....
Note

	I	II
1		
2		
3		
4		
5		
$\sum$		

Name

Vorname

Matrikelnummer

Studiengang

Obige Angaben sind richtig:

Unterschrift der Kandidatin/des Kandidaten

TECHNISCHE UNIVERSITÄT MÜNCHEN
Fakultät für Mathematik

Modulprüfung

MA8003 Einführung in die Programmierung

20.10.2017, 16:30–17:30 Uhr

Dr. Laura Scarabosio

I ....
Erstkorrektur

II ....
Zweitkorrektur

Hinweise:

Überprüfen Sie die Angabe: Es sind **5 Aufgaben** auf den **Seiten 1 bis 5** !

Zum Bestehen ist die Hälfte der Punkte hinreichend. Damit die Klausur als bestanden gewertet werden kann, muss das Deckblatt von der Kandidatin bzw. dem Kandidaten unterschrieben sein!

Nur von der Aufsicht auszufüllen:

Hörsaal verlassen von .... bis ....

Vorzeitig abgegeben um ....

Besondere Bemerkungen:

Anleitung: (30 Punkte)

Bei den Multiple-Choice-Aufgaben (MC) gibt es immer **exakt eine** korrekte Antwort. MC-Fragen mit mehreren richtigen Antworten sind nicht vorgesehen. Eine Aufgabe, bei der keine Antwort gegeben wurde, gilt als nicht bearbeitet und wird mit 0 Punkten gewertet. Ebenso werden bei mehreren Antworten auf eine Aufgabe automatisch 0 Punkte vergeben, selbst wenn die richtige Lösung genannt wurde. Sollten Sie im MC-Teil versehentlich ein Kreuz an die falsche Stelle machen, füllen Sie das betreffende Kästchen vollständig aus (☒ → ■) um dies deutlich zu machen. **Antworten in grün, rot oder mit Bleistift werden nicht gewertet!**

Falls nicht anders angegeben, gehen Sie bei der Beantwortung der Fragen davon aus, dass Matlab sich vorher in seinem initialen Zustand befand, d.h., der Variablenspeicher ist als leer anzunehmen und reservierte Bezeichner haben ihre ursprüngliche Bedeutung.

Verwenden Sie bei den Lückenaufgaben Matlab-Notation, wenn die Antwort Vektoren oder Matrizen sind. Beispiele: Schreiben Sie `[4; -1]` für den Vektor $\begin{pmatrix} 4 \\ -1 \end{pmatrix}$, `[3, 1, 2]` für den Vektor $(3, 1, 2)$ und `[1 2; 3 4]` für die Matrix $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$.

Bei allen Aufgaben gibt es **keine negativen Punkte** bei falscher Antwort.

---

1. Beantworten Sie die folgenden Fragen. | 4 Pkte.

(a) Sei `A` die Matrix, die vom Befehl

```
>> A = [1 6 5; 3 2 7; 0 1 0];
```

angelegt wird. Kreuzen Sie den korrekten Befehl an, um die Submatrix `B = [6 5; 2 7; 1 0]` aus `A` zu extrahieren:

- ☐ `B = A(1:2,:)`
- ☐ `B = A(1:end,1:end-1)`
- ☐ `B = A(:,2:3)`
- ☐ `B = A(end-1:end,1:2)`

(b) Sei `x=[1 9 3 5 0 8]`. Geben Sie an, welcher Befehl das gleiche Ergebnis wie

```
>> v = (x>2 & x<7)
```

ergibt.

- ☐ `v = find(x>2 & x<7)`
- ☐ `v = any(x>2) + any(x<7)`
- ☐ `v = all(x>2 & x<7)`
- ☐ `v = ~(x<=2 | x>=7)`

Seitenpunkte: | 4 Pkte.

(c) Wir möchten die 10×10 Matrix

$$A = \begin{pmatrix} 4 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 4 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 4 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 4 & -2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 4 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2 & 4 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 & 4 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2 & 4 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 4 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 4 \end{pmatrix}$$

im *dünnbesetzten* Format erstellen und die Lösung des linearen Systems $Ax = b$ mit $b \in \mathbb{R}^{10}$ berechnen. Nehmen Sie an, dass die rechte Seite `b=[3 2 2 2 2 2 2 2 2 3]'` schon als Variable gespeichert ist. Kreuzen Sie den korrekten Befehl für die Zuweisung von `A` und die Lösung des Systems an.

- ☐ `A = zeros(10) + diag(4*ones(10,1)) + diag(-2*ones(9,1),1) + diag(-2*ones(9,1),-1);`
  `x = A\b;`
- ☐ `e = ones(10,1); A = spdiags([-2*e 4*e -2*e],-1:1,10,10); x = A\b';`
- ☐ `e = ones(10,1); A = spdiags([-2*e 4*e -2*e],-1:1,10,10); x = A\b;`
- ☐ `B = zeros(10) + diag(4*ones(10,1)) + diag(-2*ones(9,1),1) + diag(-2*ones(9,1),-1);`
  `A = sparse(B); x = A\b';`

(d) Sei eine Struktur (Datentyp `struct`) `Student` gegeben. Wir wollen dem Feld `Matrikelnummer` dieser Struktur den Wert 136897025 zuweisen. Kreuzen Sie den korrekten Befehl an.

- ☐ `Student.Matrikelnummer = 136897025;`
- ☐ `Student[Matrikelnummer] = 136897025;`
- ☐ `Student(Matrikelnummer) = 136897025;`
- ☐ `Matrikelnummer.Student = 136897025;`

2. Gegeben sei die `for`-Schleife:

4 Pkte.

```
s = 0;
n = round(rand(1,100)*100);
for i=1:length(n)
    if(n(i)<50)
        s = s + n(i);
    end
end
```

In der zweiten Zeile erzeugt der Befehl `n = round(rand(1,100)*100)` hundert ganzzahlige Zufallszahlen zwischen 0 und 100.

Schreiben Sie den obigen Codeteil um, ohne Schleifen und bedingte Ausführungen zu benutzen.

Seitenpunkte:

4 Pkte.

3. Die folgenden Befehle geben nicht das gewünschte Ergebnis aus. Sie geben entweder eine Fehlermeldung oder ein falsches Ergebnis aus. Geben Sie die korrekten Befehle an, die die gewünschten Ergebnissen ergeben.

3 Pkte.

1. Die Vektoren `x=[1 2 3]';` und `y=[2 7 9]';` sind gegeben. Wir wollen den Vektor `z=[x(1)*y(1),x(2)*y(2),x(3)*y(3)]'` ohne Schleifen berechnen:

   Falscher Befehl: `z = x*y;`

   Korrekter Befehl: ____________________

2. Sei `A = gallery('poisson',10)` eine 100×100 dünnbesetzte Matrix. Wir wollen `A` als eine 100×100 vollbesetze Matrix ausgeben.

   Falscher Befehl: `full(A(:))`

   Korrekter Befehl: ____________________

3. Sei `A = [-1 5 9; 8 2 5; 1 -8 -3]` eine Matrix. Wir wollen alle positiven Einträge von `A` summieren.

   Falscher Befehl: `sum(A>0);`

   Korrekter Befehl: ____________________

4. Sei `f` ein Handle zu einer Funktion $f : \mathbb{R}^2 \to \mathbb{R}$ (z.B. `f = @(x,y) -exp(-x.^2 - y.^2);`). Nehmen Sie an, dass `f` schon vektorisiert ist. Schreiben Sie eine Matlab-Funktion

9 Pkte.

```
function cut_surfmod(f,tol)
```

die das Funktionengebirge von $|f(x, y)|$ (Absolutwert von f) für $x \in [-1, 1]$ und $y \in [-1, 1]$ plottet. Dabei sollen die Werte von $|f(x, y)|$ (an den Stellen wo $|f(x, y)|$ evaluiert wird), die kleiner als eine Toleranz `tol` sind, durch Null ersetzt werden. Die Art des Plots ist nicht wichtig, solange er ein drei dimensionales Bild der Funktion $f : \mathbb{R}^2 \to \mathbb{R}$ ergibt.

Die Matlab-Funktion `cut_surfmod` soll die folgenden Überprüfungen haben:

Seitenpunkte: 12 Pkte.

- sie soll eine Fehlermeldung liefern, wenn sie mit keinen Eingaben aufgerufen wird;
- wenn nur die Eingabe `f` übergeben wird, soll `tol=0` gesetzt werden;
- sie soll eine Fehlermeldung liefern, wenn die Eingabe für `tol` eine negative Zahl ist.

Sei `f = @(x,y) -exp(-x.^2 - y.^2)` gegeben. Rufen Sie die Funktion `cut_surfmod` für dieses `f` und eine Toleranz von 0.1 auf.

5. Das Fixpunktverfahren ist ein iterativer Algorithmus zur näherungsweisen Bestimmung von Lösungen einer Gleichung $x = f(x)$, wobei $f : \mathbb{R} \to \mathbb{R}$ eine geeignete Funktion ist. Sei ein Startwert x_0 gegeben. Für $n \geq 1$ wird in jeder Iteration des Fixpunktverfahrens der neue Wert x_{n+1} als

10 Pkte.

$$x_{n+1} = f(x_n)$$

berechnet. Die Iterationen werden bis zum Erreichen einer Toleranz oder einer maximalen Iterationsanzahl ausgeführt.

Schreiben Sie eine Matlab-Funktion

Seitenpunkte: 10 Pkte.

```
function x = fixpunkt(f,x0,tol,maxiter)
```

für die Bestimmung der Lösung von $x = f(x)$. Die Funktion f wird durch das Funktionhandle `f` als Eingabeargument übergeben. Die Eingabe `x0` ist der Startwert x_0. Das Eingabeargument `tol` ist die erwünschte Toleranz (als $|x_{n+1} - x_n|$ in jeder Iteration gemessen) und `maxiter` ist die maximale Iterationsanzahl. Das Ausgabeargument `x` ist die Approximation der Lösung von $x = f(x)$. Die Abbruchbedingung soll das Erreichen der Toleranz oder der maximalen Iterationsanzahl sein.

Geben Sie die Matlab-Befehle an, um die Nullstelle der Gleichung

$$\sqrt{x} + \sqrt[3]{x} - 2x + 1 = 0$$

mit Hilfe der Funktion `fixpunkt` zu berechnen. Wenn Sie die Funktion `fixpunkt` aufrufen, benutzen Sie $x_0 = 3/2$ als Startwert, `1e-8` als Toleranz und `1000` als maximale Iterationsanzahl.

Tipp: Versuchen Sie, die obige Gleichung als $x = f(x)$ für eine geeignete Funktion f umzuschreiben und diese in Matlab einzugeben.

Seitenpunkte: 0 Pkte.