

Klausur zur BSc-Vorlesung „Rechnergestützte Modellierung“ des WS 2011 1. Termin

Die Klausur besteht aus drei Aufgaben. In der ersten Aufgabe sind 20 Punkte zu erreichen, in der zweiten Aufgabe 5 und in der letzten 10 Punkte.

Am Ende der Klausur finden Sie zusätzlichen Platz für Nebenrechnungen. Sollte Ihnen bei der Bearbeitung von Aufgabe 3 ein Fehler unterlaufen, finden Sie am Ende der Klausur zusätzlich die Möglichkeit Ihre Lösung nochmal gut leserlich zu präsentieren. Vergewissern Sie sich bitte bei der Abgabe, dass Sie alle ungültigen Lösungsversuche durchgestrichen haben!

Füllen Sie bitte zunächst die folgende Tabelle aus. Schreiben Sie bitte anschließend Ihren Namen auf jede Seite dieser Klausur!

Vorname	
Nachname	
Matrikelnummer	

Viel Erfolg!

Die nachfolgende Tabelle ist nur für die interne Verwendung gedacht! Nehmen Sie hier keine Eintragungen vor!

Aufgabe 1	Aufgabe 2		Aufgabe 3	Summe	Note
	a	b - d			

Name: _____

Aufgabe 1 (20 Punkte)

Untersuchen Sie das folgende Programm auf Fehler. Tragen Sie anschließend den jeweils korrekten Befehl **vollständig** mit der zugehörigen Zeilenzahl in der nachfolgenden Tabelle ein.

Hinweis: Die Zeilenzahlen sind nicht Bestandteil des Programms und dienen nur der Übersichtlichkeit! **Es ist nicht gestattet, die Reihenfolge der Befehle zu verändern, weitere Befehle hinzuzufügen oder vorhandene Befehle vollständig zu entfernen!**

```
01 function Oligopolmarkt[sw]
02 Einlesen(4);
03 [GG,fval,exitflag] = tfsolve(&RK,[sw(1).sw(2)]);
04 if exitflag=1
05     clc;
06     fprintf('Im Gleichgewicht beträgt die Produktion des ersten');
07     fprintf(' Oligopolisten:\n\nq1 = %0.2f';GG(1));
08     fprintf('\n\nund jene des zweiten Oligopolisten ist \n\n ');
09     fprintf('q2 = %0.2f\n',GG(2));
10 else
11     fprintf('Es konnte keine Lösung gefunden werden! \n');
12     fprintf('Da es sich um eine nichtlineares Gleichungssystem handelt,');
13     fprintf('\nmacht es Sinn, einen anderen Startwert zu probieren!\n');
14 end;
15 end
16
17 function OM==RK(qq)
18 global c1,c2;
19 Q=qq(1)+qq(2);
20 R1=N[Q]-N(Q)*qq(1)/Q-c1;
21 R2=N(Q)-N(Q)*qq(2)/Q-c2;
22 RK=[R1;R2];
23 end
24
25 function P=N(Q)
26 global a;E;
27 P=a*E/qq;
28 end
29
30 function Einlesen
31 global a E c1 c2;
32 a1=input('Bitte geben Sie die Konsumelastizität des Gutes ein> ','s');
33 E=input("Bitte geben Sie das Einkommen des Konsumenten ein> ");
34 c1=input('Bitte geben Sie die Stückkosten des Oligopolisten 1 ein> ');
35 c2=fprintf('Bitte geben Sie die Stückkosten des Oligopolisten 2 ein> ');
36 endf
```


Name: _____

Aufgabe 2 (5 Punkte)

Nehmen Sie an, Sie hätten die folgende Funktion programmiert und in Ihrem Arbeitsverzeichnis gespeichert:

```
01 function z=mach_was(x)
02 m=size(x,1);
03 n=size(x,2);
04 if m>1 && n>1
05     fprintf('Bitte andere Übergabe wählen');
06 else
07     for i=1:m-1
08         k=i;
09         while k>0 && x(k+1)>x(k)
10             p=x(k);
11             x(k)=x(k+1);
12             x(k+1)=p;
13             k=k-1;
14         end;
15     end;
16     z=x;
17 end;
18 end
```

a) Beschreiben in maximal 3 Sätzen den Sinn der Funktion. (2P)

b) Was wäre die Bildschirmausgabe, wenn man die folgenden Befehle ausführen würde: (1P)

```
Vektor=[1;5;4];
y=mach_was(Vektor)
```

Name: _____

- Platz für Nebenrechnungen -

A large, empty rectangular box with a thin black border, occupying most of the page. It is intended for students to write down their calculations or work.

Name: _____

- Fortsetzung Aufgabe 2 -

```
01 function z=mach_was(x)
02 m=size(x,1);
03 n=size(x,2);
04 if m>1 && n>1
05     fprintf('Bitte andere Übergabe wählen');
06 else
07     for i=1:m-1
08         k=i;
09         while k>0 && x(k+1)>x(k)
10             p=x(k);
11             x(k)=x(k+1);
12             x(k+1)=p;
13             k=k-1;
14         end;
15     end;
16     z=x;
17 end;
18 end
```

c) Was wäre die Bildschirmausgabe, wenn man die folgenden Befehle ausführen würde: (1P)

```
Vektor=[1 5 4];
y=mach_was(Vektor)
```

d) Was wäre die Bildschirmausgabe, wenn man die folgenden Befehle ausführen würde: (1P)

```
Vektor=[1 5;4 7];
y=mach_was(Vektor)
```

Name: _____

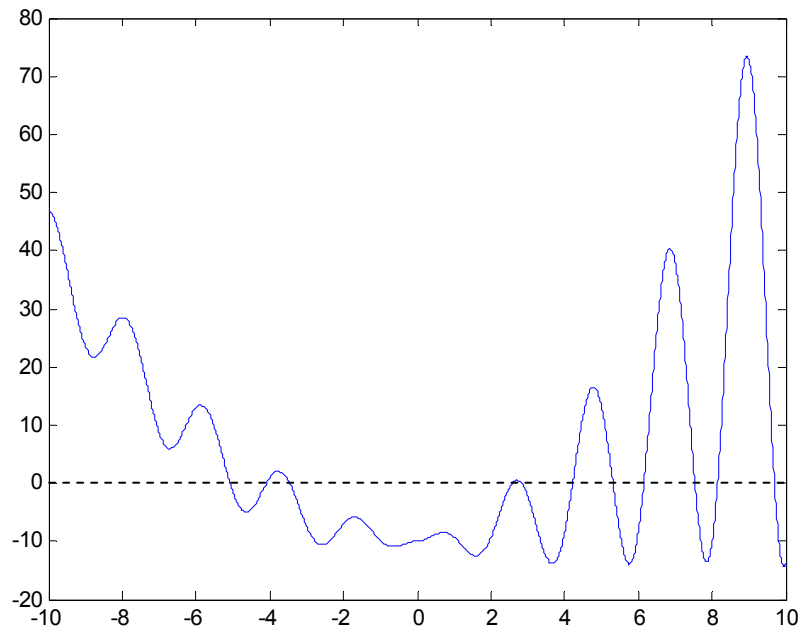
- Platz für Nebenrechnungen -

A large, empty rectangular box with a thin black border, occupying most of the page. It is intended for students to write down their calculations or work.

Name: _____

Aufgabe 3 (10 Punkte)

Das folgende Programm soll die Nullstellen einer sehr komplizierten Funktion:



bestimmen. Wie Sie aus der Vorlesung wissen, ist MatLab in der Lage für Sie diese Nullstellenbestimmung zu übernehmen. Allerdings geben Ihnen die in der Vorlesung vorgestellten Befehle immer **nur eine** Nullstelle zurück.

Aufgabe: Um mehrere Nullstellen bestimmen zu können, schreiben Sie eine Funktion mit einer Übergabe- und einer Rückgabevariablen. Die Übergabevariable nennen Sie "Start" und die Rückgabevariable "Erg". Die Variable "Start" soll für das Übergeben eines Vektors verwendet werden, mit dem der User mehrere Startwerte in Vektorform übergeben kann. Für jeden dieser derart übergebenen Startwerte lassen Sie dann MatLab die Bestimmung der Nullstellen mittels der Ihnen bekannten Funktion vornehmen. Die auf diese Weise gefundenen Nullstellen geben Sie, gemeinsam mit dem zugehörigen Startwert, dem Nutzer über die Rückgabevariable "Erg" wieder zurück. Achten Sie dabei darauf dem Nutzer nur dann eine Lösung zurück zu geben, wenn der Algorithmus auch wirklich eine Nullstelle finden konnte. Bestimmen Sie dabei die Nullstellen der bereits völlig fertig programmierten Funktion $f(x)$.

Bei der Bearbeitung der Aufgabe ist es Ihnen nicht erlaubt den vorgegeben Code zu modifizieren! Nutzen Sie ausschließlich die vorgegebenen Stellen, um eigenen Code zu ergänzen! Das fertige Programm soll für einen Vektor beliebigen Formates funktionieren!

Wichtig: Sie müssen nicht zwingend alle möglichen Stellen zur Bearbeitung dieser Aufgabe verwenden. Vielmehr ist es durch eine effiziente Programmierung möglich auf einige Platzhalter zu verzichten! Sollten Sie mehr Platz benötigen, können Sie selbstverständlich mehrere Befehle in eine Reihe schreiben!

Name: _____

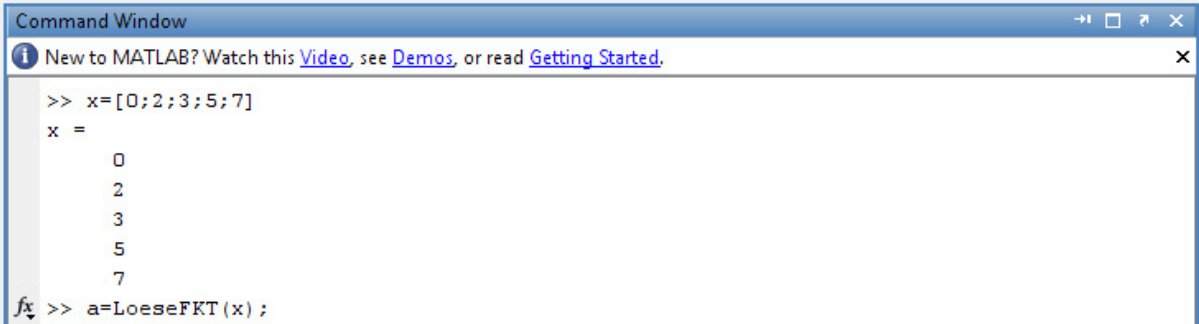
Ein Zahlenbeispiel:

Der Nutzer Ihrer Funktion möchte die Startwerte 0, 2, 3, 5 und 7 verwenden.

Um diese Beispielwerte Ihrer Funktion zu übergeben, stellen Sie diese als den folgenden Vektor dar:

$$\begin{bmatrix} 0 \\ 2 \\ 3 \\ 5 \\ 7 \end{bmatrix}$$

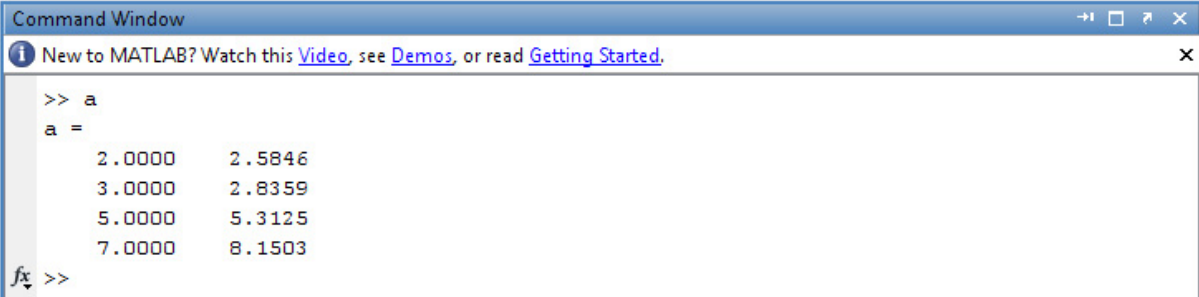
Diesen Vektor speichern Sie in der Variable x und übergeben diese anschließend der von Ihnen programmierten Funktion *LoeseFKT*. Betrachten Sie dazu folgende Abbildung:



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> x=[0;2;3;5;7]
x =
     0
     2
     3
     5
     7
fx >> a=LoeseFKT(x);
```

Ihre Funktion *LoeseFKT* wird nun der Reihe nach die den Startwerten zugehörigen Nullstellen ermitteln und in einer Matrix speichern. Anschließend wird Sie, wie im vorprogrammierten Programmteil ersichtlich, den Bildschirm löschen, und die oben beschriebene Matrix zurückgeben.

Um sich diese Rückgabematrix anzusehen, tätigen Sie die folgenden Eingaben in dem Command-Fenster:



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> a
a =
     2.0000     2.5846
     3.0000     2.8359
     5.0000     5.3125
     7.0000     8.1503
fx >>
```

Auf der linken Seite der Rückgabematrix sehen Sie die von Ihnen vorgegebenen Startwerte, auf der rechten die von MatLab bestimmten Nullstellen. Berücksichtigen Sie dabei, dass der Startwert Null hier nicht mehr angezeigt wird. Dies liegt daran, dass bei diesem Startwert keine sinnvolle Lösung gefunden werden konnte!

Viel Erfolg!

Name: _____

```
function _____ = LoeseFKT _____
```

```
_____
```

```
_____
```

```
_____
```

```
for _____
```

```
_____
```

```
_____
```

```
if exitflag _____
```

```
_____
```

```
_____
```

```
_____
```

```
_____
```

```
end
```

```
_____
```

```
_____
```

```
_____
```

```
clc;
```

```
end
```

```
function funk = g(x)
```

```
funk=0.5*x^2+2*abs(x)*exp(x/10)*sin(3*x)-10;
```

```
end
```

Name: _____

- Platz für erneute Bearbeitung der Aufgabe 3 -

```
function _____ = LoeseFKT _____  
  
_____  
  
_____  
  
for _____  
_____  
_____  
  
    if exitflag _____  
_____  
_____  
_____  
_____  
_____  
  
end  
  
_____  
  
_____  
  
_____  
  
clc;  
  
end  
  
function funk = g(x)  
funk=0.5*x^2+2*abs(x)*exp(x/10)*sin(3*x)-10;  
  
end
```

Name: _____

- Platz für erneute Bearbeitung der Aufgabe 3 -

```
function _____ = LoeseFKT _____  
  
_____  
  
_____  
  
for _____  
_____  
_____  
  
    if exitflag _____  
_____  
_____  
_____  
_____  
_____  
  
end  
  
_____  
  
_____  
  
_____  
  
clc;  
  
end  
  
function funk = g(x)  
funk=0.5*x^2+2*abs(x)*exp(x/10)*sin(3*x)-10;  
  
end
```

Name: _____

- Platz für Nebenrechnungen –

A large, empty rectangular box with a thin black border, occupying most of the page. It is intended for students to perform calculations or show their work.

Name: _____

- Platz für Nebenrechnungen -

A large, empty rectangular box with a thin black border, occupying most of the page. It is intended for students to perform calculations or show their work.