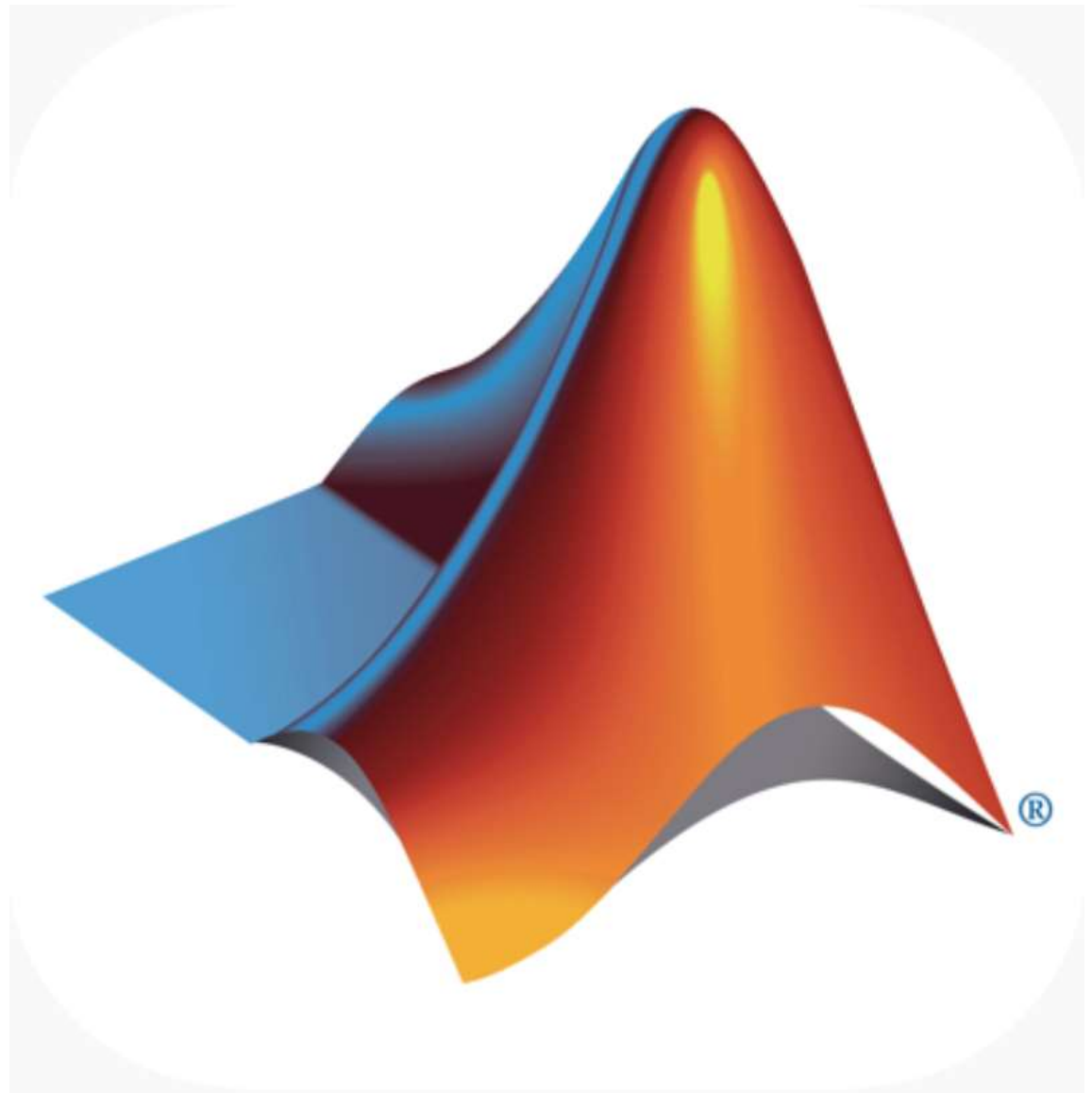


# MATLAB Survival Guide



Engineering Peer Helper Program

# MATLAB CHEAT SHEET

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

## Basics

|                                   |                             |
|-----------------------------------|-----------------------------|
| <code>clc</code>                  | Clear command window        |
| <code>clear</code>                | Clear all variables         |
| <code>clf</code>                  | Clear all plots             |
| <code>close all</code>            | Close all plots             |
| <code>doc function</code>         | Open help page for function |
| <code>% This is a comment.</code> | Comments                    |
| <code>ctrl-c</code>               | Abort the current operation |
| <code>format short</code>         | Display 4 decimal places    |
| <code>format long</code>          | Display 15 decimal places   |
| <code>disp('text')</code>         | Print text                  |

## Defining and Changing Variables

|  |   |
|--|---|
| <code>a = 3</code>   | Define variable $a$ to be 3                   |
| <code>x = [1, 2, 3]</code>                                       | Set $x$ to be the row vector $[1, 2, 3]$      |
| <code>x = [1; 2; 3]</code>                                       | Set $x$ to be the column vector $[1, 2, 3]^T$ |
| <code>A = [1, 2, 3, 4;<br/>5, 6, 7, 8;<br/>9, 10, 11, 12]</code> | Set $A$ to be a $3 \times 4$ matrix           |
| <code>x(2) = 7</code>  | Change $x$ from $[1, 2, 3]$ to $[1, 7, 3]$    |
| <code>A(2,1) = 0</code>  | Change $A_{2,1}$ from 5 to 0                  |

## Basic Arithmetic and Functions

|                                 |                                    |
|---------------------------------|------------------------------------|
| <code>3*4, 7+4, 2-6, 8/3</code> | multiply, add, subtract and divide |
| <code>3^7</code>                | Compute $3^7$                      |
| <code>sqrt(5)</code>            | Compute $\sqrt{5}$                 |
| <code>log(3)</code>             | Compute $\ln(3)$                   |
| <code>log10(100)</code>         | Compute $\log_{10}(100)$           |
| <code>abs(-5)</code>            | Compute $ -5 $                     |
| <code>sin(5*pi/3)</code>        | Compute $\sin(5\pi/3)$             |
| <code>floor(3.8)</code>         | Compute $\lfloor 3.8 \rfloor$      |

## Constructing Matrices and Vectors

|                                       |  |
|---------------------------------------|--|
| <code>zeros(12, 5)</code>             | Make a $12 \times 5$ matrix of zeros   |
| <code>ones(12, 5)</code>              | Make a $12 \times 5$ matrix of ones  |
| <code>eye(5)</code>                   | Make a $5 \times 5$ identity matrix  |
| <code>eye(12, 5)</code>               | Make a $12 \times 5$ identity matrix   |
| <code>linspace(1.4, 6.3, 1004)</code> | Make a vector with 1004 elements evenly spaced between 1.4 and 6.3                                     |
| <code>logspace(1.4, 6.3, 1004)</code> | Make a vector with 1004 elements where the log of the spacing is evenly increasing between 1.4 and 6.3 |
| <code>7:15</code>                     | Row vector of 7, 8, ..., 14, 15  |

## Operations on Matrices and Vectors

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

## Entries of Matrices and Vectors

|                         |   |
|-------------------------|---|
| <code>x(2:12)</code>    | The 2 <sup>nd</sup> to the 12 <sup>th</sup> elements of $x$ |
| <code>x(2:end)</code>   | The 2 <sup>nd</sup> to the last elements of $x$             |
| <code>x(1:3:end)</code> | Every third element of $x$ from the first to last           |
| <code>A(5,:)</code>     | Get the 5 <sup>th</sup> row of $A$                          |
| <code>A(:,5)</code>     | Get the 5 <sup>th</sup> column of $A$                       |
| <code>A(5, 1:3)</code>  | Get the first to third elements in the 5 <sup>th</sup> row  |

## Plotting

|                                   |   |
|-----------------------------------|---|
| <code>plot(x,y)</code>            | Plot $y$ versus $x$ (must be the same length)                               |
| <code>loglog(x,y)</code>          | Plot $y$ versus $x$ on a log-log scale (both axes have a logarithmic scale) |
| <code>semilogx(x, y)</code>       | Plot $y$ versus $x$ with $x$ on a log scale                                 |
| <code>semilogy(x, y)</code>       | Plot $y$ versus $x$ with $y$ on a log scale                                 |
| <code>axis equal</code>           | Force the $x$ and $y$ axes to be scaled equally                             |
| <code>title('A Title')</code>     | Add a title to the plot   |
| <code>xlabel('x label')</code>    | Add a label to the $x$ axis   |
| <code>ylabel('y label')</code>    | Add a label to the $y$ axis   |
| <code>legend('foo', 'bar')</code> | Label 2 curves for the plot   |
| <code>grid</code>                 | Add a grid to the plot  |
| <code>hold on</code>              | Multiple plots on single figure   |
| <code>figure</code>               | Start a new plot  |

## Constants

|                      |  |
|----------------------|--|
| <code>pi</code>      | $\pi = 3.141592653589793$  |
| <code>NaN</code>     | Not a number (i.e. 0/0)  |
| <code>Inf</code>     | Infinity   |
| <code>realmax</code> | Largest positive floating-point number $1.7977 \cdot 10^{308}$   |
| <code>realmin</code> | Smallest positive floating-point number $2.2251 \cdot 10^{-308}$ |

# MATLAB CHEAT SHEET

## For loops

```
for k = 1:5
    disp(k);
end
```

## While loops

```
k = 0;
while k < 7
    k = k + 1;
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
```

## Conditional Statements

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

## Functions

```
function output = addNumbers(x, y)
    output = x + y;
end
addNumbers(10, -5)
5
```

## Function Handles

```
f = @(x) sin(x.^2)./(5*x);
f(pi/2)
0.0795
f([-pi/2, 0, pi/2])
-0.0795 NaN 0.0795
```

## Plotting

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

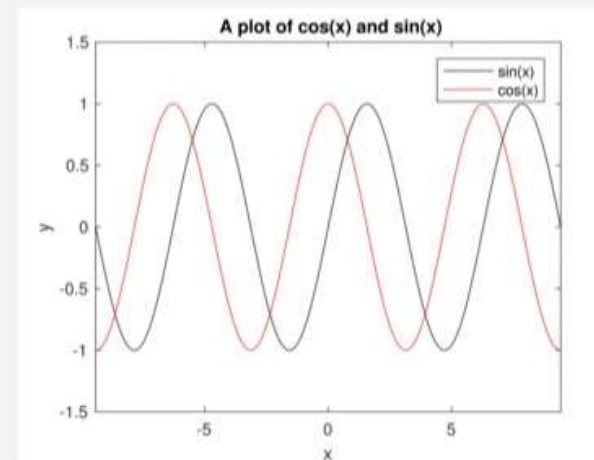
plot(x, y1, 'k-'); % Plot sin(x) as a black line
hold on % Now we can add another curve
plot(x, y2, 'r-'); % Plot cos(x) as a red line

% Set the axis limits
axis([-3*pi, 3*pi, -1.5, 1.5])

% Add axis labels
xlabel('x');
ylabel('y');

% Add a title
title('A plot of cos(x) and sin(x)');

% Add a legend
legend('sin(x)', 'cos(x)');
```





Ali Habibnia Eghbal Rahimikia

Last update: May, 2017  
MATLAB r2017a

## Preliminaries

**func** : This function or command requires a toolbox to execute.

## Ways to get help

|                            |  |
|----------------------------|--|
| <code>doc</code>           | Display documentation.                   |
| <code>doc command</code>   | Display documentation for function.      |
| <code>help</code>          | Display documentation in command window. |
| <code>help command</code>  | Display help text in command window.     |
| <code>lookfor (X)</code>   | Search all M-files for X.                |
| <code>docsearch (X)</code> | Search documentation for X.              |
| <code>demo</code>          | Access demonstration examples.           |
| <code>which command</code> | Locate functions.                        |

## File extensions

|                            |                                      |
|----------------------------|--------------------------------------|
| <code>.n</code>            | A MATLAB script, function, or class. |
| <code>.mat</code>          | A MATLAB data, stores workspace.     |
| <code>.fig</code>          | MATLAB figure or GUI template.       |
| <code>.p</code>            | MATLAB protected function file.      |
| <code>.mlx</code>          | MATLAB live script.                  |
| <code>.nex</code>          | MATLAB executable.                   |
| <code>.nlapp</code>        | MATLAB App Designer template.        |
| <code>.mdl .slx</code>     | Simulink model.                      |
| <code>.ndlp .slxp</code>   | Simulink protected model.            |
| <code>.nlappinstall</code> | MATLAB app installer file.           |
| <code>.nltx</code>         | MATLAB toolbox file.                 |

## Common data types

|                      |  |
|----------------------|--|
| <code>single</code>  | Single precision numerical data (32 bits). |
| <code>double</code>  | Double precision numerical data (64 bits). |
| <code>char</code>    | Character array.                           |
| <code>string</code>  | String array.                              |
| <code>logical</code> | True (1) or false (0).                     |
| <code>struct</code>  | Structure array.                           |
| <code>cell</code>    | Cell array.                                |

`map container` Map values to unique keys (dictionary).

## Data import/export

|                                     |   |
|-------------------------------------|---|
| <code>xlsread/xlswrite</code>       | Read/write Excel spreadsheet.                     |
| <code>load/save</code>              | Load/save MATLAB variables.                       |
| <code>load/save -ascii</code>       | Load/save text files (.txt, .csv).                |
| <code>dlmread/dlmwrite</code>       | Read/write ASCII-delimited file.                  |
| <code>readtable/writetable</code>   | Create/write table from file.                     |
| <code>fscanf/fprintf</code>         | Read/write data from/to text file.                |
| <code>textscan</code>               | Read formatted data from text file.               |
| <code>fgetl</code>                  | Read line from file, removing newline characters. |
| <code>fgets</code>                  | Read line from file, keeping newline characters.  |
| <code>fread/fwrite</code>           | Read/write from/to binary file.                   |
| <code>fopen/fclose</code>           | Open/close file.                                  |
| <code>importdata</code>             | Load data from file.                              |
| <code>readall</code>                | Read data from data-store.                        |
| <code>imread/imwrite</code>         | Read/write image file.                            |
| <code>save filename</code>          | Save all variables to .mat file.                  |
| <code>save filename x,y</code>      | Save x,y variables to .mat file.                  |
| <code>load filename</code>          | Load all variables from .mat file.                |
| <code>webread/webwrite (URL)</code> | Read/write content from/to URL.                   |
| <code>websave (URL)</code>          | Save data from URL to file.                       |

## Basic commands

|                        |  |
|------------------------|--|
| <code>clc</code>       | Clear command window.                      |
| <code>clear</code>     | Clear workspace.                           |
| <code>clear (X)</code> | Clear (X) from memory.                     |
| <code>close (X)</code> | Close figure (X).                          |
| <code>close all</code> | Close all figures.                         |
| <code>...</code>       | Continue entering statement.               |
| <code>clf</code>       | Clear current figure.                      |
| <code>whos (X)</code>  | Size, bytes, class, and attributes of (X). |
| <code>ver</code>       | List MATLAB version and toolboxes.         |
| <code>dir</code>       | List current folder contents.              |
| <code>tic/toc</code>   | Start/stop stopwatch timer.                |
| <code>beep</code>      | Produce system beep sound.                 |
| <code>ans</code>       | Last answer.                               |
| <code>pwd</code>       | Current directory.                         |
| <code>path</code>      | View/change search directory.              |
| <code>pathtool</code>  | Open set path window.                      |
| <code>mkdir</code>     | Make new directory.                        |
| <code>cd</code>        | Change current directory.                  |
| <code>what</code>      | List of MATLAB files in folder.            |
| <code>which</code>     | Find directory of functions.               |
| <code>lasterr</code>   | Last error message.                        |
| <code>lastwarn</code>  | Last warning message.                      |
| <code>rehash</code>    | Refresh caches.                            |
| <code>home</code>      | Send cursor home.                          |
| <code>exit</code>      | Close MATLAB.                              |

## Create basic variables

|                              |   |
|------------------------------|---|
| <code>x=5</code>             | Define variable x to be 5.                        |
| <code>x=nan</code>           | Define variable x to be Not-a-Number.             |
| <code>j:k</code>             | Row vector from j to k (step size: 1).            |
| <code>j:i:k</code>           | Row vector from j to k (step size: i).            |
| <code>linspace(a,b,n)</code> | n numbers linearly spaced between a and b.        |
| <code>logspace(a,b,n)</code> | n numbers logarithmically spaced between a and b. |
| <code>NaN(a,b)</code>        | a x b matrix of NaN values.                       |
| <code>ones(a,b)</code>       | a x b matrix of 1 values.                         |
| <code>zeros(a,b)</code>      | a x b matrix of 0 values.                         |
| <code>eye(a)</code>          | Identity matrix of size a.                        |
| <code>sparse(a,b)</code>     | a x b sparse matrix.                              |
| <code>rand(a,b)</code>       | Uniform a x b random numbers in [0,1].            |
| <code>randi(imax,a,b)</code> | Uniform a x b random integers in [1,imax].        |
| <code>randn(a,b)</code>      | Gaussian a x b random numbers.                    |
| <code>randperm(a)</code>     | Integer random permutation in [1,a].              |
| <code>diag(x)</code>         | Square matrix (vector x: diagonal elements).      |

## Basic math functions

|                           |  |
|---------------------------|--|
| <code>abs(x)</code>       | Absolute value of x.                                   |
| <code>sqrt(x)</code>      | Square root of x.                                      |
| <code>sign(x)</code>      | Sign of x.   |
| <code>round(x)</code>     | Round of x.  |
| <code>ceil(x)</code>      | Round x toward positive infinity.                      |
| <code>fix(x)</code>       | Round x toward zero.                                   |
| <code>floor(x)</code>     | Round x toward negative infinity.                      |
| <code>complex(a,b)</code> | Complex array (z = a + bi).                            |
| <code>real(x)</code>      | Real part of complex number.                           |
| <code>image(x)</code>     | Imaginary part of complex number.                      |
| <code>conj(x)</code>      | Complex conjugate of x.                                |
| <code>log(x)</code>       | Natural logarithm of x.                                |
| <code>log10(x)</code>     | Common logarithm of x.                                 |
| <code>exp(x)</code>       | Exponential of x (e <sup>x</sup> ).                    |
| <code>rem(a,b)</code>     | Remainder after division of a by b.                    |
| <code>mod(a,b)</code>     | Remainder after division of a by b (modulo operation). |
| <code>lcm(a,b)</code>     | Least common multiples of a and b.                     |
| <code>gcd(a,b)</code>     | Greatest common multiples of a and b.                  |
| <code>nthroot(a,n)</code> | Real n-th root of a.                                   |

## Trigonometric functions

|   |  |
|---|--|
| <code>#: sin, cos, tan, sec, or cot.</code> | sine, cosine, tangent, secant, or cotangent.       |
| <code>#/#d(x)</code>                        | # of x in radians/degrees.                         |
| <code>#h(x)</code>                          | Hyperbolic # of x.                                 |
| <code>a#/#d(x)</code>                       | Inverse # of x in radians/degrees.                 |
| <code>a#h(x)</code>                         | Inverse hyperbolic # of x.                         |
| <code>atan2/atan2d(x)</code>                | Four-quadrant inverse tan of x in radians/degrees. |
| <code>hypot(x)</code>                       | Square root of sum of squares of x.                |
| <code>deg2rad(x)</code>                     | Convert x from degrees to radians.                 |
| <code>rad2deg(x)</code>                     | Convert x from radians to degrees.                 |



# Matlab Cheat Sheet

## Some nifty commands

|                         |  |
|-------------------------|--|
| clc                     | Clear command window                   |
| clear                   | Clear system memory                    |
| clear x                 | Clear x from memory                    |
| commandwindow           | open/select commandwindow              |
| whos                    | lists data structures                  |
| whos x                  | size, bytes, class and attributes of x |
| ans                     | Last result                            |
| close all               | closes all figures                     |
| close(H)                | closes figure H                        |
| winopen(pwd)            | Open current folder                    |
| class(obj)              | returns objects class                  |
| save filename           | saves all variables to .mat file       |
| save filename x,y       | saves x,y variables to .mat file       |
| save -append filename x | appends x to .mat file                 |
| load filename           | loads all variables from .mat file     |
| ver                     | Lists version and toolboxes            |
| beep                    | Makes the beep sound                   |
| doc function            | Help/documentation for function        |
| docsearch string        | search documentation                   |
| web google.com          | opens webaddress                       |
| inputdlg                | Input dialog box                       |
| methods(A)              | list class methods for A               |

## Statistical commands

|   |                          |
|---|--------------------------|
| distrnd   | random numbers from dist |
| distpdf   | pdf from dist            |
| distcdf   | cdf dist                 |
| distrnd   | random numbers from dist |
| hist(x)   | histogram of x           |
| histfit(x)  | histogram and            |
| *Standard distributions (dist): norm, t, f, gam, chi2, bino |                          |
| *Standard functions: mean,median,var,cov(x,y),corr(x,y),    |                          |
| *quantile(x,p) is <u>not</u> textbook version.              |                          |
| (It uses interpolation for missing quantiles.               |                          |

## Keyboard shortcuts

|                   |  |
|-------------------|--|
| edit filename     | Opens filename in editor                           |
| Alt               | Displays hotkeys                                   |
| F1                | Help/documentation for <u>highlighted</u> function |
| F5                | Run code   |
| F9                | Run <u>highlighted</u> code                        |
| F10               | Run code line                                      |
| F11               | Run code line, enter functions                     |
| Shift+F5          | Leave debugger                                     |
| F12               | Insert break point                                 |
| Ctrl+Page up/down | Moves between tabs                                 |
| Ctrl+shift        | Moves between components                           |
| Ctrl+C            | Interrupts code                                    |
| Ctrl+D            | Open <u>highlighted</u> codes file                 |
| Ctrl+ R/T         | Comment/uncomment line                             |
| Ctrl+N            | New script   |
| Ctrl+W            | Close script                                       |
| Ctrl+shift+d      | Docks window                                       |
| Ctrl+shift+u      | Undocks window                                     |
| Ctrl+shift+m      | max window/restore size                            |

## Built in functions/constants

|   |                           |
|---|---------------------------|
| abs(x)  | absolute value            |
| pi  | 3.1415...                 |
| inf   | ∞                         |
| eps   | floating point accuracy   |
| 1e6   | 10 <sup>6</sup>           |
| sum(x)  | sums elements in x        |
| cumsum(x)   | Cummulative sum           |
| prod  | Product of array elements |
| cumprod(x)  | cummulative product       |
| diff  | Difference of elements    |
| round/ceil/fix/floor                                  | Standard functions..      |
| *Standard functions: sqrt, log, exp, max, min, Bessel |                           |
| *Factorial(x) is only precise for x < 21              |                           |

## Cell commands

|                                       |                             |
|---------------------------------------|-----------------------------|
| A cell can contain any variable type. |                             |
| x=cell(a,b)                           | a x <b>x</b> cell array     |
| x(n,m)                                | access cell n,m             |
| cell2mat(x)                           | transforms cell to matrix   |
| cellfun('fname',C)                    | Applies fname to cells in C |

## Strings and regular expressions

|          |                                      |
|----------|--------------------------------------|
| strcmp   | compare strings (case sensitive)     |
| strcmpi  | compare strings (not case sensitive) |
| strncomp | as strcmp, but only n first letters  |
| strfind  | find string within a string          |
|          | , gives start position               |
| regex    | Search for regular expression        |

## Logical operators

|   |                     |
|---|---------------------|
| &&  | Short-Circuit AND.  |
| &   | AND                 |
|   | Short-Circuit or    |
|   | or                  |
| -   | not                 |
| ==  | Equality comparison |
| ~=  | not equal           |
| isa(obj, 'class_name')                                      | is object in class  |
| *Other logical operators: <,>,>=,<=                         |                     |
| *All <u>above</u> operators are <u>elementwise</u>          |                     |
| *Class indicators: isnan, isequal, ischar, isinf, isvector  |                     |
| , isempty, isscalar, iscolumn                               |                     |
| *Short circuits only evaluate second criteria if            |                     |
| first criteria is passed, it is therefore faster.           |                     |
| And useful fpr avoiding errors occurring in second criteria |                     |
| *non-SC are bugged and short circuit anyway                 |                     |

## Variable generation

|                      |                            |
|----------------------|----------------------------|
| j:k                  | row vector [j,j+1,...,k]   |
| j:i:k                | row vector [j,j+1,...,k],  |
| linspace(a,b,n)      | n points linearly spaced   |
|                      | and including a and b      |
| NaN(a,b)             | a x b matrix of NaN values |
| ones(a,b)            | a x b matrix of 1 values   |
| zeros(a,b)           | a x b matrix of 0 values   |
| meshgrid(x,y)        | 2d grid of x and y vectors |
| [a,b]=deal(NaN(5,5)) | declares a and b           |
| global x             | gives x global scope       |

## Tables

|   |                             |
|---|-----------------------------|
| T=table(var1,var2,...,varN)                               | Makes table*                |
| T(rows,vars)  | get sub-table               |
| T(rows,vars)  | get data from table         |
| T.var or T.(varindex)                                     | all rows of var             |
| T.var(rows)   | get values of var from rows |
| summary(T)  | summary of table            |
| T.var3(T.var3>5)=5  | changes some values         |
| T.Properties.Varnames                                     | Variable names              |
| T = array2table(A)  | ! make table from array     |
| T = innerjoin(T1,T2)                                      | innerjoin                   |
| T = outerjoin(T1,T2)                                      | outerjoin !                 |
| Rows and vars indicate rows and variables.                |                             |
| tables are great for large datasets, because they         |                             |
| use less memory and allow faster operations.              |                             |
| *rowfun is great for tables, much faster than eg. looping |                             |

## matrix and vector operations/functions

|                |                                       |
|----------------|---------------------------------------|
| x=[1, 2, 3]    | 1x3 (Row) vector                      |
| x=[1; 2; 3]    | 3x1 (Column) vector                   |
| x=[1, 2; 3, 4] | 2x2 matrix                            |
| x(2)=4         | change index value nr 2               |
| x(:)           | All elements of x (same as x)         |
| x(j:end)       | j'th to last element of x             |
| x(2:5)         | 2nd to 5th element of x               |
| x(:,j)         | all j row elements                    |
| x(:,j)         | all j column elements                 |
| diag(x)        | diagonal elements of x                |
| x.*y           | Element by element multiplication     |
| x./y           | Element by element division           |
| x+y            | Element by element addition           |
| x-y            | Element by element subtraction        |
| A^n            | normal/Matrix power of A              |
| A.^n           | Elementwise power of A                |
| A'             | Transpose                             |
| inv(A)         | Inverse of matrix                     |
| size(x)        | Rows and Columns                      |
| eye(n)         | Identity matrix                       |
| sort(A)        | sorts vector from smallest to largest |
| eig(A)         | Eigenvalues and eigenvectors          |
| numel(A)       | number of array elements              |
| x(x>5)=0       | change elemnts >5 to 0                |
| x(x>5)         | list elements >5                      |
| find(A>5)      | Indices of elements >5                |
| find(isnan(A)) | Indices of NaN elements               |
| [A,B]          | concatenates horizontally             |
| [A;B]          | concatenates vertically               |

For functions on matrices, see `bsxfun`, `arrayfun` or `repmat`  
 \*if `arrayfun/bsxfun` is passed a `gpuArray`, it runs on GPU.  
 \*Standard operations: `rank`, `rref`, `kron`, `chol`  
 \*Inverse of matrix `inv(A)` should almost never be used, use `RREF` through `\` instead: `inv(A)b = A\b`.



# Cheatography

## Linear Algebra - MATH 232 Cheat Sheet

by fionaw via [cheatography.com/124375/cs/23750/](http://cheatography.com/124375/cs/23750/)

### Basic Equations

#### Network Flows

1. the flow in an arc is only in one directions
2. flow into a node = flow out of a node
3. flow into the network = flow out of the network

#### Balancing Chemical Equations

1. add x's before each combo and both side
2. carbo = x1 + 2(x3), set as system, solve

#### Matrix

|                    |                                  |
|--------------------|----------------------------------|
| augmented matrix   | variables and solution(rhs)      |
| coefficient matrix | coefficients only, no rhs matrix |

### Vectors, Norm, Dot Product

magnitude (norm) of vector v is  $\|v\|$ ;  $\|v\| \geq 0$

if  $k > 0$ , kv same direction as v      magnitude =  $k\|v\|$

if  $k < 0$ , kv opposite direction to v      magnitude =  $|k|\|v\|$

vectors in  $R^n$  (n = dimension)       $v = (v_1, v_2, \dots, v_n)$

$v = P_1P_2 = OP_2 - OP_1$       displacement vector

norm/magnitude of vector  $\|v\|$        $\text{sqrt}((v_1)^2 + (v_2)^2 + \dots)$

$\|v\| = 0$  iff  $v = 0$        $\|kv\| = |k|\|v\|$

unit vector u in same direct as v       $u = (1/\|v\|)v$

$e_1 = (1, 0, \dots)$  ...  $e_n = (0, \dots, 1)$  in  $R^n$       standard unit vector

$d(u, v) = \text{sqrt}((u_1 - v_1)^2 + (u_2 - v_2)^2 + \dots + (u_n - v_n)^2) = \|u - v\|$

$d(u, v) = 0$  iff  $u = v$

### Vectors, Norm, Dot Product (cont)

$u \cdot v = u_1v_1 + u_2v_2$       dot product

$\dots + u_nv_n$

$\|u\| \|v\| \cos(\theta)$

u and v are orthogonal if  $u \cdot v = 0$  ( $\cos(\theta) = 0$ )

a set of vectors is an orthogonal set iff  $v_i \cdot v_j = 0$ , if  $i \neq j$

a set of vectors is an orthonormal set iff  $v_i \cdot v_j = 0$ , if  $i \neq j$ , and  $\|v_i\| = 1$  for all i

$(u \cdot v)^2 \leq \|u\|^2 \|v\|^2$       Cauchy-Schwarz Inequality

or

$|u \cdot v| \leq \|u\| \|v\|$

$d(uv) \leq d(u, w) + d(w, v)$       Triangle Inequality

$\|u + v\| \leq \|u\| + \|v\|$

$\|v_1 + v_2 + \dots + v_k\| \leq \|v_1\| + \|v_2\| + \dots + \|v_k\|$

### Lines and Planes

a vector equation with parameter t       $x = x_0 + tv$ ,  $-\infty < t < +\infty$

solutin set for 3 dimension linear equation is a plane

if x is a point on this plane       $n \cdot (x - x_0) = 0$

(point-normal equation)  $A(x - x_0) + B(y - y_0) + C(z - z_0) = 0$

general/algebraic equation  $Ax + By + Cz = D$

two planes are parallel if  $n_1 = kn_2$ , orthogonal if  $n_1 \cdot n_2 = 0$

### Matrix Algebra, Identity and Inverse Matrix

$(A + B)_{ij} = (A)_{ij} + (B)_{ij}$        $(A - B)_{ij} = (A)_{ij} - (B)_{ij}$

$(cA)_{ij} = c(A)_{ij}$        $(A^T)_{ij} = (A)_{ji}$

$(AB)_{ij} = a_i b_{1j} + a_i b_{2j} + \dots + a_i b_{kj}$

Inner Product (number) is  $u^T v = u \cdot v$ , u and v same size

Outer Product (matrix) is  $uv^T$ , u and v can be any size

$(A^T)^T = A$        $(kA)^T = k(A)^T$

$(A+B)^T = A^T + B^T$        $(AB)^T = B^T A^T$

$\text{tr}(A^T) = \text{tr}(A)$        $\text{tr}(AB) = \text{tr}(BA)$

$u^T v = \text{tr}(uv^T)$        $\text{tr}(uv^T) = \text{tr}(vu^T)$

$\text{tr}(A) = a_{11} + a_{22} + \dots + a_{nn}$        $(A^T)_{ij} = A_{ji}$

Identity matrix is square matrix with 1 along diagonals

If A is  $m \times n$ ,  $A^T$  is  $n \times m$  and  $mA = A^T A$

a square matrix is invertible(nonsingular)  $AB = BA$

if:

B is the inverse of A       $B = A^{-1}$

if A has no inverse, A is not invertible (singular)

$\det(A) = ad - bc \neq 0$  is invertible

if A is invertible:  $(AB)^{-1} = B^{-1}A^{-1}$

$(A^n)^{-1} = A^{-n} = (A^{-1})^n$        $(A^T)^{-1} = (A^{-1})^T$

$(kA)^{-1} = 1/k(A^{-1})$ ,  $k \neq 0$

### Elementary Matrix and Unifying Theorem

elementary matrices are invertible

$A^{-1} = E_k E_{k-1} \dots E_2 E_1$

$[A | I] \rightarrow [I | A^{-1}]$

(how to find inverse of A)

$Ax = b$ ;  $x = A^{-1}b$