

Programmieraufgabe III (5+5+5+5 = 20 Punkte)

Abgabe in der Woche 3.-7. Dezember

```
In [16]: import numpy as np
import matplotlib.pyplot as plt
```

Teilaufgabe a)

Implementieren Sie die **einfache Vektoriteration nach von Mises**. Input soll sein

- eine Funktion $Afun$, die die Anwendung eines linearen Operators beschreibt,
- ein Startvektor z_0 (nicht notwendigerweise normiert),
- eine maximale Anzahl von Iterationen $itmax$ sowie eine Toleranz $epsilon$, derart dass die Iteration stoppt, wenn zwei aufeinanderfolgende Eigenvektorapproximationen sich nur um $Norm < epsilon$ unterscheiden.

Output sollen Näherungen des gesuchten Eigenwerts und des zugehörigen Eigenvektors sein. Zusätzlich sollen jeweils Listen mit allen Iterierten (für Eigenwert und Eigenvektor) ausgegeben werden. Falls die Toleranz nicht erreicht wird, soll eine Fehlermeldung ausgegeben werden.

```
In [380]: def vectoriteration(Afun, z0, itmax, epsilon):
# to be filled in
return eigval, z, eigvlist, zlist
```

Testen Sie Ihre Implementierung an der 4×4 -Matrix $A = \begin{pmatrix} 1 & & & \\ & 2 & & \\ & & 2 & \\ & & & 2 \end{pmatrix}$. Welche Konvergenzgeschwindigkeit für

die Eigenwertnäherungen können Sie beobachten?

Wie ändert sich das Konvergenzverhalten, wenn Sie eine Nebendiagonale mit den Einträgen 0, 1, 1 hinzufügen, d.h. für

$$A = \begin{pmatrix} 1 & 0 & & \\ & 2 & 1 & \\ & & 2 & 1 \\ & & & 2 \end{pmatrix}?$$

In []:

Teilaufgabe b)

Sei B diejenige $n \times n$ -Matrix mit Einträgen b auf der Hauptdiagonalen und Einträgen -1 auf den beiden Nebendiagonalen. Testen Sie Ihre Implementierung aus Aufgabe a), indem Sie den betragsmäßig größten Eigenwert von B für verschiedene $b > 2$ bzw. $n \in \mathbb{N}$ berechnen.

Welche Konvergenzgeschwindigkeit hinsichtlich der Eigenwerte können Sie beobachten? Interpretieren Sie Ihre Ergebnisse.

In []:

Teilaufgabe c)

Implementieren Sie die **inverse Iteration** zur Bestimmung des betragsmäßig kleinsten Eigenwerts einer invertierbaren Matrix. Input darf hier also eine Matrix als numpy-array sein; die restlichen Inputs sollen wie in a) gewählt werden.

Wenden Sie Ihre Funktion auf die Matrix B aus der vorhergehenden Aufgabe an und plotten Sie erneut die Konvergenzgeschwindigkeiten für verschiedene b und n .

```
In [385]: def inverseiteration(A, z0, itmax, epsilon):
          # to be filled in
          return
```

Teilaufgabe d)

Gegeben Sei nun die Diagonalmatrix $D_m \in \mathbb{R}^{10 \times 10}$ mit den ersten m Diagonaleinträgen 10 und den letzten $(10 - m)$ Diagonaleinträgen 2. Ferner sei die Matrix $C = C(c) \in \mathbb{R}^{10 \times 10}$ für $c \in (-1, 1)$ definiert durch $C_{10,j} = C_{j,10} = c$ für $j = 1, \dots, 9$; alle anderen Einträge sind Null.

Bestimmen Sie jeweils einen Eigenwert $\lambda(c)$ der gestörten Matrix $D_m + C(c)$, der möglichst nahe am Eigenwert 2 der ursprünglichen Matrix D_m liegt und plotten Sie den Verlauf von $\lambda(c)$ für variierendes $c \in (-0.01, 0.01)$.

Implementieren und verwenden Sie dazu die **inverse/gebrochene Iteration nach Wielandt** mit entsprechend sinnvollen Parametern.

Deuten Sie das im Plot zu beobachtende Verhalten von $\lambda(c)$ für variierendes m .

```
In [387]: def inverseiterationwielandt(A, z0, mu, itmax, epsilon):
          # to be filled in
          return
```