Technische Universität Berlin **Wintersemester 2018/2019**
Fakultät II – Mathematik und Naturwissenschaften
Institut für Mathematik
Reinhold Schneider, Benjamin Kutschan

Numerische Mathematik I

6. Übungsblatt: Rotationen, SVD

Hausaufgaben: (Abgabe vor der Vorlesung am 5.12.2018)

Aufgabe 1: **(3 Punkte)**
Beschreiben Sie den Algorithmus zur Überführung einer quadratischen Matrix in eine obere Bidiagonalform mittels Householder-Spiegelungen.

Aufgabe 2: **(3 Punkte)**
Sei $A \in \mathbb{R}^{n\times n}$ mit Spalten a_i, $i = 1, \dots, n$. Zeige mit Hilfe der QR-Zerlegung die Abschätzung

$$|\det(A)| \leq \prod_{i=1}^{n} \|a_i\|_2.$$

Bestimmen Sie weiterhin die Menge $S \subset \mathbb{R}^{n\times n}$, sodass $A \in S$ genau dann, wenn die obere Abschätzung mit Gleichheit erfüllt ist. Beweisen Sie diese Aussage für Ihre Menge S.

Aufgabe 3: **(2 Punkte)**
Zeigen Sie, dass falls $A \in \mathbb{R}^{n\times n}$, dann

$$\|A\|_{\mathrm{F}} \leq \sqrt{\operatorname{rank}(A)}\|A\|_2.$$

Aufgabe 4: **(2 Punkte)**
Zeige, dass für die von der euklidischen 2-Norm induzierte Operatornorm und eine invertierbare Matrix A

$$\kappa_2(A^T A) = (\kappa_2(A))^2$$

gilt (Tip: Matrixzerlegung). Zeige außerdem, dass für beliebige Normen

$$k\,(\kappa(A))^2 \leq \kappa(A^T A) \leq K\,(\kappa(A))^2$$

für nur von der Norm abhängige Konstanten k und K gilt.

Programmieraufgabe 1: (Abgabe in den Rechnersprechstunden bis zum 21. Dezember 2018, **Terminbestellung bis 7. Dezember**)

Es soll die Singulärwertzerlegung einer $n \times n$-Matrix berechnet werden. Dies ist eine Zerlegung

$$A = USV^T,$$

wobei U und V orthogonale Matrizen sind und S diagonal. Dazu ist die Matrix zunächst mittels Givens-Rotationen in eine obere Bidiagonalform zu bringen (nur die Hauptdiagonale und die erste obere Nebendiagonale sind besetzt). Dazu müssen die Rotationen von links und rechts angewendet werden.

Danach soll mehrmals/iterativ das folgende Verfahren angewendet werden: Überführe die Matrix mittels Givens-Rotationen von rechts in *untere* Bidiagonalform. Danach mittels Givens-Rotationen von links wieder in obere Bidiagonalform. Mit den Rotationen soll immer oben links in der Matrix begonnen werden.

Die Iterationen können dann beendet werden, wenn die Summe der Quadrate der Nichtdiagonalelemente der Matrix auf $n \cdot tol^2$ gefallen ist, wobei n die Größe der Matrix ist und z.Bsp. `tol=0.0001` `norm(A-diag(diag(A)),'fro')**2 < n*tol**2`.

Die SVD soll mit

```
[U,S,V] = mysvd(A)
```

aufgerufen werden können.

Die Orthogonalität von U kann zum Beispiel durch `norm(U*U'-eye(100))` überprüft werden. Testen Sie ihren Algorithmus mit zufälligen Matrizen `A=rand(n)`. Wie kann die Korrektheit der Zerlegung verifiziert werden? Ob und warum der Algorithmus konvergiert, muss diesmal nicht bewiesen werden.

Hinweise:
Für die Implementierung ist es sinnvoll, die Funktion

```
A = givensLeft(i,j,k,v,A)
```

zu implemtieren, die den Eintrag $A_{j,k}$ eliminiert, indem die Zeilen i und j rotiert werden.

Analog kann die Funktion

```
A = givensRight(i,j,k,v,A)
```

den Eintrag $A_{k,j}$ eliminieren, indem die *Spalten* i und j rotiert werden.

Eine Funktion

```
A = bidiagonal(A)
```

soll die Matrix A bidiagonalisieren.

Und eine Funktion

```
A = iteration(A)
```

soll eine der oben beschriebenen Iterationen ausführen.

Wenn das alles funktioniert und nach einigen Iterationen die Matrix immer kleine Einträge außerhalb der Diagonalen hat, könnt ihr anfangen die U und V einzusammeln. Dazu dürfen für U und V explizit Matrizen gespeichert werden, auf die die Givensrotationen in jedem Schritt auch angewandt werden. Tipp: Starte mit $U = I$ und $V = I$. Die Funktionen sollten entsprechend zu

```
[U,A] = givensLeft(i,j,k,v,U,A),

[A,V] = givensRight(i,j,k,v,A,V),

[U,A,V] = bidiagonal(A),
```

und

```
[U,A,V] = iteration(U,A,V)
```

geändert werden. Viel Spaß!