MATLAB Introduction

Introduction to Communication Engineering

SSY121

2021

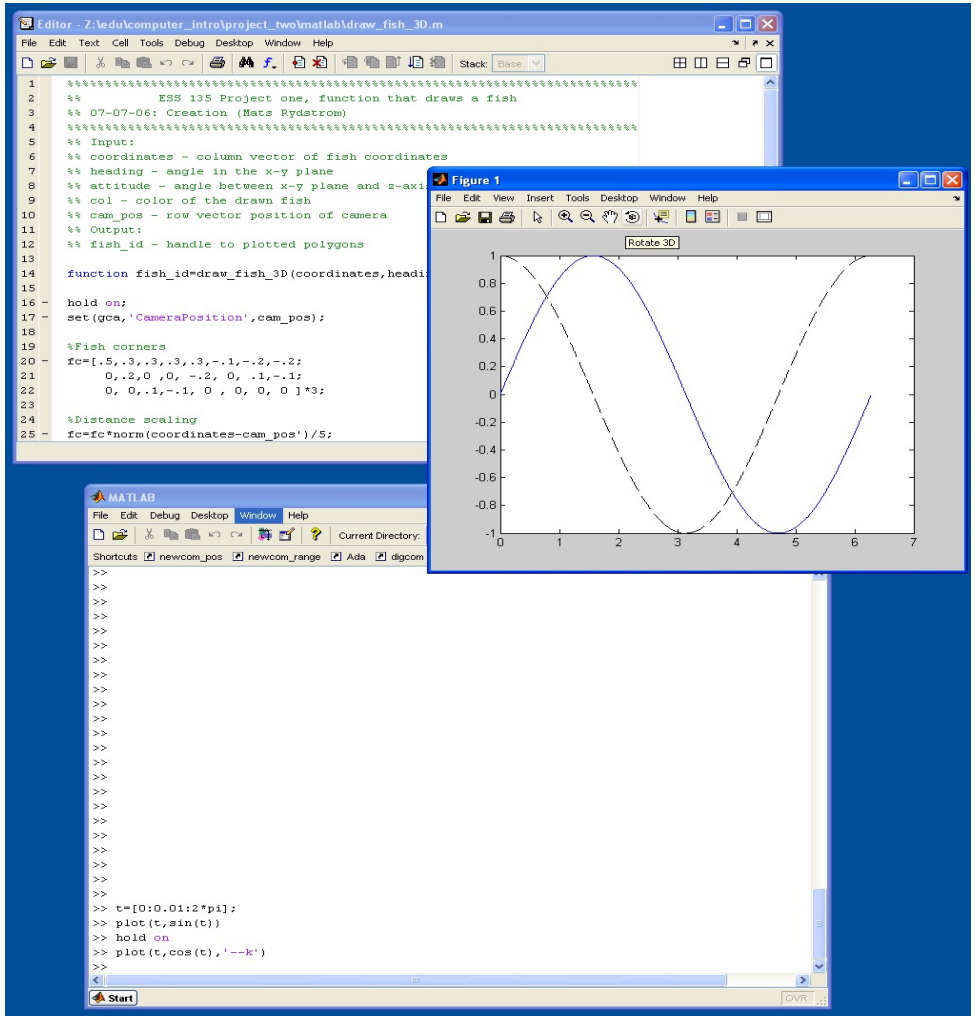
Acknowledgments to K. Nagalapur, M. Reza Khanzadi, J. Karout, C. Botella, Y. Sui, and W. Yibo

Practical stuff

- Slides can be downloaded from the course homepage
- Information about where to get the MATLAB with a student license can be found on: https://it.portal.chalmers.se/itportal/Student/Student?setlang=en
- Quick link for Chalmers to download MATLAB (Need license from above): https://se.mathworks.com/academia/tah-portal/chalmers-30650541.html
- Most computer rooms already have MATLAB: ES61, ES62, ES63, HB105, HB110, etc.

What is MATLAB?

- MATrix LABoratory
- Mathematical tool
- Used for simulations, visualization, and much more
- Widely used

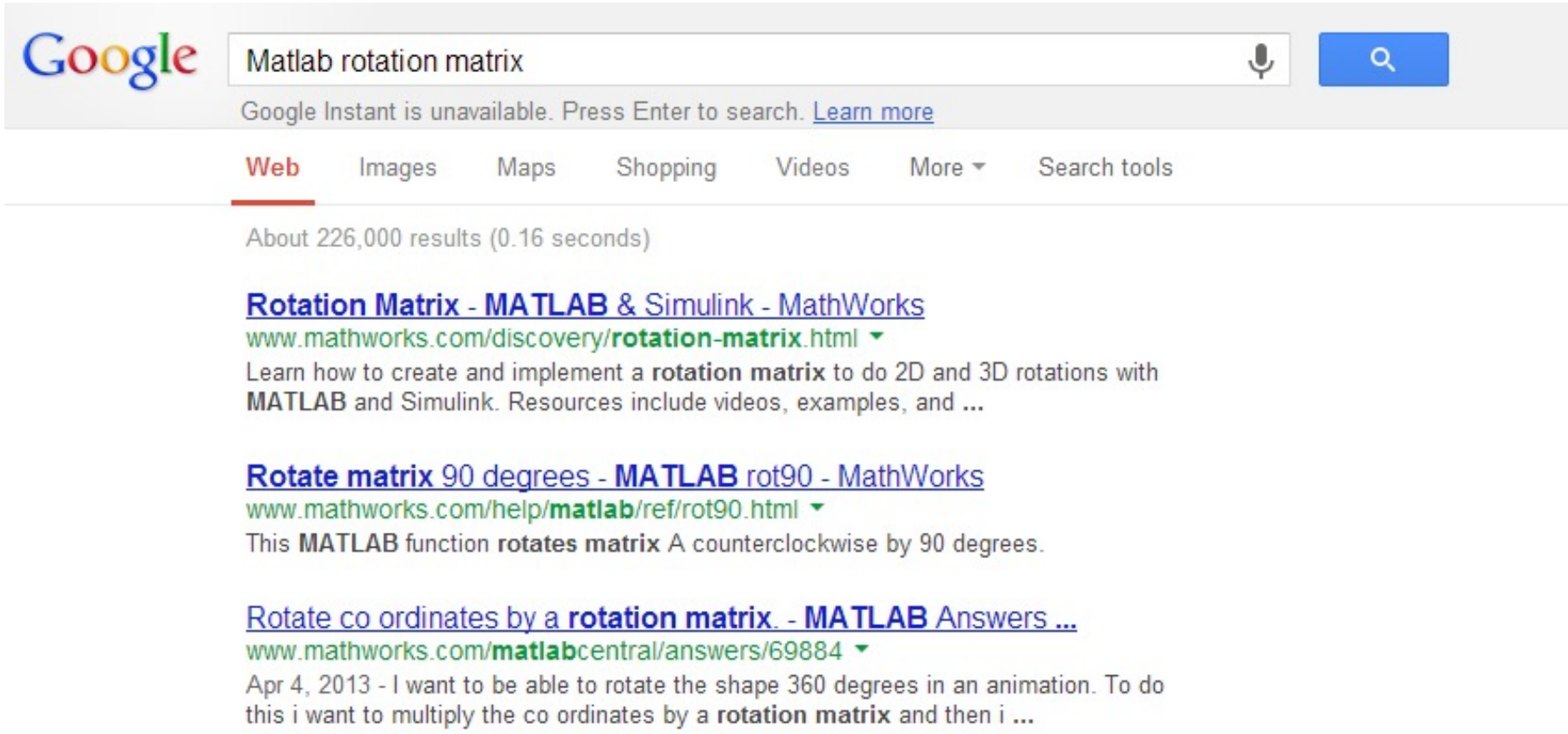


How to learn MATLAB

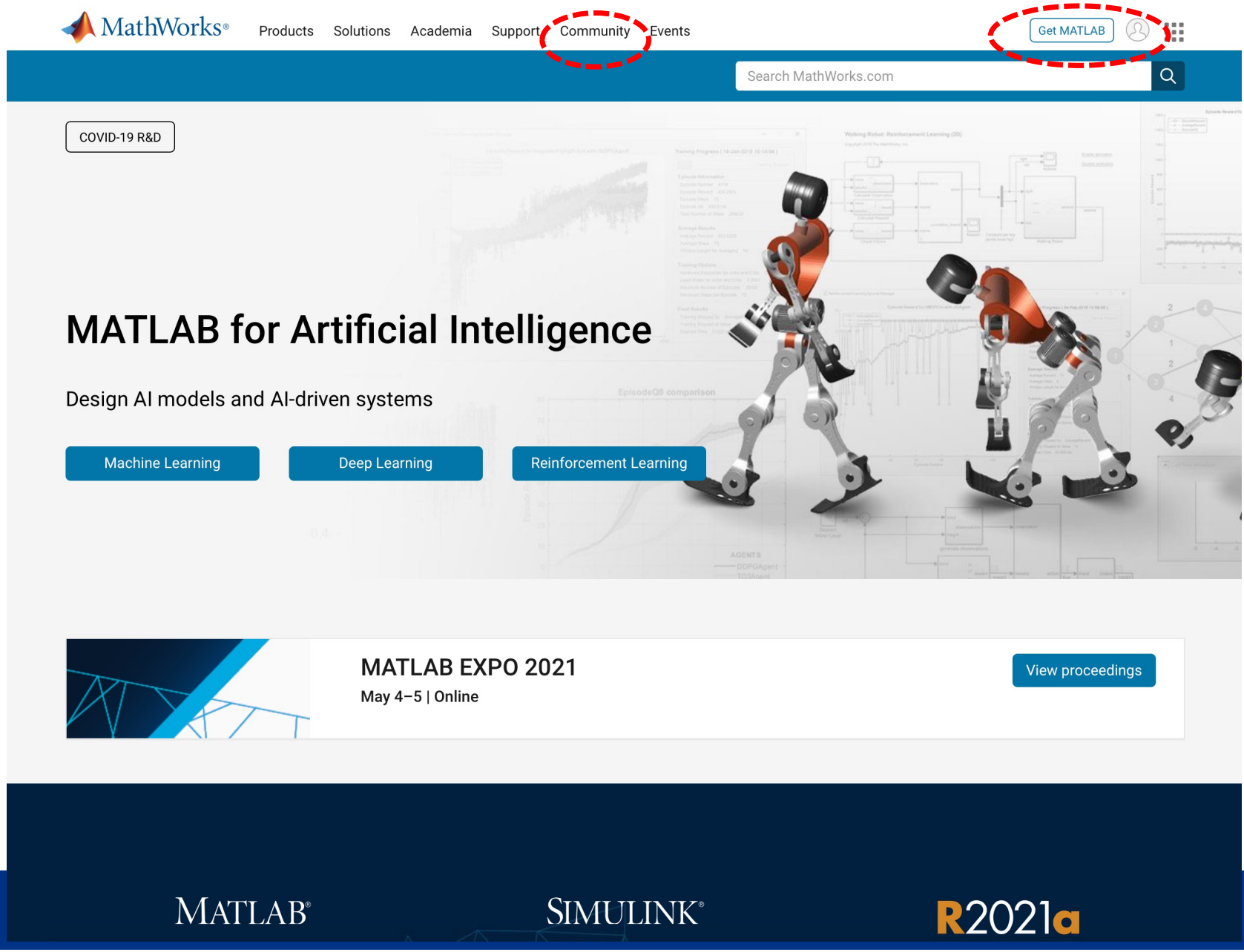
- Learning everything is just impossible!
- Getting some hands-on experience is necessary
- Learning how to learn is more important
  - Help
  - MATLAB central
  - Search engines

On-line resources

- Google is your friend!



www.mathworks.com



An open exchange for the MATLAB and Simulink user community

For you | Newest | Trending | Activities | All Community

ANNOUNCEMENT

Redesign of File Exchange Review Section

The community team is pleased to release a new design for File Exchange review section. The new design has addressed major pain points we heard from File Exchange users and improves the overall experience. . ...

Processing files using a for loop

MH Matt in Answers on 21 February 2012

I am trying to write a program to read in files and analyze each file one by one. The files are wav files, and I want to read them in, filter them with a filter I have already designed, plot frequency vs. time and do a spectrogram of each file. I am supposed to use uigetdir to find the directory of...

2 answers

Tags: file processing, sound waves, for loop

0 1129 0

Passing array as an input argument

MN Mayank Nautiyal in Answers on 06 October 2019

I want to pass an array [p1 p2 p3] in the function "arithmetic_decoding" as an input. function arithmetic_decoding(tag,n,[p1,p2,p3]) % this shows error "Unbalanced or unexpected parenthesis or bracket" Please help? %p1=0.4; %p2=0.5; %p3=0.1;

1 answer

Tags: array, function

0 324 0

YW

Yibo Wu

My Community Profile

Earn Your First Badges

Introduction to Community

Explore Community Areas

MATLAB Answers

Ask & Answer questions about

https://se.mathworks.com/videos.html#matlabgetstarted

Videos and Webinars

Search Videos | Videos

Videos Home | Search

Contact sales | Trial software

MATLAB Videos

- Introduction
- Getting Started
- What's New
- Data Science and Machine Learning
- Deep Learning
- Image Processing and Computer Vision
- Computational Finance
- Signal Processing and Wireless Communications

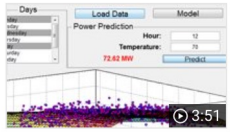
Introduction

What Is MATLAB?

Get an overview of MATLAB, the language of technical computing.

Analyzing and Visualizing Data with MATLAB

Programming and Developing Algorithms with MATLAB

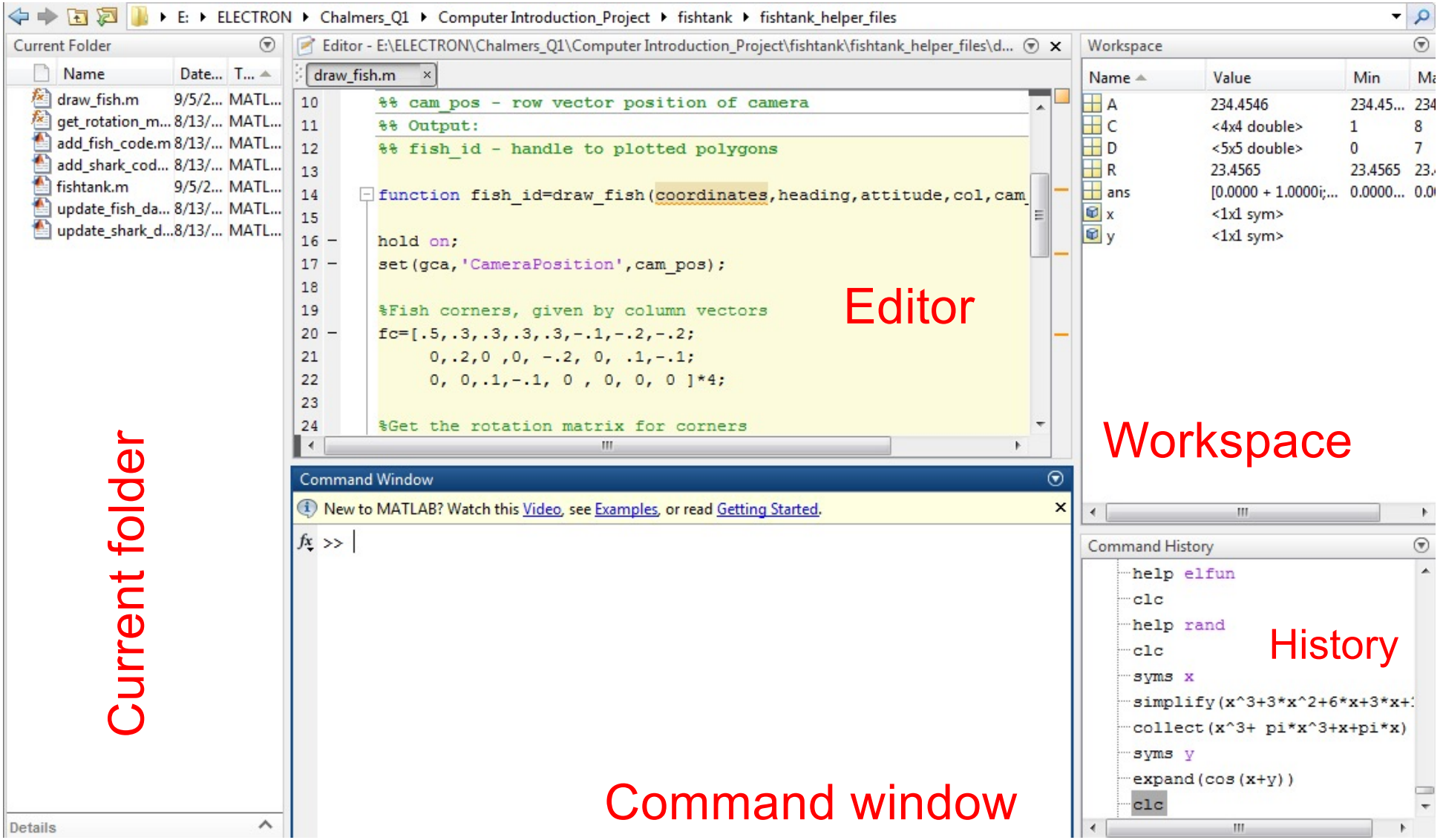


Developing and Deploying Applications with MATLAB

Technical Computing with MATLAB

Simulink Videos

Interface


E: ▸ ELECTRON ▸ Chalmers_Q1 ▸ Computer Introduction_Project ▸ fishtank ▸ fishtank_helper_files
Current Folder
Name Date... T...
draw_fish.m 9/5/2... MATL...
get_rotation_m... 8/13/... MATL...
add_fish_code.m 8/13/... MATL...
add_shark_cod... 8/13/... MATL...
fishtank.m 9/5/2... MATL...
update_fish_da... 8/13/... MATL...
update_shark_d... 8/13/... MATL...
Details
Editor - E:\ELECTRON\Chalmers_Q1\Computer Introduction_Project\fishtank\fishtank_helper_files\d...
draw_fish.m
10 %% cam_pos - row vector position of camera
11 %% Output:
12 %% fish_id - handle to plotted polygons
13
14 function fish_id=draw_fish(coordinates,heading,attitude,col,cam
15
16 hold on;
17 set(gca,'CameraPosition',cam_pos);
18
19 %Fish corners, given by column vectors
20 fc=[.5,.3,.3,.3,.3,-.1,-.2,-.2;
21 0,.2,0 ,0, -.2, 0, .1,-.1;
22 0, 0,.1,-.1, 0 , 0, 0, 0]*4;
23
24 %Get the rotation matrix for corners
Editor
Workspace
Name Value Min Ma
A 234.4546 234.45... 234
C <4x4 double> 1 8
D <5x5 double> 0 7
R 23.4565 23.4565 23.
ans [0.0000 + 1.0000i;... 0.0000... 0.0
x <1x1 sym>
y <1x1 sym>
Workspace
Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.
>>
Command window
Command History
help elfun
clc
help rand
clc
syms x
simplify(x^3+3*x^2+6*x+3*x+
collect(x^3+ pi*x^3+x+pi*x)
syms y
expand(cos(x+y))
clc
History
Current folder

MATLAB vectors and matrices

```
>> v=[1,2,3]
```
Row vector

```
v =

    1    2    3
```

```
>> v2=[1;2;3]
```
Column vector

```
v2 =

    1
    2
    3
```

```
>> v2'
```
Vector transpose (conjugate transpose for complex numbers)

```
ans =

    1    2    3
```

```
>> A=[1,2,3;4,5,6;7,8,9]
```
Matrix

```
A =

    1    2    3
    4    5    6
    7    8    9
```

```
>> B=[v2,v2,v2]
```
Another matrix

```
B =

    1    1    1
    2    2    2
    3    3    3
```

```
>> C=[v;v;v];
```
; suppresses output!

Working with vectors and matrices

```
>> v=[0:0.1:10];      Row vector from 0 to 10
>> v(51)              The 51th element

ans =

    5

>> v(51:55)          Element 51 to 55

ans =

   5.0000   5.1000   5.2000   5.3000   5.4000
```

```
>> A=[1,2,3;4,5,6;7,8,9]

A =

     1     2     3
     4     5     6
     7     8     9

>> A(1,2)     A(row,column)

ans =

     2

>> A(:,2)     : means all rows

ans =

     2
     5
     8
```

Working with vectors and matrices

```
>> v=[1;2;3];
>> v'*v          Inner product

ans =

    14

>> A=[1,2,3;4,5,6;7,8,9];
>> A*A                      Matrix product

ans =

    30    36    42
    66    81    96
   102   126   150
```

```
>> zeros(2,3)    Matrix of zeros

ans =

     0     0     0
     0     0     0

>> ones(2,2)     Matrix of ones

ans =

     1     1
     1     1

>> eye(3)        Identity matrix

ans =

     1     0     0
     0     1     0
     0     0     1
```

Getting help

```
>> help
HELP topics

matlab\general    - General purpose commands.
matlab\ops        - Operators and special characters.
matlab\lang       - Programming language constructs.
matlab\elmat      - Elementary matrices and matrix manipulation.
matlab\elfun      - Elementary math functions.
matlab\specfun    - Specialized math functions.
matlab\matfun     - Matrix functions - numerical linear algebra.
…
```

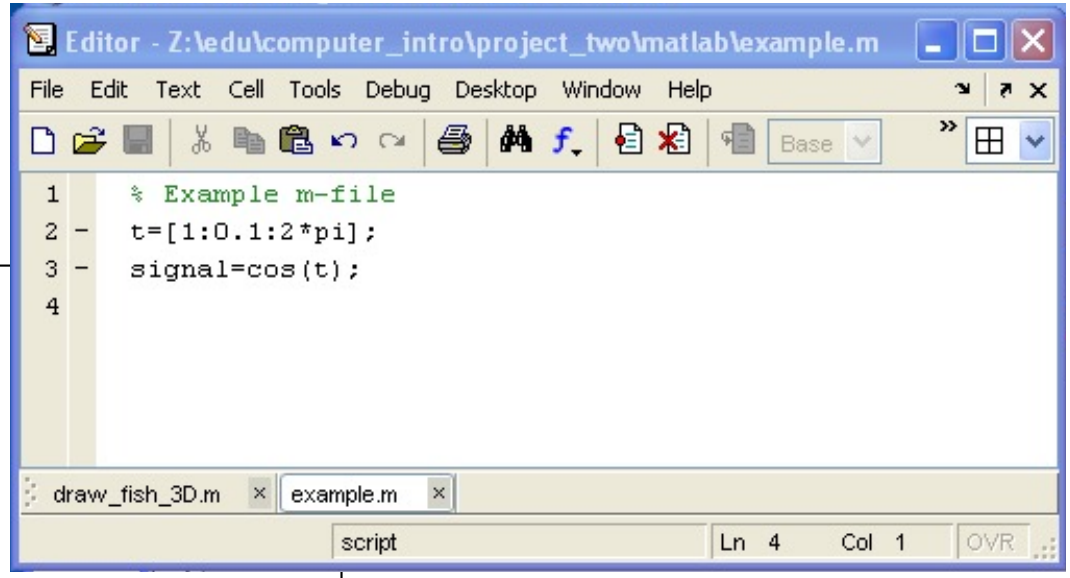
```
>> help elmat
  Elementary matrices and matrix
manipulation.

  Elementary matrices.
    zeros       - Zeros array.
    ones        - Ones array.
    eye         - Identity matrix.
    repmat      - Replicate and tile array.
…
```

```
>> lookfor permutation
PERMS  All possible permutations.
COLAMD Column approximate minimum
degree permutation.
COLMMD Column minimum degree
permutation.
COLPERM Column permutation.
DMPERM Dulmage-Mendelsohn permutation
….
```

```
>> help zeros

  ZEROS  Zeros array.

  ZEROS(N) is an N-by-N matrix of zeros.

  ZEROS(M,N) or ZEROS([M,N]) is an M-by-N matrix of zeros.

  ZEROS(M,N,P,...) or ZEROS([M N P ...]) is an M-by-N-by-P-by-... array of
  zeros.

  ZEROS(SIZE(A)) is the same size as A and all zeros.

  ZEROS with no arguments is the scalar 0.

  ZEROS(M,N,...,CLASSNAME) or ZEROS([M,N,...],CLASSNAME) is an
  M-by-N-by-... array of zeros of class CLASSNAME.

  Similarly,
  ONES(N)
```

M-files

- Very useful when executing a sequence of commands
- All variables are global




```
>> pwd

ans =

Z:\edu\computer_intro\lecture_slides

>> ls
.                Lecture_1b_2009.ppt   Lecture_3_2009.ppt   example.m
..               Lecture_1a_2009.ppt   Lecture_2_2009.ppt   my_function.m
ITS_masters_intro.pdf

>> example

signal =

  Columns 1 through 9

    0.5403    0.4536    0.3624    0.2675    …
```

Programming constructs

- MATLAB supports all common constructs: conditionals and loops.
- Useful when writing your m-file scripts
- Notice the double equal sign for comparisons.
- The loop variable can be noninteger

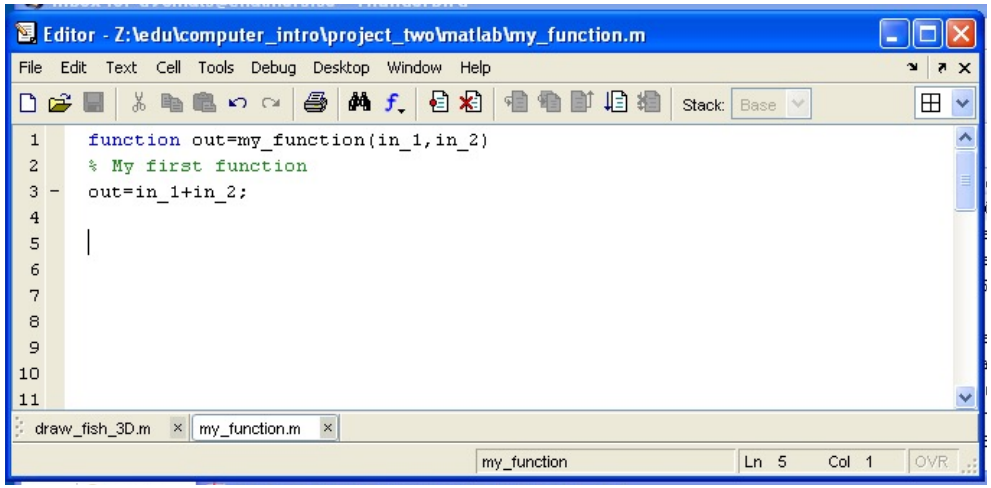
```
for J=1:10,
    var=var+1;
end

while J<10,
    var=var+1;
End
```

```
if a==b,
    var=var+1;
else
    var=var-1;
end

switch a,
    case 1,
        do
something;
    case 2,
        do
something;
    otherwise
        do
something;
end
```

User-defined functions

- Works like any in-built function
- There can be arbitrarily number of input/output arguments
- All variables are local
- The file should have the same name as the function
- The comment (preceded by %) at the beginning of the function will show up when user asks for help:

```
>> help my_function
```



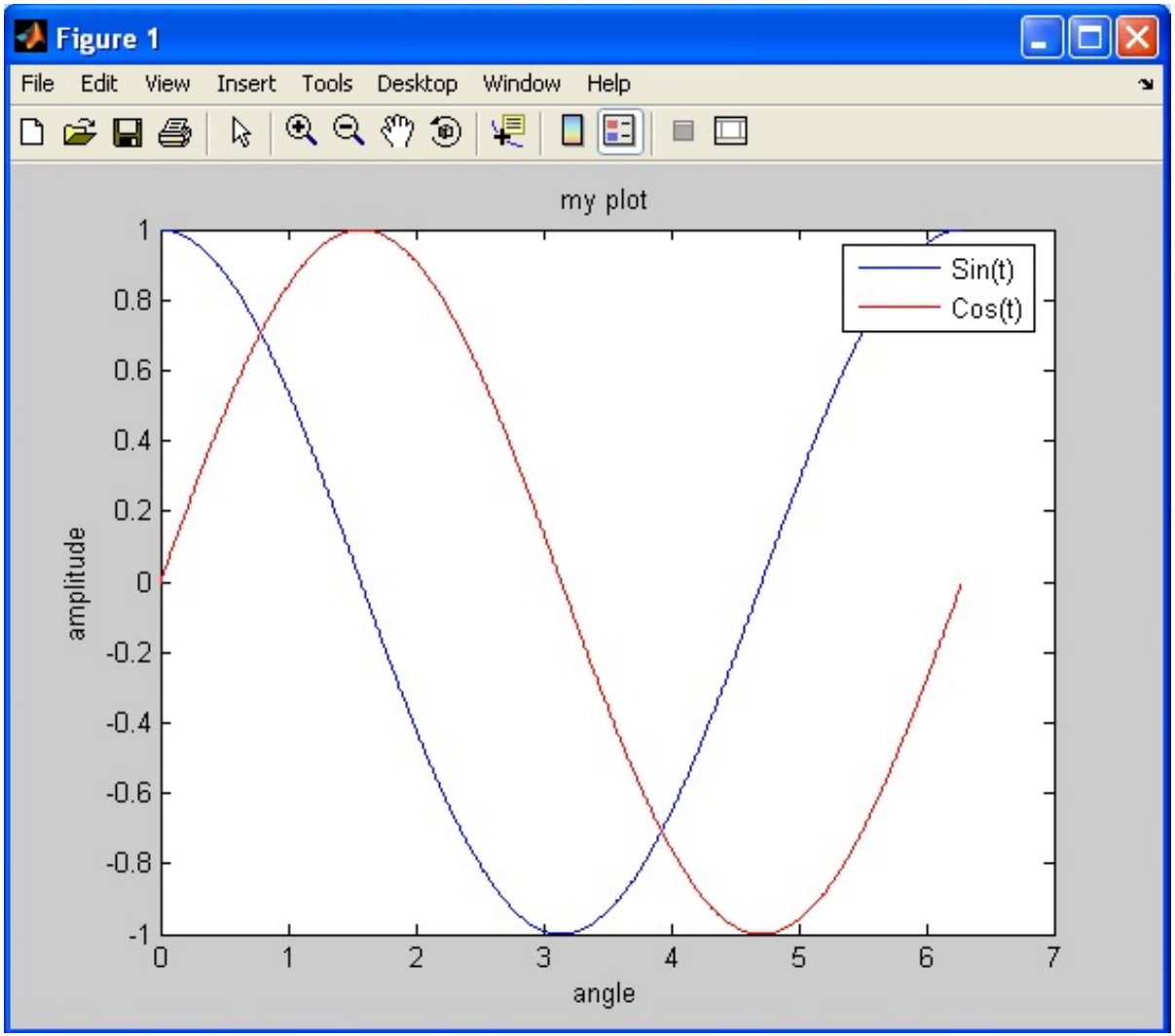

```
>> s=my_function(2,3)

s =

     5
```

Good programming practice:
Split your program into several functions that can be tested separately!

Debugging your code

- MATLAB has numerous functions for debugging code, see help codetools
- Insert breakpoints and step
- Some of the most useful include:
  - keyboard; placed in an M-file stops execution and gives control to the user
  - dbstop if <condition> terminates execution and gives control to the user to examine the local workspace
  - dbquit quits debug mode
- The MATLAB editor has more tools, check them out!

```
>> t=[0:0.01:2*pi];
>> plot(t,cos(t),'b')
>> hold on
>> plot(t,sin(t),'r')
>> xlabel('angle')
>> ylabel('amplitude')
>> title('my plot')
>> legend('Sin(t)','Cos(t)')
```



Type >>help plot for more information!

Session II

Directories and the file system

```
>> pwd        Print current directory
ans =
Z:\edu\computer_intro\lecture_slides

>> ls                 List files in directory
>> mkdir newdir       Make new directory
>> cd newdir          Change directory
>> exit               Exit MATLAB
```

```
>> who          List all variables in memory
>> whos         Long form of WHO
>> clear all    Clear all variables
```

```
>> load file                  Loads file.mat into MATLAB
>> load file.txt              Loads a text file into MATLAB
>> save filename              Saves workspace variables to 'filename.mat'
>> save filename x y          Saves only variables x and y
>> save filename x -ASCII     Saves x to text file
```

Looking for functions and getting help

>> lookfor integral	List commands related to integral
>> help	List all help topics
HELP topics	
matlab\general	General purpose commands
matlab\ops	Operators and special characters
>> demo	Run demostrations

Check out the following:

elmat – elementary matrices and matrix manipulation
elfun – elementary math functions
lang – programming language
graph2d – 2-D plotting

And the rest of help matlab* when you find the time, it is time well spent!

Complex numbers

```
>> x=3+5*i
>> x=3+5*j
>> x=complex(3,5)
x =
   3.0000 + 5.0000i
```

The expression A+i*B or A+j*B will give identical results if i or j has not been assigned.

```
>> x*(1+2*i)

ans =

  -7.0000 +11.0000i

>> sqrt(x)

ans =

   2.1013 + 1.1897i

x*(1+2*1i)

ans =

  -7.0000 +11.0000i
```

Use **1i** or **1j** instead!

Be careful when using **i** and **j** as variables

```
>> real(x)      Complex real part
>> imag(x)      Complex imaginary part
>> abs(x)       Complex modulus
>> angle(x)     Complex argument
>> conj(x)      Complex conjugate
```

Transpose and Hermitian Transpose

```
>> A = randn(3,3)+1i*randn(3,3)

A =

  -1.2141 + 0.0326i   1.5326 + 1.5442i  -0.2256 - 0.7423i
  -1.1135 + 0.5525i  -0.7697 + 0.0859i   1.1174 - 1.0616i
  -0.0068 + 1.1006i   0.3714 - 1.4916i  -1.0891 + 2.3505i

>> A_transpose = A.'  % Transpose

A_transpose =

  -1.2141 + 0.0326i  -1.1135 + 0.5525i  -0.0068 + 1.1006i
   1.5326 + 1.5442i  -0.7697 + 0.0859i   0.3714 - 1.4916i
  -0.2256 - 0.7423i   1.1174 - 1.0616i  -1.0891 + 2.3505i

>> A_hermitian = A'     % Conjugate  transpose

A_hermitian =

  -1.2141 - 0.0326i  -1.1135 - 0.5525i  -0.0068 - 1.1006i
   1.5326 - 1.5442i  -0.7697 - 0.0859i   0.3714 + 1.4916i
  -0.2256 + 0.7423i   1.1174 + 1.0616i  -1.0891 - 2.3505i
```

Basic vector math

```
>> a=[1:4]'; b=[4:7]';      Transposed row vectors
>> c=a'*b;                  Inner product
>> C=M*A;                   Matrix product
>> d=M*b;                   Matrix-vector product
>> a=a+b;
>> M=M^2;
>> M=M.^2;                  Element-wise operations
>> c=a.*b;
>> c=1./b;
>> c=b.^2;
```

```
>> [1 2 3] *[1 2 3]'
>> [1 2 3]'* [1 2 3]
>> [1 2 3] *[1 2 3]
>> [1 2 3].*[1 2 3]
>> [1 2 3]./[4 5 6]
```

Useful commands

```
>> x=1:12;
>> n=length(x);          Length of vector
>> X=reshape(x,4,3);     Change size
>> X=reshape(x,3,4);
>> n=size(x);            Size of array
>> [m,n] = size(X);      Number of rows and columns in X
>> isequal(x,x);         True if arrays are numerically equal
>> isempty(x);           True for empty array
>> isreal(x);            True for real array
>> strcmp(str1,str2)     Compare strings
```

Some constants

```
>> pi, Inf, NaN, i, ans, eps, e

>> R=214.123;
>> format long, R
>> format short, R          Default mode
>> format short e, R
>> format                   Restore the default mode
>> format compact           Reduces the number of blank lines on the screen

>> linspace(x1,x2,n)        Generates n linearly equally spaced points
                            between x1 and x2.
```

```
>> linspace(1,4,5)
>> v1=1:5
>> v2=1:0.5:3
>> v3=5:-1:-2
```

Matrix operations

```
>> inv(A)          Inverse of a square matrix
>> pinv(B)         Pseudoinverse
>> E=eig(A)        Eigenvalues of a square matrix
>> [V,D]=eig(A)    Eigenvalues and eigenvectors
>> det(A)          Determinant of a square matrix
>> norm(A)         Matrix norm
>> norm(A,1)
>> norm(A,2)
>> norm(A,'fro')
>> norm(A,p)
>> norm(A,inf)
>> [Q,R]=qr(A)     Orthogonal-triangular decomposition
>> rank(A)         Matrix rank
```

Efficient solution of matrix equations: Ax=b

```
>> x=A\b;          Same as inv(A)*b (solved by least
                   squares for non-square matrices)
>> x=b/A;          Same as b*inv(A)
```

Notice that b/A=(A ' \b ') '

```
>> C=diag([4:7])
>> D=diag([4:7],-1)
>> C=C+ones(size(C))
>> diag(C)
>> tril(C)
>> triu(C)
>> 5*ones(size(C))
>> 5*diag([ones(1,5)])
>> 5*eye(5)
```

```
>> C=[1 2 3;1 2 3;1 2 3]
>> sum(C,1)
>> sum(C,2)
```

Useful functions

Most math functions are implemented, see help elfun etc.

>> m=mean(a);	Average or mean value
>> v=var(a);	Variance
>> f=fft(a);	Discret Fourier transform
>> abs(a);	Absolute value
>> norm(a);	Vector norm
>> sum(a)	Sum of elements
>> a=randn(n)	Normally distributed random numbers.
>> a=rand(n)	Uniformly distributed pseudo-random numbers Read carefully the property 'state'!

Generate uniformly distributed values from the interval [a, b]

```
>> r = a + (b-a).*rand(100,1);
```

Generate normally distributed values with mean m_x and standard deviation s_x.

```
>> r = m_x + s_x.*randn(100,1);
```

Read carefully the input/output parameters of a function!

Useful functions

Count elements in vector that fulfil some condition:

```
>> v=[1,2,3,4,5,6,7,8,9,10];
>> length(find(v>5))
>> [i,j]=find(v>5)
>> A=randn(3)
>> [i,j]=find(A>1)
```

Find the five smallest elements in a vector v:

```
>> [sorted_v, idx]=sort(v);
>> v(idx(1:5))
>> y = sort(X,dim,mode)
>> [Y,I] = sort(X,dim,mode)
```

```
>> C=randint(3,3,[-2 2])
>> C=sort(C,1,'ascend')
>> C=sort(C,1,'descend')
>> C=sort(C,2,'ascend')
>> C=sort(C,2,'descend')
```

Useful constructs

Do something to all elements in a vector v:

```
for element=1:length(v),
        v(element)= v(element)+1;
end
```

Do something to all columns in a matrix M:

```
for col=1:size(M,2),
        M(:,col)= M(:,col)*2;
end
```

>> break	Terminate execution of WHILE or FOR innermost loop
>> return	Return to invoking function
>> continue	Pass control to the next iteration of FOR or WHILE loop

User control

User input to scripts and functions

```
>> x=input('Type x value: ')
>> str=input('Type a string ','s'); %for strings
```

Get N points from the current axes and return

the X- and Y-coordinates in length N vectors X and Y.

```
>> [X,Y] = ginput(N)
```

Pause for a while

```
>> disp('pause for 5 seconds');
>> pause(5)
>> disp('pause until key is pressed');
>> pause
```

MATLAB string handling

```
>> str='This is an example. '
```

Notice the single quotation marks

```
>> str(4)
ans =
s
>> length(str)
ans =
    18
>> >> A=['Hello ' str ' Bye.']
A =
Hello, This is an example. Bye.
```

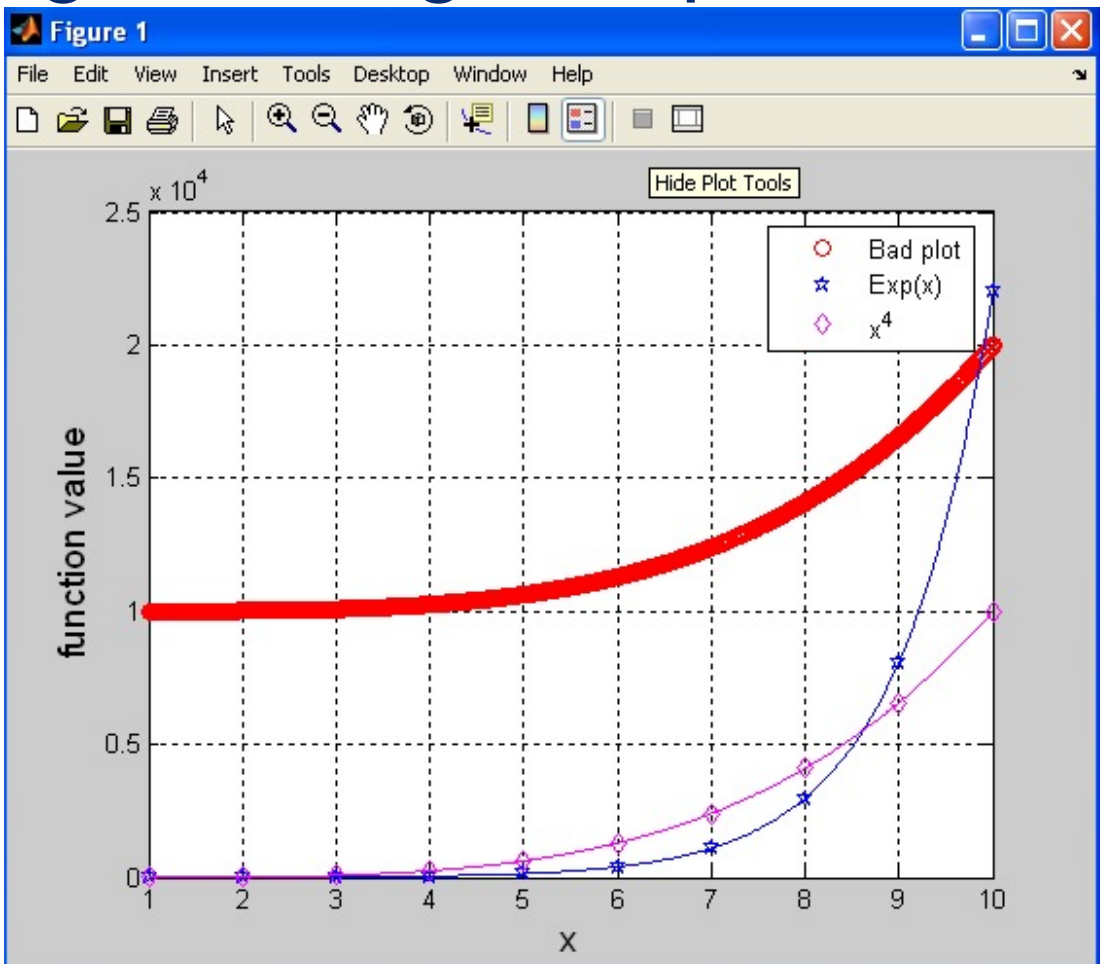
```
>> num2str      Convert numbers to a string
>> int2str      Convert integer to string
>> disp         Display a string
>> eval         Symbolic evaluation
>> for i=1:10
disp(['Iteration number ' int2str(i)]);
end
>> x=3;
>> eval('x+2');
>> for i=1:10
eval(['A',num2str(i),'=zeros(i);'])
end
```

eval([]) – Useful to generate variable names dynamically during the execution of a program

Tips and tricks for generating nice plots

- Remember that reports are often printed in black and white, so don't use weak colors (like yellow)
- Separate different curves with **markers** and **line styles** not only with different colors
- Use a sufficient font size
- Carefully consider whether to plot in log-scale

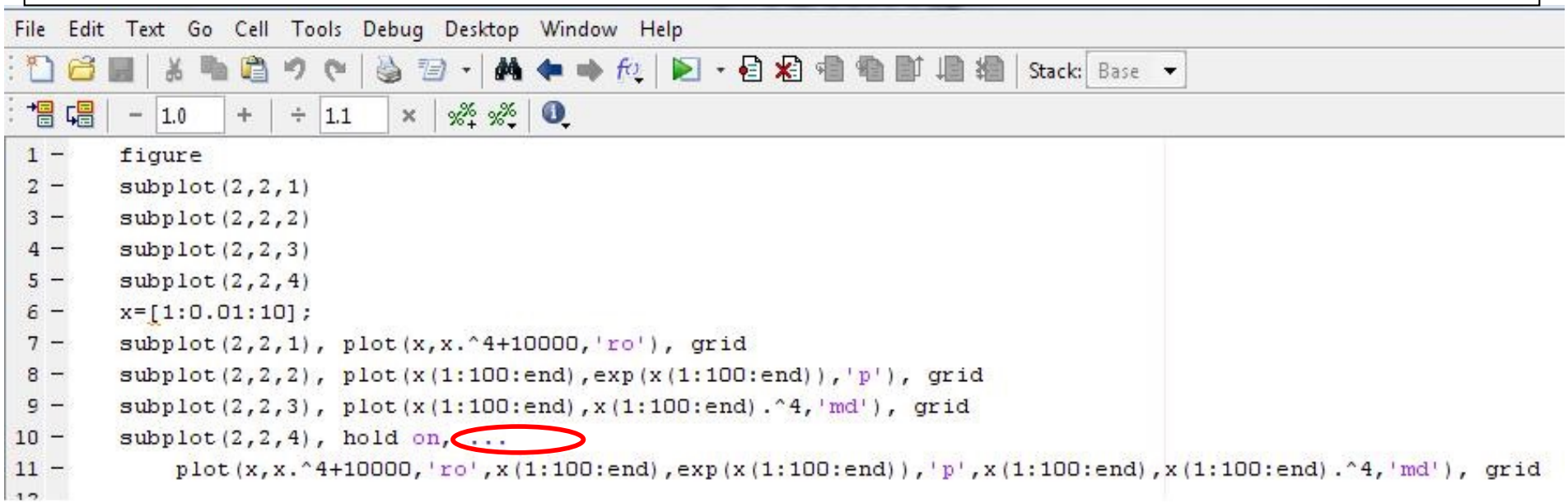
```
>> x=[1:0.01:10];
>> plot(x,x.^4+10000,'ro')
>> hold on;
>> plot(x(1:100:end),exp(x(1:100:end)),'p')
>> plot(x(1:100:end),x(1:100:end).^4,'md')
>> plot(x,exp(x))
>> plot(x,x.^4,'m')
>> legend('Bad plot','Exp(x)','x^4')
>> xlabel('x','FontSize',14)
>> ylabel('function value','FontSize',14)
>> grid on
```



- Check the 'edit' menu
- Check the different options for saving the figure (.fig by default)
- .fig file, can be edited later

Plotting

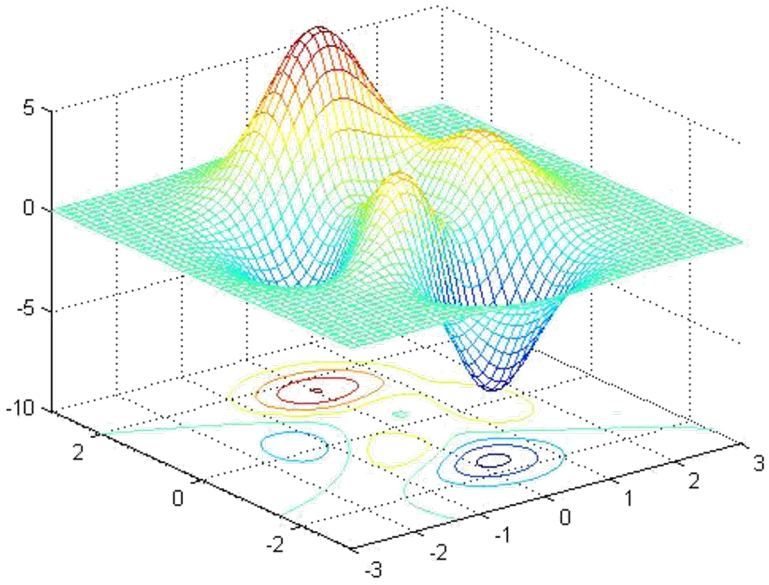
>> axis([xmin xmax ymin ymax])	Change the axis scaling on plots
>> hold on	
>> hold off	
>> subplot(m,n,p)	Breaks the Figure window into an m-by-n matrix
>> semilogx(t,x,' r')	Plot with logarithmic X axis
>> semilogy(t,x,' r')	Plot with logarithmic Y axis
>> loglog (t,x,' r')	Plot with logarithmic X and Y axis
>> daspect([X Y Z])	Sets the data aspect ratio

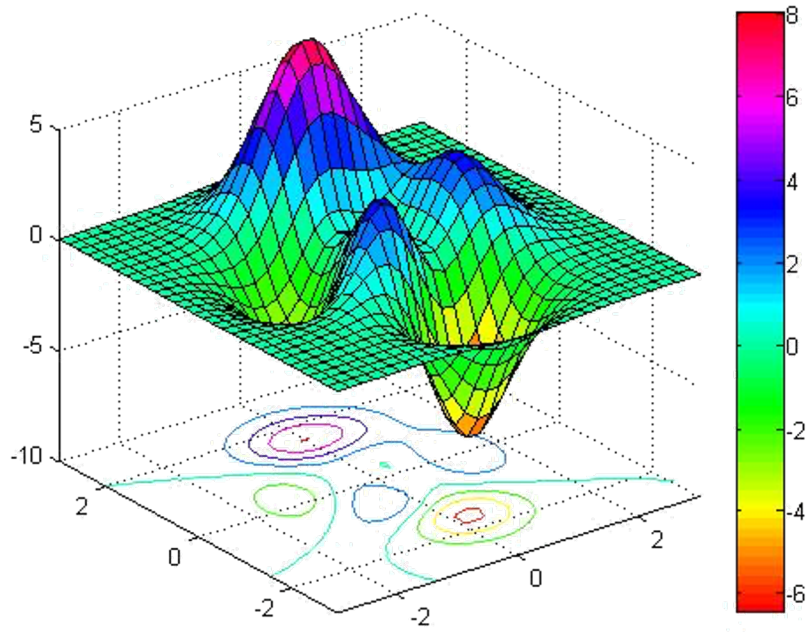


Plotting

>> stem	Discrete sequences
>> bar	Bar graph
>> hist	Histogram
>> stairs	Stairstep plot

For 3D plots: surf, mesh…



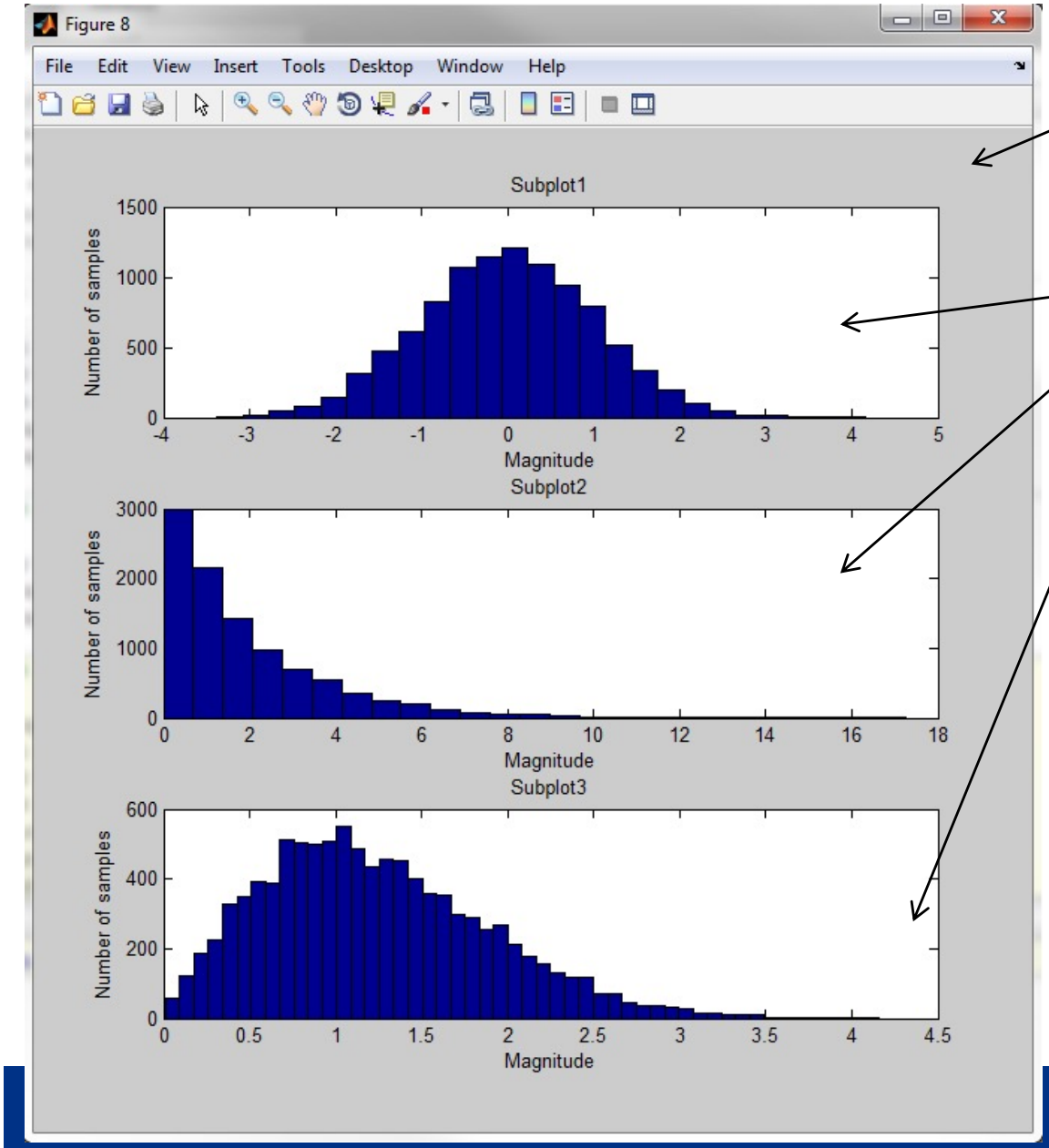


Readability and comments

- % Add comments for improving understanding and future reference

- Use %% to divide the script into blocks

Components of a MATLAB figure (Advanced)



Figure

Axes

```
>> get(gcf)
>> set(gcf,'Name','Histograms')
>> get(gca)
>> axes_hd = get(gcf,'children');
>> title(axes_hd(3),'SUB-PLOT 1')
```

Some advanced commands

- Use of global variables: GLOBAL
  - A single copy of the variable is shared between functions and workspace. Any assignment to that variable, in any function, is available to all the other functions declaring it GLOBAL.
- Use of STRUCT arrays:

```
>> field1 = 'name'; field2 = 'phone';
>> values1 = {'Keerthi','Kumar'};
>> values2 = {'1232','1890'};
>> contacts = struct(field1,values1,field2,values2);
```

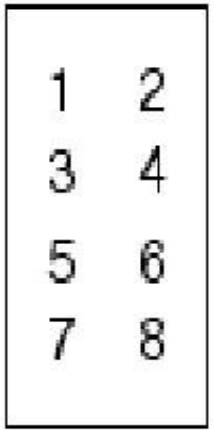
```
>> contacts(2)
ans =
    name: 'Kumar'
    phone: '1890'
```

```
>> contacts(1).phone
ans =
1232
```

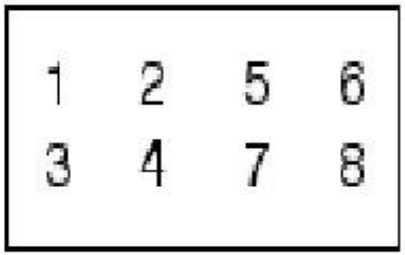
Some advanced commands

- Concatenations:

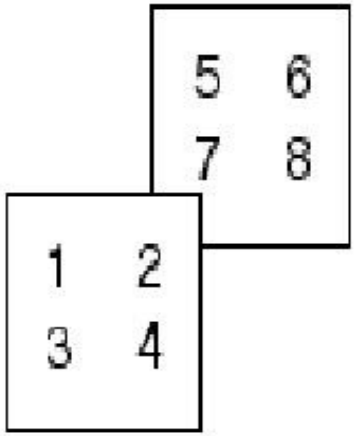
```
>> A=[1 2; 3 4]; B=[5 6; 7 8];
```



cat(1,A,B)



cat(2,A,B)



cat(3,A,B)

```
>> A=[ ]
>> for i=1:3
A=[A ones(3)]
end
```

Stores an empty matrix in A.

MATLAB toolboxes

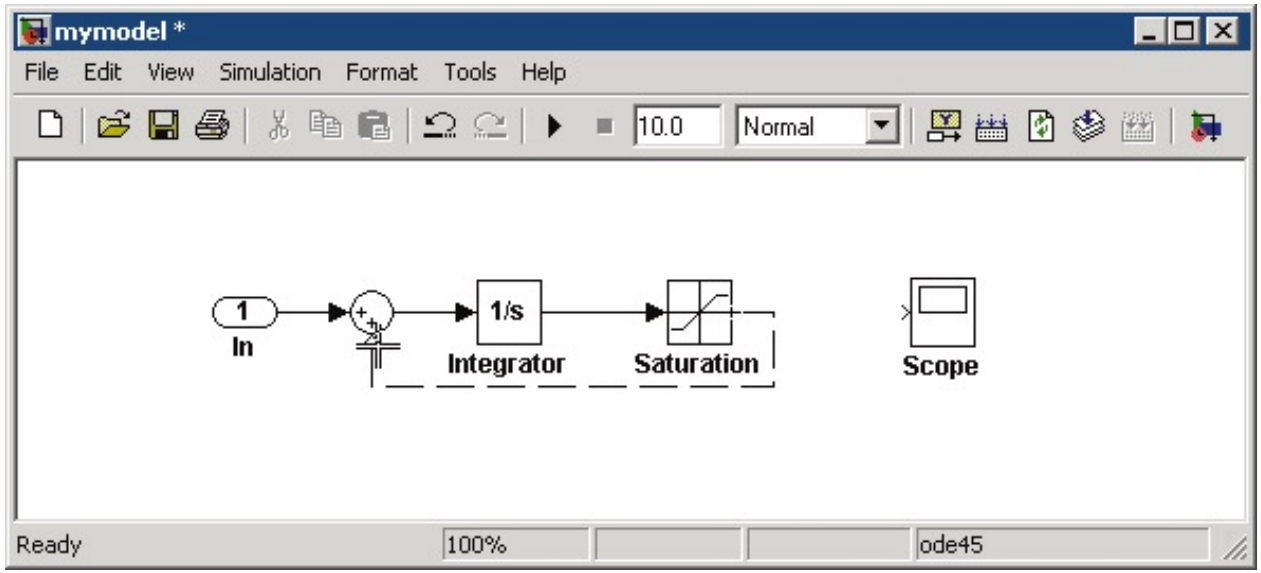
- Toolboxes are specialized collections of M-files (MATLAB language programs) built for solving particular classes of problems.
- Your particular installation might not include all toolboxes.
- Of most interest to you is perhaps:
- **Math and Optimization:**
  - Optimization Toolbox
  - Symbolic Math Toolbox
  - Partial Differential Equation Toolbox
- **Statistics and Data Analysis:**
  - Statistics Toolbox
  - Curve Fitting Toolbox
- **Control System Design and Analysis:**
  - Control System Toolbox
  - System Identification Toolbox

MATLAB toolboxes

- **Signal Processing and Communications**
  - Signal Processing Toolbox
  - Communications Toolbox
  - Filter Design Toolbox
  - RF Toolbox
- **Image Processing**
  - Image Processing Toolbox
  - Image Acquisition Toolbox

MATLAB Simulink

- Simulink lets you visualize your system in a nice way:



Try it yourself...

1) Generate N = 10000 samples from Gaussian distribution with mean = 0 and variance = 4, plot the histogram

2) Generate N = 10000 samples from chi-squared distribution with 2-degrees of freedom (dof), plot the histogram (Hint: sum of two squared normal random variables)

3) Generate N = 10000 samples from Rayleigh distribution, plot the histogram (Hint: Square root of chi-squared random variable with 2-dof)

Try it yourself...

1) Generate

a) x, a sinusoid of frequency, f1 = 3.2 kHz

b) y, a cosinusoid of f2 = 4.8 kHz

c) z, a cosinusoid of f3 = 1.6 kHz

Sampled at fs = 10.24 kHz and duration 3s.

Form s = x + y + z and plot.

2) Compute N = 256 point DFT of s and plot the magnitude and phase against the frequency

3) Change N and fs, observe the DFT plots. Relate the observations to sampling theorem.

Symbolic math toolbox (Advanced)

- Incorporates symbolic computation into the numeric environment of MATLAB
- **Calculus**: differentiation, integration, limits, summation, and Taylor series
- **Linear Algebra**: Inverses, determinants, eigenvalues, singular value decomposition, and canonical forms of symbolic matrices
- **Simplification**: Methods of simplifying algebraic expressions
- **Solution of Equations**: Symbolic and numerical solutions to algebraic and differential equations
- **Special Mathematical Functions**
- **Variable-Precision Arithmetic**
- **Transforms**: Fourier, Laplace, z-transform, and corresponding inverse transforms

Useful commands

```
>> syms x y                    Short-cut for constructing symbolic objects
>> x=sym('x')
>> syms                        Lists the symbolic objects in the workspace
    'x'    'y'
>> y+2
 ans =
 y+2
>> f=x^2+3*x-exp(x)
>> t=4; syms x y
>> findsym(x^3+t*x+y)          Finds the symbolic variables in a symbolic expression
ans =
x, y
```

Useful commands

```
>> pretty(x^2 +log(x))                    Pretty print a symbolic expression.
>> syms alpha t
>> A = [alpha t alpha*t];
>> latex(A) =                             LaTeX representation
        \left [\begin {array}{ccc} \alpha & t & \alpha\,t\end {array}\right ]
>> syms n
>> limit(n/(n+1),n,inf)                   Limit of the expression as n -> inf
>> diff(log(x),x)                         Differentiation
>> int(x^2,x)                             Integration
>> simplify(x^3+3*x^2+6*x+3*x+1)          Symbolic simplification
>> collect(x^3+ pi*x^3+x+pi*x)            Collect coefficients
>> expand(cos(x+y))                       Symbolic expansion
```

Useful commands

>> factor(x^2+2*x+1)	Symbolic factorization
>> [z,how_to]=simple(2*cos(x)^2-sin(x)^2)	Search for simplest form
>> subs(x^2+x*y,x,pi)	Replaces variables in the expression
>> solve(x^4+2*x^2+1,x)	Solution of algebraic equations
>> double(solve(x^4+2*x^2+1,x))	
>> [x,y]=solve(x+y-1,2*x-y)	
>> s=solve('x+y-1','2*x-y')	

…That's it!