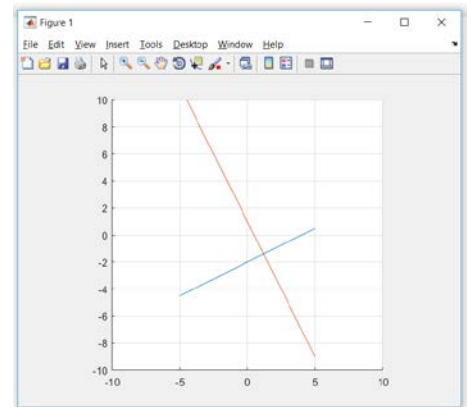


# Fachkonferenz Mathematik 2017



29. Oktober 2017

## Matlab im BM-Unterricht

### Mathematik jenseits des Taschenrechners

```
%Cosinussatz
%Ch. Fäs
%06.10.17
%
a= input ('Seitenlänge a = ');
b= input ('Seitenlänge b = ');
phi= input ('Zwischenwinkel in Grad = ');
c=sqrt(a^2+b^2-2*a*b*cosd(phi))
```

# 1 Inhalt

1	Warum Matlab?	1
1.1	Vorteile von Matlab	1
1.2	Nachteile von Matlab	1
2	Arbeiten im Command Window	2
2.1	Einführung in einfache Rechenoperationen	2
2.1.1	Grundoperationen	2
2.2	Rechnen mit sehr grossen und sehr kleinen Zahlen	3
2.3	Trigonometrischen Funktionen und Logarithmus Sin, cos, tan, acosd	4
2.4	Befehle: Solve, Expand, Factor	6
2.5	Vektorrechnung	7
2.6	Darstellen von Kurven plot	8
3	Script-Dateien: M-Files	10
4	Matlab in IDAF Sequenzen.	11
5	Vergleich mitGeogebra	12
6	Anhang	13

# 1 Warum Matlab?

- Matlab findet oft in Fachhochschulen und Industrie Verwendung. Darum sind viele beim Eintritt in eine technische Fachhochschule bereits mit Matlab etwas vertraut.
- Beim Erstellen von Script-Dateien, im Matlab die sogenannten m-files, zeigen auf, wie ohne Grafikerunterstützung Programme geschrieben werden können.

## 1.1 Vorteile von Matlab

- Mit einigen wenigen Matlab Befehlen wie solve, factor, expand usw. und vorallem in der Vektorgeometrie (dot, cross, norm usw.) können Rechenoperationen sehr schnell, sicher und zeitsparen ausgeführt werden. Dadurch kann man sich vorallem bei IDAF-Sequenzen auf die eigentliche Aufgabenstellung besser konzentrieren.
- Matlab kann teilweise verdeutlichen was im Hintergrund eines Programms abläuft.
- Matlab wird zu Beginn als bedienerunfreundlich empfunden.
- Deshalb ist Matlab wahrscheinlich weniger für die BM Typ Wirtschaft geeignet.
- Matlab bietet aber noch sehr viel mehr, als im Regelunterricht behandelt werden kann.

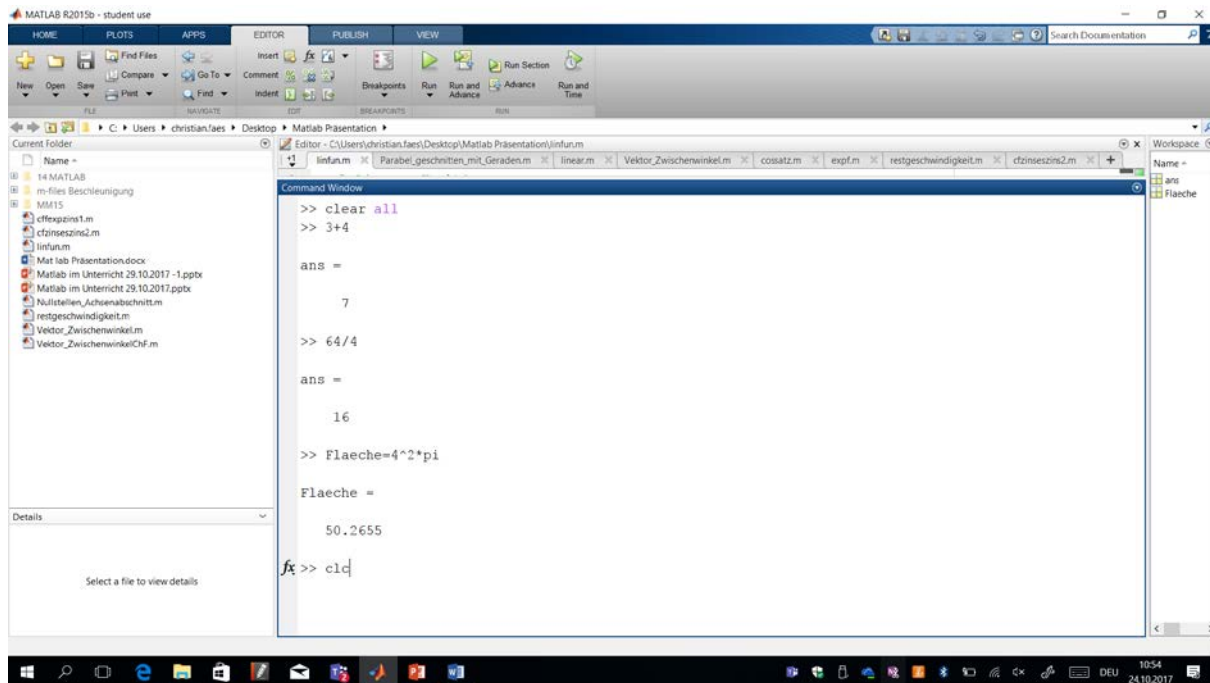
## 1.2 Nachteile von Matlab

- Matlab weist keine komfortable Grafikoberfläche auf.
- Hier ist ein Programm wie Geogebra eher geeignet.

## 2 Arbeiten im Command Window

### 2.1 Einführung in einfache Rechenoperationen

#### 2.1.1 Grundoperationen



#### Command Window

```
>> pi
.....
ans =
.....
    3.1416
.....
>> format long
>> pi
.....
ans =
.....
    3.141592653589793
.....
>> format short
>> pi
.....
ans =
.....
    3.1416
.....
fx >>
```

## 2.2 Rechnen mit sehr grossen und sehr kleinen Zahlen

```
Command Window
```

```
>> 1.9899*10^30/(5.95*10^24)
```

```
ans =
```

```
3.3444e+05
```

```
>> 1.6*10^-19*4*10^6
```

```
ans =
```

```
6.4000e-13
```

```
fx >>
```

```
Command Window
```

```
>> c=1.6*10^-19*6.022*10^23
```

```
c =
```

```
9.6352e+04
```

```
>> fprintf('Ladung = %0.4f As\n', c)
```

```
Ladung = 96352.0000 As
```

```
fx >>
```

## 2.3 Trigonometrischen Funktionen und Logarithmus Sin, cos, tan, acosd....

```
Command Window
>> sin(pi)
ans =
    1.2246e-16
>> sin(pi/2)
ans =
    1
>> sind(30)
ans =
    0.5000
>> acosd(sqrt(2)/2)
ans =
    45
```

```
Command Window
>> clear all
>> log(2.7182818)
ans =
    1.0000
fx >>
```

```
Command Window
>> clear all
>> log(2.7182818)
ans =
    1
fx >>
```

Evaluate Selection	F9
Open Selection	Strg+D
Help on Selection	F1
Function Browser	Umschalt+F1
Show Function Browser Button	
Function Hints	Strg+F1
Cut	Strg+X
Copy	Strg+C
Paste	Strg+V
Select All	Strg+A
Find...	Strg+F
Print...	Strg+P
Print Selection...	
Page Setup...	
Clear Command Window	

```
Command Window
>> clear all
>> log (2.7182818)

ans =

    1.0000

>> log10(1000000)

ans =

     6

fx >>
```



```
Command Window
>> clear all
>> log (2.7182818)

ans =

    1.0000

>> log10(1000000)

ans =

     6

>> log2(256)

ans =

     8

fx >> |
```

## 2.4 Befehle: Solve, Expand, Factor

```
Command Window
>> clear all
>> syms x y
>> [x y]=solve(3*x+y==73,2*x-y==32)

x =

21

y =

10

fx >>
```

```
Command Window
>> clear all
>> syms x
>> expand((2*x+7)*(x-11)*(x+3)*(x-5)*(x+6)*(x-5))

ans =

2*x^6 - 17*x^5 - 156*x^4 + 872*x^3 + 3844*x^2 - 10215*x - 34650

fx >> |
```

```
Command Window
>> clear all
>> syms x
>> factor(2*x^6 - 17*x^5 - 156*x^4 + 872*x^3 + 3844*x^2 - 10215*x - 34650)

ans =

[ x - 11, 2*x + 7, x + 6, x + 3, x - 5, x - 5]

fx >> |
```



## 2.5 Vektorrechnung

```
Command Window
>> a=[1 5 6]
a =
     1     5     6
>> b=[3 3 4]
b =
     3     3     4
>> c=a+b
c =
     4     8    10
fx >> |
```

```
Command Window
a =
     1
     5
     6
>> b=[3 3 4]'
b =
     3
     3
     4
>> c=a+b
c =
     4
     8
    10
```

```
Command Window
>> clear all
>> a=[1 5 6]'
a =
     1
     5
     6
>> b=[3 3 4]
b =
     3     3     4
>> c=a+b
Error using +
Matrix dimensions must agree.
fx >>
```

```
Command Window
>> clear all
>> a=[1 5 6]'
a =
     1
     5
     6
>> b=[3 3 4]'
b =
     3
     3
     4
>> Zwischenwinkel=acosd(dot(a,b)/(norm(a)*norm(b)))
Zwischenwinkel =
    23.8263
```

## 2.6 Darstellen von Kurven plot

Command Window

```
>> clear all
>> x=[-5;0.1;5]

x =

-5.0000
 0.1000
 5.0000
```

fx &gt;&gt; |

Command Window

```
>> clear all
>> x=[-5:0.1:5]

x =

Columns 1 through 10
-5.0000 -4.9000 -4.8000 -4.7000 -4.6000 -4.5000 -4.4000 -4.3000 -4.2000 -4.1000

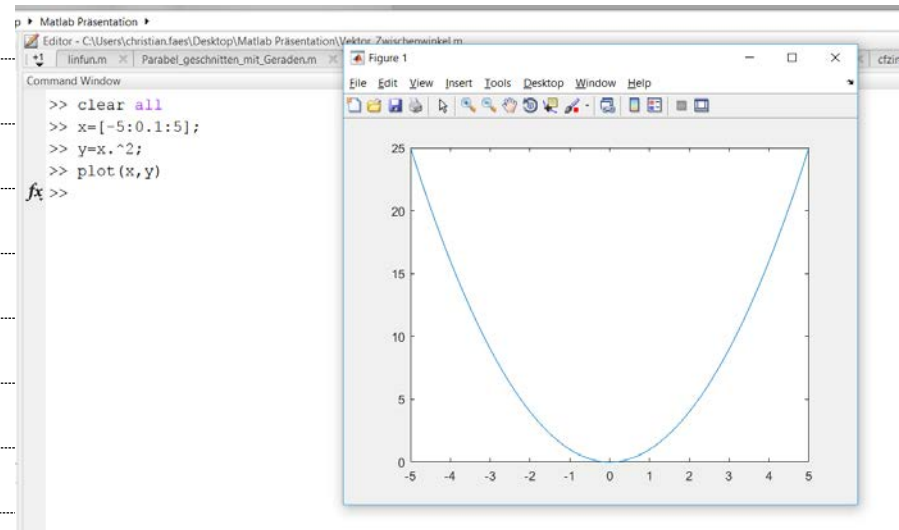
Columns 11 through 20
-4.0000 -3.9000 -3.8000 -3.7000 -3.6000 -3.5000 -3.4000 -3.3000 -3.2000 -3.1000

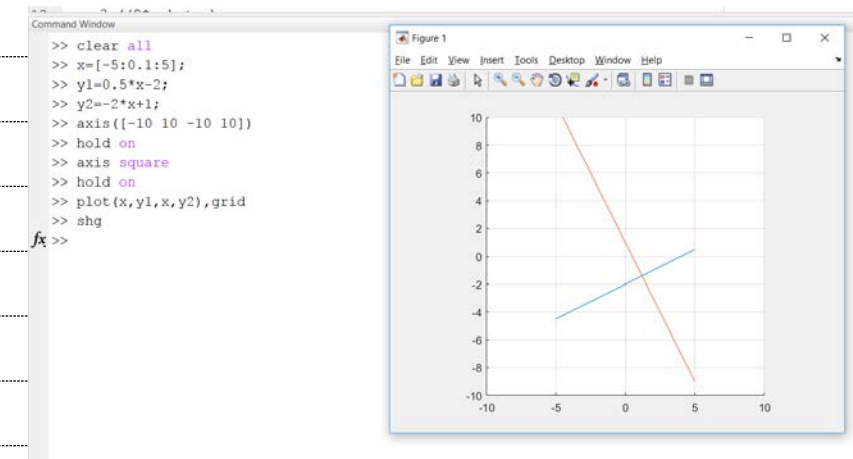
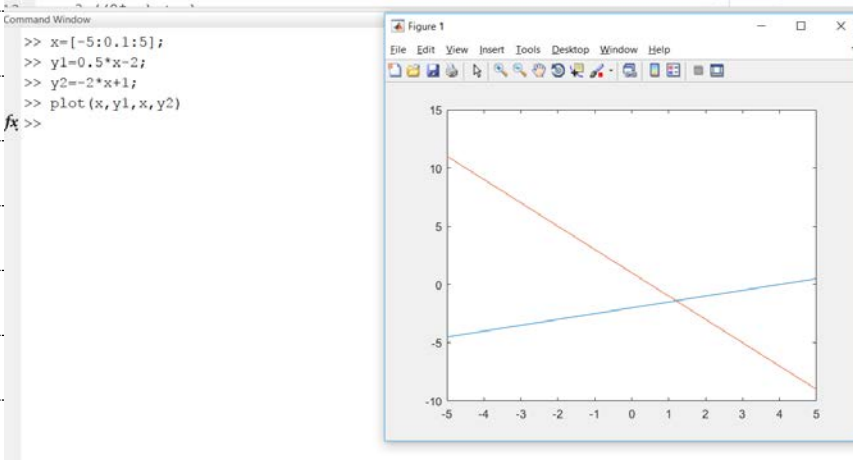
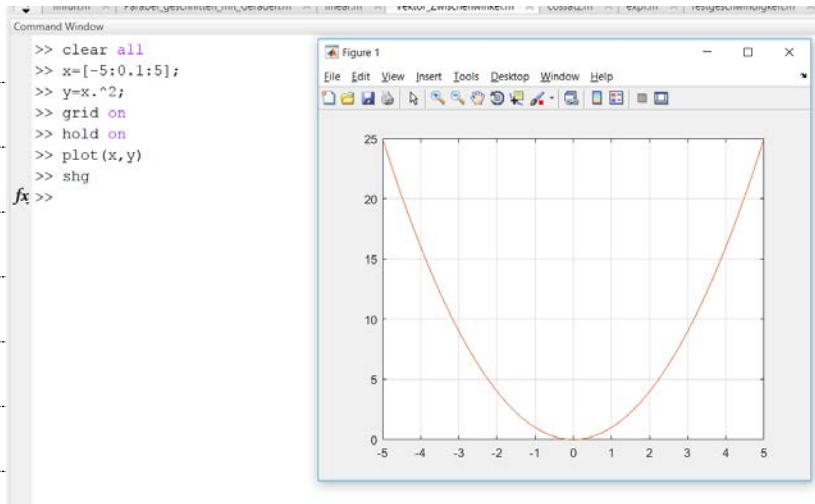
Columns 21 through 30
-3.0000 -2.9000 -2.8000 -2.7000 -2.6000 -2.5000 -2.4000 -2.3000 -2.2000 -2.1000

Columns 31 through 40
-2.0000 -1.9000 -1.8000 -1.7000 -1.6000 -1.5000 -1.4000 -1.3000 -1.2000 -1.1000

Columns 41 through 50
-1.0000 -0.9000 -0.8000 -0.7000 -0.6000 -0.5000 -0.4000 -0.3000 -0.2000 -0.1000
```

fx











## 6 Anhang

```
%Cosinussatz
%Ch. Fäs
%06.10.17
%
a= input ('Seitenlänge a = ');
b= input ('Seitenlänge b = ');
phi= input ('Zwischenwinkel in Grad = ');
c=sqrt(a^2+b^2-2*a*b*cosd(phi))
```

```
% Zinsrechnen Exponentialfunktion
% 24.10.17
% cfa
clc
clear all
disp('Zinseszins')
disp('Eingabe Kn=K0*(1+(p/100))^n')
K0=input('Grundkapital K0= ');
p=input('Zinsfuß p= ');
x=input('Anzahl Jahre n= ');
n=[0:1:x];
Kn=K0*(1+(p/100)).^n;
plot(n,Kn),grid
disp('*****')
disp('Resultate')
disp('')
disp('kapital nach n Jahren')
format long
Kn
disp('')
disp('*****')
disp('')
shg
```



```
% Parabel geschnitten mit Geraden
% Parabel geschnitten mit Geraden.m
%07.06.2017
%Jan Weibel
clc
disp ('Parabel geschnitten mit Geraden')
disp (' ')
disp ('Eingabe Parabel a1, b1, c1 Eingabe Gerade
a2, b2')
a1=input ('Wert a1= ');
b1=input ('Wert b1= ');
c1=input ('Wert c1= ');
a2=input ('Wert a2= ');
b2=input ('Wert b2= ');
b= b1-a2;
c= c1-b2;
D=b^2-4*a1*c;
x1=( (-b+sqrt(D))/(2*a1));
x2=( (-b-sqrt(D))/(2*a1));
y1= a2*x1+b2;
y2= a2*x2+b2;
disp (' ')
disp ('*****')
disp ('Resultate')
disp (['A=(',num2str(x1), '/', num2str(y1), ')'])
disp (['B=(',num2str(x2), '/', num2str(y2), ')'])
disp (' ')
disp ('*****')
% Ende
```

```
% Zwischenwinkel zwischen zwei Vektoren berechnen
% Vektor_Zwischenwinkel.m
% 31.05.2012
% P. Lüscher
% v1.0
clear
clc
disp ('Zwischenwinkel zwischen zwei Vektoren')
disp ('v1.0')
disp ('*****')
disp (' ')
ax=input ('1. Vektor x-Wert = ');
ay=input ('1. Vektor y-Wert = ');
az=input ('1. Vektor z-Wert = ');
bx=input ('2. Vektor x-Wert = ');
by=input ('2. Vektor y-Wert = ');
bz=input ('2. Vektor z-Wert = ');
disp (' ')
disp ('*****')
disp (' ')
%Resultat
a=[ax ay az];
b=[bx by bz];
c=acosd(dot(a,b)/(norm(a)*norm(b)));
fprintf('Winkel = %0.3f°\n', c)
disp (' ')
disp ('*****')
```

```
% Lineare Funktion
% linfo.m
% 01.06.2017
% mri
clear
clc
disp ( '*****' )
disp ( 'Lineare Funktion' )
disp ( ' ' )
v=input ( 'Variante 1: 2 Punkte; Variante 2:
Steigung a und 1 Punkt. v=      ' );
if v==1
    Ax=input ( 'x-Koordinate von Punkt A=      ' );
    Ay=input ( 'y-Koordinate von Punkt A=      ' );
    Bx=input ( 'x-Koordinate von Punkt B=      ' );
    By=input ( 'y-Koordinate von Punkt B=      ' );
    a=(Ay-By)/(Ax-Bx);
    b=Ay-a*Ax;
elseif v==2
    Ax=input ( 'x-Koordinate von Punkt A=      ' );
    Ay=input ( 'y-Koordinate von Punkt A=      ' );
    a=input ( 'Steigung a=      ' );
    b=Ay-a*Ax;
else
    disp ( 'Falsche Eingabe: Abbruch' )
end
xN=-b/a;
w=input ( 'Berechnung der zur ersten Funktion
senkrechten Geraden durch P: 1=Ja:      ' );
if w==1
    Px=input ( 'x-Koordinate von Punkt P=      ' );
    Py=input ( 'y-Koordinate von Punkt P=      ' );
    m=-1/a;
    n=Py-Px*m;
else
    disp ( 'ohne weitere Berechnung' )
end
clc
```

```
disp ( '*****' )
disp ( 'Resultate' )
disp ( ' ' )
disp ( ['Steigung a= ', num2str(a)] )
disp ( ['y-Achsenabschnitt b= ', num2str(b)] )
disp ( ' ' )
disp ( 'Funktionsgleichung y1' )
if sign (b)==1
    disp ( ['y1=', num2str(a), '*x+', num2str(b)] )
elseif sign (b)==0
    disp ( ['y1=', num2str(a), '*x'] )
else
    disp ( ['y1=', num2str(a), '*x', num2str(b)] )
end
disp ( ' ' )
disp ( ['Nullstelle N0 = (', num2str(xN), ' / 0)'] )
disp ( ' ' )
if w==1
    disp ( 'senkrechte Funktionsgleichung y2' )
    if sign (n)==1
        disp
        ( ['y2=', num2str(m), '*x+', num2str(n)] )
    elseif sign (n)==0
        disp ( ['y2=', num2str(m), '*x'] )
    else
        disp
        ( ['y2=', num2str(m), '*x', num2str(n)] )
    end
else
end
disp ( ' ' )
disp ( '*****' )
```

```
%Exponentialfunktionen darstellen
%M. Richner
%19.09.07
%expf.m
clc
disp('Exponentialfunktionen')
disp(' ')
x1=[-2:0.05:1.2]
y1=10.^(x1)
x2=[-2:0.05:3]
y2=2.^(x2)
plot(x1,y1,x2,y2), grid
title('Exponentialfunktion')
hold on
gtext('y1=10^x')
hold on
gtext('y2=2^x')
hold on
shg
```

```
% restgeschwinigkeit
% IDAF
% Claude Spirgi, Florian Wirz, Jonathan Falk
% Berechnen der Restgeschwindigkeit

% Das Programm berücksichtigt folgende 3 Situa-
tionen
% - Das schnellere Fahrzeug rammt den Fussgänger
% - Das schnellere Fahrzeug rammt den Fussgänger
ohne zu bremsen
% - Beide Fahrzeuge können rechtzeitig bremsen

clear
clc

% Eingaben

disp('-----')
disp('          --- Restgeschwindigkeit ---')
disp('-----')
disp(' ')
disp('Eingaben')
disp('-----')
disp('Fahrzeug 1 kann immer rechtzeitig brem-
sen')
disp(' ')
disp('Fahrzeug 1')
v01=input('Anfangsgeschwindigkeit [km/h]: ');
t1=input('Reaktionsgeschwindigkeit [s]: ');
a1=input('Bremsverögerung [m/(s^2)]: ');
disp(' ')
disp('Fahrzeug 2')
v02=input('Anfangsgeschwindigkeit [km/h]: ');
t2=input('Reaktionsgeschwindigkeit [s]: ');
a2=input('Bremsverögerung [m/(s^2)]: ');
disp(' ')

```

```
disp('-----')
disp(' ')

% Berechnungen

% Geschwindigkeit in m/s umwandeln
v1=v01/3.6;
v2=v02/3.6;

% von Fahrzeug 1 gefahrene Strecke
s1=v1*t1+v1^2/(2*a1);
s2=v2*t2+v2^2/(2*a2);
sr1=v1*t1;
sr2=v2*t2;
vr=sqrt(v2^2-2*a2*(s1-v2*t2))*3.6;
if (v2^2-2*a2*(s1-v2*t2))>0

% Visualisierung
y=[0:0.01:v02+10];
x=0*y+s1;
% max Weg
plot(x,y,'r','LineWidth',2.5), grid
title('Restgeschwindigkeit');
xlabel('Strecke [m]')
ylabel('Geschwindigkeit [km/h]')
hold on

% Fahrzeug 1
x=[0:0.01:sr1];
y=0*x+v01;
plot(x,y,'g','LineWidth',2)
hold on
x=[sr1:0.01:s1];
y=(0-v01)/(s1-sr1)*x+v01-(0-v01)/(s1-
sr1)*sr1;
plot(x,y,'g','LineWidth',2)
hold on

% Fahrzeug 2
```

```
x=[0:0.01:sr2];
y=0*x+v02;
plot(x,y,'b','LineWidth',2)
hold on
x=[sr2:0.01:s1];
y=(vr-v02)/(s1-sr2)*x+v02-(vr-v02)/(s1-
sr2)*sr2;
plot(x,y,'b','LineWidth',2)
hold on
legend('Distanz zum Fussgängerstreifen')

% Ausgabe
% Ungebremster Aufprall berücksichtigen
if s1<sr2
    disp('Resultat')
    disp('-----')
    disp('Der Reaktionsweg ist zu gross.')
    disp('Das Fahrzeug kollidiert mit dem Fuss-
gänger ungebremst.')
else
    disp('Resultat')
    disp('-----')
    disp(['Restgeschwindigkeit: ',num2str(vr),'
km/h.'])
end
else
    % Visualisierung
    y=[0:0.01:v02+10];
    x=0*y+s1;
    % max Weg
    plot(x,y,'r','LineWidth',2.5), grid
    title('Restgeschwindigkeit');
    xlabel('Strecke [m]')
    ylabel('Geschwindigkeit [km/h]')
    hold on

    % Fahrzeug 1
    x=[0:0.01:sr1];
```



```
y=0*x+v01;
plot(x,y,'g','LineWidth',2)
hold on
x=[sr1:0.01:s1];
y=(0-v01)/(s1-sr1)*x+v01-(0-v01)/(s1-
sr1)*sr1;
plot(x,y,'g','LineWidth',2)
hold on

% Fahrzeug 2
x=[0:0.01:sr2];
y=0*x+v02;
plot(x,y,'b','LineWidth',2)
hold on
x=[sr2:0.01:s2];
y=(0-v02)/(s2-sr2)*x+v02-(0-v02)/(s2-
sr2)*sr2;
plot(x,y,'b','LineWidth',2)
hold on
legend('Distanz zum Fussgängerstreifen')

disp('Resultat')
disp('-----')
disp('Beide Autos können rechtzeitig brem-
sen. ')
end
disp('-----
-----')
```