

Matlab Quick Reference¹

1 Comments

`%` a comment starts with a `'%'` symbol

Note: Every nontrivial step of a program should be documented (explained) by a nearby comment!

2 Defining Variables

```
x=1.0           % x is defined as 1 and the result is shown
x=1.0;         % x is defined as 1, the result is not shown
u=[1 0 0]      % u is the row-vector (1,0,0)
v=[1;0;0]     % v is the column vector (1,0,0)^T
A=[2 1 0; 0 2 0; 0 0 2]
                % A is a 3x3 matrix; a_11=2, a_12=1,...
A(1,2)=1      % set entry a_12 of A to 1

whos          % show all defined variables
help whos    % help for the command whos

str='Hello'   % set variable str to 'Hello'
disp(str);   % show string str
disp(x);     % show value of variable x
```

3 Operators

```
w=A*v         % w is result of matrix-vector-multiplication A*v
A'            % A' is the transpose of A
v+v          % vector addition
2*v          % scalar multiplication

2^3          % 2 to the power of 3
A^2          % matrix A squared
m=size(A,1)  % m is the number of rows of A
n=size(A,2)  % n is the number of columns of A
sqrt(v'*v)   % euclidean norm of v
```

¹Original author: Dr. Oliver Rheinbach

```

norm(v)          % euclidean norm of v

inv(A)           % inverse of A
eig(A)           % eigenvalues of A
det(A)           % determinate of A
x=A\[1;1;1]      % is mathematically equivalent to
                 % x=inv(A)*[1;1;1] but much more efficient

rank(A)          % rank of A
null(A)          % nullspace of A

x=A(1,:)         % x becomes the 1st row of A
y=A(:,2)         % y becomes the 2nd column of A

```

4 Predefined Matrices

```

zeros(3,4);      % 3x4 zero matrix
ones(3,4);       % 3x4 matrix of ones
eye(4);          % 4x4 identity matrix

```

5 Additional Operators and Functions

```

floor(3.5)       % round to lower
ceil(3.5)        % round to higher
round(3.5)       % round correctly

s=num2str(x)     % writes the value of x into string s
t=[ 'x is ' s ] % concatenates the strings 'x is' and s
disp(t)          % display t
[1;2;3].*[2;3;4] % multiply vectors componentwise

sin(2*pi)
cos(2*pi)
exp(2)           % exponential function

clear i          % clear the variable i
x=2+3*i          % x becomes the complex number 2+3*i
x*x              % now uses complex number arithmetic
1:5              % gives the vector [1 2 3 4 5]
0:.2:1           % gives [0 .2 .4 .6 .8 1]

format long      % display 15 digits of numbers

```

6 Drawing

```

% 2D graphs

```

```

x=[0:.1:2*pi];
y=sin(x);
plot(x,y);
plot(x,y,x,sin(x+1));

% 3D surfaces
x=[0;1;0];
y=[0;0;1];
z=[0;1;1];
triangle=[1 2 3];% indices for x,y,z
trisurf(triangle,x,y,z,'facecolor','interp');
           % draws the triangle in 3D (patch)
           % (0,0,0) (1,0,1) (0,1,1)

help trisurf      % gives more information of trisurf
                  % trisurf can draw many triangles in one call

close all        % close all graphic windows

```

7 Programming

```

for i=1:10        % for-loop: i runs from 1 to 10
    disp(i);      % and i is displayed every time
end

if i==1          % branch
    disp('i is 1');
else
    disp('i is not 1');
end

% Comparison operators:
% ==: is equal to; ~=: is not equal to
% < > <= >=

x=[ 1 3 7 11];
while norm(x)>1  % while norm(x)>1
    x=.5*x;      % multiply x by .5
end

a=1;
(a<5.0)&&(a>1.0) % logical AND
(a<5.0)|| (a>1.0) % logical OR
~(a<5.0)         % logical NOT

```