Familienname:

Vorname:

Matrikelnummer:

Aufgabe 1 (8 Punkte):
Aufgabe 2 (2 Punkte):
Aufgabe 3 (10 Punkte):
Aufgabe 4 (10 Punkte):

Gesamtpunktzahl:

Schriftlicher Nachtest zu Matlab (90 Minuten)
VU Einführung ins Programmieren für TM (SS 2009)

01. Oktober 2009

Aufgabenstellung. Die sogenannte *Power-Iteration* approximiert (unter gewissen Voraussetzungen) den betragsgrößten Eigenwert $\lambda \in \mathbb{R}$ einer symmetrischen Matrix $A \in \mathbb{R}^{n\times n}$ sowie einen dazugehörigen Eigenvektor $x \in \mathbb{R}^n$. Dazu wählt man einen Startvektor $x^{(0)} \in \mathbb{R}^n\backslash\{0\}$, z.B. $x^{(0)} = (1, \dots, 1) \in \mathbb{R}^n$, und definiert induktiv für $k \in \mathbb{N}$ die Folgen

$$x^{(k)} := \frac{Ax^{(k-1)}}{\|Ax^{(k-1)}\|_2} \quad \text{und} \quad \lambda_k := \sum_{j=1}^{n} x_j^{(k)} (Ax^{(k)})_j,$$

wobei $\|y\|_2 := \big(\sum_{j=1}^n y_j^2\big)^{1/2}$ die euklidische Norm bezeichne. Dann konvergiert die Folge (λ_k) gegen λ, und $(x^{(k)})$ konvergiert gegen einen Eigenvektor zu λ.

Aufgabe 1 (8 Punkte). Schreiben Sie eine Funktion `euklidnorm`, die die euklidische Norm

$$\|y\|_2 := \Big(\sum_{j=1}^{n} y_j^2\Big)^{1/2}$$

eines (Spalten-) Vektors $y \in \mathbb{R}^n$ berechnet. Realisieren Sie die Funktion über eine Schleife. Dabei sollen lediglich skalare Addition und Multiplikation sowie die Funktion `sqrt` verwendet werden.

Lösung zu Aufgabe 1.

Aufgabe 2 (2 Punkte). Realisieren Sie die Funktion `euklidnorm` mit möglichst wenig MATLAB-Code unter Verwendung aller Stärken der MATLAB-Arithmetik.

Lösung zu Aufgabe 2.

Aufgabe 3 (10 Punkte). Schreiben Sie eine Funktion `poweriteration`, die eine Matrix A, eine Toleranz $\tau > 0$ und einen Startvektor $x^{(0)}$ übernimmt und schließlich die Folgen $x^{(k)}$ und λ_k gemäß

$$x^{(k)} := \frac{Ax^{(k-1)}}{\|Ax^{(k-1)}\|_2} \quad \text{und} \quad \lambda_k := \sum_{j=1}^{n} x_j^{(k)} (Ax^{(k)})_j,$$

berechnet, bis gilt

$$\|Ax^{(k)} - \lambda_k x^{(k)}\|_2 \leq \tau \quad \text{sowie} \quad |\lambda_{k-1} - \lambda_k| \leq \begin{cases} \tau & \text{für } |\lambda_k| \leq \tau, \\ \tau\,|\lambda_k| & \text{anderenfalls.} \end{cases}$$

Die Funktion liefere in diesem Fall den Vektor $(\lambda_0, \ldots, \lambda_k)$ sowie die Matrix $(x^{(0)}, \ldots, x^{(k)})$ zurück.

Lösung zu Aufgabe 3.

Aufgabe 4 (10 Punkte). Schreiben Sie eine möglichst rechenökonomische Variante der Funktion `poweriteration`, d.h. vermeiden Sie unnötige Berechnungen, indem Sie Ergebnisse ggf. zwischenspeichern. Ferner sollen *nicht* mehr die gesamten Folgen der λ_j sowie $x^{(j)}$ gespeichert werden, sondern nur noch die jeweils letzten beiden Werte, d.h. $(\lambda_{k-1}, \lambda_k)$ sowie $(x^{(k-1)}, x^{(k)})$. Insbesondere gebe die Funktion `poweriteration` also lediglich die zuletzt berechneten Werte λ_k und $x^{(k)}$ zurück.