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

Algorithmik kontinuierlicher Systeme — 3. August 2017

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
Max. Punktzahl	9	15	9	13	8	12	10	14
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, 3. August 2017 ______________________________

(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 tridiagonale $(n \times n)$-Matrix und b ein n-Vektor. Welche Komplexität hat die Matrix-Vektor-Multiplikation Ab?	$\mathcal{O}(\qquad)$
Welche Komplexität hat die Berechnung der Euklidischen Norm eines n-Vektors?	$\mathcal{O}(\qquad)$
Sei A eine $(n \times n)$-Matrix und b ein n-Vektor. Welche Komplexität hat das Lösen eines Gleichungssystems $Ax = b$, wenn die LR-Zerlegung $A = PLR$ bereits vorliegt?	$\mathcal{O}(\qquad)$
Sei A eine nichtsinguläre $(n \times n)$-Matrix und b ein n-Vektor. Welche Komplexität hat die Berechnung der QR-Zerlegung $A = QR$?	$\mathcal{O}(\qquad)$
Sei A eine dünnbesetzte $n \times n$ Matrix mit höchstens sieben von Null verschiedenen Einträgen pro Zeile. Welche Komplexität hat die Berechnung eines einzelnen Iterationsschritts des SOR-Verfahrens?	$\mathcal{O}(\qquad)$

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

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

Aufgabe 2 — QR-Zerlegung (15 Punkte)

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

$$A = \frac{1}{3}\begin{bmatrix} 1 & -5 & 8 \\ -2 & 1 & -16 \\ 2 & 8 & -11 \end{bmatrix} = \underbrace{\frac{1}{3}\begin{bmatrix} 1 & -2 & -2 \\ -2 & 1 & -2 \\ 2 & 2 & -1 \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} 1 & 1 & 2 \\ 0 & 3 & -6 \\ 0 & 0 & 3 \end{bmatrix}}_{R}$$

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

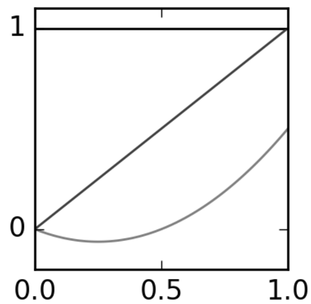
b) Berechnen Sie nun $|\det(A)|$, also den Betrag der Determinante von A.

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

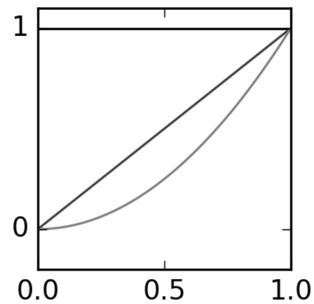
```
import numpy as np

def backward_substitute(R, y):
    """Solve R @ x = y for x."""
```

Aufgabe 3 — Interpolation (9 Punkte)

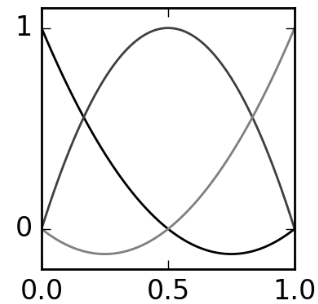
a) Jedes dieser Bilder beschreibt eine Basis für Polynome zweiten Grades mit den Stützstellen 0, $\frac{1}{2}$ und 1. Schreiben Sie unter jedes Bild, um welche Art von Basis es sich handelt (der Name genügt).



Typ:



Typ:



Typ:

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

i	0	1	2
x_i	0	2	4
y_i	1	2	-1

Aufgabe 4 — Python (13 Punkte)

a) Lösen Sie die folgenden Probleme jeweils mit **einer** Zeile Python!

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 )`

- Eine Zeichenkette `str` viermal aneinander hängen, e.g. `"AB"` → `"ABABABAB"`.

  `lambda str: (                                        )`

- Prüfen, ob die Zeichenkette `str` ein Palindrom ist, d.h. vorwärts und rückwärts gelesen identisch.

  `lambda str: (                                        )`

- Den Median einer beliebigen Liste von Ganzzahlen `L` berechnen (Das Element, bei dem genau die Hälfte der Elemente kleiner ist und die andere Hälfte größer). Sie dürfen annehmen, dass `L` ungerade Länge hat.

  `lambda L: (                                        )`

- Die Euklidische Norm eines Vektors `v` (repräsentiert als eindimensionales NumPy-Array aus Gleitkommazahlen) berechnen. Das Modul NumPy sei unter dem Namen `np` verfügbar.

  `lambda v: (                                        )`

- Prüfen, ob die gegebene Zeichenkette `month` ein `'R'` oder `'r'` enthält.

  `lambda month: (                                        )`

b) Was geben folgende Codeschnipsel aus?

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

print(a)
```

Ausgabe: _

```
import numpy as np
L = [0]
A = np.array([L, L])
a,b = A.shape

print(b - a)
```

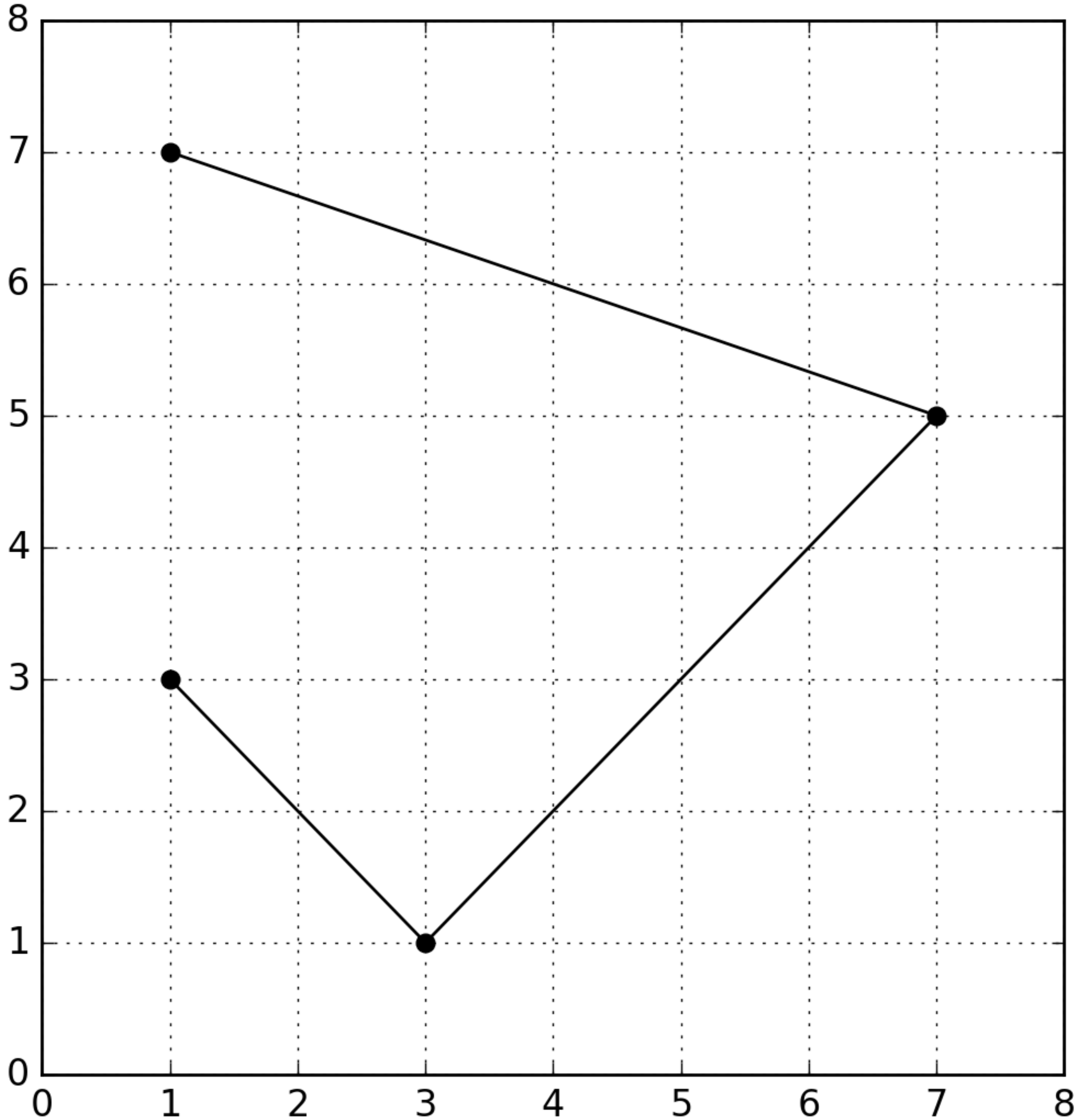
Ausgabe: _

```
import numpy as np
A = np.array([1, 2, 3])
B = A[:2]
C = B[::-1]
C[0] = 4
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 obere Hälfte der ursprünglichen Bézierkurve beschreiben.

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

c) Nennen Sie drei Formeigenschaften von Bézierkurven.

-
-
-

Aufgabe 6 — Iterative Lösungsverfahren (12 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.

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) In welchem Bereich muss der Relaxationsparameter ω beim SOR-Verfahren liegen, damit das Verfahren für positiv definite Matrizen konvergiert?

d) Begründen Sie jeweils knapp, wieso das Gauß-Seidel-Verfahren für folgende Matrizen konvergiert:

$\begin{bmatrix} 4 & -1 & 0 & -1 \\ -1 & 4 & -1 & 0 \\ 0 & -1 & 4 & -1 \\ -1 & 0 & -1 & 4 \end{bmatrix}$	
$\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}$	
$\begin{bmatrix} 1 & 2 & 4 \\ 0 & 100 & 0 \\ 0 & 0 & 20 \end{bmatrix}$ mit den Eigenwerten $\lambda_1 = 1$, $\lambda_2 = 100$ und $\lambda_3 = 20$	

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 von A den Rang, sowie die Konditionszahl κ_2 bezüglich der Spektralnorm $||.||_2$.

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 $B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$. 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 (14 Punkte)

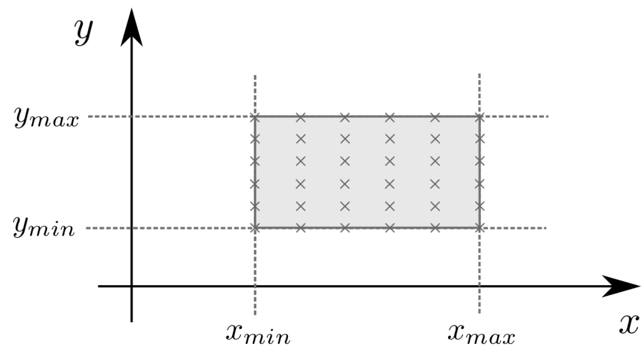
Sie versuchen, das Minimum der folgenden Funktion zu bestimmen:

$$F(x, y) = 2x^2 + 3y^2 + 2x - y + 42$$

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

b) Führen Sie für das gleiche Problem einen Schritt des Newton-Verfahrens durch. Wählen Sie als Startwert erneut $[x_0, y_0] = [1, 0]$. **Hinweis:** Gesucht ist das Minimum von F, nicht die Nullstellen.

c) Von der Mathematik frustriert beschließen Sie, das Minimum einer Funktion anzunähern, indem Sie die Funktion einfach für zahlreiche x- und y-Koordinaten auswerten. Implementieren Sie eine Funktion `minimize`, welche eine Funktion $F(x, y)$ auf einem $n \times n$ Gitter über dem Gebiet $(x_{\min}, x_{\max}) \times (y_{\min}, y_{\max})$ auswertet (siehe Skizze) und die die Koordinaten des kleinesten Funktionswertes auf dem Gitter als Tupel (x, y) zurückgibt. Bei den Koordinaten und Funktionswerten handelt es sich stets um Gleitkommazahlen.



Beispiel für $n = 6$

```
import numpy as np

def minimize(F, n, xmin, xmax, ymin, ymax):
```