```
% Comment
“;” after the command does not show result on the screen
clc % clear screen
clear % clear all variables
clear x % clear variable x
```

Help

```
% help function_name
%Examples
help elfun % elementary functions
help matfun % functions with special matrices
```

Vectors and matrices

```
% row-vector
a1 = [1 2 3]
% row-vector
a2 = [1, 2, 3]
% column-vector
a3 = [1; 2; 3]
```

How to define a matrix

```
% Matrix 2x3
b1 = [1 2 3; 4 5 6]
% Matrix 3x2
b2 = [1 2; 3 4; 5 6]
```

How to set a vector with a given step between values

```
% set a vector
% begin value:step: end value
x = 0:0.01: 6
% Compute vector y (as a function from x)
y = sin(x)
```

Dimensions of vectors and matrices

```
% Define a vector
d = 1:12
% size of vector
size(d)
% Define a matrix
A = [1 2 3; 1 2 3; 1 2 3]
% size of matrix
size(A)
[m,n]=size(A)
```

How to select elements, rows and columns of a matrix

```
% Define a matrix
A = [1 2 3 4; 4 5 6 7; 7 8 9 10]
% Change 1st entry
A(1, 1) = 100
% Change all entries in 3rd column
A(:, 3) = 50
% Change all entries in 2nd row
A(2, :) = 33
% Change part of the matrix
```

```
A(2:4, 2 : 4) = 55
% Use of keyword end
A(3:end, 3 : end) = -21
% Delete rows or columns
% Delete 1st row
A(1, :) = []
% Delete 2nd column
A(:, 2) = []
% Define new matrix
A = [1 1 1; 2 2 2; 7 3 3]
% Delete two first rows
A(1:2, : ) = []
% Delete two last elements
A(2:end) = []
```

How to define matrix of zeros and matrix of ones

```
% Matrix of ones 7x7
A = ones(7)
% Matrix of ones 2x3
B = ones(2, 3)
% Zero matrix 3x3
C = zeros(3)
% Zero matrix 2x6
D = zeros(2, 6)
```

Magic square

```
% Magic square 3x3
M3 = magic(3)
% Magic square 5x5
M5 = magic(5)
```

Identity matrix

```
% Identity matrix 3x3
E = eye(3)
% Identity matrix 3x4
F = eye(3, 4)
```

Matrix inverse

```
% Define a matrix
A = [1 2; 0 2]
% Matrix inverse
inv(A)
% or
A ^ (-1)
```

Random matrix

```
%random scalar drawn from the uniform distribution in the interval (0,1)
n=rand
% pseudorandom matrix 5x5 with the values from the standard uniform
distribution in the open interval (0,1)
A = rand(5)
% random matrix 2x3 with the values from the standard uniform distribution in
the open interval (0,1)
A=rand(2,3)
```

Random integer matrix

```
% random integer scalar integer between 1 and 20
n =  randi(20)
% random matrix 3 x 4 with integer values between -5 and 7
A= randi([-5 7],3,4)
% random integer matrix 5x5 with integer values between 1 and 10
A =  randi(10,5)
```

Matrix product

```
% Define two matrices
M1 = [1 1 1; 2 2 2]
M2 = [3 4; 3 5; 3 6]
% Matrix product
M1*M2
```

Pointwise operations with vectors

```
% Define two vectors of equal lengths
v1 = 10:10 : 50, v2 = 1 : 5
% Pointwise product of vectors
r_1 = v1.*v2
% Pointwise division of vectors
r_2 = v1. / v2
% Sum and multiplication by a scalar
r_3 = 0.1*v1 + 100 * v2
% Define two vectors of equal lengths
q1 = [1 2 3 4], q2 = [10 20 30 40]
% Pointwise right division of vectors
p_1 = q1. / q2
% Pointwise left division of vectors
p_2 = q1.\q2
```

Pointwise operations with matrices

```
% Define two matrices of equal dimensions
m1 = [2 4 6; 3 7 9], m2 = [6 4 2; 9 7 3]
% Pointwise matrix product
z_1 = m1.*m2
% Pointwise matrix division
z_2 = m2. / m2
% Sum and multiplication by a scalar
z_3 = m1 + 10 * m2
% Define two matrices of equal dimensions
h1 = [10 20; 30 40], h2 = [5 10; 15 20]
% Pointwise right matrix division
w_1 = h1. / h2
% Pointwise left matrix division
w_2 = h1.\h2
```

Power and transposition

```
% Define a matrix
A = [1 2 3; 0 2 0; 0 0 3]
% Power of 2
A ^ 2
% Define a matrix
```

```
A = [1 1 1; 2 2 2; 4 5 6]
% Matrix transpose
A'
% Define complex matrix
A = [1 - i 1 + i; 2 + 3i 2 - 3i]
% Matrix transpose
A.'
```

How to combine matrices

```
% Define matrices
M1 = [1 2; 3 4], M2 = [5 6 7; 8 9 10]
% combine horizontally
 [M1 M2]
% combine horizontally using cat function
cat(2, M1, M2)

% Define matrices
M3 = [1 2 3], M4 = [5 6 7; 8 9 10]
% combine vertically
 [M3; M4]
% combine vertically using cat function
cat(1, M3, M4)
```

Sum of entries

```
% Define a matrix
A = [1 2; 3 4]

%  a row vector containing the sum of each column
sum(A)
% a column vector containing the sum of each row
sum (A,2)
% sum of all entries
sum(sum(A))
% trace of a matrix
trace(A)
```

Selecting diagonals of matrix

```
% Define a matrix
A = [1 2 3; 1 2 3; 1 2 3]
% Main diagonal
diag(A)
% Diagonal below main diagonal
diag(A, -1)
% Diagonal above main diagonal
diag(A, 1)
```

How to define a diagonal matrix

```
% define a vector
d1 = [1 2 3]
% Define diagonal matrix with entries of vector d1 on the main diagonal
diag(d1)
% below main diagonal
diag(d1(2:3), -1)
% above main diagonal
```

```
diag(d1(2:3), 1)
```

How to flip and rotate matrices

```
a1 = [10 2 3; 40 5 6; 70 8 9]
fliplr(a1)
a2 = [10 20 30; 4 5 6; 7 8 9]
rot90(a2)
```

How to sort matrices

```
% Define a matrix
A =rand(4,5)
% sort elements of each column
sort(A)
% sort elements of each row
sort(A,2)
```

Sparse matrix

```
% random sparse matrix nxn with density 0.05
n=10;
As=sprand(n,n,0.05)
% plot sparse matrix pattern
spy(As)
% convert to dense matrix
full(As)
```

Matrix left division mldivide, \: A\B ~ inv(A)*B

Matrix left division solves

- the system of linear equations A*x = b for x, where A is the matrix, b is a right-hand side vector, x is the vector of unknows
- the system of matrix equations A*X=B for X, where A and B are the known matrices (must have the same number of rows) and X is unknown matrix

```
% Define matrix A and vector b
A = [1 0 0; 0 2 0; 0 0 3]
b = [10; 40; 150]
% Solve linear system Ax = b
% \ is mldivide command (matrix left devision)
x = A\b
% or
x = mldivide(A,b)
% the system Ax=b can be also solved using matrix inverse:
x = A ^ (-1)*b
% or
x = inv(A)*b
```

Matrix right division B/A = B*inv(A)

Matrix left division solves the system of linear equations X*A = B for X

```
% Define two square random matrices
A=rand(4)
B=rand(4)
A/B
```