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

Algorithmik kontinuierlicher Systeme — 02. August 2019

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	10	12	7	11	10	9	12	10	9
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 Zusatzseiten am Ende der Klausur. Fügen Sie einen Hinweis in Ihre Lösung ein, wenn die Lösung auf den Zusatzseiten fortgesetzt wurde und beschriften Sie diese mit der richtigen 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 (22 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, 02. August 2019

____

(Unterschrift)

Aufgabe 1 — Theoriefragen (10 Punkte)

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

Welche Komplexität hat die Bestimmung des Betrags der Determinante einer $(n \times n)$-Matrix A, wenn die QR-Zerlegung von A bereits bekannt ist?	$\mathcal{O}(\qquad\qquad)$
Welche Komplexität hat die Matrixmultiplikation von zwei $(n \times n)$-Matrizen mit dem Standardalgorithmus?	$\mathcal{O}(\qquad\qquad)$
Sei f eine ausreichend glatte, eindimensionale Funktion auf $\mathbb{R}$. Was ist die Ordnung des Interpolationsfehlers bei linearer Interpolation von f mit äquidistanter Schrittweite h?	$\mathcal{O}(\qquad\qquad)$
Welche Komplexität hat das Lösen eines Gleichungssystems $Rx = b$, wenn R eine nichtsinguläre obere $(n \times n)$-Dreiecksmatrix ist?	$\mathcal{O}(\qquad\qquad)$
Welche Komplexität hat die Berechnung der Diskreten Fouriertransformation (DFT) eines Vektors der Länge n mit dem Algorithmus von Cooley und Tukey?	$\mathcal{O}(\qquad\qquad)$

b) Kreuzen Sie die zutreffenden Matrixeigenschaften an:

	strikt diagonaldominant	symmetrisch positiv definit	singulär
$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	☐	☐	☐
$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$	☐	☐	☐
$\begin{bmatrix} 2 & 1 \\ 2 & 1 \end{bmatrix}$	☐	☐	☐
$\begin{bmatrix} 3 & 1 \\ 2 & 4 \end{bmatrix}$	☐	☐	☐
$\begin{bmatrix} 2 & 3 \\ 3 & 10 \end{bmatrix}$	☐	☐	☐

Aufgabe 2 — LR-Zerlegung (12 Punkte)

a) Bestimmen Sie für die Matrix A die LR-Zerlegung mit Spalten-Pivotsuche, d.h., eine Permutationsmatrix P, eine untere Dreiecksmatrix L und eine obere Dreiecksmatrix R, sodass gilt $A = PLR$.

$$A = \begin{bmatrix} 4 & 5.5 & 1 \\ 10 & 5 & 8 \\ 2 & 8 & 1 \end{bmatrix}$$

b) Rückwärtssubstitution 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.

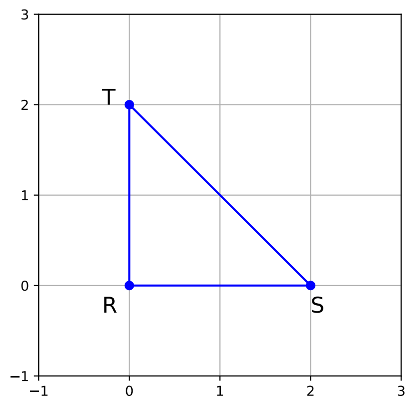
```
import numpy as np

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

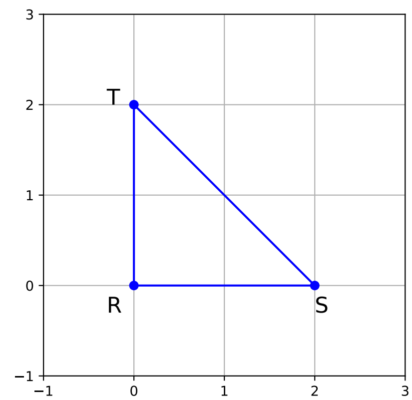
Aufgabe 3 — Baryzentrische Koordinaten (7 Punkte)

Im Folgenden ist ρ das Gewicht von R, σ das Gewicht von S und τ das Gewicht von T.

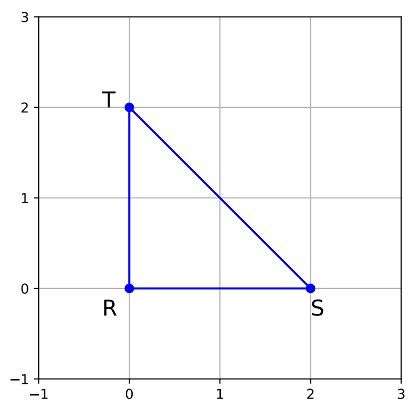
a) Zeichnen Sie die Bereiche in die Skizzen ein für die die gegebenen Gewichte gelten.



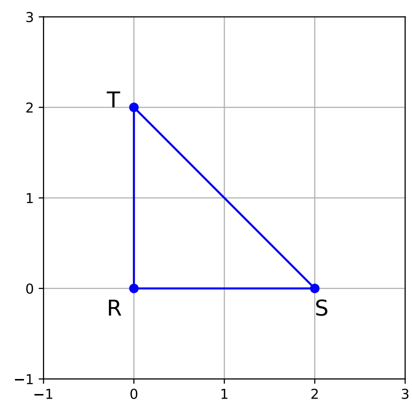
$$\rho \geq 0 \quad \wedge \quad \sigma \geq 0 \quad \wedge \quad \tau \geq 0$$



$$\rho \leq 0 \quad \wedge \quad \sigma \leq 0 \quad \wedge \quad \tau \geq 0$$



$$\rho = 0 \quad \wedge \quad \sigma \geq 0 \quad \wedge \quad \tau \leq 0$$



$$\rho \leq 0 \quad \wedge \quad \sigma \geq 0 \quad \wedge \quad \tau \geq 0$$

b) Berechnen Sie bezüglich des Dreiecks mit den Koordinaten $R = [-2, 0]^T, S = [0, -1]^T, T = [2, 2]^T$ die Koordinaten für die Punkte P_1, P_2, P_3 mit folgenden baryzentrischen Koordinaten

	ρ	σ	τ
P_1	0.5	0.25	0.25
P_2	-1	$2/3$	$1/3$
P_3	0	$1/3$	$2/3$

Aufgabe 4 — Python (11 Punkte)

a) Lösen Sie die folgenden Probleme mit je einer Funktion in Python!

Hinweis: Vergessen Sie nicht, Ihren Code richtig einzurücken! Orientieren Sie sich dabei an den gepunkteten Hilfslinien.

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

```
def ganzzahl(n):
    return n % 2 == 0
```

- Das Minimum einer Liste von Ganzzahlen `L` zurückgeben.

- Eine neue Liste `L` zurückgeben in der die gleichlangen Listen `R1` und `R2` elementweise addiert sind.

- Zählen wie oft in zwei gleich langen Listen a und b gilt: $a_i < b_i$.

- Prüfen, ob eine Zeichenkette `S` ein Palindrom ist.

b) Was geben folgende Codeschnipsel aus?

```
a = [3]
for i in range(10):
  if i % 3 == 0:
    a = a + [i]
print(a)
```

Ausgabe:

```
import numpy as np
A = np.zeros((2,2))
A[:,0] += 1
A[:,1] += 2
print(A[0] + A[1])
```

Ausgabe:

```
A = [1,3,5,7,9,11,13,16]
print (A[1:7:2])
```

Ausgabe:

```
A = [1,1,1,1]
A[0:2], A[3] = [4,4], 5
print(A)
```

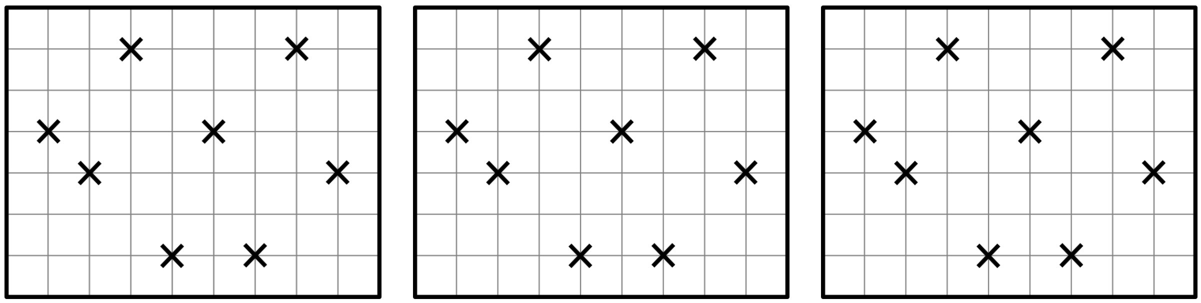
Ausgabe:

```
L = [[1]]*2
L[0].append(3)
print(L)
```

Ausgabe:

Aufgabe 5 — Farben und Formen (10 Punkte)

a) Median Cut Ziel dieser Aufgabe ist es, acht Datenpunkte mit je zwei Komponenten mit dem Median Cut Algorithmus auf vier representative Datenpunkte zu vereinfachen. Zeichnen Sie dazu in die folgenden drei Bilder die umgebenden Boxen des jeweils ersten, zweiten und dritten Schritts des Median Cut Algorithmus ein. Markieren Sie dann im dritten Bild die resultierenden vier Datenpunkte.



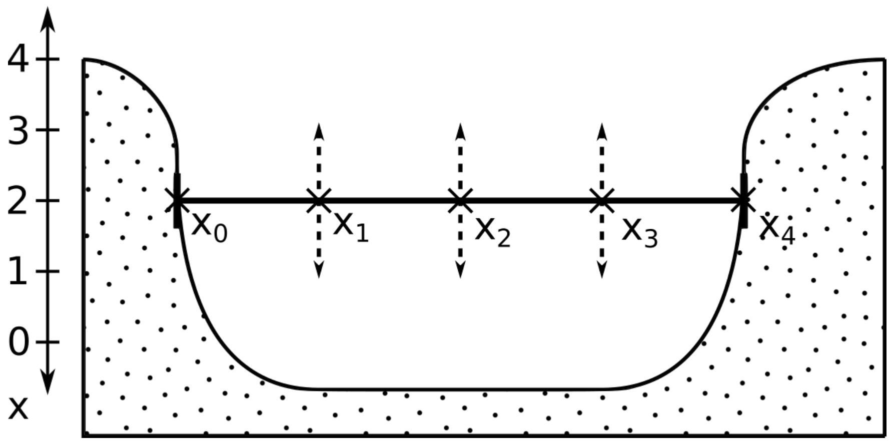
Schritt 1 - Eine Box **Schritt 2** - Zwei Boxen **Schritt 3** - Vier Boxen

b) Polynominterpolation Bestimmen Sie das Interpolationspolynom zu den Folgenden Punkten:

i	0	1	2
x_i	0	2	4
y_i	1	3	-1

Aufgabe 6 — Numerik in der Praxis (9 Punkte)

Ziel dieser Aufgabe ist es, die Auslenkung einer Brücke zu bestimmen. Der Einfachheit halber modellieren Sie Ihre Brücke durch nur 5 diskrete x-Koordinaten $x_0, \dots, x_4$.



a) Gleichungssystem Aufstellen Aus einem alten Buch über Brückenbau entnehmen Sie folgende Formel für die Koordinaten der inneren Punkte ihrer Brücke.

$$3x_i - x_{i+1} - 2x_{i-1} = f_i, \quad i \in \{1, 2, 3\}$$

Aus Ihren Messungen kennen Sie die Randwerte $x_0 = x_4 = 2$, sowie die zu erwartende Belastung der Brücke $f_1 = -1$, $f_2 = -3$, $f_3 = -1$. Stellen Sie nun das zugehörige Gleichungssystem auf:

$$\begin{bmatrix} \text{—} & \text{—} & \text{—} \\ \text{—} & \text{—} & \text{—} \\ \text{—} & \text{—} & \text{—} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \text{—} \\ \text{—} \\ \text{—} \end{bmatrix} \qquad \text{(System I)}$$

b) Lösbarkeit Zur Sicherheit bitten Sie eine befreundete Mathematikerin um Hilfe, die Ihnen stattdessen folgendes Gleichungssystem vorschlägt:

$$\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -2 \\ 4 \end{bmatrix} \qquad \text{(System II)}$$

Begründen Sie, wieso das Gauss-Seidel Verfahren geeignet ist, um System II zu lösen:

c) Gleichungssystem Lösen Führen Sie nun für das System II zwei Iterationen des lexikographischen Gauss-Seidel Verfahrens mit dem Startwert $\mathbf{x}^0 = [x_1, x_2, x_3] = [0, 0, 0]$ durch. Geben Sie die resultierenden Vektoren $\mathbf{x}^1$ und $\mathbf{x}^2$ an.

Aufgabe 7 — Lineare Algebra (12 Punkte)

Gegeben sei die Matrix

$$A = \begin{pmatrix} 6 & 0 & 4 \\ 0 & 2 & a \\ 0 & 4 & 2a \end{pmatrix} \in \mathbb{R}^{3\times 3}$$

mit $a \in \mathbb{R}$.

a) Bestimmen Sie das charakteristische Polynom sowie alle Eigenwerte von A.

b) Geben Sie die Definition des Spektralradius $\rho(M)$ einer Matrix $M \in \mathbb{C}^{n\times n}$ an und bestimmen Sie $\rho(A)$ in Abhängigkeit von a.

c) Für welche Werte von a ist die Matrix A singulär?

d) Bestimmen Sie alle Eigenvektoren der Matrix A für den Fall $a = 2$.

Aufgabe 8 — Nichtlineare Optimierung (10 Punkte)

a) Sie versuchen das Minimum der folgenden Funktion zu bestimmen

$$F(x,y) = 2x^2y^2 + 3x^2 - 3x + 2y^3 + y^2 + 3y - 1$$

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

b) Rechnen Sie nun mit folgender Jacobi und Hessematrix weiter:

$$J_F(x,y) = \begin{bmatrix} 3x^2 + 6xy + 6x - y^2 - 4 \\ 3x^2 - 2xy + 3 \end{bmatrix}^T$$

$$H_F(x,y) = \begin{bmatrix} 6x + 6y + 6 & 6x - 2y \\ 6x - 2y & -2x \end{bmatrix}$$

Führen Sie einen Schritt des Gradientenverfahrens mit Startwert $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$ und Schrittweite $t = 0.5$ durch.

c) Rechnen Sie nun mit folgender Jacobi und Hessematrix weiter:

$$J_F(x,y) = \begin{bmatrix} 6xy + 8x - y^2 - 4 \\ 3x^2 - 2xy + 4 \end{bmatrix}^T$$

$$H_F(x,y) = \begin{bmatrix} 6y+8 & 6x-2y \\ 6x-2y & -2x \end{bmatrix}$$

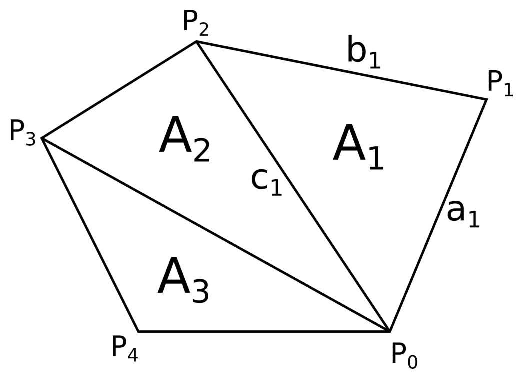
Außerdem ist gegeben

$$J_F(1,3) = \begin{bmatrix} 13 \\ 1 \end{bmatrix}$$

Führen Sie einen Schritt des Newton-Verfahrens durch. Wählen Sie als Startwert $\begin{bmatrix} 1 \\ 3 \end{bmatrix}$. Hinweis: Gesucht ist das Minimum von F , nicht die Nullstellen.

Aufgabe 9 — Fläche eines konvexen Polygons (9 Punkte)

Ziel dieser Aufgabe ist es, mit Python die Fläche eines beliebigen konvexen Polygons zu bestimmen.



a) Seitenlänge Schreiben Sie eine Funktion `distance`, die den Abstand zwischen zwei Punkten P1 und P2 zurückgibt. Die Punkte seien durch `(x, y)`-Tupel aus Gleitkommazahlen repräsentiert.

```
from math import sqrt
def distance(P1, P2):
  ⋮   ⋮   ⋮   ⋮   ⋮
  ⋮   ⋮   ⋮   ⋮   ⋮
  ⋮   ⋮   ⋮   ⋮   ⋮
  ⋮   ⋮   ⋮   ⋮   ⋮
  ⋮   ⋮   ⋮   ⋮   ⋮
  ⋮   ⋮   ⋮   ⋮   ⋮
  ⋮   ⋮   ⋮   ⋮   ⋮
```

b) Dreiecksfläche Schreiben Sie eine Funktion `triangle_area`, die die Fläche eines beliebigen Dreiecks, das durch seine drei Eckpunkte gegeben ist, mit dem Satz des Heron berechnet. Dieser lautet

$$A_\Delta = \sqrt{s\,(s - a)\,(s - b)\,(s - c)}, \tag{1}$$

wobei a, b und c die Seitenlängen des Dreiecks sind, und $s = (a + b + c)/2$ der halbe Umfang des Dreiecks.

```
from math import sqrt
def triangle_area(P1, P2, P3):
  ⋮   ⋮   ⋮   ⋮   ⋮
  ⋮   ⋮   ⋮   ⋮   ⋮
  ⋮   ⋮   ⋮   ⋮   ⋮
  ⋮   ⋮   ⋮   ⋮   ⋮
  ⋮   ⋮   ⋮   ⋮   ⋮
  ⋮   ⋮   ⋮   ⋮   ⋮
  ⋮   ⋮   ⋮   ⋮   ⋮
  ⋮   ⋮   ⋮   ⋮   ⋮
```

c) Gesamtfläche Schreiben Sie nun eine Funktion `polygon_area`, die die Fläche eines konvexen Polygons berechnet. Dazu wird das Polygon in Dreiecke aufgeteilt und deren Fläche aufsummiert. Das Polygon wird als Liste aus Punkten übergeben. Die einzelnen Punkte werden durch `(x,y)`-Tupel aus Gleitkommazahlen repräsentiert.

```
def polygon_area(points):
```