Probeklausur

Dies ist eine Probeklausur.

(a) Lösen Sie jede Aufgabe in der dafür vorgesehenen Python-Datei (Aufgabe 1 in Aufgabe1.py, Aufgabe 2 in Aufgabe2.py usw.). Sollte es Teilaufgaben geben, so schreiben Sie Ihre Lösung bitte in den dafür vorgesehenen Abschnitt.

(b) Im untersten Abschnitt haben Sie Platz für eigene Tests. Diese werden NICHT gewertet.

(c) Verändern Sie NICHT die Kommentare, die schon in den Dateien stehen und fügen Sie keine zusätzlichen #%%-Kommentare hinzu.

(d) Ihre Lösung sollte mittels des „run“-Befehls (F5) lauffähig sein.

(e) Sie dürfen alle Dateien in diesem Ordner (Python-Lektionen, Tafelanschrieb und Folien aus den Vorlesungen, sowie Musterlösungen) und ein doppelseitig handbeschriebenes DIN A4 Blatt zum Lösen verwenden. In diesem Ordner befindet sich die html-Seite *index.html*, in der alle Vorlesungen und Materialien aufgelistet sind. Die Dateien in diesem Ordner sind momentan auf dem Stand des 24.01.

Besprechung der Probeklausur in der Vorlesung am 31.01.

MATHEMATISCHES INSTITUT
PROF. DR. ACHIM SCHÄDLE
MARINA FISCHER

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

31. JANUAR 2019

Schriftliche Prüfung zur Computergestützten Mathematik zur linearen Algebra

Probeklausur

Aufgabe 1: (6 Punkte)

(a) Schreiben Sie eine PYTHON-Funktion `y = f(x)`, die

$$f(x) = \frac{1}{4(x-2)^2}$$

auf dem Array `x` auswertet.

(b) Schreiben Sie eine PYTHON-Funktion `y = g(x)`, die die stückweise definierte Funktion

$$g(x) = \begin{cases} (x-1)^2, & x \leq 1 \\ 1 + \sin(\frac{\pi}{2} + \pi x), & x > 1 \end{cases}$$

auf dem Array `x` auswertet.

(c) Plotten Sie beide Funktionen in einen Plot im Intervall $[0, 4]$. Achten Sie darauf, dass $g(x)$ auch an der wichtigen Stelle $x = 1$ ausgewertet wird. Schränken Sie die y-Achse auf das Intervall $[0, 2]$ ein.
Die Funktion $f(x)$ soll in einer grünen, gestrichelten Linie und $g(x)$ in einer schwarzen, durchgezogenen Linie geplottet werden.
Erstellen Sie eine Legende so, dass sie sich in der oberen linken Ecke befindet.

Aufgabe 2: (6 Punkte)

Die *integrierten Legendre-Polynome* $L_k(x)$ sind wie folgt definiert:

$$(k+1)L_{k+1}(x) = (2k-1)xL_k(x) - (k-2)L_{k-1}(x) \text{ für } k \geq 2$$

mit

$$L_1(x) = x \text{ und } L_2(x) = \frac{1}{2}(x^2 - 1).$$

(a) Schreiben Sie eine Funktion `intlepore(n)`, die das `n`-te integrierte Legendre-Polynom auf rekursive Weise bestimmt. Die Polynome sollen hierbei als Objekte der Klasse `Polynom` definiert sein, die Sie aus dem Modul `polynompy` importieren können. Geben Sie eine aussagekräftige Fehlermeldung aus, falls `n` keine ganze Zahl ist.

(b) Plotten Sie für $n = 2, 3, \ldots, 6$ die Polynome $L_n(x)$ mit Hilfe einer Schleife und der `plot`-Methode der Klasse `Polynom` gut sichtbar im Intervall $[-1, 1]$. Jedes Polynom soll mit einer durchgezogenen Linie in einer anderen Farbe dargestellt werden.

Aufgabe 3: (6 Punkte)

Gegeben sei folgender Quelltext für die Eliminierung der Einträge der ersten Spalte einer Matrix `A` $\in \mathbb{R}^{\mathtt{m}\times\mathtt{n}}$. Eliminiert wird mit einem betragsgrößten Element der Spalte, das zuvor nach oben getauscht wird. `A` sei vom `type` Array mit `float`-Werten. Zurückgegeben werden sollen die Matrix, die Faktoren für die Elimination und der ursprüngliche Index der nach oben getauschten Zeile.

```
def EliminiereSpalte(A):
    m, n = A.shape;
    index = np.argmax(abs[A[:, 0]]);
    A[[0, index], :] = A[[0, index], :];
    l = A[1:, [ 0 ] ] * A[0, 0];
    A[1:, :] = A[1:, :] - l @ A[[ 0 ], :];
    return A, l, index;
```

(a) Finden und korrigieren Sie die drei Fehler im Quelltext. Markieren Sie die gefundenen Stellen mit Kommentaren.

(b) Kommentieren Sie, zusätzlich zu den Kommentaren aus (a), die Funktion Zeile für Zeile.

Aufgabe 4: (6 Punkte)

Sei $m \geq 1$ und $A \in \mathbb{R}^{m\times m}$ symmetrisch und invertierbar. Gesucht ist der betragsmäßig kleinste Eigenwert von A und ein zugehöriger Eigenvektor. Dazu dient der folgende Algorithmus:

> Gegeben sei eine Toleranz $\varepsilon > 0$. Wähle den Startwert $v_0 = [1, \ldots, 1]^T \in \mathbb{R}^m$.
> Für $n = 1, 2, \ldots$
> Setze $\widetilde{v}_n = A^{-1}v_{n-1}$ und $v_n = \widetilde{v}_n / \|\widetilde{v}_n\|_2$
> Setze $\mu_n = (v_n)^T A v_n$
> Brich ab, falls $\|Av_n - \mu_n v_n\|_2 < \varepsilon$

(a) Schreiben Sie eine Funktion `mu, v = invPot(A, eps)`, die den obigen Algorithmus implementiert. `A` ist die Matrix A, `eps` die Toleranz für das Abbruchkriterium. `mu` und `v` sind der approximierte Eigenwert bzw. ein dazugehöriger normierter Eigenvektor. Geben Sie die Anzahl der benötigten Iterationen mit `print` aus. Sie dürfen den Array `A` in Ihrer Implementierung NICHT invertieren.

(b) Setzen Sie `eps` $= 10^{-5}$ und testen Sie Ihre Implementierung an der Matrix

$$A = \texttt{np.diag}([10]*19 + [1]) \in \mathbb{R}^{20\times 20}.$$

Vergleichen Sie das Ergebnis mit dem von `numpy.linalg.eig`.

Aufgabe 5: (6 Punkte)

Gegeben sind die x-Werte `x` und die gestörten Funktionswerte `y`, die Sie in der `.py`-Datei zu dieser Aufgabe finden.
Aus sicherer Quelle wissen Sie, dass die Werte gestörte Auswertungen der Funktion

$$y = f(x) = \lambda_1 \log(x) + \lambda_2 \sin(2 \cdot (1 - x)) + \lambda_3 x$$

sind, wobei Sie die Koeffizienten λ_1, λ_2 und λ_3 nicht kennen. Dabei sei $\log = \ln$ der natürliche Logarithmus.
Gesucht sind Koeffizienten $\widetilde{\lambda}_1$, $\widetilde{\lambda}_2$ und $\widetilde{\lambda}_3$, welche die ungestörten Koeffizienten λ_1, λ_2 und λ_3 von f im Sinne der kleinsten Fehlerquadrate möglichst gut approximieren.

(a) Stellen Sie die Matrix A und den Vektor b des zugehörigen Minimierungsproblems der Form

$$\min_{\widetilde{\lambda} \in \mathbb{R}^3} \|A\widetilde{\lambda} - b\|_2 \tag{1}$$

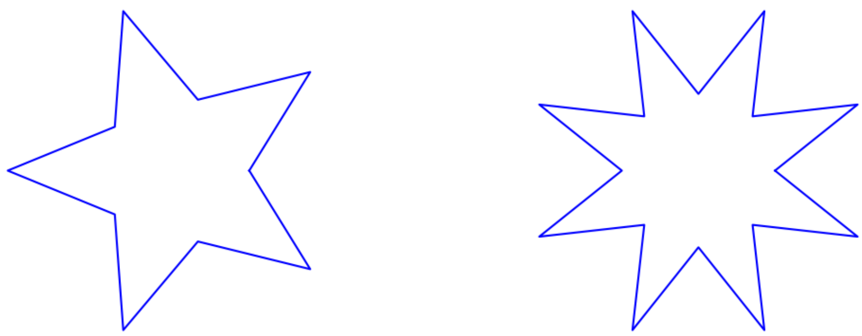
auf. Dabei sei $\widetilde{\lambda} = [\widetilde{\lambda}_1, \widetilde{\lambda}_2, \widetilde{\lambda}_3]^T$.

(b) Bestimmen Sie mit Hilfe der QR-Zerlegung von A einen Vektor $\widetilde{\lambda}$, der (1) löst. Lösen Sie nicht die Normalengleichung, sondern verwenden Sie die Befehle `scipy.linalg.qr` und `scipy.linalg.solve_triangular`.

(c) Plotten Sie die Funktion $\widetilde{f}(x)$, welche die Koeffizienten $\widetilde{\lambda}_i$ statt λ_i in f verwendet, im Intervall $[1, 20]$ zusammen mit den gestörten Funktionswerten in einen gemeinsamen Plot. $\widetilde{f}$ soll an mindestens 100 Punkten im Intervall ausgewertet werden.

Aufgabe 6: (6 Punkte)

Schreiben Sie eine Funktion `Stern(n)`, die die Eckpunkte eines Sterns mit `n` Zacken berechnet und diesen etwa wie unten dargestellt zeichnet. Hierbei sollen nur die Umrisse des Sterns geplottet werden, jedoch keine Linien, die sich durch den Stern ziehen. Ist kein `n` festgelegt, soll ein fünfzackiger Stern entstehen.



Beispiele für einen fünf- und einen achtzackigen Stern.
Die Orientierung und die Größen sind hier nur beispielhaft.