

Ingenieurinformatik

Numerik für Ingenieure

Name	Vorname	Semestergruppe	Hörsaal

	Aufgabe 1	Aufgabe 2	Aufgabe 3	Aufgabe 4	Summe	

Studienbeginn vor WS13/14 (Kombinationsprüfung) **	<input type="checkbox"/>
Studienbeginn ab WS13/14 bis WS15/16 **	<input type="checkbox"/>
Studienbeginn ab SS16 (Kombinationsprüfung)	<input type="checkbox"/>
Diplomstudiengang Maschinenbau**	<input type="checkbox"/>

**** Die Prüfung ist nur dann gültig, wenn Sie die Zulassungsvoraussetzung erworben haben (erfolgreiche Teilnahme am Praktikum).**

Aufgabensteller: Dr. Reichl, Dr. Küpper und Kollegen

Bearbeitungszeit: 60 Minuten

- Hilfsmittel:**
- Taschenrechner nicht zugelassen
 - PC/Notebook nicht zugelassen
 - Sonstige eigene Hilfsmittel sind erlaubt
 - Bearbeitung mit Bleistift ist erlaubt

***** Viel Erfolg!!! *****

Aufgabe 1: (ca. 21 Punkte)

1.1. Durch Aufruf der eingebauten MATLAB-Funktion **isprime(x)** kann überprüft werden, ob x eine Primzahl ist oder nicht:

- **isprime(13)** liefert als Rückgabewert eine 1, denn 13 ist eine Primzahl,
- **isprime(14)** liefert als Rückgabewert eine 0, denn 14 ist keine Primzahl.

Programmieren Sie eine eigene MATLAB-Funktion mit dem Namen **ist_prim(x)**, die dasselbe Ergebnis liefert wie die eingebaute Funktion **isprime(x)**: Falls x eine Primzahl ist, wird 1 zurückgegeben. Sonst wird 0 zurückgegeben.

Sie dürfen ohne weitere Überprüfung davon ausgehen, dass mit dem Parameter x eine positive ganze Zahl übergeben wird (also kein Vektor und auch keine Matrix, keine Zahl mit Nachkommastellen, keine negative Zahl oder null).

Hinweise: - Eine Primzahl ist eine natürliche Zahl größer als 1,
die nur durch sich selbst und durch 1 ohne Rest teilbar ist.

- Die eingebaute Funktion **isprime(x)** darf nicht verwendet werden!

- 1.2. Schreiben Sie ein MATLAB-Skript, welches die ersten 100 Primzahlen ermittelt und hintereinander in dem Vektor **vec** speichert. Sie dürfen in Ihrem Skript die eingebaute Funktion **isprime(x)** oder die selbst erstellte Funktion **ist_prim(x)** verwenden.

```
% MATLAB-Skript zur Berechnung von 100 Primzahlen
```

```
vec = [];
```

```
% Ausgabe der 100 Primzahlen im Vektor "vec"
```

```
disp('Die ersten 100 Primzahlen sind: ')
```

```
disp(vec)
```

Aufgabe 2: (ca. 14 Punkte)

Die Variablen A und x sind mit den folgenden Werten belegt:

$$A = \begin{bmatrix} -1 & 2 \\ 2 & -1 \end{bmatrix};$$

$$x = \begin{bmatrix} 1 \\ -1 \end{bmatrix};$$

Wie lauten die Ausgaben der folgenden MATLAB-Befehle?

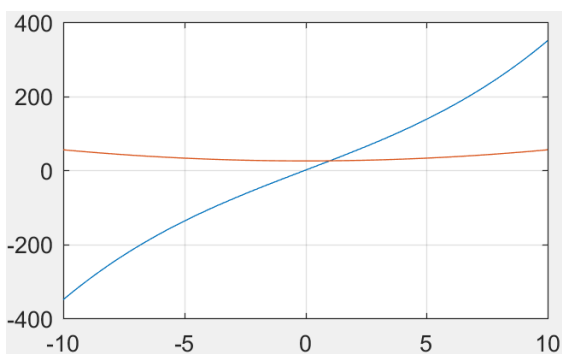
<code>disp(x.*x)</code>	(1 Pkt.)
<code>disp(A.^3)</code>	(2 Pkt.)
<code>disp(A.*A)</code>	(2 Pkt.)
<code>disp(A*A)</code>	(3 Pkt.)
<code>disp(A*x)</code>	(2 Pkt.)
<code>disp((A*x)')</code>	(1 Pkt.)
<code>disp([x; x])</code>	(1 Pkt.)
<code>disp(A + [x x])</code>	(2 Pkt.)

Aufgabe 3: (ca. 12 Punkte)

Schreiben Sie ein MATLAB-Skript, welches die folgenden Aufgaben löst:

- Der Anwender gibt die Koeffizienten a, b, c, d des Polynoms $y = ax^3 + bx^2 + cx + d$ ein.
- Das Polynom und seine 1. Ableitung werden im Bereich $x = -10 \dots +10$ grafisch ausgegeben.
- Die beiden Nullstellen der 1. Ableitung werden (wie in der Abbildung gezeigt) im Command Window ausgegeben. Achtung: Die Nullstellen können auch komplex sein.

```
disp('Polynom y = a*x^3 + b*x^2 + c*x + d eingeben')
```

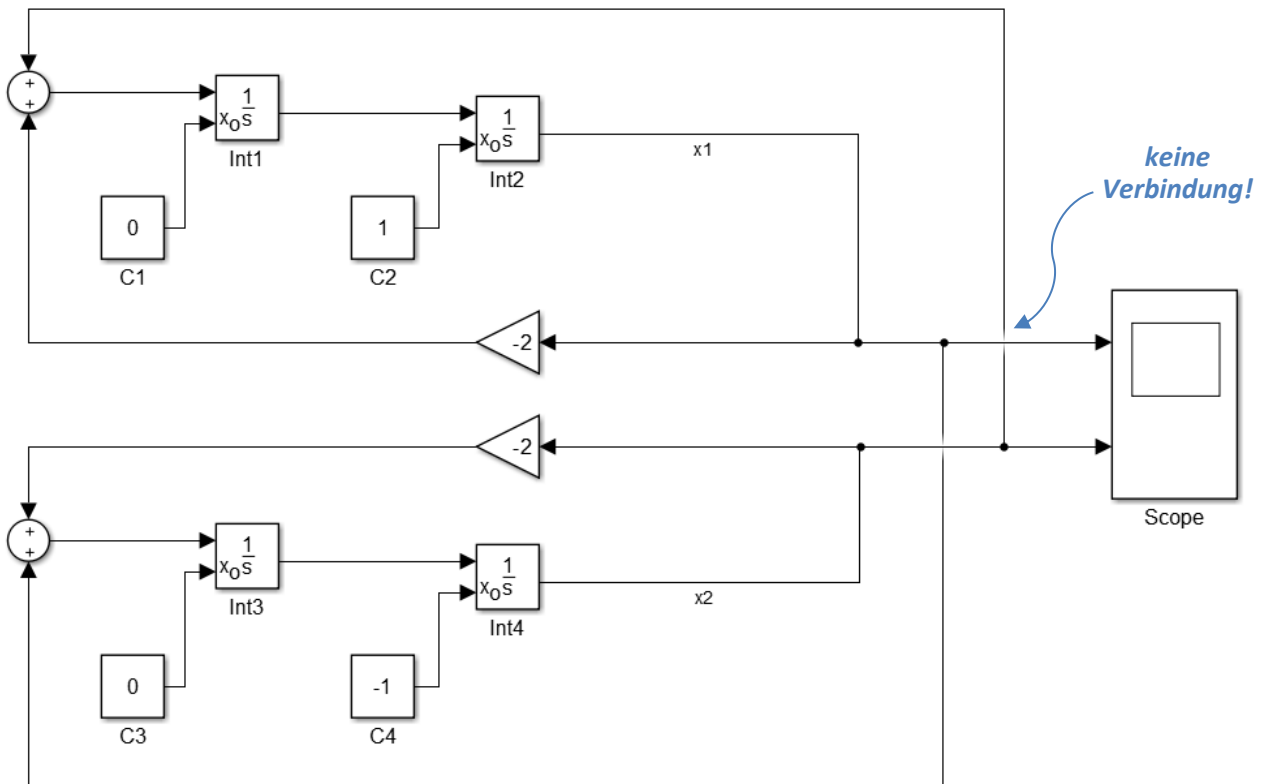


```
Command Window
Polynom y = a*x^3 + b*x^2 + c*x + d eingeben
a: 0.1
b: 0
c: 25
d: 0
} Eingabe

Nullstellen der 1. Ableitung:
0.0000 + 9.1287i
0.0000 - 9.1287i
} Ausgabe
```

Aufgabe 4: (ca. 20 Punkte)

Ein System von zwei Differentialgleichungen 2. Ordnung wird mit Simulink gelöst.



4.1. Wie lauten die beiden Differentialgleichungen, die im Simulink-Modell definiert sind?

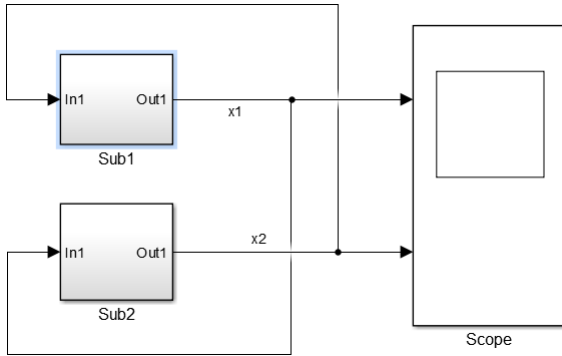
$$\ddot{x}_1 =$$

$$\ddot{x}_2 =$$

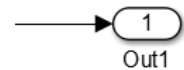
4.2. Wie lauten die Anfangsbedingungen dieser beiden Differentialgleichungen?

4.3. Das Differentialgleichungssystem soll für den Zeitraum $t = 0 \dots 25$ s numerisch gelöst werden. Wo wird dieser Zeitraum in Simulink eingestellt?

4.4. Zur Vereinfachung des Simulinkmodells werden zwei Subsysteme definiert:



Wie muss der innere Aufbau des Subsystems „Sub1“ aussehen, damit sich das Verhalten im Vergleich zum ursprünglichen System nicht verändert?



4.5. Das System von zwei Differentialgleichungen **2. Ordnung** aus Aufgabe 4.1. soll in MATLAB mittels ode45 gelöst werden. Dazu muss es zunächst in ein System von vier Differentialgleichungen **1. Ordnung** umgewandelt werden.

Wie lauten die vier Differentialgleichungen 1. Ordnung inkl. Anfangsbedingungen?

(Platz für Nebenrechnungen)