Prof. Dr. Ulrich Rüde
Lehrstuhl für Systemsimulation
FAU Erlangen-Nürnberg

Algorithmik kontinuierlicher Systeme — 13. Februar 2018

Angaben zur Person (Bitte in DRUCKSCHRIFT ausfüllen!):

Name, Vorname: ______________________________

Geburtsdatum: ______________________________

Matrikelnummer: ______________________________

Studienfach: ______________________________

Nicht von der Kandidatin bzw. vom Kandidaten auszufüllen!

Aufgabe	1	2	3	4	5	6	7	8	9
Max. Punktzahl	9	8	17	13	8	9	10	10	6
Erreichte Punkte									

Gesamtpunktzahl	
Note	

Organisatorische Hinweise

Die folgenden Hinweise bitte aufmerksam lesen und die Kenntnisnahme durch Unterschrift bestätigen!

- Bitte legen Sie Ihren Studentenausweis und einen Lichtbildausweis zur Personenkontrolle bereit.
- Hilfsmittel (außer Schreibmaterial und Lineal) sind nicht zugelassen. Elektronische Geräte sind auszuschalten.
- Fragen zu den Prüfungsaufgaben werden grundsätzlich nicht beantwortet.
- Die Lösung einer Aufgabe muss auf das jeweilige Aufgabenblatt geschrieben werden. Sollte der Platz nicht reichen, verwenden Sie die Zusatz-Seiten am Ende der Klausur. Fügen Sie einen Hinweis in Ihre Lösung ein, wenn die Lösung auf den Zusatz-Seiten fortgesetzt wurde und beschriften Sie diese mit Namen und Aufgabennummer.
- Es können durch die Aufsicht zusätzlich Seiten eingeheftet werden, sollte mehr Platz benötigt werden. Bitte beschriften Sie den Kopf dieser Seiten mit Ihrem Namen und Matrikelnummer. Streichen Sie alles, was nicht verwendet werden soll, doppelt aus.
- Die Programmieraufgaben sind in der Programmiersprache Python 3 zu bearbeiten.
- Wenn Sie die Prüfung aus gesundheitlichen Gründen abbrechen müssen, so muss Ihre Prüfungsunfähigkeit durch eine Untersuchung bei einem Vertrauensarzt nachgewiesen werden. Melden Sie sich bei der Aufsicht und lassen Sie sich das entsprechende Formular aushändigen.
- Die Bearbeitungszeit beträgt 90 Minuten.
- Überprüfen Sie die Prüfungsaufgaben auf Vollständigkeit (18 Seiten inklusive Deckblatt) und einwandfreies Druckbild.
- Vergessen Sie nicht, auf dem Deckblatt die Angaben zur Person einzutragen und die **Erklärungen auf dieser Seite zu unterschreiben**.
- Viel Erfolg!

Erklärungen

Durch meine Unterschrift bestätige ich den Empfang der vollständigen Klausurunterlagen und die Kenntnisnahme der obigen Informationen.

Erlangen, 13. Februar 2018

____

(Unterschrift)

Aufgabe 1 — Theoriefragen (9 Punkte)

a) Beantworten Sie die folgenden Fragen! Schreiben Sie Ihre Antwort in die rechte Spalte der Tabelle!

Sei A eine $(n \times n)$-Matrix und b ein n-Vektor. Welche Komplexität hat das Lösen des Gleichungssystems $Ax = b$ mittels Gauß-Elimination.	$\mathcal{O}(\quad\quad)$
Sei D eine invertiarbare $(n \times n)$-Diagonalmatrix. Welche Komplexität hat die Berechnung der Inversen von D.	$\mathcal{O}(\quad\quad)$
Sei f eine ausreichend glatte, eindimensionale Funktion auf $\mathbb{R}$. Was ist Ordnung des Interpolationsfehlers bei linearer Interpolation von f mit äquidistanter Schrittweite h.	$\mathcal{O}(\quad\quad)$
Sei A eine nichtsinguläre $(n \times n)$-Matrix und b ein n-Vektor. Welche Komplexität hat die Bestimmung der LR-Zerlegung $A = PLR$?	$\mathcal{O}(\quad\quad)$
Sei A eine dünnbesetzte $(n \times n)$-Matrix mit höchstens fünf von Null verschiedenen Einträgen pro Zeile. Welche Komplexität hat die Berechnung von zwei Iterationsschritten des Jacobi-Verfahrens zur Lösung von $Ax = b$.	$\mathcal{O}(\quad\quad)$

b) Sind folgende Gleichungssysteme $Ax = b$ überbestimmt, unterbestimmt oder eindeutig lösbar?

	überbestimmt	unterbestimmt	eindeutig lösbar
$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$	☐	☐	☐
$\begin{bmatrix} 1 & 3 \\ 2 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}$	☐	☐	☐
$\begin{bmatrix} 1 & 2 & 1 \\ 3 & 6 & 3 \\ 2 & 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 9 \\ 6 \end{bmatrix}$	☐	☐	☐
$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix}$	☐	☐	☐

Aufgabe 2 — QR-Zerlegung (15 Punkte)

a) Von der Matrix A ist folgende QR-Zerlegung bekannt:

$$A = \frac{1}{3}\begin{bmatrix} 1 & -7 & 2 \\ -2 & 5 & 5 \\ -2 & -4 & 5 \end{bmatrix} = \underbrace{\frac{1}{3}\begin{bmatrix} 1 & -2 & 2 \\ -2 & 1 & 2 \\ -2 & -2 & -1 \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} 1 & -1 & -2 \\ 0 & 3 & -1 \\ 0 & 0 & 1 \end{bmatrix}}_{R}$$

Lösen Sie das lineare Gleichungssystem $Ax = b$ für $b = [-3, 3, -3]^T$.

b) Berechnen Sie nun $|\det(A)|$, also den Betrag der Determinante von A.

c) Implementieren Sie nun die Funktion `forward_substitute`, die eine untere Dreiecksmatrix L (als $n \times n$ NumPy-Array) und einen n-Vektor b (als $n \times 1$ NumPy-Array) übergeben bekommt und die das Gleichungssystem $Ly = b$ nach y löst. Nutzen Sie dabei die besondere Struktur der Matrix L.
Hinweis: Vergessen Sie nicht, Ihren Code richtig einzurücken! Orientieren Sie sich dabei an den gepunkteten Hilfslinien.

```
import numpy as np

def forward_substitute(L, b):
    """Solve L @ y = b for y."""
```

Aufgabe 3 — Interpolation (10 Punkte)

a) Nennen Sie jeweils zwei Eigenschaften der folgenden Interpolationsverfahren:

Catmull-Rom-Interpolation

-
-

Nearest-Neighbor-Interpolation

-
-

b) Bestimmen Sie die Koeffizienten des Newton-Interpolationspolynoms zu den Folgenden Punkten:

i	0	1	2
x_i	0	1	3
y_i	1	-1	7

Aufgabe 4 — Python (13 Punkte)

a) Lösen Sie die folgenden Probleme jeweils mit **einer** Zeile Python!
`numpy` ist als `np` importiert.

Zur Erinnerung: `lambda X: ( AUSDRUCK )` funktioniert wie `def NAME_EGAL(X): return AUSDRUCK`

- **Beispiel**: Prüfen, ob die Ganzzahl `n` gerade ist.

  `lambda n: ( n % 2 == 0 )`

- Den Mittelwert einer beliebigen Liste `L` berechnen.

  `lambda L: (                                    )`

- Das größte Element einer beliebigen Liste `L` zurück geben.

  `lambda L: (                                    )`

- Eine Zeichenkette in der Mitte tauschen, e.g. `"ALGOKS"` → `"OKSALG"`. (die Länge der Zeichenkette ist gerade. Listenzugriff ist nur mit Ganzzahlen möglich.)

  `lambda str: (                                    )`

- Die Anzahl der Elemente `e` in einer beliebigen Liste `L` berechnen.

  `lambda L, e: (                                    )`

- Prüfen, ob die gegebene Zeichenkette `day` ein `'S'` enthält.

  `lambda day: (                                    )`

b) Was geben folgende Codeschnipsel aus?

```
a = 3
for i in range(5):
    a = i*i

print(a)
```

Ausgabe: _

```
A = [2,3,1,4]

print(A[3:1:-1])
```

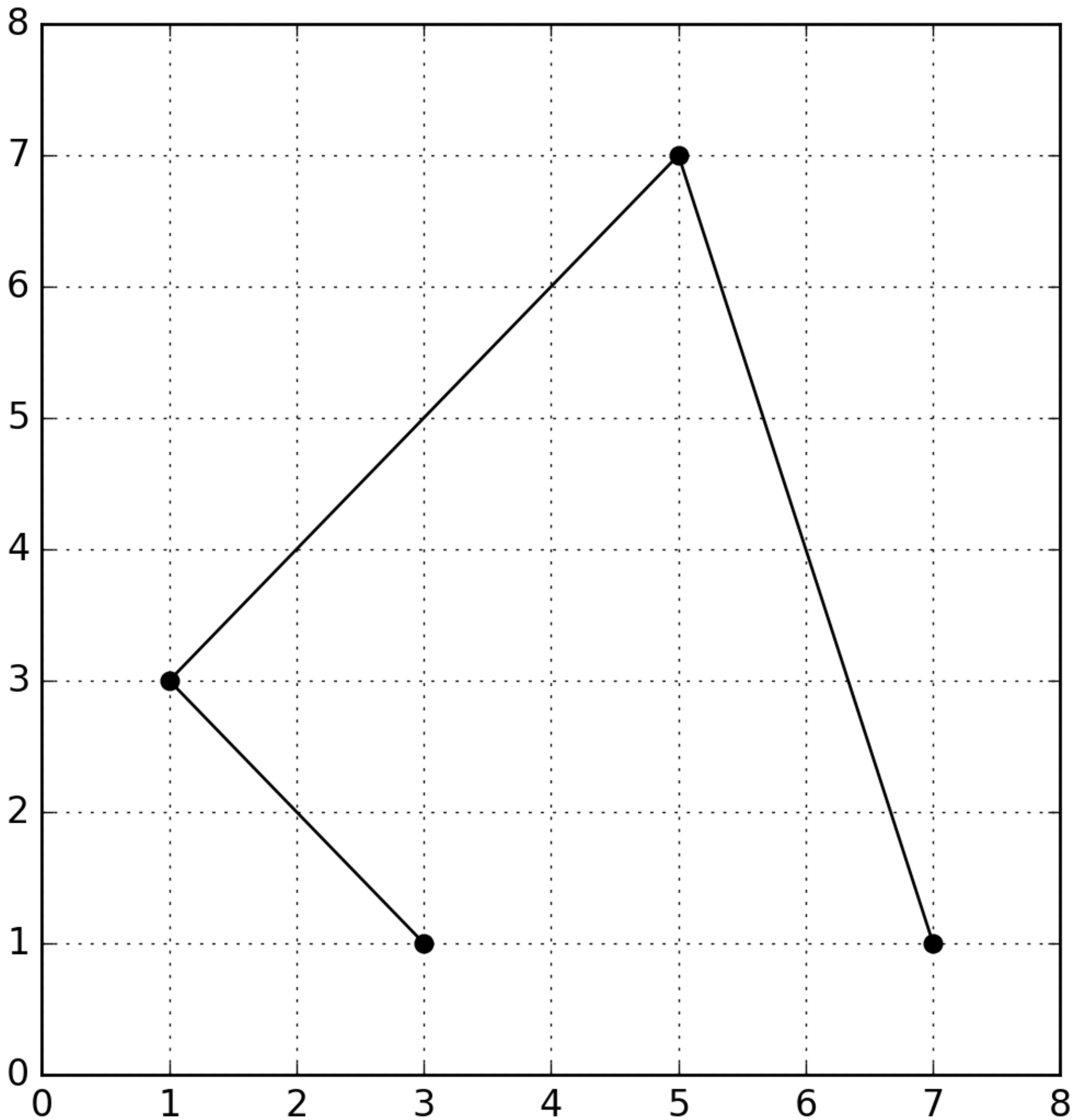
Ausgabe: [_ , _]

```
import numpy as np
A = np.array([1, 2, 3])
B = A[:2]
C = B[::-1]
C[1] = 5
print(A)
```

Ausgabe: [_ , _ , _]

Aufgabe 5 — Bézierkurven (8 Punkte)

a) Bestimmen Sie graphisch zum gegebenen Kontrollpolygon den Punkt der zugehörigen Bézierkurve mit Parameter $t = \frac{1}{2}$.



b) Lesen Sie aus Aufgabe a) die Koordinaten von vier Punkten ab, die als Kontrollpolygon genau **die rechte Hälfte** der ursprünglichen Bézierkurve beschreiben.

$(x_1, y_1) = (\qquad , \qquad)$ $(x_2, y_2) = (\qquad , \qquad)$ $(x_3, y_3) = (\qquad , \qquad)$ $(x_4, y_4) = (\qquad , \qquad)$

c) Nennen Sie drei Formeigenschaften von Bézierkurven.

-
-
-

Aufgabe 6 — Iterative Lösungsverfahren (9 Punkte)

Gegeben sei folgendes Gleichungssystem $Ax = b$:

$$\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 1 \end{bmatrix}$$

a) Führen Sie einen Iterationsschritt des Jacobi-Verfahrens mit dem Startwert $[0, 0, 0]^T$ aus. Der Rechenweg muss erkennbar sein.

Zur Erinnerung: Gegeben sei das Gleichungssystem $Ax = b$:

$$\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 1 \end{bmatrix}$$

b) Führen Sie einen Iterationsschritt des Gauß-Seidel-Verfahrens (Reihenfolge x_1, x_2, x_3) mit Startwert $[0, 0, 0]^T$ aus. Der Rechenweg muss erkennbar sein.

c) Nennen Sie eine hinreichende Voraussetzung dafür, dass das Jacobi-Verfahren zur Lösung von $Ax = b$ nach x konvergiert:

Aufgabe 7 — Singulärwertzerlegung (10 Punkte)

Von der Matrix A sei die Singulärwertzerlegung bekannt:

$$A = \frac{1}{5}\begin{bmatrix} 15 & -8 \\ -20 & -6 \end{bmatrix} = \frac{1}{5}\begin{bmatrix} 3 & 4 \\ -4 & 3 \end{bmatrix}\begin{bmatrix} 5 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

a) Bestimmen Sie den Rang und die Eigenwerte von AA^T.

b) Geben Sie die Definitionen der Spektralnorm $||\cdot||_2$, der Zeilensummennorm $||\cdot||_\infty$ und der Frobeniusnorm $||\cdot||_F$ an und bestimmen Sie $||A||_2$, $||A||_\infty$ und $||A||_F$.

c) Gegeben sei nun die Matrix $B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix}$. Bestimmen Sie eine Singulärwertzerlegung von B.

Hinweis: Die Singulärwerte müssen absteigend sortiert sein, d.h. $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$.

Aufgabe 8 — Nichtlineare Optimierung (10 Punkte)

Sie versuchen, das Minimum der folgenden Funktion zu bestimmen:

$$F(x, y) = x^2 + 2y^2 - 2xy + 4x + 42$$

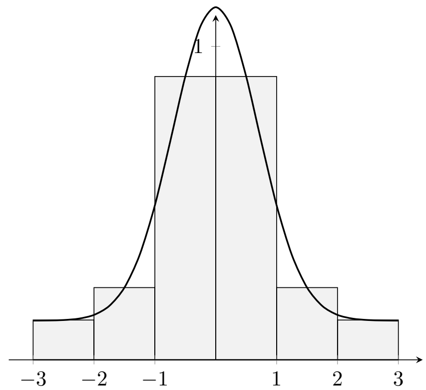
a) Berechnen Sie die Jacobi- und die Hessematrix von $F(x, y)$.

b) Führen Sie einen Schritt des Gradientenabstiegs-Verfahren durch. Verwenden Sie dazu die Schrittweite $t_0 = 1$ und den Startwert $[x_0, y_0] = [2, 0]$.

c) Führen Sie für das gleiche Problem einen Schritt des Newton-Verfahrens durch. Wählen Sie als Startwert erneut $[x_0, y_0] = [2, 0]$. **Hinweis:** Gesucht ist das Minimum von F, nicht die Nullstellen.

Aufgabe 9 — Riemannsches Integral (6 Punkte)

Implementieren Sie eine Funktion `integrate`, welche das Integral einer Funktion $f(x)$ auf einem Intervall $[x_{\min}, x_{\max}]$ abschätzt. Dazu soll das Intervall in N Rechtecke gleicher Breite unterteilt werden, wobei die Höhe des Rechtecks der Funktionswert am Mittelpunkt des Teilintervalls ist. Zurückgegeben wird die Summe der Flächen aller Rechtecke. Bei den Koordinaten und Funktionswerten handelt es sich stets um Gleitkommazahlen.



Beispiel für $f(x) = e^{-x^2} + \frac{1}{8}, x_{min} = -3, x_{max} = 3, N = 6$

```
import numpy as np

def integrate(f, xmin, xmax, N):
```