

2. Programmierprojekt zu Algorithmen der Numerischen Mathematik

(Eigenwerte)

- a) Programmieren Sie die direkte Potenzmethode. Plotten Sie den Fehler für die folgendermaßen definierten Testmatrizen:

```
n = length(d);  
S = triu(diag(n:-1:1,0) + ones(n,n));  
A = S*diag(d,0)*inv(S);
```

und

```
n = length(d);  
z = diag(sqrt(1:n),0) + ones(n,n);  
[Q R] = qr(z);  
B = Q*diag(d,0)*Q';
```

mit

1. $d = (1:10)'$;
2. $d = [\text{ones}(9,1); 2]$;
3. $d = 1 - 2 \cdot (-(1:0.5:5))$;

- b) Programmieren Sie die inverse Potenzmethode. Plotten Sie den Fehler für die Testmatrizen aus a).
- c) Programmieren Sie den Algorithmus aus Aufgabe 17 b) (s. Blatt 5).
- d) Verwenden Sie Ihren Code aus Teil c) (oder entsprechende Matlab-Funktionen) um den QR-Algorithmus mit Shift für Tridiagonalmatrizen zu implementieren. Verwenden Sie dazu einmal den Shift wie in der Vorlesung und einmal den sogenannten *Wilkinson-Shift*:

$$\mu = h_{n,n} + d - \text{sign}(d) \sqrt{d^2 + h_{n-1,n}^2}$$

mit $d = (h_{n-1,n-1} - h_{n,n})/2$ und $\text{sign}(0) \in \{1, -1\}$ (μ ist der Eigenwert der unteren rechten 2×2 -Untermatrix von H , der näher an $h_{n,n}$ liegt).

Testen Sie Ihr Programm an $A = \text{tridiag}(1, -2, 1) \in \mathbb{R}^{n \times n}$ mit $n = 20$. Stellen Sie die Konvergenz des ersten berechneten Eigenwertes dar, indem Sie den exakten Fehler in jedem QR-Schritt berechnen und diesen graphisch gegen die Iterationszahl auftragen.

Hinweis: Die exakten Eigenwerte von A sind $\lambda_j = 2 \cos \frac{j\pi}{n+1} - 2$, $j = 1, \dots, n$.

- e) (BONUS) Sei eine $m \times n$ Matrix A und deren Singulärwertzerlegung $A = U\Sigma V^T$ mit $\Sigma = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix}$, Σ_r nicht singulär, gegeben.

Dann ist die Lösung des linearen Ausgleichsproblems

$$\|Ax - b\|_2 = \min, \quad \|x\|_2 = \min \quad (1)$$

(Ohne Beweis) gegeben durch

$$x = A^+b, \quad A^+ = V\Sigma^+U^T, \quad \Sigma^+ = \begin{bmatrix} \Sigma_r^{-1} & 0 \\ 0 & 0 \end{bmatrix}.$$

Schreiben Sie ein Matlab-Programm, welches das Problem (1) auf die genannte Weise löst.

Hinweis: Wenn sie in der Matlab-Hilfe nach 'svd' (wie 'singular value decomposition') suchen, brauchen Sie die Singulärwertzerlegung nicht selbst zu programmieren.

Besprechung der Programmieraufgabe in den Übungen am 12.06.2013.
Jede Teilaufgabe ist ein Kreuz wert.