

# Numerik Cheat Sheet

## 1 Basics

### 1.1 Sortieren

### 1.2 FFT

## 2 Lineare Gleichungssysteme

### 2.1 Allgemeine Aufgabenstellung

Geg.:  $\mathbf{A} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{b} \in \mathbb{R}^n$

Ges.:  $\mathbf{x} \in \mathbb{R}^n$

$$\mathbf{Ax} = \mathbf{b}$$

### 2.2 Dreiecksmatrizen

Untere Dreiecksmatrix  $\mathbf{L} \in \mathbb{R}^{n \times n}$  und obere Dreiecksmatrix  $\mathbf{R} \in \mathbb{R}^{n \times n}$ .

REGULÄRE/INVERTIERBARE/NICHT-SINGULÄRE MATRIX  
Matrix  $\mathbf{A}$  ist regulär, wenn  $\det \mathbf{A} \neq 0$ . Determinante einer  $\Delta$ Matrix ist das Produkt ihrer Diagonalelemente.  $\mathbf{L}$  und  $\mathbf{R}$  sind regulär, wenn alle Diagonalelemente  $\neq 0$ .

VORWÄRTSEINSETZEN

$$\mathbf{Ly} = \mathbf{b}$$

Rechenaufwand:  $n^2$  AO

Um Speicher zu sparen  $b_i \leftarrow y_i$ .

forward\_subst

```
for j = 1 : n
    x_j ← b_j / l_jj
    for i = j + 1 : n
        b_i ← b_i - l_ij x_j
```

RÜCKWÄRTSEINSETZEN

$$\mathbf{Rx} = \mathbf{y}$$

Rechenaufwand:  $n^2$  AO

Um Speicher zu sparen  $b_i \leftarrow x_i$ .

backward\_subst

```
for j = n : 1
    x_j ← b_j / r_jj
    for i = 1 : j - 1
        b_i ← b_i - r_ij x_j
```

### 2.3 LR-Zerlegung

Sei  $\mathbf{A} \in \mathbb{R}^{n \times n}$  und  $\mathbf{L}, \mathbf{R} \in \mathbb{R}^{n \times n}$

$$\mathbf{A} = \mathbf{LR}$$

Ansatz:

Matrizen  $\mathbf{A}$ ,  $\mathbf{L}$ ,  $\mathbf{R}$  in Teilmatrizen  $\mathbf{A}_{**}$ ,  $\mathbf{A}_{*1}$ ,  $\mathbf{A}_{1*}$ ,  $\mathbf{L}_{**}$ ,  $\mathbf{L}_{*1}$ ,  $\mathbf{R}_{**}$ ,  $\mathbf{R}_{1*}$  zerlegen.

Es folgen 4 Gleichungen aus  $\mathbf{A} = \mathbf{LR}$ :

$$\begin{aligned} a_{11} &= l_{11} r_{11} \\ \mathbf{A}_{*1} &= \mathbf{L}_{*1} r_{11} & \Leftrightarrow \mathbf{L}_{*1} &= \mathbf{A}_{*1} / r_{11} \\ \mathbf{A}_{1*} &= l_{11} \mathbf{R}_{1*} & \Leftrightarrow \mathbf{R}_{1*} &= \mathbf{A}_{1*} \\ \mathbf{A}_{**} &= \mathbf{L}_{*1} \mathbf{R}_{1*} + \mathbf{L}_{**} \mathbf{R}_{**} \end{aligned}$$

Per Def.  $l_{11} = 1$  und damit  $r_{11} = a_{11}$ , sodass

$$\mathbf{A}_{**} - \mathbf{L}_{*1} \mathbf{R}_{1*} = \mathbf{L}_{**} \mathbf{R}_{**} \quad (1)$$

PRAKTISCHE UMSETZUNG

Elemente von  $\mathbf{A}$  überschreiben, sodass:

$$\begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ l_{21} & r_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & r_{n-1,n} \\ l_{n1} & \cdots & l_{n,n-1} & r_{nn} \end{pmatrix} \quad (2)$$

KRITERIUM

Sei  $\mathbf{A}$  regulär,  $\mathbf{A}$  besitzt eine LR-Zerlegung  $\Leftrightarrow$  Alle Hauptuntermatrizen regulär.

MODELLPROBLEM: BANDMATRIX

Irgendwas bzgl Effizienz.

LR-DECOMP

Aufwand: *kubisch*

↓ Aufwand: Nur 1x für jede Matrix betreiben. Sobald LR-Decomp vorliegt nur noch *quadratischer* Aufwand  
↓ Aufwand: Tridiagonalmatrix. Erster Schritt mit 3 AOPs. Restmatrix bleibt tridiagonal. Aufwand  $3n + 6n$  für R- und F-Einsetzen.

lr\_decomp

```
for k = 1 : n
    for i = k + 1 : n
        a_ik ← a_ik / a_kk
    for j = k + 1 : n
        a_ij ← a_ij - a_ik a_kj
```

PROBLEM DER EXISTENZ EINER LR-Z

Falls  $\mathbf{A}$  oder  $\mathbf{A}_{**}$  eine 0 auf der Diagonalen hat, existiert keine LR-Zerlegung. Lösung: Permutiere die Zeilen von  $\mathbf{A}$  so, dass das Ergebnis eine LR-Z besitzt.

PERMUTATIONSMATRIX

Sei  $\mathbf{P} \in \mathbb{R}^{n \times n}$ . Falls in jeder Zeile und Spalte von  $\mathbf{P}$  genau ein Eintrag 1 und alle anderen 0, dann ist  $\mathbf{P}$  eine Permutationsmatrix.  $\mathbf{P}$  ist orthogonal. Ein Produkt zweier Permutationsmatrizen  $\mathbf{PQ}$  ist auch eine Permutationsmatrix.

PERMUTATION

Bijektive Abbildung  $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ .

LR-Z MIT PIVOTSUCHE

$\mathbf{A}$  regulär. Es existiert  $\mathbf{P} \in \mathbb{R}^{n \times n}$  sodass  $\mathbf{PA} = \mathbf{LR}$  gilt.

Pivotisierung: Finde betragsmaximalstes Element in der aktuellen Spalte, welches unter dem aktuellen Diagonalelement von  $\mathbf{A}$  liegt und tausche die aktuelle Zeile mit der Zeile in der das betragsmaximalste Element ist, mit Hilfe von  $\mathbf{P}$ .  $a_{11} \neq 0$ , da betragsgrößtes Element.

LÖSEN EINES GLEICHUNGSSYSTEMS MIT PIVOTSUCHE

$$\mathbf{Ax} = \mathbf{b} \Leftrightarrow \mathbf{PAx} = \mathbf{Pb} \Leftrightarrow \mathbf{LRx} = \mathbf{Pb}.$$

1.)  $\mathbf{Ly} = \mathbf{b}$  2.)  $\mathbf{Rx} = \mathbf{y}$

lr\_pivot

```
for k = 1 : n
    i_* ← k // Finde max. Element
    for i = k + 1 : n
        if |a_ik| > |a_i_*k|: i_* ← i
    p_k ← i_*
    for j = 1 : n // Tausche Zeilen
        γ ← a_kj, a_kj ← a_i_*j, a_i_*j ← γ
    for i = k + 1 : n
        a_ik ← a_ik / a_kk
    for j = k + 1 : n
        a_ij ← a_ij - a_ik a_kj
```

$\mathbf{p}$  protokolliert, welche Vertauschungen durchgeführt wurden, um sie später auf  $\mathbf{b}$  anwenden zu können.

Aufwand:  $\frac{2}{3}n^3$ .

SONDERFALL:  $\mathbf{A}$  POSITIV DEFINIT

TODO.

## 2.4 Fehlerverstärkung

NORM DES MATRIX-VEKTOR-PRODUKTS

Wie stark ändert sich die Länge eines Vektors wenn er mit  $\mathbf{A}$  multipliziert wird. Mapping von Einheitskreis auf Ellipse..

Für  $\mathbf{A} \in \mathbb{R}^{n \times n}$  gilt:

$$\begin{aligned} \alpha_2(\mathbf{A}) &= \min\{\|\mathbf{Ay}\|_2 : \mathbf{y} \in \mathbb{R}^n, \|\mathbf{y}\|_2 = 1\} \\ \beta_2(\mathbf{A}) &= \max\{\|\mathbf{Ay}\|_2 : \mathbf{y} \in \mathbb{R}^n, \|\mathbf{y}\|_2 = 1\} \end{aligned}$$

und

$$\alpha_2(\mathbf{A}) \|\mathbf{z}\|_2 \leq \|\mathbf{Az}\|_2 \leq \beta_2(\mathbf{A}) \|\mathbf{z}\|_2$$

Eigenschaften der Norm:

$$\|\mathbf{x}\| = 0 \Leftrightarrow \mathbf{x} = \mathbf{0}$$

$$\|\lambda \mathbf{x}\| = |\lambda| \|\mathbf{x}\|$$

$$\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$$

TODO

## 2.5 QR-Zerlegung

Für jede Matrix gibt es eine QR-Z.

LR-Z: SCHLECHT KONDITIONIERTES PROBLEM

$\kappa_2(\mathbf{A}) \gg 1$ ,  $\kappa_2(\mathbf{A}) \leq \kappa_2(\mathbf{L})\kappa_2(\mathbf{R})$

Kritisch falls  $\kappa_2(\mathbf{L})\kappa_2(\mathbf{R}) \gg \kappa_2(\mathbf{A})$ .

Ziel: Suche Transformationen  $\mathbf{Q} \in \mathbb{R}^{n \times n}$ , die die Norm unverändert lassen:

$$\|\mathbf{Q}\mathbf{y}\|_2 = \|\mathbf{y}\|_2$$

Mit Hinzunahme des Skalarprodukts:

$$\langle \mathbf{y}, \mathbf{y} \rangle_2 = \|\mathbf{y}\|_2^2 = \|\mathbf{Q}\mathbf{y}\|_2^2 = \langle \mathbf{Q}\mathbf{y}, \mathbf{Q}\mathbf{y} \rangle_2 = \langle \mathbf{y}, \mathbf{Q}^* \mathbf{Q} \mathbf{y} \rangle_2$$

muss  $\mathbf{Q}^* \mathbf{Q} = \mathbf{I}$  gelten.

Gesucht:  $\mathbf{A} = \mathbf{Q}\mathbf{R}$ .

Konditionszahl bzw. Fehlerverstärkung wird nicht

verschlechtert:  $\alpha_2(\mathbf{A}) = \alpha_2(\mathbf{R})$ ,  $\beta_2(\mathbf{A}) = \beta_2(\mathbf{R})$

$\kappa_2(\mathbf{A}) = \kappa_2(\mathbf{R})$ .

GIVENS-ROTATION

Mit Hilfe von Givens-Rotationen können wir beliebige

$\mathbf{A} \in \mathbb{R}^{m \times n}$  auf obere  $\Delta$ gestalt bringen.

$$\mathbf{Q} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \text{ und } \mathbf{Q}\mathbf{y} = \begin{pmatrix} cy_1 + sy_2 \\ 0 \end{pmatrix}$$

Konsekutiv Givens-Rotationen  $\mathbf{Q}_{ij}$   $i$ -te und  $j$ -te Zeile anwenden um Eintrag  $a_{ij}$  zu beseitigen. Bsp.  $\mathbf{A} \in \mathbb{R}^{4 \times 3}$ :

$$\begin{matrix} \mathbf{R} = \mathbf{Q}_{43}\mathbf{Q}_{32}\mathbf{Q}_{42}\mathbf{Q}_{21}\mathbf{Q}_{31}\mathbf{Q}_{41}\mathbf{A} \\ \underbrace{\mathbf{Q}_{41}^*\mathbf{Q}_{31}^*\mathbf{Q}_{21}^*\mathbf{Q}_{42}^*\mathbf{Q}_{32}^*\mathbf{Q}_{43}^*}_{\mathbf{Q}}\mathbf{R} = \mathbf{A} \end{matrix}$$

KOMPAKTE DARSTELLUNG

Verwende Nulleinträge von  $\mathbf{A}$  bzw.  $\mathbf{R}$  um  $\mathbf{Q}_{ij}$  zu beschreiben.

Finde Givens-Rotation:

$$\rho = \begin{cases} s = \rho, c = \sqrt{1-s^2} & \text{falls } |\rho| < 1 \\ c = 1/\rho, s = \sqrt{1-c^2} & \text{falls } |\rho| > 1 \\ c = 1, s = 0 & \text{falls } \rho = 1 \end{cases}$$

Speichern der QR-Z in  $\mathbf{A}$ :

$$\begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ \rho_{21} & r_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & r_{n-1,n} \\ \rho_{n1} & \cdots & \rho_{n,n-1} & r_{nn} \end{pmatrix} \quad (3)$$

Qr Decomp von  $\mathbf{A} \in \mathbb{R}^{m \times n}$

---

```

for  $k = 1 : \min(m, n)$  // Loop über Diagonale
  for  $i = k + 1 : m$  // Loop über Elemente unter Diagonalen
    if  $a_{ik} = 0$ 
       $\rho \leftarrow 1, c \leftarrow 1, s \leftarrow 0$ 
    else if  $|a_{kk}| \geq |a_{ik}|$  // Vgl. mit Diag.element
       $\tau \leftarrow a_{ik}/a_{kk}, \rho \leftarrow \tau/\sqrt{\tau^2 + 1}, s \leftarrow \rho, c \leftarrow \sqrt{1-s^2}$ 
    else // Vgl. mit Diag.element
       $\tau \leftarrow a_{kk}/a_{ik}, \rho \leftarrow \sqrt{\tau^2 + 1}/\tau, c \leftarrow 1/\rho, s \leftarrow \sqrt{1-c^2}$ 

```

```

// Diag.element aktual., Giv.-Rot. in aktueller It. speichern
 $a_{kk} \leftarrow ca_{kk} + sa_{ik}, a_{ik} \leftarrow \rho$ 
for  $j = k + 1 : n$  // Loop über Elemente in der  $k$ -ten Zeile
  // Giv.-Rot auf Zeile anwenden
   $\alpha \leftarrow a_{kj}, a_{kj} \leftarrow c\alpha + sa_{ij}, a_{ij} \leftarrow -s\alpha + ca_{ij}$ 

```

Aufwand:  $6n^2 + 2n^3$  (quadratische Matrix) 3x mehr als LR-Z.

LÖSEN GLEICHUNGSSYSTEM  $\mathbf{A}\mathbf{x} = \mathbf{b}$

$\mathbf{b} = \mathbf{A}\mathbf{x} = \mathbf{Q}\mathbf{R}\mathbf{x} = \mathbf{Q}\mathbf{y} \Leftrightarrow$  1.)  $\mathbf{y} = \mathbf{Q}^*\mathbf{b}$  2.)  $\mathbf{y} = \mathbf{R}\mathbf{x}$ .

1.) Qr-transform: Über einzelne  $G_{ij}$  (oben links angefangen) iterieren und auf  $b$  multiplizieren.

2.) Rückwärtseinsetzen.

EFFIZIENTERE QR-Z

Householder-Spiegelungen: Aufwand 2x mehr als LR-Z.

Mit Optimierungen bei Speicherzugriffen bei QR-Z ähnlich

schnell wie LR-Z.

## 2.6 Ausgleichsprobleme

Wir suchen  $\mathbf{x}$  so, dass alle Gleichungen möglichst gleich gut erfüllt werden.

## 3 Nichtlineare Gleichungssysteme

Wir untersuchen nichtlineare Gleichungssysteme der Form

Gegeben eine stetige Funktion  $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ , finde  $\mathbf{x}^* \in \mathbb{R}^n$  mit  $f(\mathbf{x}^*) = \mathbf{0}$

Transformieren in ein Nullstellenproblem.

### 3.1 Bisektionsverfahren

Einfache Technik, die in jedem Schritt den Fehler mindestens halbiert. Basierend auf dem Zwischenwertsatz für stetige Funktionen.

Funktioniert nur für 1D.

ZWISCHENWERTSATZ

Eine reelle Funktion  $f$ , die in  $[a, b]$  stetig ist, nimmt jeden Wert zwischen  $f(a)$  und  $f(b)$  an. Haben  $f(a)$  und  $f(b)$  verschiedene Vorzeichen, so ist eine Existenz mindestens einer Nullstelle in  $[a, b]$  garantiert.

VERFAHREN IN MATHEMATISCHER NOTATION

$$(a^{(0)}, b^{(0)}) = (a, b)$$

$$x^{(m)} = \frac{a^{(m)} + b^{(m)}}{2}$$

$$(a^{(m+1)}, b^{(m+1)}) = \begin{cases} (a^{(m)}, x^{(m)}) & \text{if } f(a^{(m)})f(x^{(m)}) < 0 \\ (x^{(m)}, b^{(m)}) & \text{sonst.} \end{cases}$$

Bisection

$f_a \leftarrow f(a), f_b \leftarrow f(b)$

while  $b - a > \epsilon$

$x \leftarrow (a + b)/2$

$f_x \leftarrow f(x)$

```

if  $f_a f_x < 0$ 
   $b \leftarrow x, f_b \leftarrow f_x$ 
else
   $a \leftarrow x, f_a \leftarrow f_x$ 

```

Aufwand:  $m + 2$  Auswertungen von  $f$  und  $2m$

Rechenoperationen, mit  $m = \lceil \log_2((b - a)/\epsilon) \rceil$  Schritten. Hohe Stabilität und jedes konstruierte Intervall muss eine Nullstelle enthalten. Nur auf reellwertige Funktionen auf geeigneten Intervallen anwendbar.

## 3.2 Allgemeine Fixpunktiterationen

ITERATION

$U \subseteq \mathbb{R}^n$  eine abgeschlossene Teilmenge und  $\Phi : U \rightarrow U$  eine (Selbst-)Abbildung. Dann ist  $\Phi$  eine Iteration auf  $U$ . Folge der Iterierten  $\mathbf{x}^{(m+1)} = \Phi(\mathbf{x}^{(m)})$ . Kontruiere Iteration so, dass  $\Phi$  gegen gesuchte Lösung  $\mathbf{x}^*$  konvergiert. Es soll  $\phi(\mathbf{x}^*) = \mathbf{x}^*$  gelten. Die Lösung muss ein Fixpunkt von  $\Phi$  sein.

MITTELWERTSATZ DER DIFFERENTIALRECHNUNG

Zwischen  $a$  und  $b$  von  $f$  gibt es mindestens einen

Kurvenpunkt, für den die Tangente an  $\eta$  parallel zur Sekante durch  $a$  und  $b$  ist:  $(b - a)f'(\eta) = f(b) - f(a)$ .

TODO: WÄHLEN DER RICHTIGEN ITERATION

FIXPUNKTSATZ VON BANACH

Sei  $\Phi$  eine Iteration auf einer **abgeschlossenen** Menge

$U \subseteq \mathbb{R}^n$  (**Selbstabbildung**). Sei  $L \in [0, 1)$  so gegeben, dass:  $\|\Phi(\mathbf{x}) - \Phi(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$  für alle  $\mathbf{x}, \mathbf{y} \in U$ . **Kontraktion**.  $\Phi$  besitzt **genau** einen Fixpunkt und die Folge der Iterierten konvergiert für jeden Startwert  $\mathbf{x}^{(0)} \in U$  gegen diesen Fixpunkt.

Fehlerabschätzung *a-priori* (Vorhersagen wieviele Schritte) und *a-posteriori* (Prüfen ob Näherung schon genau genug):

$$\|\mathbf{x}^{(m)} - \mathbf{x}^*\| \leq \frac{L^m}{1 - L} \|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|$$

$$\|\mathbf{x}^{(m)} - \mathbf{x}^*\| \leq \frac{1}{1 - L} \|\mathbf{x}^{(m)} - \mathbf{x}^{(m+1)}\| \quad \forall m \in \mathbb{N}_0$$

## 3.3 1D-Newton-Verfahren

$U \subseteq \mathbb{R}^n$  offene Menge und  $f : U \rightarrow \mathbb{R}^n$  zweimal stetig differenzierbar mit Nullstelle  $v\mathbf{x}^* \in U$ , so dass  $f(\mathbf{x}^*) = \mathbf{0}$ . Ziel: Konstruiere Iteration  $\Phi$ , die  $\mathbf{x}^*$  als Nullstelle besitzt.

TAYLOR

TODO

$$0 = f(x^*) = f(x) + f'(x)(x^* - x) + \frac{f''(\eta)}{2}(x^* - x)^2$$

EINDIMENSIONALES NEWTON-VERFAHREN

$$\Phi : U \rightarrow \mathbb{R} \quad x \rightarrow x - \frac{f(x)}{f'(x)}$$

Indem der dritte Term der Taylorreihe *wegfällt*, approximieren wir die Funktion  $f$  durch ihre Tangente im

Punkt  $x$ . Die Nullstelle der Tangente ist die nächste Iterierte  $\Phi(x)$ .

KONVERGENZ

Sei  $r \in \mathbb{R}_{>0}$  und  $U = (x^* - r, x^* + r)$  und gelte  $f \in C^2(U)$ ,  $|1/f'(x)| \leq C_1 \forall x \in U$ ,  $|f''(x)| \leq C_2 \forall x \in U$  und  $r \leq \frac{2}{C_1 C_2}$ , dann ist die Abbildung  $\Phi$  für das Newton Verfahren eine Selbstabbildung auf  $U$ , sodass gilt:

$$|\Phi(x) - x^*| \leq \frac{C_1 C_2}{2} |x - x^*|^2 \quad \forall x \in U$$

Newton-Verfahren konvergiert, falls  $x^{(0)}$  in  $U$  liegt.

Konvergenz ist umso schneller, je näher die Iterierten an der Lösung liegen. **Quadratische Konvergenz.**

### 3.4 ND-Newton-Verfahren

HAUPTSATZ DER INTEGRAL- UND DIFFERENTIALRECHNUNG

$$f(b) - f(a) = \int_a^b f'(t) dt$$

NEWTON-VERFAHREN

Sei  $U \subseteq \mathbb{R}^n$  und  $Df(\mathbf{x})$  für alle  $\mathbf{x} \in U$  regulär (also invertierbar). Es gilt:

$$\Phi : U \rightarrow \mathbb{R}^d, \quad \mathbf{x} \mapsto \mathbf{x} - Df(\mathbf{x})^{-1} f(\mathbf{x})$$

KONVERGENZ

Sei  $r \in \mathbb{R}_{>0}$  und  $U = K(\mathbf{x}^*, r)$  und gelte

$f \in C^1(U, \mathbb{R}^b)$ ,

$\|Df(\mathbf{x})^{-1}\|_2 \leq C_1 \forall \mathbf{x} \in U$ ,

$\|Df(\mathbf{x}) - Df(\mathbf{y})\|_2 \leq C_2 \|\mathbf{x} - \mathbf{y}\|_2 \forall \mathbf{x}, \mathbf{y} \in U$  und  $r \leq \frac{2}{C_1 C_2}$ .

Dann ist  $\Phi$  eine Selbstabbildung auf der Kugel  $U$  und es gilt

$$\|\Phi(\mathbf{x}) - \mathbf{x}^*\|_2 \leq \frac{C_1 C_2}{2} \|\mathbf{x} - \mathbf{x}^*\|_2^2 \quad \forall \mathbf{x} \in U$$

Quadratische Konvergenz unter schwächeren Voraussetzungen, denn  $f'$  muss Lipschitz-stetig sein.

UMSETZUNG

Anstatt Inverse Jacobimatrix zu berechnen, lineares Gleichungssystem nach  $\mathbf{d}$  lösen (erhöhte numerische Stabilität):

$$\mathbf{x}^{(m+1)} = \mathbf{x}^{(m)} + \mathbf{d}^{(m)} \quad \mathbf{d}^{(m)} = -Df(\mathbf{x}^{(m)})^{-1} f(\mathbf{x}^{(m)})$$

$$Df(\mathbf{x}^{(m)}) \mathbf{d}^{(m)} = -f(\mathbf{x}^{(m)})$$

GEDÄMPFTES NEWTON-VERFAHREN

Um Divergenz zu vermeiden Newton-Richtung mit Dämpfungsparameter  $\sigma^{(m)}$  multiplizieren. Sorgt dafür, dass der Fehler nicht größer als im vorangehenden Schritt werden kann.

$$\mathbf{x}^{(m+1)} = \mathbf{x}^{(m)} + \sigma^{(m)} \mathbf{d}^{(m)}$$

## 4 Eigenwertprobleme

Im Allgemeinen werden Eigenwertprobleme in der Form  $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ ,  $\mathbf{x} \neq 0$  untersucht.

### 4.1 Vektoriteration

Eignet sich für die Berechnung des größten Eigenwerts.

Eigenwertproblem in die Form eines linearen

Gleichungssystems bringen:

$$(\mathbf{A} - \lambda \mathbf{I})\mathbf{x} = \mathbf{0}$$

**Annahmen:**  $\mathbf{A} = \mathbf{A}^*$ . Mit Hauptachsentransformation auf Diagonalgestalt bringen, sodass eine orthogonale Matrix  $\mathbf{U} \in \mathbb{R}^n$  existiert, wo  $\mathbf{U}^* \mathbf{A} \mathbf{U} = \mathbf{D}$  gilt, sodass die Eigenwerte  $\lambda_n$  in absteigendem Betrag auf der Diagonalen von  $\mathbf{D}$  liegen.

**Motivation:** Wenn ein Vektor  $\hat{\mathbf{x}}^{(0)} \in \mathbb{R}^n$  mit der  $m$ -ten Potenz der Matrix  $\mathbf{D}$  multipliziert wird, erhält man neue Vektoren

$$\hat{\mathbf{x}}^{(m)} = \mathbf{D}^m \hat{\mathbf{x}}^{(0)} = \begin{pmatrix} \lambda_1^m \hat{x}_1^{(0)} \\ \vdots \\ \lambda_n^m \hat{x}_n^{(0)} \end{pmatrix}$$

sodass sich für große Werte von  $m$  (aufgrund der absteigenden Sortierung) die ersten Komponenten gegenüber dem Rest durchsetzen.

DOMINANTER EIGENWERT

Gilt  $|\lambda_1| > |\lambda_2| \leq \dots \leq |\lambda_n|$  so werden  $\hat{\mathbf{x}}^{(m)}$  gegen ein Vielfaches von  $\hat{\mathbf{e}}^{(1)} = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}^*$  konvergieren. Der erste Einheitsvektor ist ein Eigenvektor zu  $\lambda_1$ , sodass wir Konvergenz gegen einen Eigenvektor erhalten.

KONVERGENZ

$\mathbf{e}^{(1)} = \mathbf{U} \hat{\mathbf{e}}^{(1)}$  ist Eigenvektor zu Eigenwert  $\lambda_1$ . Es gilt  $\tan(\mathbf{x}^{(0)}, \mathbf{e}^{(1)}) < \infty$ , also  $\mathbf{x}^{(0)}$  soll nicht senkrecht auf  $\mathbf{e}^{(1)}$  stehen. Dann gilt

$$\tan(\mathbf{x}^{(m)}, \mathbf{e}^{(1)}) \leq \left( \frac{|\lambda_2|}{|\lambda_1|} \right)^m \tan(\mathbf{x}^{(0)}, \mathbf{e}^{(1)}) \quad \forall m \in \mathbb{N}_0$$

Sind  $|\lambda_2|$  und  $|\lambda_1|$  ähnlich groß, gibt es langsame Konvergenz.

DIAGONALISIERBARE MATRIZEN

**TODO**

**Numerische Probleme:** Iterationsfolge führt zu Vektoren mit sehr großen ( $|\lambda_1| > 1$ ) oder sehr kleinen Einträgen ( $|\lambda_1| < 1$ ).

**Lösung:** Normalisieren mit der Norm (also nur Skalierung):

$$\mathbf{y}^{(m)} = \mathbf{A} \mathbf{x}^{(m-1)} \quad \gamma^{(m)} = \|\mathbf{y}^{(m)}\|_2 \quad \mathbf{x}^{(m)} = \mathbf{y}^{(m)} / \gamma^{(m)} \quad \forall m \in \mathbb{N}$$

RAYLEIGH-QUOTIENT

Rayleigh-Quotient zu  $\mathbf{A}$  ist gegeben durch

$$\Lambda_A : \mathbb{R}^n \setminus \{\mathbf{0}\} \rightarrow \mathbb{R} \quad \mathbf{x} \mapsto \frac{\langle \mathbf{A} \mathbf{x}, \mathbf{x} \rangle_2}{\langle \mathbf{x}, \mathbf{x} \rangle_2}$$

Falls  $\mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$  ein Eigenvektor von  $\mathbf{A}$  zu  $\lambda \in \mathbb{R}$  ist gilt  $\Lambda_A = \lambda$ . Mit der Cauchy-Schwarz-Ungleichung

$|\langle \mathbf{x}, \mathbf{y} \rangle_2| \leq \|\mathbf{x}\|_2 \|\mathbf{y}\|_2 \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$  lässt sich die Genauigkeit der Näherung des Eigenwerts abschätzen über:

$$|\Lambda_A(\mathbf{x}) - \lambda| \leq \|\mathbf{A} - \lambda \mathbf{I}\|_2 \sin(\mathbf{x}, \mathbf{e}) \leq \|\mathbf{A} - \lambda \mathbf{I}\|_2 \tan(\mathbf{x}, \mathbf{e})$$

mit  $\mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$  als Näherung des Eigenvektors  $\mathbf{e} \in \mathbb{R}^n$  von  $\mathbf{A}$  zu  $\lambda \in \mathbb{R}$ .

**Quadratische Konvergenz:** Falls  $\mathbf{A} = \mathbf{A}^*$ , ergibt sich die Abschätzung

$$|\Lambda_A(\mathbf{x}) - \lambda| \leq \|\mathbf{A} - \lambda \mathbf{I}\|_2 \sin^2(\mathbf{x}, \mathbf{e})$$

Näherung des Eigenwerts kann wesentlich schneller als die des Eigenvektors konvergieren.

power\_adaptive

$\gamma \leftarrow \|\mathbf{x}\|_2$ ,  $\mathbf{x} \leftarrow \mathbf{x} / \gamma$

$\mathbf{y} \leftarrow \mathbf{A} \mathbf{x}$

$\lambda \leftarrow \langle \mathbf{y}, \mathbf{x} \rangle_2$

while  $\|\lambda \mathbf{x} - \mathbf{y}\|_2 > \epsilon \|\mathbf{y}\|_2$

$\gamma \leftarrow \|\mathbf{y}\|_2$ ,  $\mathbf{x} \leftarrow \mathbf{y} / \gamma$

$\mathbf{y} \leftarrow \mathbf{A} \mathbf{x}$

$\lambda \leftarrow \langle \mathbf{y}, \mathbf{x} \rangle_2$

$\mathbf{y}$  wird für die Berechnung von  $\Lambda_A$ , die Prüfung auf Konvergenz und für die Bestimmung der nächsten Iterierten genutzt.

### 4.2 Inverse Iteration

Eignet sich für die Berechnung des kleinsten Eigenwerts (geben die niedrigsten Frequenzen für Resonanzeffekte an).

### 4.3 Orthogonale Iteration

### 4.4 QR-Iteration

### 4.5 Praktische QR-Iteration

## 5 Approximation von Funktionen

### 5.1 Polynominterpolation

### 5.2 Neville-Aitken-Verfahren

### 5.3 Newtons dividierte Differenzen

### 5.4 Approximation von Funktionen

## 6 Numerische Integration

### 6.1 Quadraturformeln

### 6.2 Fehleranalyse

