Dipl.-Ing. Irina Ikkert, M.Eng.

Ostfalia
Hochschule für angewandte
Wissenschaften

Einführung in die Modellierung

Klausur

19. Juni 2017

Name: ____________________ Mat.-Nr.: ____________________

Vorname: ____________________

Aufgabe:	1	2	3	4	Summe
erreichbare Punkte:	12	8	25	35	80
erreichte Punkte:					
Note:					

Hinweise:

- Bearbeitungszeit: **60 Minuten**
- Erlaubte Hilfsmittel: **Formelsammlung**, **Vorlesungsfolien** auf dem **V-Laufwerk**.
- Bearbeiten Sie bitte jede Aufgabe in einem eigenen m-file und arbeiten Sie stets mit Kommentaren.
- Speichern Sie rechtzeitig vor dem Ende der Klausur alle relevanten Daten auf dem **U-Laufwerk** ab, da sonst die Gefahr eines Datenverlustes besteht.
- Melden Sie Probleme mit dem Rechner sofort der Aufsicht.
- In die Bewertung der Aufgaben fließen u.a. die Vollständigkeit, Korrektheit und Programmlesbarkeit (incl. Kommentare) ein.

Anmelden am Rechner:

Zum Anmelden am Rechner ist unbedingt der spezielle Klausur-Account für diese Klausur zu verwenden (siehe Seite 2).

Registrierung:

Öffnen Sie die Datei `U:\Bitte_ausfuellen!.txt` und tragen Sie in die vorgesehenen Zeilen Ihren Namen, Vornamen, Matrikelnummer und Anwendernamen (Zugangskennung) ein. Speichern Sie diese Datei.

Dipl.-Ing. Irina Ikkert, M.Eng.

Ostfalia
Hochschule für angewandte
Wissenschaften

Einführung in die Modellierung

Klausur

19. Juni 2017

Matrizen und Vektoren

1. Gegeben ist ein Zeilenvektor komplexer Zahlen $\boldsymbol{v1} = [\mathbf{3}+\mathbf{2}\boldsymbol{i} \quad \mathbf{7}-\mathbf{4}\boldsymbol{i} \quad \mathbf{12}]$. Erzeugen Sie einen weiteren Spaltenvektor $\boldsymbol{v2}$ aus 7 Elementen mit ganzen Zufallszahlen von 1 bis 10,

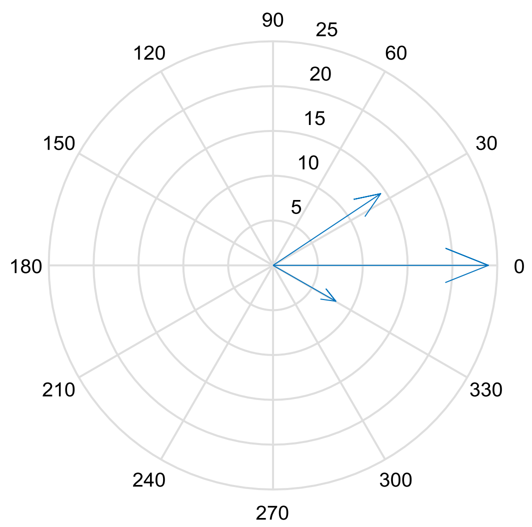
   - multiplizieren Sie die letzten drei Elemente des Zufallsvektors $\boldsymbol{v2}$ mit dem Vektor $\boldsymbol{v1}$
   - stellen Sie die Hauptdiagonalelemente dieser Multiplikation mit **compass** grafisch dar.

<u>Lösung</u>:

```
clear all
clc

v1 = [3+2i 7-4i 12]
v2 = round(10*rand(1,7))'
v3 = v2(5:7)*v1

compass([v3(1,1) v3(2,2) v3(3,3)]) % compass(diag(v3))
```



Dipl.-Ing. Irina Ikkert, M.Eng.

Einführung in die Modellierung

Klausur

19. Juni 2017

2. Berechnen Sie die Lösung $\boldsymbol{x}$ des Gleichungssystems

$$\begin{cases} 2x_1 - 5x_2 + 3x_3 = 9 \\ -x_1 + 2x_2 + 2x_3 = 13 \\ 4x_1 + x_2 + x_3 = -7 \end{cases}$$

Lösung:

```
clear all
clc

A = [2 -5 3; -1 2 2; 4 1 1]
b = [9; 13; -7]

x = inv(A)*b
```

Polynome

3. Gegeben ist ein Polynom fünften Grades:

$$p = 0{,}02x^5 - 0{,}24x^4 + 0{,}7x^3 + 0{,}4x^2 - 3x + 2$$

Stellen Sie in einem Diagrammfenster mit drei Diagrammen im Bereich von -2 bis 7:

- den Verlauf der Polynomfunktion mit ihren Nullstellen
- den Verlauf der zweimal abgeleiteten Polynomfunktion
- den Verlauf der integrierten Polynomfunktion

Ergänzen Sie Ihre Diagramme mit den Überschriften (evtl. Legende), Achsenbeschriftung und Gitternetzlinien.

Lösung:

```
clear all
clc
p = [0.02,-0.24,0.7,0.4,-3,2]   % Polynom
r = roots(p) % Nullstellen berechnen
yr = polyval(p,r)

x = -2:0.1:7; % Darstellungsbereich definieren
yp = polyval(p,x) %Polynomwerte an der Stelle x berechnen
d = polyder(p) %Koeffizienten der Ableitung

d2 = polyder(d) % Koeffizienten der zweiten Ableitung
yd = polyval(d2,x) % Polynomwerte der zweiten Ableitung an der Stelle x
```

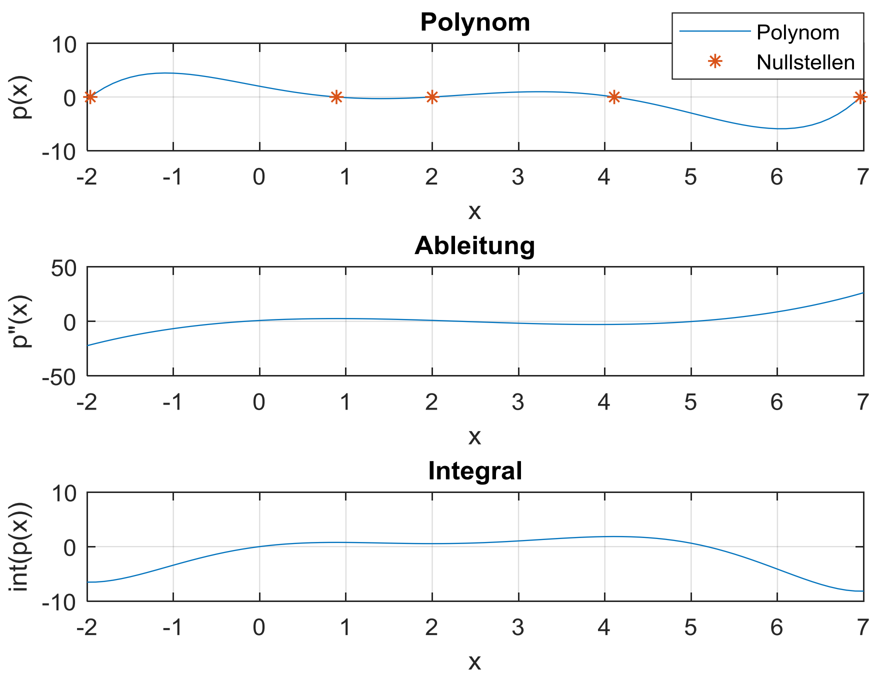
Einführung in die Modellierung

Klausur

19. Juni 2017

```
i = polyint(p) % Koeffizienten des Integrals
yi = polyval(i,x) % Polynomwerte des Integrals an der Stelle x

subplot(311) % Diagrammfenster mit 3 Diagrammen, Diagramm 1
plot(x,yp)   % Zeichnet das Polynoms
hold % im gleichen Diagramm bleiben
plot(r,yr,'*') % Zeichnet die Nullstellen
title('Polynom') % Titel
legend('Polynom','Nullstellen') % Legende
xlabel('x') % x Achse beschriften
ylabel('p(x)') % y Achse beschriften
grid on % Gitternetzlinien
subplot(312) % Diagrammfenster mit 3 Diagrammen, Diagramm 2
plot(x,yd) % Zeichnet die zweite Ableitung der Polynomfunktion
title('Ableitung')
grid on
xlabel('x')
ylabel('p"(x)')
subplot(313)
plot(x,yi) % Zeichnet die integrierte Polynomfunktion
title('Integral')
grid on
xlabel('x')
ylabel('int(p(x))')
```



Einführung in die Modellierung

Klausur

19. Juni 2017

Function Files

4. Schreiben Sie eine Funktion ***schnittpunkte***, die mögliche Schnittpunkte zweier Funktionen symbolisch berechnet und ausgibt:

   function [Xs,Ys] = schnittpunkte (f1,f2,a,b)

   mit:
   f1 – erste Funktion (function handle)
   f2 – zweite Funktion (function handle)
   a – untere Grenze des Definitionsbereichs
   b – obere Grenze des Definitionsbereichs

Die Funktion soll:

- beim Aufruf mit nur zwei Funktionen f1 und f2 deren Schnittpunkte symbolisch berechnen und ausgeben,
- beim Aufruf mit vier Parametern f1, f2, a, b sollen die beiden Funktionen mit den Schnittpunkten zusätzlich in einem gemeinsamen Diagramm im Bereich von a bis b mit plot numerisch dargestellt werden,
- für alle anderen Aufrufe soll das Programm mit einer Fehlermeldung abgebrochen werden.

Testen Sie Ihr Programm für die Funktionen $f_1 = x^3 - 4x - 5$ und $f_2 = 3x - 3$ im Bereich zwischen -3 und 3.

Speichern Sie die Funktion und den Aufruf der Funktion auf dem U-Laufwerk ab.

<u>Lösung:</u>

```
% Definition der beiden Testfunktionen
f1 = @(x) x^3-4*x-5
f2 = @(x) 3*x-3
% Definitionsbereich
a = -3
b = 3

% Aufruf der Funktion schnittpunkte
[Xs,Ys] = schnittpunkte(f1,f2,a,b)
```

Einführung in die Modellierung

Klausur

19. Juni 2017

```
% Funktion schnittpunkte
function [ Xs,Ys ] = schnittpunkte( f1,f2,a,b )

% Definition symbolischer Variablen und Funktionen
syms x

f1 = f1(x);
f2 = f2(x);

Xs = double(solve(f1 == f2)); % Schnittpunkte der beiden Funktionen
Ys = double(subs(f1,x,Xs)); % Ordinate der Schnittpunkte berechnen

if nargin == 4 % falls Aufruf mit 4 Parametern

    xn = linspace(a,b,1000); % Darstellungsbereich festlegen
    f1 = double(subs(f1,x,xn)); % Funktionswerte f1 für xn
    f2 = double(subs(f2,x,xn)); % Funktionswerte f2 für xn
    % Zeichnet die beiden Funktionen mit Schnittpunkten
    plot(xn,f1,xn,f2,Xs,Ys,'*')
    title('Schnittpunkte zweier Funktionen'); % Titel
    xlabel('x'); % x Achse beschriften
    ylabel('y'); % y Achse beschriften
    grid on % Gitternetzlinien

end
% falls Aufruf weder mit 4 noch mit 2 Parametern
if nargin ~= 4 && nargin ~= 2
    error('ungültige Anzahl der Eingabeparameter. Das Programm wird
abgebrochen'); % Beenden des Programms mit Fehlermeldung
end
end
```