MATLAB CHEAT SHEET for Numerical Analysis

Throughout this document x and y will be either row or column vectors and A will always be a matrix.

Basics

`clc`	Clear command window
`clear`	Clear all variables
`clf`	Clear all plots
`close all`	Close all plots
`doc function name`	Open help page for function
`% This is a comment`	Comments
`Ctrl+c`	Abort the current operation
`format short`	Display 4 decimal places
`format long`	Display 15 decimal places
`disp('text')`	Print text
`fprintf('x=%1.2f',x)`	Formatted outputs
`whos`	List variables
`timeit or tic...toc`	Measure CPU time of codes
`profile on/viewer/off`	Profile execution time for functions

Defining and Changing Variables

`a = 3`	Define variable a to be 3
`x = [1, 2, 3]`	Set x to be the row vector $[1, 2, 3]$
`x = [1; 2; 3]`	Set x to be the column vector $[1, 2, 3]^T$

`A = [1, 2, 3, 4;`
`     5, 6, 7, 8;`
`     9, 10, 11, 12]`
	Set A to be a 3×4 matrix

`x(2) = 7`	Change x from $[1, 2, 3]$ to $[1, 7, 3]$
`A(2,1) = 0`	Change $A_{2,1}$ from 5 to 0

Basic Arithmetic and Functions

`3*4, 7+4, 2-6, 8/3`	multiply, add, subtract and divide
`3^7`	Compute 3^7
`sqrt(5)`	Compute $\sqrt{5}$
`log(3)`	Compute $\ln(3)$
`log10(100)`	Compute $\log_{10}(100)$
`abs(-5)`	Compute $\lvert -5\rvert$
`sin(5*pi/3)`	Compute $\sin(5\pi/3)$
`floor(3.8)`	Compute $\lfloor 3.8 \rfloor$
`rem(23,5)`	Remainder of 23/5 after division
`integral(fun,a,b)`	Numerical integration of a function
`interp1(x,y,xp)`	1-D data interpolation
`polyfit(x,y,n)`	Polynomial curve fitting
`roots(p)`	Roots of a Polynomial
`fzero(fun,x0)`	Root of nonlinear function
`ode45/ode15s`	ODE solvers

Constructing Matrices and Vectors

`zeros(12, 5)`	Make a 12×5 matrix of zeros
`ones(12, 5)`	Make a 12×5 matrix of ones
`eye(5)`	Make a 5×5 identity matrix
`eye(12, 5)`	Make a 12×5 identity matrix
`diag(A)`	Diagonal matrix or elements
`linspace(1.4, 6.3, 1004)`	Make a vector with 1004 elements evenly spaced between 1.4 and 6.3
`7:15`	Row vector of $7, 8, \ldots, 14, 15$
`speye(n)`	Sparse identity matrix
`spdiags(A,d,m,n)`	Sparse matrix by columns of A.

Operations on Matrices and Vectors

`3 * x`	Multiply every element of x by 3
`x + 2`	Add 2 to every element of x
`x + y`	Element-wise addition of two vectors x and y
`A * y`	Product of a matrix and vector
`A * B`	Product of two matrices
`A .* B`	Element-wise product of two matrices
`A ^ 3`	Square matrix A to the third power
`A .^ 3`	Every element of A to the third power
`cos(A)`	Compute the cosine of every element of A
`abs(A)`	Compute the absolute values of every element of A
`A'`	Transpose of A
`inv(A)`	Compute the inverse of A
`det(A)`	Compute the determinant of A
`eig(A)`	Compute the eigenvalues of A
`size(A)`	Get the sizes of A
`length(x)`	Get the dimension of x

Entries of Matrices and Vectors

`x(2:12)`	The 2^{nd} to the 12^{th} elements of x
`x(2:end)`	The 2^{nd} to the last elements of x
`x(1:3:end)`	Every third element of x from the first to last
`A(5,:)`	Get the 5^{th} row of A
`A(:,5)`	Get the 5^{th} column of A
`A(5, 1:3)`	Get the first to third elements in the 5^{th} row

2D Plotting

`plot(x,y)`	Plot y versus x (must be the same length)
`loglog(x,y)`	Plot y versus x on a log-log scale (both axes have a logarithmic scale)
`semilogx(x, y)`	Plot y versus x with x on a log scale
`semilogy(x, y)`	Plot y versus x with y on a log scale
`axis equal`	Force the x and y axes to be scaled equally
`title('A Title')`	Add a title to the plot
`xlabel('x label')`	Add a label to the x axis
`ylabel('y label')`	Add a label to the y axis
`legend('foo', 'bar')`	Label 2 curves for the plot
`grid`	Add a grid to the plot
`hold on`	Multiple plots on single figure
`axis([xa, xb, ya, yb])`	Set the x and y ranges
`figure`	Start a new plot
`subplot(2,2,index)`	Split the graph into 2*2 parts
`print(filename,'-djpeg')`	Save figure to jpeg format

Special Constants

`pi,exp(1)`	$\pi = 3.141592653589793, e = 2.718281828459046$
`NaN`	Not a number (i.e. $0/0$ or ∞/∞)
`Inf`	Infinity (e.g., $1/0$)
`realmax`	Largest positive floating-point number (1.7977e+308)
`realmin`	Smallest positive floating-point number (2.2251e-308)
`eps`	Floating-point relative accuracy (2.2204e-16)

Originally by Steven E. Thornton **Revised by Jun Liu** **Updated:** January 10, 2020

For loops

```
for k = 1:5:100 %start:step:end
    disp(k);
end
tic %starting timer
A=zeros(1000,1000);%define A for storage
for i=1:1000  %nested loop
    for j=1:1000
        A(i,j)=1/(i+j); %A changes size if not predefined.
    end
end
cpu=toc %measure CPU time aout 0.0343 Seconds
```

While loops

```
k = 0;
while k < 7
    k = k + 1;
    if(k > 5)
        break; %Terminate execution of for or while loop
    end
end
```

Logicals

```
a = 10;  % Assign a the value of 10
a == 5   % Test if a is equal to 5
    false
a == 10  % Test if a is equal to 10
    true
a >= 5   % Test if a is greater than or equal to 5
    true
a < 11   % Test if a is less than 11
    true
a ~= 4   % Test if a is not equal to 4
    true
a > 1 && a ~= 10 % Test if a is greater than 1 AND
    false        % not equal to 10
a > 1 || a ~= 10 % Test if a is greater than 1 OR
    true         % not equal to 10
```

If Conditional Statements

```
if a > 10
    disp('Greater than 10');
elseif a == 5 %optional
    disp('a is 5');
else %optional
    disp('Neither condition met');
end
```

Switch-Case Statements

```
method='Newton';
switch method
   case 'Newton'
      % Call Newton codes
   case 'Secant'
      % Call Secant codes
   otherwise %optional
      % Run this only when no case is true
end
```

Functions (save as addNumbers.m)

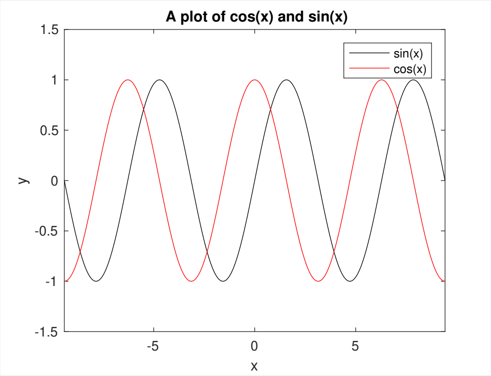
```
%function name matches with file name
function output = addNumbers(x, y)
    output = x + y;
end
>> addNumbers(10, -5) %Call function in Command Window
```

Anonymous Function Handles

```
>> f = @(x) sin(x.^2)./(5*x); %vectorized in Element-wise operations

>> f(pi/2)
    0.0795
>> f([-pi/2, 0, pi/2])
    -0.0795 NaN 0.0795
>> integral(f,0,pi/2) %integral of f from 0 to pi/2
    0.1770
```

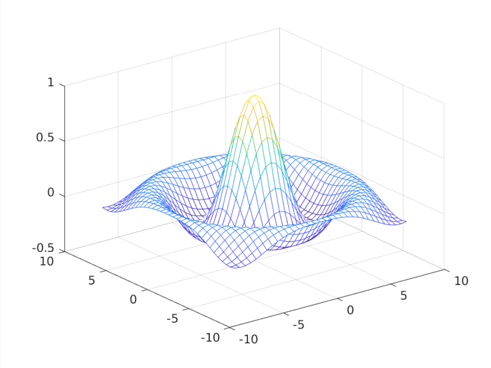
2D Plotting

```
x = linspace(-3*pi, 3*pi, 1000);
y1 = sin(x); y2 = cos(x);

plot(x, y1, 'k-');  % Plot sin(x) as a black line
hold on             % Now we can add another curve
plot(x, y2, 'r-');  % Plot cos(x) as a red line

axis([-3*pi, 3*pi, -1.5, 1.5]) % Set the axis limits
xlabel('x'); ylabel('y'); % Add axis labels
title('A plot of cos(x) and sin(x)'); % Add a title
legend('sin(x)', 'cos(x)'); % Add a legend in order
```



3D Plotting

```
[X,Y] = meshgrid(-8:.5:8);
R = sqrt(X.^2 + Y.^2) + eps;
Z = sin(R)./R;
mesh(X,Y,Z)
```



MATLAB Reserved Syntax Keywords (can not be redefined by you)

```
break,case,catch,classdef,continue,
else,elseif,end,for,function,global,
if,otherwise,parfor,persistent,return,
spmd,switch,try,while
```