8. Eigenwertprobleme

a) spezielles Eigenwertproblem: Berechne die Eigenwerte (EW) und Eigenvektoren (EV) einer reellen, quadratischen Matrix A:

$$A\vec{x} = \lambda \vec{x}$$

b) allgemeines EWP:

$$A\vec{x} = \lambda B \vec{x}$$

8.1 Transformationsmethoden

Strategie: Finde eine Ähnlichkeitstransformation

$$C = T^{-1} A T$$

die einfach zu berechnen ist und so, dass die EW von C einfacher zu berechnen sind

→ Transformation auf obere Hessenbergform:

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} & \cdots & h_{1n} \\ h_{21} & h_{22} & h_{23} & \cdots & h_{2n} \\ 0 & h_{32} & h_{33} & \cdots & h_{3n} \\ & 0 & \ddots & \ddots & \vdots \\ & 0 & & h_{n,n-1} & h_{nn} \end{pmatrix}$$

also $h_{ij} = 0 \quad \forall\, i > j+1$. Benutze die Givens-Transformation.

8.2 QR-Algorithmus

Gesucht ist ein Verfahren, um die EW und EV einer Hessenbergmatrix H zu bestimmen. Ein solches Verfahren ist durch den QR-Algorithmus gegeben.

Def. QR-Transformation

$$A = QR \to A' = RQ$$

Dann ist $A' = RQ = Q^{-1} A Q$ orthogonal ähnlich zu A. Außerdem ist A' ebenfalls eine Hessenbergmatrix, falls A eine ist.

QR-Verfahren für reelle EW

a) $A \to H = Q^T A Q$

b) $H_1 = H$, $H_k = Q_k R_k$, $H_{k+1} = R_k Q_k$, $k = 1, 2, \dots$

Die Folge $\lim_{k \to \infty} H_k$ konvergiert (unter bestimmten Voraussetzungen) gegen eine Rechtsdreiecksmatrix mit

$$\lim_{k \to \infty} h_{ii}^{(k)} = \lambda_i \qquad i = 1, 2, \dots, n$$

Die Konvergenz kann sehr langsam sein, besser verhält sich das Verfahren mit Spektralverschiebung.

b) $H_k - \sigma \mathbb{1} = Q_k R_k$, $H_{k+1} = R_k Q_k + \sigma \mathbb{1}$, $k = 1, 2, \dots$

8.3 QR-Algorithmus für tridiagonale Matrizen

In der Physik trifft man häufig auf symmetrische (hermitische) Matrizen. Die Givens-Transformation führt dann auf tridiagonalform:

$$J = \begin{pmatrix} \alpha_1 & \beta_1 & & & 0 \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_{n-1} \\ 0 & & & \beta_{n-1} & \alpha_n \end{pmatrix}$$

Die β_i seien ungleich Null, da das Problem ansonsten in Unterprobleme zerfällt, die unabhängig voneinander gelößt werden können. Der Algorithmus ist formal identisch zum Vorausgegangenen:

$$J_k - \sigma \mathbb{1} = Q_k R_k \quad , \; J_{k+1} = R_k Q_k + \sigma_k \mathbb{1}$$
$$= Q_k^T (J_k - \sigma_k \mathbb{1}) Q_k + \sigma_k \mathbb{1}$$
$$= Q_k^T J_k Q_k$$

Warum hilft die Spektralverschiebung?

Für hinreichend großes k gilt für das Konvergenzverhalten des QR-Algorithmus:

$$|\beta_i^{(k)}| \approx \left| \frac{\lambda_{i+1}}{\lambda_i} \right|^k \quad , \; i = 1, 2, \dots, n$$

- wir nehmen hier an, dass $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$
- β_i konvergiert gegen 0
- $|\lambda_{i+1}/\lambda_i|$ kann beliebig nah an 1 liegen
  $\Rightarrow$ langsame Konvergenz

Für $\tilde{J} = J - \sigma \mathbb{1}$ gilt allerdings

$$|\tilde{\beta}_i^{(k)}| \approx \left| \frac{\lambda_{i+1} - \sigma}{\lambda_i - \sigma} \right|^k \quad , i = 1, 2, \ldots, n-1$$

Falls $\sigma \approx \lambda_n$, dann gilt $|\lambda_n - \sigma| \ll |\lambda_i - \sigma|$, $i = 1, 2, \ldots, n-1$ und damit konvergiert $|\tilde{\beta}_n^{(k)}|$ sehr schnell gegen 0. Eine Wahl für σ_k ist z.B. $\sigma_k = \bar{j}_{nn}^{(k)}$, $k = 1, 2, \ldots$
Eine bessere Wahl für σ_k folgt aus dem EW der Untermatrix

$$C_k = \begin{pmatrix} \alpha_{n-1}^{(k)} & \beta_{n-1}^{(k)} \\ \beta_{n-1}^{(k)} & \alpha_n^{(k)} \end{pmatrix}$$

Man wählt σ_k gleich dem EW, der näher an $\alpha_n^{(k)}$ liegt.

<u>Vorgehen:</u> Wähle σ_k $(= j_{nn}^{(k)})$ und führe so viele QR-Schritte aus, bis $\beta_{n-1}^{(k)} \approx 0$. Dann ist $\bar{j}_{nn}^{(k)}$ der gesuchte EW. Verkleinere die Matrix um 1 und starte von Neuem.
Speichere α_i und β_i in zwei Vektoren $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)^T$, $\beta = (\beta_1, \beta_2, \ldots, \beta_n)^T$. Mit $c = \cos\varphi$ und $s = \sin\varphi$ transformieren jeweils zwei aufeinanderfolgende Zeilen und Spalten wie folgt:

$$\alpha_i' = \alpha_i - 2\beta_i cs - (\alpha_i - \alpha_{i+1})s^2 = \alpha_i - z$$

$$\alpha_{i+1}' = \alpha_{i+1} + 2\beta_i cs + (\alpha_i - \alpha_{i+1})s^2 = \alpha_{i+1} + z$$

$$\beta_i = (\alpha_i - \alpha_{i+1})cs + \beta_i(c^2 - s^2)$$

__Algorithmus:__ QR-Schritt mit Spektralverschiebung

```
Input: α, β, σ, τ (Maschienengenauigkeit)
x = α_1 - σ
y = β_1
for i = 1, ..., n-1
    if |x| ≤ τ |y|
        ω = -y
        c = 0
        s = 1
    else
        ω = √(x² + y²)
        c = x/ω
        s = -y/ω
    d = (α_i - α_{i+1})
    z = s (2c β_i + d s)
    α_i = α_i - z
    α_{i+1} = α_{i+1} + z
    β_i = d c s + (c² - s²) β_i
    x = β
    if i > 1 :
        β_{i-1} = ω
    if i < n-1 :
        y = -s β_{i+1}
        β_{i+1} = c β_{i+1}
```

8.4 Inverse Vektoriteration

Sei A eine reelle reguläre Matrix mit EW λ_i, $|\lambda_n| > |\lambda_{n-1}| > \ldots > |\lambda_1|$ und entsprechenden EV $\vec{u}_i$. Man kann dann jeden (beliebigen) Vektor $\vec{x}$ schreiben als:

$$\vec{x} = \sum_{i=1}^{n} c_i \vec{u}_i$$

Sei nun $\sigma \approx \lambda_j$, dann gilt:

$$\omega^{(k)} = (A - \sigma \mathbb{1})^{-k}, \quad \vec{x} = \frac{c_j}{(\lambda_j - \sigma)^k} \vec{u}_j + \sum_{i \neq j} \frac{c_i}{(\lambda_i - \sigma)^k} \vec{u}_i$$

Ist $|\lambda_j - \sigma| \ll |\lambda_i - \sigma| \ \forall i \neq j$, so hat $\omega^{(k)}$ für $k \to \infty$ einen dominanten Beitrag in Richtung des EV $\vec{u}_j$, falls $c_j \neq 0$.

<u>Algorithmus:</u> Inverse Vektoriteration mit Shift

Input: x, σ

$\omega = x / \|x\|$

for $k = 1, 2, \ldots$

  $x = (A - \sigma \mathbb{1})^{-1} \omega$ | löse $(A - \sigma \mathbb{1}) x = \omega$

  $\omega = x / \|x\|$

  $\lambda_k = (\omega, A\omega)$

output: $\lambda = \lambda_k$, $u = x$

Der Shift σ kann in jedem Iterationsschritt neu angepasst werden, um die Konvergenz zu optimieren.

$$\frac{(w, Aw)}{(w,w)} = \sigma$$

nennt man Reyleight-Quotienten und es gilt $\sigma = \lambda$, falls w ein EV von A ist.

<u>Konvergenzverhalten:</u>

$$w^{(k)} = \frac{1}{(\lambda_j - \sigma)^k} \left[c_j \vec{u}_j + \sum_{i \neq j} \frac{(\lambda_j - \sigma)^k}{(\lambda_i - \sigma)^k} \vec{u}_i \right]$$

D.h. das Verfahren konvergiert linear proportional zu

$$\frac{\lambda_j - \sigma}{\gamma} \quad \text{mit} \quad \gamma = \min_{i \neq j} |\lambda_i - \sigma|$$