



4.2. Übung. Einführung in die Programmierung (MA 8003)

Dieses Probetestat ist als Aufgabensammlung zu verstehen, und soll Ihnen dabei helfen Ihren Kenntnisstand zu überprüfen. Versuchen Sie also möglichst eigenständig und ohne Zuhilfenahme von MATLAB oder der Folien zu arbeiten. Bitte beachten Sie, dass sich Art, Schwerpunkt und Umfang der Klausur unterscheiden können. Relevant ist alles, was in der Vorlesung und den begleitenden Tutorien behandelt wurde.

Anleitung: (33 Punkte)

Bei den Multiple-Choice-Aufgaben (MC), sind mehrere richtige Antworten möglich. Bei (MC), wenn mehrere Antworten korrekt sind und nur ein Teil davon angekreuzt wird, werden 0.5 Punkte berechnet; wenn eine der falschen Antworten angekreuzt ist, werden 0 Punkte berechnet. 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 nicht (MC) 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 ($\boxtimes \rightarrow \blacksquare$) 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 die Matlab-Notation. Beispiele: Schreiben Sie $[4; -1]$ für den Vektor $\begin{pmatrix} 4 \\ -1 \end{pmatrix}$ und $[3, 1, 2]$ für den Vektor $(3, 1, 2)$.

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

1. Beantworten Sie die folgenden Fragen. Für jede Frage, ist mehr als eine korrekte Antwort möglich.

4 Pkte.

- (a) Sei $a \neq 0$. Wir haben den folgenden Matlab-Befehl:

```
f_skal = @(f,x,a) f(x/a);
```

Welche Befehle sind korrekt, um `f_skal` mit `f = @(x) sin(x).*cos(x)`, `a = 3` und `x = 1/2` anzuwenden?

- `f_skal(f,1/2,3)`
 `f_skal(@(x)sin(x).*cos(x),1/2,3)`
 `f_skal(@f,1/2,3)`
 `f_skal(sin(x).*cos(x),1/2,3)`

- (b) Sei `Artikel` ein *struct array* dessen Einträge jeweils structs mit Feld `Preis` sind. Wir wollen dem Feld `Preis` des zweiten Eintrages von `Artikel` den Wert 100 zuweisen. Kreuzen Sie den/die korrekten Befehl/e an.

Seitenpunkte:

4 Pkte.

- Artikel(2).Preis = 100
- Artikel(2)(Preis) = 100
- Artikel(Preis(2)) = 100
- Artikel.Preis(2) = 100

(c) Sei

```
>> v = ones(100,1);
```

gegeben. Geben Sie an, welche Befehle das gleiche Ergebnis ergeben wie

```
>> A = spdiags([v v+1],[0 1],100,100);
```

- I = (1:100)'; A = sparse([I;I(2:end)], [I;I(2:end)-1], [2*v;v(2:end)]);
- A = spdiags([v 2*v v],-1:1,100,100)-spdiags(v,1,100,100);
- I = (1:100)'; A = sparse([I;I(1:end-1)], [I;I(1:end-1)+1], [v;2*v(1:end-1)]);
- A = diag(v) + 2*diag(v(1:end-1),1); A = sparse(A);

(d) Geben Sie an, welche Befehle das gleiche Ergebnis ergeben wie

```
>> A = [1 0 3; 2 9 8];
```

```
>> any(A(:)>6)
```

an?

- all(~A(:))
- all([2 4 3])
- [0 1 1]'
- [0 1 1]

2. Gegeben sei die for-Schleife:

```
x = rand(10,1);
for i=1:length(x)
    if(x(i)<0)
        x(i)=0;
    else
        x(i) = 2*x(i);
    end
end
```

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

5 Pkte.

Seitenpunkte:

5 Pkte.

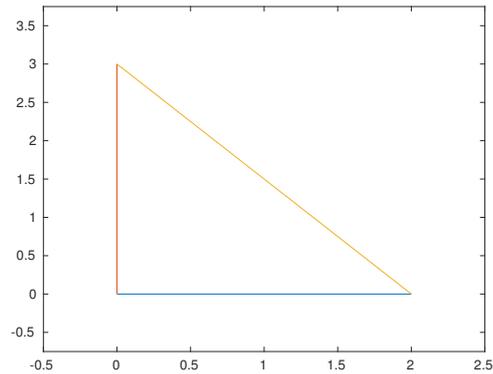


Abbildung 1: Bild zu Aufgabe 3, für $a=2$ und $b=3$.

```
x = rand(10,1);
```

```
x(x<0) = 0;
```

```
x = 2*x;
```

3. Schreiben Sie eine Matlab-Funktion

```
function x = area(a,b)
```

8 Pkte.

die die Fläche x ein rechtwinkliges Dreieck berechnet, wobei a und b die Längen der zwei Katheten sind. Diese Funktion soll die folgenden Eigenschaften haben:

- sie soll eine Fehlermeldung liefern, wenn sie mit keinen Eingaben aufgerufen wird;
- wenn nur die Eingabe a übergeben wird, soll $b = a$ gesetzt werden;
- das Dreieck muss geplottet werden, wie zum Beispiel in Abbildung 1 (die Farben der Linien und die Ränder sind nicht wichtig).

Seitenpunkte:

8 Pkte.

```
function x = area(a,b)

if(nargin<1)
    error('Mindestens eine Eingabe!')
elseif(nargin<2)
    b = a;
end

x = a*b/2;

plot([0 a],[0 0],[0 0],[0 b],[0 a],[b 0])
% oder z.B. plot([0 a 0 0],[0 0 b 0])
```

4. Geben Sie die Befehle an, die notwendig sind um das Problem 3 vom Project Euler¹ zu lösen:

8 Pkte.

The prime factors of 13195 are 5, 7, 13 and 29.

What is the largest prime factor of the number 600851475143 ?

(Übersetzung: Die Primfaktoren von 13195 sind 5, 7, 13 und 29. Welcher ist der größte Primfaktor der Zahl 600851475143 ?)

```
N = 600851475143;
p = [];
for n=1:ceil(sqrt(N))
    if(isprime(n) && rem(N,n)==0)
        p = [p n];
    end
end
max(p)

%oder N = 600851475143; max(factor(N)). Noch weitere Moeglichkeiten...
```

¹<https://projecteuler.net>

8 Pkte.

5. Das Sekantenverfahren ist ein iterativer Algorithmus um die Nullstellen einer Funktion $f : \mathbb{R} \rightarrow \mathbb{R}$ zu finden. Seien zwei Startwerte x_0, x_1 gegeben. Für $n \geq 1$, wird in jeder Iteration des Sekantenverfahrens, der neue Wert x_{n+1} wird als

8 Pkte.

$$x_{n+1} = x_n - f(x_n) \frac{x_n - x_{n-1}}{y_n - y_{n-1}}.$$

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

Schreiben Sie eine Matlab-Funktion

```
function x = sekanten(f,x0,x1,tol,maxiter)
```

für die Nullstellesuche einer Funktion f . Die Funktion f wird durch das Funktionshandle f als Argument eingegeben. Die Eingaben x_0 und x_1 sind die zwei Startwerte bzw. x_0 und x_1 . 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 Nullstelle. Die Abbruchbedingung soll das Erreichen der Toleranz oder der maximalen Iterationsanzahl sein.

```
function x = sekanten(f,x0,x1,tol,maxiter)

iter = 0;
res = abs(x1-x0);
xolder = x0;
xold = x1;

while(res>tol && iter<maxiter)
    x = xold-f(xold)/(f(xold)-f(xolder))*(xold-xolder);
    res = abs(x-xold);
    xolder = xold;
    xold = x;
    iter = iter+1;
end

% oder mit 'while(res>tol)' und 'if(iter>=maxiter), break; end'

% oder mit 'for iter=1:maxiter' und 'if(res<tol), break; end'
% (im letzten Fall ohne 'iter = iter+1;')
```

8 Pkte.