U N I V E R S I T Ä T H A M B U R G

INSTITUT FÜR
WACHSTUM UND KONJUNKTUR

Pascal Terveer

Institut für Wachstum und Konjunktur
Von-Melle-Park 5, D-20146 Hamburg

Fernsprecher: (040) 4 28 38 – 20 83
Fax-Nr: (040) 4 28 38 – 6313
E-Mail: Pascal.Terveer@wiso.uni-hamburg.de

Klausur zur BSc-Vorlesung „Rechnergestützte Modellierung“ des WS 2012
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 <u>interne Verwendung</u> 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 [ Erg ] = Tu_Was
02  clc;
03  Vari=['a' 'b' 'c 'd'];
04  func='ax^b+c*cos(dx)';
05  [a,b,c,d]=Hole(Func,Vari,Erg);
06  [x,fval,exitfläg]=tminsearch(@FNT(a,b,c,d,x),0);
07  if exitflag=0
08      fprintf('\nDas gefundene x lautet: %f0.2\n',x);
09      Erg==x;
10  else
11      fprintf('\nEs konnte keine Lösung gefunden werden!');
12      Erg=[];
13  end
14  xK=-10;0.01;15;
15  yK=FKT(a,b,c,d,x);
16  surf(xK,yK);
17  end
18
19  function [Erg]=FKT(a,b,c,d,x)
20  a*x^b+c*cos(d*x)= Erg;
21  end
22
23  function (a,b,c,d)=Hole[g,An]
24  fprintf('Bitte geben Sie die Paramater der Funktion %s' \n,g);
25  fprintf('Bitte der Reihe nach ein!\n\n');
26  Werte=0;
27  i=1;
28  while i<=Max(size(An))
29      fprintf('Bitte geben Sie %s ,An(i));
30      Werte[i]=putin('ein> ');
31      i=i+1;
32  end
33  a=Werte(1);
34  b=Werte(2);
35  c=Werte(3);
36  d=Werte(4);
37  end
38
```

Name:_____

Fehlerliste	
Zeile	**Ihre Korrektur**

Name:_____

Aufgabe 2 (5 Punkte)

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

```
01  function [a]=Finde_raus(a)
02  [r,c]=size(a);
03  a(r+1,:)=sum(a);
04  i=1;
05  b=1;
06  pos=r+1;
07  while i<c
08  b=b+1;
09  if a(pos,b)>a(pos,i)
10      y=a(:,i);
11      a(:,i)=a(:,b);
12      a(:,b)=y;
13  end
14  if c==b
15      i=i+1;
16      b=i;
17  end
18  end
19  a=a(1:r,:);
20  end
```

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

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

```
Vektor=[7 3; 5 -8];
y=Finde_raus(Vektor)
```

Name:_____

- Platz für Nebenrechnungen -

Name:_____

- Fortsetzung Aufgabe 2 -

```
01  function [a]=Finde_raus(a)
02  [r,c]=size(a);
03  a(r+1,:)=sum(a);
04  i=1;
05  b=1;
06  pos=r+1;
07  while i<c
08  b=b+1;
09  if a(pos,b)>a(pos,i)
10      y=a(:,i);
11      a(:,i)=a(:,b);
12      a(:,b)=y;
13  end
14  if c==b
15      i=i+1;
16      b=i;
17  end
18  end
19  a=a(1:r,:);
20  end
```

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

```
Vektor=[-3 22;8 -5];
y=Finde_raus(Vektor)
```

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

```
Vektor=[5 2 1];
y=Finde_raus(Vektor)
```

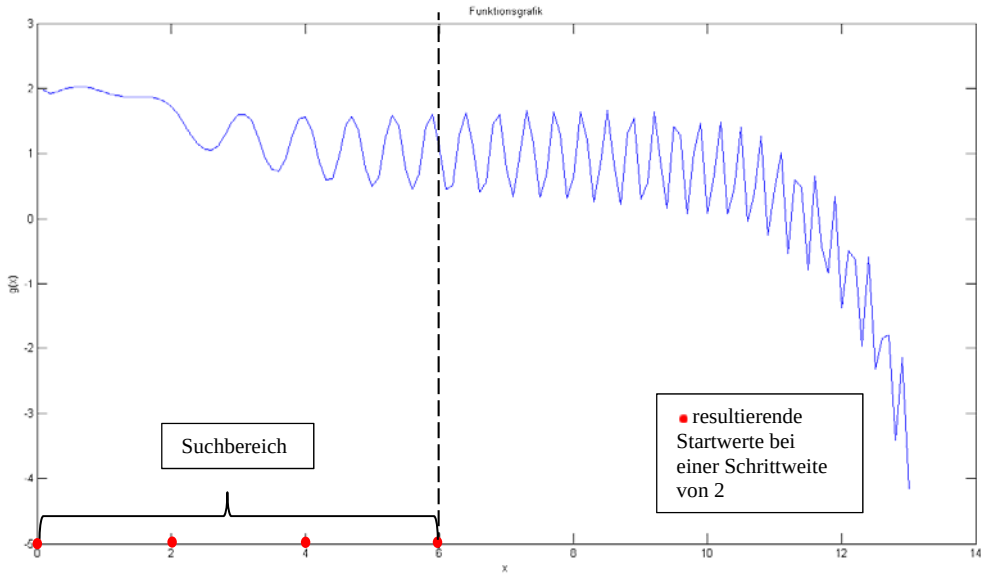
Name:_____

- Platz für Nebenrechnungen -

Name:_____

Aufgabe 3 (10 Punkte)

Das folgende Programm soll das globale Maximum einer sehr komplizierten Funktion:



bestimmen. Wie Sie aus der Vorlesung wissen, ist MatLab in der Lage für Sie die Bestimmung von Minima **im Bereich des Startwertes** zu übernehmen. Ob Sie dabei das globale oder ein lokales Minimum / Maximum finden, hängt von dem von Ihnen vorgegebenen Startwert ab.

Aufgabe: Um das globale Maximum bestimmen zu können, schreiben Sie eine Funktion mit drei Übergabe- und einer Rückgabevariablen. Die Übergabevariablen nennen Sie "lb", "ub" und "step". Der Rückgabevariable geben Sie den Namen "Lsg". Die Übergabevariablen nutzen Sie für die Vorgabe des Suchbereiches. "lb" repräsentiert dabei die untere Grenze des Suchbereiches, "ub" die obere. Nutzen Sie "step" um den Abstand der „Suchpunkte“ innerhalb des Suchbereiches vorzugeben. Verwenden Sie jeden dieser Suchpunkte als Startwert für die Bestimmung des Maximums mittels der Ihnen bekannten MatLab Funktion. Die auf diese Weise gefundenen Maxima vergleichen Sie und geben dem Nutzer anschließend die x- und y-Koordinaten des Maximums mit dem größten Funktionswert über die Rückgabevariable "Lsg" zurück. Achten Sie dabei darauf, dem Nutzer nur dann eine Lösung zurück zu geben, wenn der Algorithmus auch wirklich ein Maximum finden konnte. Ist dies nicht der Fall, geben Sie leere Mengen zurück. Bestimmen Sie dazu die **Minima** der bereits völlig fertig programmierten Funktion g(x).

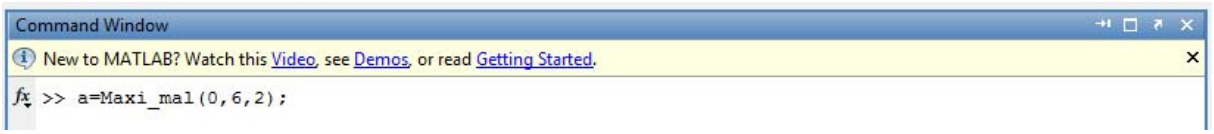
Bei der Bearbeitung der Aufgabe ist es Ihnen nicht erlaubt, den vorgegeben Code zu modifizieren! Nutzen Sie ausschließlich die vorgegebenen Stellen, um eigenen Code zu ergänzen! Das fertige Programm soll für beliebige Grenzen und Schrittweiten funktionieren!

Wichtig: Sie müssen nicht zwingend alle Zeilen 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 Zeile schreiben!

Name:_____

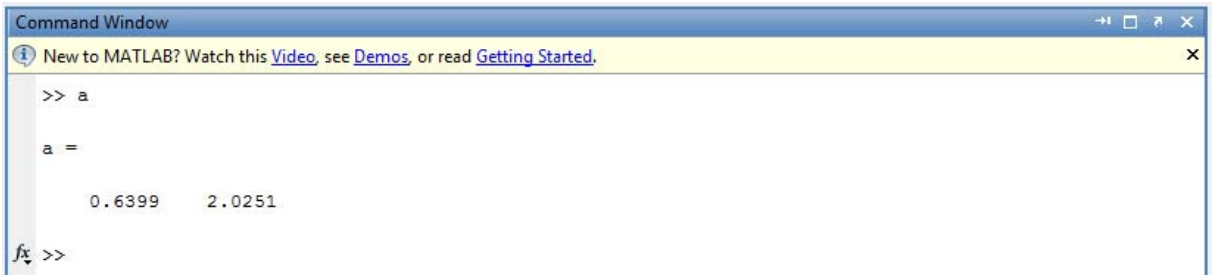
Ein Zahlenbeispiel:

Der Nutzer Ihrer Funktion möchte den in der Grafik gezeigten Suchbereich mit der dort verwendeten Schrittweite von 2 verwenden. Daher ruft er die Funktion über den Aufruf *Maxi_mal(0,6,2)* auf:



Ihre Funktion *Maxi_mal* wird nun zunächst die untere Grenze als Startwert für die Maximierung verwenden und sich die gefunden x- und y-Koordinaten merken. Da der User eine Schrittweite von zwei vorgegeben hat, wird sie die Maximierung anschließend für die Startwerte von 2, 4 und 6 durchführen. Dabei wird sie in jedem Schritt überprüfen, ob das aktuelle Maximum bei einem größeren y-Wert liegt als das Vorherige. Ist dies der Fall, wird sie sich die x- und y-Koordinate des neu gefundenen Maximums merken, andernfalls die Vorgängerwerte beibehalten. Sind alle Startwerte im vorgegebenen Bereich untersucht, wird sie, wie im vorprogrammierten Programmteil ersichtlich, den Bildschirm löschen und die Koordinaten des Maximums mit dem größten y-Wert zurückgeben.

Um sich die gefundenen x- und y-Koordinate anzusehen, tätigen Sie die folgenden Eingaben in dem Command-Fenster:



Auf der linken Seite des Rückgabevektors sehen Sie die x-Koordinate des gefundenen Maximums, auf der rechten den zugehörigen Funktionswert (also die y-Koordinate).

Hinweis: Eine Verwendung des Befehls *f(t)minbnd* ist hier nicht zielführend, da in dieser Aufgabe ein **globales Maximum** bestimmt werden soll und nicht der maximale Funktionswert in einem Intervall! Diese Werte können identisch sein, oftmals ist dies aber nicht der Fall. Daher ist die Verwendung von *f(t)minbnd* an dieser Stelle zwecklos!

Viel Erfolg!

Name:_____

```
function _________________ = Maxi_mal ____________________________________

_____________________________________________________________________________

_____________________________________________________________________________

_____________________________________________________________________________

for _________________________________________________________________________

    _____________________________________________________________________________

    _____________________________________________________________________________

    if exitflag _____________________________________________________________

    _____________________________________________________________________________

    _____________________________________________________________________________

    _____________________________________________________________________________

    _____________________________________________________________________________

    _____________________________________________________________________________

    _____________________________________________________________________________

    _____________________________________________________________________________

end

_____________________________________________________________________________

_____________________________________________________________________________

_____________________________________________________________________________

_____________________________________________________________________________

_____________________________________________________________________________

clc;

end



function y=g(x)
y=exp(2*x).^exp(-1.1*x)-0.4*log(x).*sin(cos(x.^2))-0.00001*exp(x);
y=-y;
end
```

Name:_____

- Platz für erneute Bearbeitung der Aufgabe 3 -

```
function _________________ = Maxi_mal ____________________________________

_____________________________________________________________________________

_____________________________________________________________________________

_____________________________________________________________________________

for _________________________________________________________________________

    _____________________________________________________________________________

    _____________________________________________________________________________

    if exitflag _____________________________________________________________

    _____________________________________________________________________________

    _____________________________________________________________________________

    _____________________________________________________________________________

    _____________________________________________________________________________

    _____________________________________________________________________________

    _____________________________________________________________________________

    _____________________________________________________________________________

end

_____________________________________________________________________________

_____________________________________________________________________________

_____________________________________________________________________________

_____________________________________________________________________________

_____________________________________________________________________________

clc;

end



function y=g(x)
y=exp(2*x).^exp(-1.1*x)-0.4*log(x).*sin(cos(x.^2))-0.00001*exp(x);
y=-y;
end
```

Name:_____

- Platz für erneute Bearbeitung der Aufgabe 3 -

```
function _________________ = Maxi_mal ____________________________________
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________
for _____________________________________________________________________
    _____________________________________________________________________
    _____________________________________________________________________
    if exitflag _____________________________________________________
    _____________________________________________________________________
    _____________________________________________________________________
    _____________________________________________________________________
    _____________________________________________________________________
    _____________________________________________________________________
    _____________________________________________________________________
    _____________________________________________________________________
end
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________
clc;
end


function y=g(x)
y=exp(2*x).^exp(-1.1*x)-0.4*log(x).*sin(cos(x.^2))-0.00001*exp(x);
y=-y;
end
```

Name:_____

- Platz für Nebenrechnungen –

Name:_____

- Platz für Nebenrechnungen -