

$$\rightarrow D \cdot x^{(k+1)} = -(L+R) \cdot x^{(k)} + b$$

$$\rightarrow x^{(k+1)} = D^{-1} \cdot (-(L+R) \cdot x^{(k)} + b)$$

$$\rightarrow x^{(k+1)} = \underbrace{-D^{-1} \cdot (L+R)}_B \cdot x^{(k)} + \underbrace{D^{-1} \cdot b}_c = F(x^{(k)})$$

Abbildung 10: Jacobi Verfahren Umformungen

### 10.1.2 Gauss Seidel Verfahren (= Einzelschrittverfahren)

Wird gleich wie bei Jacobi auf Fixpunktform gebacht, aber Iteration wird nacheinander durchgeführt und direkt für die neue Zeile die berechneten Werte eingesetzt.

$A \cdot x = b$  wird mit Gauss Seidel Verfahren ausgehend von einem Startvektor  $x_0$  durch die Iteration: Mit  $[D, L$  und  $R =$  Diagonal, Links und Rechtsteil von  $A]$

$$x^{(k+1)} = F(x^{(k)}) = B * x^{(k)} + c$$

$$B = -(L + D)^{-1} * R \quad \text{und} \quad c = (L + D)^{-1} * b$$

Neue Iteration mit Gauss-Seidel-Verfahren:

$$x_1^{(k+1)} = 0.25x_2^{(k)} - 0.25x_3^{(k)} + 1.25$$

$$x_2^{(k+1)} = 0.4x_1^{(k+1)} - 0.2x_3^{(k)} + 2.2$$

$$x_3^{(k+1)} = -0.2x_1^{(k+1)} + 0.4x_2^{(k+1)} + 2.4$$

$$x^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, x^{(1)} = \begin{bmatrix} 0.25 \cdot 0 - 0.25 \cdot 0 + 1.25 \\ 0.4 \cdot 1.25 - 0.2 \cdot 0 + 2.2 \\ -0.2 \cdot 1.25 + 0.4 \cdot 2.2 \end{bmatrix} = \begin{bmatrix} 1.25 \\ 2.7 \\ 3.23 \end{bmatrix}$$

$$x^{(2)} = \begin{bmatrix} 0.25 \cdot 2.7 - 0.25 \cdot 3.23 + 1.25 \\ 0.4 \cdot 1.1175 - 0.2 \cdot 3.23 + 2.2 \\ -0.2 \cdot 1.1175 + 0.4 \cdot 2.001 \end{bmatrix} = \begin{bmatrix} 1.1175 \\ 2.0004 \\ 2.5769 \end{bmatrix}$$

usw.

Abbildung 11: Gauss-Seidel Verfahren

$$\rightarrow D \cdot x^{(k+1)} = -L \cdot x^{(k+1)} - R \cdot x^{(k)} + b$$

$$\rightarrow D \cdot x^{(k+1)} + L \cdot x^{(k+1)} = -R \cdot x^{(k)} + b$$

$$\rightarrow (D+L) \cdot x^{(k+1)} = -R \cdot x^{(k)} + b$$

$$\rightarrow x^{(k+1)} = (D+L)^{-1} \cdot (-R \cdot x^{(k)} + b)$$

$$\rightarrow x^{(k+1)} = \underbrace{-(D+L)^{-1} \cdot R}_B \cdot x^{(k)} + \underbrace{(D+L)^{-1} \cdot b}_c = F(x^{(k)})$$

Abbildung 12: Gauss-Seidel Verfahren Umformungen

## 11 Vorlesung 11

### 11.1 Iterative Lösungsverfahren

#### 11.1.1 Ableitungen v Matrizen

Pro Zahl in der Matrix eine Ableitung. Da wird dann diese Zahl als unbekannte, der Rest als Konstanten gesetzt.

Dadurch entsteht dann auch Ableitungsmatrix



$$D\bar{f}(x) = \begin{bmatrix} \frac{df_1(x_1, x_2)}{dx_1} & \frac{df_1(x_1, x_2)}{dx_2} \\ \frac{df_2(x_1, x_2)}{dx_1} & \frac{df_2(x_1, x_2)}{dx_2} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = B$$

Abbildung 13: Ableitungsmatrix

Wenn LGS Diagonaldominante Matrix A: dann konvergiert sowohl Jacobi als auch Gauss-Seidel.

### 11.1.4 Diagonaldominanz

Eine quadratische Matrix ist diagonaldominant wenn **eine** der beiden Bedingungen zutrifft.

- In jeder Zeile ist das Diagonalelement betragsmässig Grösser als Summe (der Beträge) aller anderen Elemente in der Zeile
- In jeder Spalte ist das Diagonalelement betragsmässig Grösser als Summe (der Beträge) aller anderen Elemente in der Spalte

### 11.1.2 Anziehend / Abstossende Fixpunkte

Gegeben sei Funktion  $f: \mathbb{R}^n \rightarrow \mathbb{R}^n, x \mapsto f(x) = B \cdot x + c$   
 mit  $B \in \mathbb{R}^{n \times n}$  und  $c \in \mathbb{R}^n$ .  
 Dann heisst  $x^* \in \mathbb{R}^n$  Fixpunkt (vektor) von  $f$ , wenn gilt:  
 $f(x^*) = x^*$   
 Weiter heisst ein Fixpunkt  $x^*$  von  $f$  anziehend, wenn es eine Matrixnorm  $\|\cdot\|$  gibt, so dass gilt:  
 $\|B\| < 1$   
 Ansonsten heisst der Fixpunkt  $x^*$  abstossend.

Abbildung 14: Anziehende und Abstossende Fixpunkte

### 11.2 Komplexe Zahlen

$i =$  imaginäre Einheit,  $i = \sqrt{-1}$ , Erweiterung der reellen Zahlen in die komplexen Zahlen

### 11.1.3 Fehlerabschätzung

Gegeben sei eine Fixpunktgleichung der Form  $f(x) = x$  mit  $f(x) = B \cdot x + c$ .  
 Wenn bezüglich irgendeiner Norm  $\|\cdot\|$   
 $\|B\| < 1$   
 ist, dann gilt:  
 •  $f(x)$  besitzt genau einen Fixpunktvektor  $x^*$ .  
 • Die Fixpunktiteration  $x^{(k+1)} = f(x^{(k)}) = B \cdot x^{(k)} + c$  konvergiert für jeden Startvektor  $x^{(0)}$  gegen den Fixpunktvektor  $x^*$ .  
 • Der Abstand von  $x^{(k)}$  zu  $x^*$  kann abgeschätzt werden durch  
 $\|x^{(k)} - x^*\| \leq \frac{\|B\|^k}{1 - \|B\|} \cdot \|x^{(1)} - x^{(0)}\|$  a-priori-Abschätzung  
 $\|x^{(k)} - x^*\| \leq \frac{\|B\|^k}{1 - \|B\|} \cdot \|x^{(k)} - x^{(k-1)}\|$  a-posteriori-Abschätzung

Abbildung 15: Konvergenz und Fehlerabschätzung, Nötige Anzahl Iterationen

• Die Menge der komplexen Zahlen  $\mathbb{C}$  besteht aus allen Zahlen der Form  $z = x + y \cdot i$ , wobei  $x$  und  $y$  reelle Zahlen sind und  $i$  ein Symbol mit der Eigenschaft  $i^2 = -1$ .  
 • Das Symbol  $i$  heisst imaginäre Einheit. Die reelle Zahl  $x$  heisst Realteil von  $z$ , die reelle Zahl  $y$  Imaginärteil von  $z$ . Man schreibt:  $x = \text{Re}(z)$  und  $y = \text{Im}(z)$ .  
 • Eine komplexe Zahl  $z = x + iy$  kann als Punkt in der Ebene mit Koordinaten  $(x, y)$  oder als Vektor mit Komponenten  $\begin{bmatrix} x \\ y \end{bmatrix}$  dargestellt werden. Die Ebene heisst dann komplexe Zahlenebene.  
 • Die komplexe Zahl  $z^* = x - y \cdot i$  heisst die zu  $z = x + y \cdot i$  konjugiert komplexe Zahl.  $z^*$  entsteht aus  $z$  durch Spiegelung an der  $x$ -Achse.  
 • Die reelle Zahl  $|z| = \sqrt{x^2 + y^2}$  heisst Betrag von  $z = x + y \cdot i$ . Der Betrag ist gleich der Länge des Vektors  $\begin{bmatrix} x \\ y \end{bmatrix}$ .

Abbildung 16: Definition Komplexe Zahlen

Rechenregeln bleiben erhalten:

- Addition / Subtraktion:  $x - y = (2 + 4i) - (-2 + i) = 2 + 2 + 4 \cdot i - i = 4 + 3 \cdot i$
- Multiplikation / Division:  $i \cdot i = i^2 = -1$ , bei Division auf reellen Nenner erweitern versuchen.

## 12 Vorlesung 12

### 12.1 Komplexe Zahlen

#### 12.1.1 Normalform

$$z = x + y * i$$

Mit [x, y = reelle Zahlen,  $i^2 = -1$  = imaginäre Einheit, x = Realteil, y = Imaginärteil]

#### 12.1.2 Polarform = trigonometrische Form

Geometrische Darstellung von z als Punkt/Vektor in Ebene

Mit [r = Abstand Nullpunkt-z (Polarradius),  $\varphi$  = Winkel zwischen x-Achse und z (Polarwinkel) (Gegenuhrzeigersinn, in Radian ( $\pi$ ))]

- $x = r * \cos(\varphi)$
- $y = r * \sin(\varphi)$
- $r = \sqrt{x^2 + y^2}$
- 

$$\varphi = \begin{cases} \text{für } x > 0 : \tan^{-1}\left(\frac{y}{x}\right) \\ \text{für } x < 0 : \tan^{-1}\left(\frac{y}{x}\right) + \pi \end{cases}$$

#### 12.1.3 Euler'sche Formel

Entsteht aus Potenzreihen

$$e^{i*\varphi} = \cos(\varphi) + i * \sin(\varphi)$$

mit falls  $\varphi = \pi$  entsteht schönste Formel der Mathematik:  $e^{i*\pi} + 1 = 0$

#### 12.1.4 Exponentialform

$$z = x + y * i = r * \cos(\varphi) + r * \sin(\varphi) * i = r * (\cos(\varphi) + i * \sin(\varphi))$$

$$z = r * e^{i*\varphi}$$

#### 12.1.5 Division / Multiplikation / Potenz / Wurzel

- $z_1 * z_2 = r_1 * r_2 * e^{i*(\varphi_1 + \varphi_2)}$
- $z_1 / z_2 = r_1 / r_2 * e^{i*(\varphi_1 - \varphi_2)}$
- $z^n = (r * e^{i*\varphi})^n = r^n * (e^{i*\varphi})^n = r^n * e^{i*n*\varphi}$
- Wurzeln sind nicht eindeutig:

Es seien w eine komplexe Zahl mit Exponentialform  $w = r * e^{i*\varphi}$  und n eine positive ganze Zahl. Dann besitzt w die n-ten Wurzeln  $z_k = r^{1/n} * e^{i*\varphi_k}$  mit  $\varphi_k = \frac{\varphi + k*2\pi}{n}$  für  $k = 0, 1, \dots, n-1$ . Die n Wurzeln  $z_0, z_1, \dots, z_{n-1}$  bilden in der komplexen Zahlenebene die Ecken eines regelmäßigen n-Ecks.

Abbildung 17: Wurzeln Komplexe Zahlen

## 13 Vorlesung 13

### 13.1 Eigenwerte (EW) und Eigenvektoren (EV) von Matrizen

#### 13.1.1 Definition

Es seien

- $\Delta$  eine reelle  $n \times n$ -Matrix,
- $x$  ein reeller oder komplexer  $n$ -Vektor verschieden vom Nullvektor  $0$ ,
- $\lambda$  eine reelle oder komplexe Zahl.

Dann heißt  $x$  Eigenvektor zum Eigenwert  $\lambda$  von  $\Delta$ , wenn gilt:

$$\Delta * x = \lambda * x$$

Abbildung 18: Definition Eigenwerte / Eigenvektoren

#### 13.1.2 Eigenschaften

A ist reelle  $n * n$  Matrix

E ist  $n * n$  Einheitsmatrix

- Ist  $x$  EV zum EW  $\lambda$  einer Matrix  $A$ , dann ist auch jedes Vielfache  $x' = \lambda * x$  mit  $[\lambda \neq 0]$  EV zum EW  $\lambda$  von  $A$ 
  - usually wird von den EV die Vielfache voneinander sind derjenige mit 2-Norm 1 angegeben
- der Term  $p(\lambda) = \det(A - \lambda * E)$  ein Polynom (Grad  $n$  in  $\lambda$ ) und heisst charakteristisches Polynom von  $A$
- Eigenwerte  $\lambda$  von  $A$  sind Nullstellen von  $p(\lambda)$
- In  $\mathbb{C}$  besitzt  $p(\lambda)$  immer genau  $n$  Nullstellen und damit  $A$  immer genau  $n$  Eigenwerte, wobei teilweise mehrfach auftreten können
- Die Vielfachheit mit welcher EW  $\lambda$  Nullstelle von  $p(\lambda)$  ist heisst **algebraische Vielfachheit** des EW  $\lambda$
- Die Menge aller Eigenwerte heisst **Spektrum** von  $A$
- Das Maximum der Beträge aller Eigenwerte heisst **Spektralradius** von  $A$
- Alle Eigenvektoren  $x$  zu einem EW  $\lambda$  bilden zusammen mit dem Nullvektor einen Vektorraum (= Eigenraum zum EW  $\lambda$ )
- Die Dimension des Eigenraums heisst **geometrische Vielfachheit**
- geometrische Vielfachheit von EW  $\leq$  algebraische Vielfachheit EW

### 13.1.3 Berechnung

A reelle  $n \times n$  Matrix:

- Eigenwerte  $\lambda$  von  $A$  sind Lösungen von:  $\det(A - \lambda * E) = 0$  [mit  $E$  = ist  $n \times n$  Einheitsmatrix]
- zu einem Eigenwert  $\lambda$  gibt es unendlich viele Eigenvektoren  $x$ . Diese sind Lösungen des LGS:  $(A - \lambda * E) * x = 0$

## 14 Vorlesung 14

Eigenwerte (EW) und Eigenvektoren (EV) von Matrizen

### 14.1 EW von Dreiecksmatrizen

EW von Dreiecksmatrix sind die Werte auf der Diagonale

### 14.2 Ähnliche Matrizen

Zwei  $n \times n$  Matrizen  $A$  und  $B$  heissen ähnlich, wenn es eine reguläre  $n \times n$  Matrix  $I$  gibt, sodass:

$$B = I^{-1} * A * I$$

Eigenschaften v. ähnlichen Matrizen

- $A$  und  $B$  haben die gleichen EW mit den gleichen algebraischen und geometrischen Vielfachheiten
- Ist  $x$  ein EV zum EW  $\lambda$  von  $B$  dann ist  $I * x$  EV zum EW  $\lambda$  von  $A$

### 14.3 Diagonalisierbare Matrizen

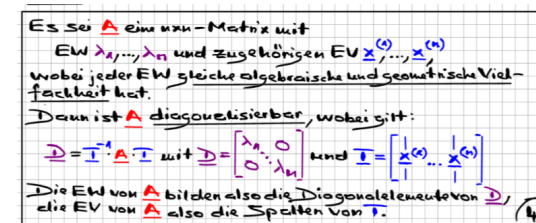


Abbildung 19: Definition Diagonalisierbare Matrizen

### 14.4 Numerische Berechnung aller EW mit QR-Verfahren

Matrix  $A$  mit QR-Zerlegung näherungsweise in eine zu  $A$  ähnliche Dreiecksmatrix  $R$  transformieren  $\Rightarrow$  EW sind näherungsweise gleich den Diagonalelementen

Durchführung

- Start: Setze  $A^{(0)} = A$
- 1. Schritt:
  - Finde für  $A^{(k)}$  die QR-Zerlegung durch, bestimme also orthogonale Matrix  $Q^{(k)}$  mit also  $(Q^{(k)})^T \cdot A^{(k)} = E$ , rechteckige Dreiecksmatrix  $R^{(k)}$ , so dass
  - $A^{(k)} = Q^{(k)} \cdot R^{(k)}$
  - Transformiere  $A^{(k)}$  mit  $Q^{(k)}$  in zu  $A^{(k+1)}$  Stückige Matrix  $A^{(k+1)}$ :  
 $A^{(k+1)} = (Q^{(k)})^{-1} \cdot A^{(k)} \cdot Q^{(k)}$
  - Es gilt:  $A^{(k+1)} = (Q^{(k)})^{-1} \cdot Q^{(k)} \cdot R^{(k)} \cdot Q^{(k)} = R^{(k)} \cdot Q^{(k)}$  Dreiecksmatrix
- 2. Schritt: Näherungsweise Dreiecksmatrix  
 Wiederhole 1. Schritt mit  $A^{(k+1)}$  statt  $A^{(k)}$ , bilde also  $A^{(k+1)} = Q^{(k+1)} \cdot R^{(k+1)}$  und  $A^{(k+2)} = R^{(k+1)} \cdot Q^{(k+1)}$
- usw.

Abbildung 20: Durchführung

### 14.5 Numerische Berechnung der betragsmäßig grössten EW und zugehörigen EV mit Vektoriteration (= Mises Iteration)

- Matrix  $A = \begin{bmatrix} 1 & 3 \\ -1 & 5 \end{bmatrix}$
- Vektoriteration von  $A$  mit  $x^{(0)} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$ :  
 $x^{(1)} = A \cdot x^{(0)} = \begin{bmatrix} 1 & 3 \\ -1 & 5 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 4 \end{bmatrix} = \begin{bmatrix} 14 \\ 10 \end{bmatrix}$ ,  
 $x^{(2)} = A \cdot x^{(1)} = \begin{bmatrix} 1 & 3 \\ -1 & 5 \end{bmatrix} \cdot \begin{bmatrix} 14 \\ 10 \end{bmatrix} = \begin{bmatrix} 28 \\ 20 \end{bmatrix}$ ,  
 $x^{(3)} = \dots = \begin{bmatrix} 56 \\ 40 \end{bmatrix}$ ,  $x^{(4)} = \dots = \begin{bmatrix} 112 \\ 80 \end{bmatrix}$ ,  $x^{(5)} = \dots = \begin{bmatrix} 224 \\ 160 \end{bmatrix}$  Wach
- Beobachtung: Je grösser  $k$ , desto näher liegen die beiden Komponenten von  $x^{(k)}$  beieinander:  
 $x^{(k)} \approx \begin{bmatrix} \alpha_k \\ \alpha_k \end{bmatrix} = \alpha_k \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix}$  für grosse  $k$
- Bedeutung von Vektor  $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$  für  $A$   
 $A \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ -1 & 5 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \end{bmatrix} = 4 \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 \\ 1 \end{bmatrix}$  ist EV von  $A$  zu EW 4
- EW und EV von  $A$

Abbildung 22: Vektoriteration

Sei  $A$  eine reelle  $n \times n$ -Matrix mit  $n$  paar einfachen EW (reelle oder komplex).

Weiter besitzt  $A$  einen reellen EW  $\lambda$ , der betragsmäßig grösster als alle anderen EW ist.

Dann gilt:

- Für jeden Startvektor  $x^{(0)}$  der Länge  $n$  konvergiert die normierte Vektoriteration  
 $x^{(k)} = \frac{A \cdot x^{(k-1)}}{\|A \cdot x^{(k-1)}\|_2}$  für  $k=0,1,2,\dots$   
 gegen einen EV  $x$  der Länge  $n$  von  $A$ .
- Die Folge  $\lambda^{(k)} = (x^{(k)})^T \cdot A \cdot x^{(k)}$  konvergiert gegen den EW  $\lambda$ .

Abbildung 23: Satz Vektoriteration

Es sei  $A$  eine reelle  $n \times n$ -Matrix. Indizes seien  $1 \leq i \leq n$  und für  $k=0,1,2,\dots$

$A^{(k)} = Q^{(k)} \cdot R^{(k)}$  mit  $Q^{(k)}$  orthogonale Matrix,  $R^{(k)}$  rechteckige Dreiecksmatrix.  
 $A^{(k+1)} = R^{(k)} \cdot Q^{(k)}$   
 $R^{(k+1)} = R^{(k)} \cdot Q^{(k)}$

Dann gilt:

- $A^{(k)}$  konvergiert für  $k \rightarrow \infty$  gegen eine reelle Matrix  $A^{(\infty)}$
- Die reellen EW von  $A$  sind die Einzelelemente entlang der Diagonalen von  $A^{(\infty)}$ .
- Die paarweise konjugiert komplexen EW von  $A$  sind die EW der  $2 \times 2$ -Blöcke entlang der Diagonalen von  $A^{(\infty)}$ .
- Wenn  $A$  eine Symmetrische Matrix ist, also  $A^T = A$  ist, dann sind alle EW reell und  $A^{(k)}$  konvergiert gegen eine Matrix  $A^{(\infty)}$  deren Spalten Zugehörige EV von  $A$  der Länge  $n$  sind.

Abbildung 21: QR - Verfahren Satz