

```
y = 5:8
y =
    5    6    7    8
```

```
x = 20:2:26
x =
    20    22    24    26
```

```
linspace(first,last,number_of_elements).

Note the use of commas (,) to separate inputs to the
linspace function.

x = linspace(0,1,5)
x =
    0    0.250    0.500    0.750    1.000
```

```
x = (1:2:5)'
x =
    1
    3
    5

[xrow,xcol] = size(x)
```

```
>> a = pi
a =
    3.1416

>> abs (absolute value) and eig (calculate eigenvalues).
>> a = sin(-5)
a =
    0.9589
```

Beispiel für if, die Zeilen unterscheiden sich in Komma oder Semikolon nach a=... :

```
>> x = 5;
>> if x<=2, a=2; elseif x>=3, a=5; else a=10; end % a=...;
>> if x<=2, a=2, elseif x>=3, a=5, else a=10, end % a=...;
a =
    5
```

Beispiel für switch, die Zeilen unterscheiden sich in Komma oder Semikolon nach a=... :

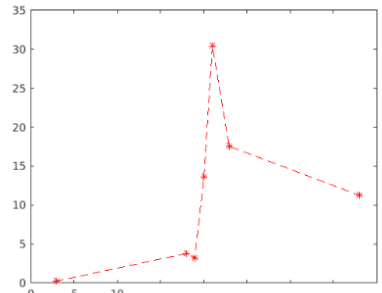
```
>> x = 5;
>> switch x case 2; a=2; case{3 4 5}; a=5; otherwise a=10; end
>> switch x case 2; a=2, case{3 4 5}; a=5, otherwise a=10, end
a =
    5
```

```
>> for k=0:2, k^2; end; >> k=0; while k<=2; k^2; k=k+1; end;
>> for k=0:2; k^2; end; >> k=0; while k<=2, k^2; k=k+1; end;
>> for k=0:2 k^2; end; >> k=0; while k<=2 k^2, k=k+1; end;
>> for k=0:2 k^2, end;
ans =
    0
ans =
    1
ans =
    4
```

```
x= randi(20,5,7)
x= randi(max,Zeilen,Spalten)

plot(x,y)
plot(sample, mass2, 'r--*')

hold on
plot(sample, mass1, 'ks')
```



```
plot(x,y,"ro-","LineWidth",5)
histogram(density,'facecolor','y')

v = v1(v1 > 6)
v =
    6.6678
    9.0698
```

You can also use logical indexing with two different vectors.

```
v = sample(v1 > 6)
v =
    18
    23
x(x==999) = 1
```

TASK
Create a variable `s` that contains the elements in `sample` corresponding to where `v1` is less than 4.

```
for idx = 1:length(density)
```

Mathematische Funktionen und Operatoren			
+ * /	Rechenoperatoren	exp(x)	Exponentialfunktion e ^x
^			
mod(x,y)	x Modulo y	log(x)	natürl. Logarithmus
rem(x,y)	Rest nach Division x/y	log10(x)	Zehner-Logarithmus
sqrt(x)	Quadratwurzel	erf(x)	Error-Funktion
abs(x)	Betrag	real(x)	Realteil
sign(x)	Signum (Vorzeichen, {-1, 0, 0})	imag(x)	Imaginärteil
round(x)	Runden	conj(x)	kompl. Konjugation
ceil(x)	Aufrunden	angle(x)	Phase kompl. Zahl
floor(x)	Abrunden		
Trigonometrische Funktionen			
sin(x)	Sinus	atan(x)	Arcus-Tangens ± π/2
cos(x)	Cosinus	atan2(x)	Arcus-Tangens ± π
tan(x)	Tangens	sinh(x)	Sinus Hyperbolicus
cot(x)	Cotangens	sinc(x)	sin(πx)/(πx)

Anweisung	Beschreibung
[x11 x12 ... ; x21 x22 ...]	Eingabe von Vektoren und Matrizen
Startwert : [Schrittweite:] Zielw.	":"-Operator erzeugt Zeilenvektor, Default-Schritt ist 1
linspace (Start, Ziel, Anzahl)	Erzeugung linearer Vektoren mit Anzahl Elementen
logspace (Start, Ziel, Anzahl)	Erzeugung logarithm. Vektoren mit Anzahl Elementen
eye (n)	Einheitsmatrix der Größe n x n, a _{ii} =1 für i=j, sonst 0
zeros (Zeilen, Spalten [, Typ])	Zeilen x Spalten Matrix, alle Elemente a _{ij} =0
ones (Zeilen, Spalten [, Typ])	Zeilen x Spalten Matrix, alle Elemente a _{ij} =1
rand (Zeilen, Spalten)	Zeilen x Spalten Matrix, 0 ≤ a _{ij} ≤ 1 gleichvert. Zufallswerte
randn (Zeilen, Spalten)	Zeilen x Spalten Matrix, normalvert. Zufallswerte (Gauß)

Anweisung	Beschreibung
* ^ \	Matrixoperationen: Multiplikation, Potenz, Linksdivision
. * . ^ . /	Elementweise Operatoren
Matrix.' , transpose (Matrix)	Transponierte einer Matrix (Vektor ist 1D-Matrix)
Matrix', ctranspose (Matrix)	Transponierte mit konjugiert-komplexen Elementen
diff (Vektor [, n])	n-facher Differenzvektor, Default ist n=1.
conv (Vektor1 , Vektor2)	Faltung (convolution), z.B. Polynom-Multiplikation
f([x1 x2 x3 ...])	= [f(x1) f(x2) f(x3) ...], elementweise Anwend.

Anweisung	Beschreibung	Anweisung	Beschreibung
max (vec)	größtes Vektorelement	inv(matrix)	Inverse einer Matrix: A ⁻¹
min (vec)	kleinstes Element	det(matrix)	Determinante
mean (vec)	Mittelwert	eig(matrix)	Eigenwerte
std (vec)	Standardabweichung	rank(matrix)	Rang
sum (vec)	Summe der Elemente	cumsum (vec)	Kumulierte Summe
prod (vec)	Produkt der Elemente	cumprod (vec)	Kumuliertes Produkt

Anweisung	Beschreibung
size (Matrix)	Dimensionen einer Matrix oder Länge eines Vektors
length (Matrix)	Größe Dim. einer Matrix oder Länge eines Vektors
clear	Löscht alle Variablen im Workspace
clear all	Löscht zusätzlich alle globalen Variablen
clear [var1, var2, ...]	Löscht die ausgewählten Variablen
who	Liste aller Variablen im Workspace
whos	Detaillierte Liste aller Variablen im Workspace mit name, Dimension, Speicherbedarf und Datentyp

Operator	Beschreibung
== , eq (a,b)	gleich
~= , ne (a,b)	ungleich
< , lt (a,b)	kleiner
<= , le (a,b)	kleiner oder gleich
> , gt (a,b)	größer
>= , ge (a,b)	größer oder gleich
~ , not (a)	Negation
& , and (a,b)	UND
&&	Shortcut-UND (bricht ab sobald Ergebnis eindeutig)
, or (a,b)	ODER
	Shortcut-ODER (bricht ab sobald Ergebnis eindeutig)
xor (a,b)	XOR
all (Vektor)	=1 wenn jedes Vektorelement wahr
any (Vektor)	=1 wenn mindestens ein Vektorelement wahr
logical (a)	Typ-Umwandlung
exist ('x')	Existenz von x
find (Vektor)	Indizes wahrer Elemente

Line Style	Description
-	Solid line (default)
--	Dashed line
:	Dotted line
-. .	Dash-dot line
Marker	Description
o	Circle
+	Plus sign
*	Asterisk
.	Point
x	Cross
s	Square
d	Diamond
^	Upward-pointing triangle
v	Downward-pointing triangle
>	Right-pointing triangle
<	Left-pointing triangle
p	Pentagram
h	Hexagram
Color	Description
y	yellow
m	magenta
c	cyan
r	red
g	green
b	blue
w	white
k	black

```
t=0:0.1:10;
s= sin(2*pi*5*t).^2;
plot(t, s, 'k-')
grid
xlabel ('Frequenz [rad/s]') ;
ylabel ('Amplitude') ;
legend('location','best'); % north-, south-, west- east-, best
```

```
t = ( 0 : 0.01 : 2 ) ;
f1 = exp(-t / 2 ) ;
f2 = exp(-2* t / 5 ) ;
f3 = exp(-t+1)
plot ( t , f1 , 'b' , t , f2 , 'r' , t , f3 , 'g' ) ;
subplot (221)
plot ( t , f1 , 'b' ) ;
subplot (222)
plot ( t , f2 , 'r' ) ;
subplot (223)
plot ( t , f3 , 'g' ) ;
```

Die Excel-Importfunktion `xlsread` kann etwa wie folgt verwendet werden:

```
» xlswrite ( 'ExcelExport' , AusSpalte1 )
```

```
» B = E5; % Umspeichern der Matrix E5
» B(:,2) = [ ] % Leeren der 2. Spalte
```

```
B =
    1     0     0     0
    0     0     0     0
    0     1     0     0
    0     0     1     0
    0     0     0     1
```

```
» [zeilen , spalten] = size(B) % Größe feststellen
```

```
zeilen =
    5
spalten =
    4
```

```
function [y1,y2] = my_fun(x1,x2)
%MY_FUN berechnet meine Funktion.
% [y1,y2] = my_fun(x1,x2)
% x1: erstes Input Argument
% x2: zweites Input Argument

% fldr / ZHAW / IMS / 2017

a = x1+x2;
y1 = std(a);
y2 = mean(a);
end
```

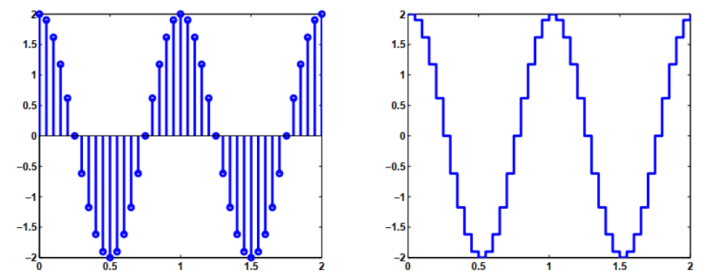
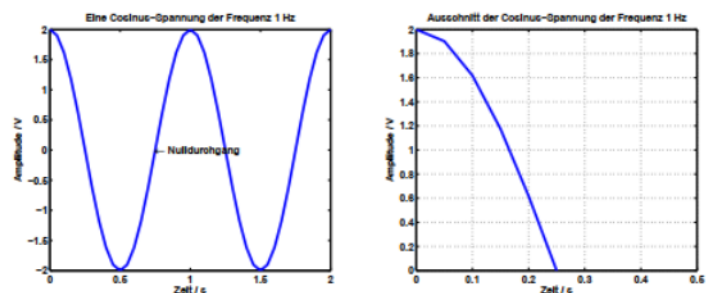


Abb. 1.5: x, y -Grafik mit den Plotfunktionen `stem` (links) und `stairs` (rechts)

```
» t=(0:0.05:2);
» cosfkt=2*cos(2*pi*t);
» plot(t,cosfkt)
» xlabel('Zeit / s')
» ylabel('Amplitude / V')
» text (0.75,0,'<math>\leftarrow</math> Nulldurchgang', 'FontSize',18)
» title('Eine Cosinus-Spannung der Frequenz 1 Hz')
» figure % Neues Fenster öffnen !!!!
» plot(t,cosfkt)
» xlabel('Zeit / s')
» ylabel('Amplitude / V')
» grid % Gitterraster setzen
» axis([0, 0.5, 0, 2]) % Ausschnitt im Intervall [0, 0.5],
% Amplituden im Intervall [0, 2]
» title('Ausschnitt der Cosinus-Spannung der Frequenz 1 Hz')
```



```
» N = repmat(M,2,2)
```

```
N =
```

```
 1  2  1  2
 3 -2  3 -2
-1  4 -1  4
 1  2  1  2
 3 -2  3 -2
-1  4 -1  4
```

```
» zVec(end) = [ ]
```

```
zVec =
```

```
 0  3 -1  0  1
```

```
function [t, sinfkt, cosfkt, expfkt]= funkbsp2(f1, f2, damp)
```

```
%
% Funktion funkbsp2
%
% Aufruf: [t, sinfkt, cosfkt, expfkt] = funkbsp2(f1, f2, damp)
% oder   funkbsp2(f1, f2, damp)
%
% Erstes Beispiel für eine MATLAB-Funktion
%
% Hier gehört normalerweise auch die Beschreibung der Ein- und
% Ausgangsparameter sowie der wesentlichen Funktionsmerkmale hin!
```

```
t=(0:0.01:2);
sinfkt=sin(2*pi*f1*t);
cosfkt=2*cos(2*pi*f2*t);
expfkt=exp(-damp*t);
plot(t,[sinfkt;cosfkt; expfkt])
xlabel('Zeit / s')
ylabel('Amplitude')
title('Drei wunderschöne Signale')
```

```
function [erg, rest] = LsgWhile2(a,b)
```

```
%
% Funktion LsgWhile2
%
% Aufruf: [erg, rest] = LsgWhile2(a,b)
%
% Aufrufbeispiel: [erg, rest] = LsgWhile2(10,3)
%
% ...
```

```
erg = 0;
rest = a;
```

```
% Berechnungsschleife
```

```
while rest >= b
    rest = rest - b;
    erg = erg + 1;
end;
```

sign block für minus und plus bei quadrat und fcn für
funktion abhängig von u

Simulink Shortcuts

