

Blatt 1: Matlab Crashkurs

MNEU

Felder

Aufgabe 1:

$$V = (1, 4, 6, 2)$$

$$W = \begin{pmatrix} 0 \\ 4 \\ -1 \\ -3 \end{pmatrix}$$

Matlab-Source:

Aufgabe 2:

$$V \cdot W$$

Matlab-Source:

$$V * W =$$

$$V' * W =$$

$$V * W' =$$

$$V' * W' =$$

Aufgabe 3:

$$W \cdot V$$

Matlab-Source:

$$W * V =$$

$$W' * V =$$

$$W * V' =$$

$$W' * V' =$$

Aufgabe 4:

$$U = (4, 5, 6, 7, 8)$$

$$W = \begin{pmatrix} 2 \\ 2.2 \\ \vdots \\ 3.2 \\ 3.4 \end{pmatrix}$$

Matlab-Source:

Aufgabe 5:

(wie oben)

Matlab-Source:

length(U)= size(U)=

length(W)= size(W)=

Aufgabe 6:

$$U = (1, 4, \underbrace{5, 6, 7, 8, 9}_{=:W}, 10)$$

Matlab-Source:

Aufgabe 7:

$$B = (1, 2), C = \begin{pmatrix} 3 \\ -1 \end{pmatrix}$$

$$A_1 = \begin{pmatrix} 1 & 2 \\ 3 & -1 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 1 & 3 \\ 2 & -1 \end{pmatrix}$$

$$A_3 = \begin{pmatrix} 3 & -1 \\ 1 & 2 \end{pmatrix}$$

Matlab-Source:

Aufgabe 8:

$$A_1 + A_2$$
$$A_3 + 4$$
$$A_1 \cdot A_2$$
$$(A_{1ij} \cdot A_{2ij})_{ij}$$

Matlab-Source:**Aufgabe 9:**

Für $i = 1, 2$, $j = 1, 4$ und $k = 1, 3$ seien die Komponenten der Matrix A gegeben durch:

$$A_{ijk} = \sum_{l=1}^k \frac{i+j}{l}$$

Weisen Sie den Komponenten

$$A_{111}, A_{241} \text{ und } A_{223}$$

den jeweiligen Wert zu.

Matlab-Source:

Matlab Skripte

Aufgabe 10:

Erzeugen Sie das Unterverzeichnis `MatlabCrashkurs`, wechseln Sie dahin und erzeugen dort die Datei `first.m`. Überzeugen Sie sich davon, dass die Datei vorhanden und im richtigen Verzeichnis zu finden ist.

Wiederholen Sie Aufgabe 7 und 8, indem Sie alle Anweisungen in die Datei `first.m` schreiben.

Matlab-Source:

```
>> mkdir MatlabCrashkurs
>> cd MatlabCrashkurs
>> edit first.m
>> ls
>> pwd
```

For-Schleifen

Erstellen Sie eine Datei namens `loop.m`. Setzen Sie immer ein `clear all` in die erste Zeile; sicherheitshalber.

Aufgabe 11:

Implementieren Sie mit Hilfe einer `for`-Schleife den Wert der Summe

$$\sum_{i=1}^{10} i.$$

Matlab-Source:

Aufgabe 12:

Und jetzt zwei verschachtelte `for`-Schleifen:

$$\sum_{i=1}^5 \sum_{j=4}^{10} i + j$$

Matlab-Source:

Aufgabe 13:

Für $i = 1, \dots, 20$, $j = 1, \dots, 40$ und $k = 1, \dots, 30$ seien die Komponenten der Matrix A gegeben durch:

$$A_{kij} = \sum_{l=1}^k \frac{i+j}{l!}$$

Wieviele `for`-Schleifen benötigen Sie, wenn Sie die Matlab-Funktion `l!=factorial(l)` verwenden?

Matlab-Source:

Aufgabe 14:

(wie oben)

Matlab-Source:

`size(A)=`

Aufgabe 15:

$$B = A_{1,i=4,8,j=10,12}$$

Matlab-Source:

Graphen

Versuchen Sie mal, eine Datei `plot.m` im Editor zu öffnen.

```
>> edit plot.m
```

Diese Datei ist bei Matlab bereits vorhanden. Es ist der Hilfetext zur `plot`-Funktion. Sie sollten immer einen Dateinamen wählen, der noch nicht "vorhanden" ist. Auf der sicheren Seite sind Sie wenn Sie das Präfix `My` voranstellen, also

```
>> edit MyPlot.m
```

MyPlot.m:

```
clear all  
close all
```

Bei der graphischen Darstellung wird ein neues Fenster geöffnet. Wenn Sie wieder in den Editor wechseln bringen Sie das `plot`-Fenster in den Hintergrund. Bei erneutem Aufruf bleibt es hinten und Sie müssen es mühsam nach vorne klicken. Bequemer ist es, wenn Sie bei jedem Aufruf alle Graphikfenster zunächst schließen. Das erreichen Sie mit der Anweisung `close all`.

Aufgabe 16:

Definieren Sie einen Parameter $N = 10$. Es ist das Intervall $I = [0, 1]$ gegeben und zerlegt in $N - 1$ Teilintervalle

$$I = \cup_{i=1}^{N-1} [x_i, x_{i+1}],$$

wobei $x_1 = 0$ und $x_N = 1$ ist. Erzeugen Sie das Feld

$$x = (0, \dots, 1) \in \mathbb{R}^N.$$

Matlab-Source:

Aufgabe 17:

Belegen Sie das Feld

$$u = e^x$$

mit $x = (0, \dots, 1) \in \mathbb{R}^N$ wie oben und stellen Sie es graphisch dar.

Matlab-Source:

Aufgabe 18:

Belegen Sie zusätzlich das Feld

$$y = x^2$$

und mit u von oben das Feld

$$v = u \cdot y$$

komponentenweise und stellen Sie dieses auf das Feld x aufgetragen graphisch dar.

Matlab-Source:

Aufgabe 19:

Lesen Sie in der Hilfe (`>> help plot` oder ausführlicher `>> doc plot`) wie Sie die Farb- und Formdarstellungen der Graphen selbst bestimmen können und Titel, sowie Achsenbeschriftungen hinzufügen (`title`, `xlabel`, `ylabel`). Versuchen Sie eine Graphik, ähnlich (!!!) wie in [Abbildung 1](#) zu erzeugen.

Matlab-Source:

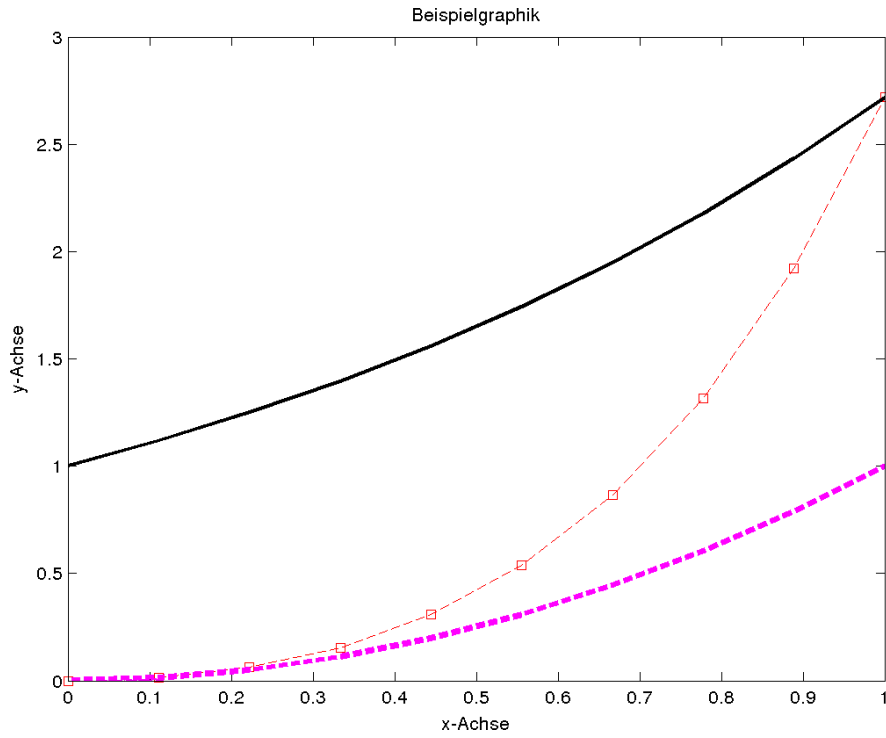


Abbildung 1: Beispielgraphik (schwarz (k), rot (r), magenta (m))

Aufgabe 20:

Lesen Sie den Hilfetext zur Matlab-funktion `subplot` und versuchen Sie die graphische Darstellung in Abbildung 2 nachzubilden.

Matlab-Source:

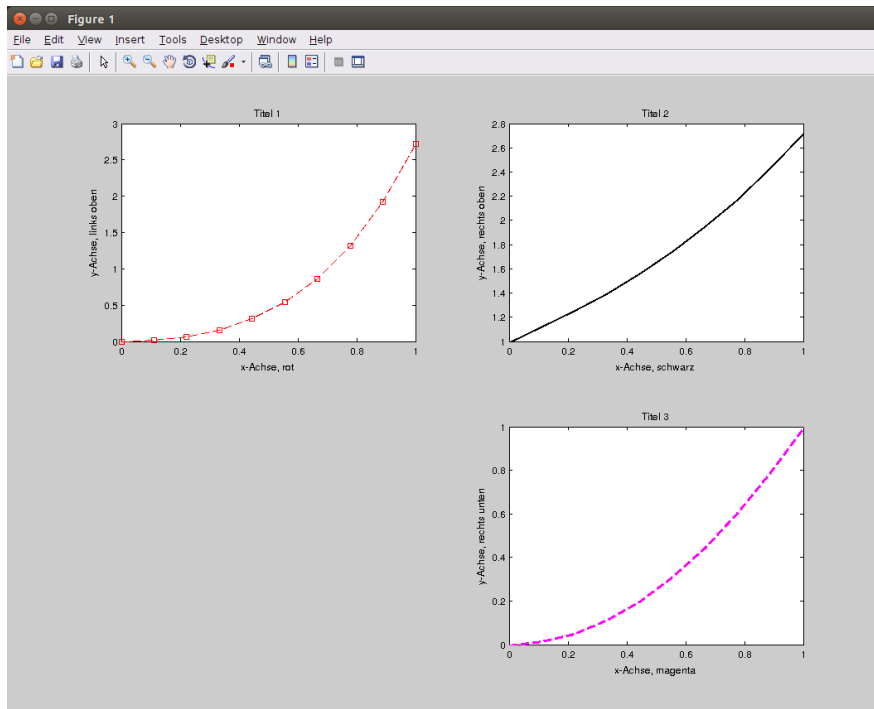


Abbildung 2: Beispielgraphik

Spezielle Matrizen

Aufgabe 21:

Probieren Sie die Befehle rechts aus und informieren Sie sich über

```
>> help ones  
und  
>> help zeros.
```

Matlab-Source:

```
ones(2,3)=
```

```
ones(4)=
```

```
zeros(1,5)=
```

Aufgabe 22:

Gegeben sind

$$A = \begin{pmatrix} 2 & 1 & -4 & 8 \\ 0 & 3 & 0 & -2 \\ 5 & 4 & 6 & 2 \\ 6 & -1 & 1 & 0 \end{pmatrix}$$

$$V = \begin{pmatrix} 1 \\ 4 \\ 6 \\ 8 \end{pmatrix}$$

Probieren Sie die Befehle rechts aus und informieren Sie sich über

```
>> help diag
```

und

```
>> help eye.
```

Matlab-Source:

```
diag(A)=
```

```
diag(V)=
```

```
diag(diag(A))=
```

```
diag(V,1)=
```

```
diag(V,-2)=
```

```
eye(3)=
```

```
eye(2,3)=
```

Aufgabe 23:

Stellen Sie nun folgende Matrizen auf:

$$A = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$C = \begin{pmatrix} 0 & 3 & 0 & 0 \\ -1 & 0 & 3 & 0 \\ 0 & -1 & 0 & 3 \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

Matlab-Source:

if-else-Anweisung**Aufgabe 24:**

Spielen Sie mit logischen Operatoren

$\&\&$, $\|\|$ und \sim .

Matlab-Source:

```
if 1
    display('1 ist true')
end

if 0
    display('0 ist true (das sollten sie
nicht lesen koennen)')
else if 1 || 0
    display('1 oder 0 ist true')
end

if 0 && 1
    ...

if ~0 && 1
    ...

if 2
    display('2 ist true')
```

Aufgabe 25:

Untersuchen Sie

$$((a \& \sim (a|b)) | \sim a) \& b = ?$$

für verschiedene Werte von a und b .

Matlab-Source:

Aufgabe 26:

Spielen Sie mit Vergleichsoperatoren

`==`, `~=`, `>`, `>=`, `<`

und

`<=`.

Matlab-Source:

```
if 1<=2
    ...
if 1<=2 || 0~=3
    ...
etc
```

Ein- und Ausgabe

Aufgabe 27:

Lesen aus und schreiben an stdio
(Bildschirm)

Matlab-Source:

```
a = input('Enter a:');  
b = input('Enter b:');  
  
c=a^b;  
fprintf('%f^%f=%e \n',a,b,c);
```

Aufgabe 28:

Formatierte Ausgabe
Probieren Sie Folgendes:

Matlab-Source:

```
fprintf('%f\n',2/3);  
fprintf('%.2f\n',2/3);  
fprintf('%8.2f\n',2/3);  
  
fprintf('Die %d ist eine natuerliche Zahl.\n',2);  
fprintf('Die %4d ist eine natuerliche Zahl.\n',2);  
fprintf('Die %04d ist eine natuerliche Zahl.\n',2);  
  
fprintf('Jetzt kommt ein String: %s \n','hallo');  
Wort = 'hallo';  
fprintf('Oder versteckt in einer Variablen: %s \n',Wort);
```

Aufgabe 29:

Die Anweisungen rechts erzeugen eine 100x100-Matrix A mit zufällig gewählten Einträgen aus $\{0, 1, 2\}$. Lesen Sie dazu die Hilfetexte zu `randi` und `mod`. Die Anweisung `spy` stellt graphisch die Belegung von nicht Nulleinträgen der Matrix dar. Mit `save` werden die Werte von A in einer Datei namens `A.mat` gespeichert. Überzeugen Sie sich mit `ls` von der Existenz der Datei und mit `whos` davon, dass A tatsächlich definiert ist. Löschen Sie alles aus dem Speicher mit `clear all`. `load A` liest die Werte in `A.mat` wieder ein und belegt die Matrix A wieder. Überzeugen Sie sich davon.

Matlab-Source:

```
A=mod(randi(100,100),3);
spy(A);
save A;
ls
whos
clear all
whos
load A
whos
```


strukturiertes Programmieren

Aufgabe 30:

Erstellen Sie eine Datei namens

`MyProg.m`

und definieren Sie darin die Funktion

$$f(x) = \sin(e^x)$$

mit dem Namen `MySinExp`. Auf einem Intervall $I = [1, 4]$ gebe es die äquidistante Zerlegung in $N = 100$ Teilintervalle. Erzeugen Sie das Feld x und $y = \text{MySinExp}(x)$. Stellen Sie y auf x aufgetragen graphisch dar.

Matlab-Source:

Aufgabe 31:

Erzeugen Sie eine Funktionsdatei für die Abbildung

$$g(x) = \sin x.$$

mit dem Namen `MySin`.

Aufgabe 32:

Definieren Sie innerhalb der Funktionsdatei `MySin` eine Unterfunktion

$$v(x) = e^x$$

mit dem Namen `MyExp`, welche dann in `MySin` gemäß

$$\text{MySin}(\text{MyExp}(x))$$

zur Anwendung kommt. `MySin` liefert dann den Funktionswert $\sin(e^x)$.

Belegen Sie in `MyProg` das Feld y mit `MySinExp` und das Feld u mit `MySin`, stellen Sie beide Felder und die Differenz graphisch dar.

Aufgabe 33:

Erstellen Sie eine Funktionsdatei namens `dSin(x,k)` mit den Übergabeparametern x , und k und der Rückgabe

$$\frac{d^k}{dx^k} \sin x .$$

Also

$$\text{dSin}(x,0) = \sin x$$

$$\text{dSin}(x,1) = \sin' x = \cos x$$

$$\text{dSin}(x,2) = \sin'' x = -\sin x$$

etc.

Erzeugen Sie auf dem Intervall $I = [0, 2\pi]$ Felder mit Funktionswerten zu 0-ter, 1-ter und 2-ter Ableitung von `sin` und stellen Sie diese graphisch dar.