Advanced Engineering Research Preparation

Introduction to Matlab: Data Presentation & Manipulation

Presented by: Prof. Eric Larkins (Coates A46)
Optics and Photonics Group
Department of Electrical and Electronic Engineering

Prepared by: Dr. Steve Greedy
George Green Institute for Electromagnetics Research
Department of Electrical and Electronic Engineering

Overview

- Objectives
  - Introduction to Matlab
  - Data manipulation
  - Data presentation

- No prior experience of Matlab Assumed
- Matlab available in all the engineering computer labs

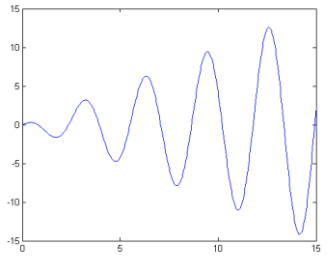
Overview

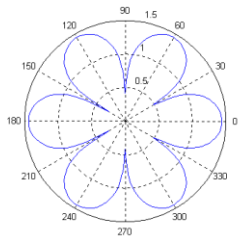
The University of Nottingham

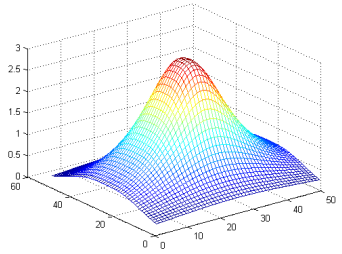
- What is Matlab
- The Matlab Environment
- A Brief Introduction
  - Variables
  - Operators
  - Functions
- Plotting with Matlab

What is Matlab

- Matlab – **Mat**rix **Lab**oratory
- A software package for numerical computation and visualization
  - Matrix manipulation
  - Algorithm implementation
  - Data and function plotting
- Can be enhanced through Toolboxes
  - e.g. Image Processing
  - Interfaces with Simulink for system design
- Has its own language







Why Use Matlab?

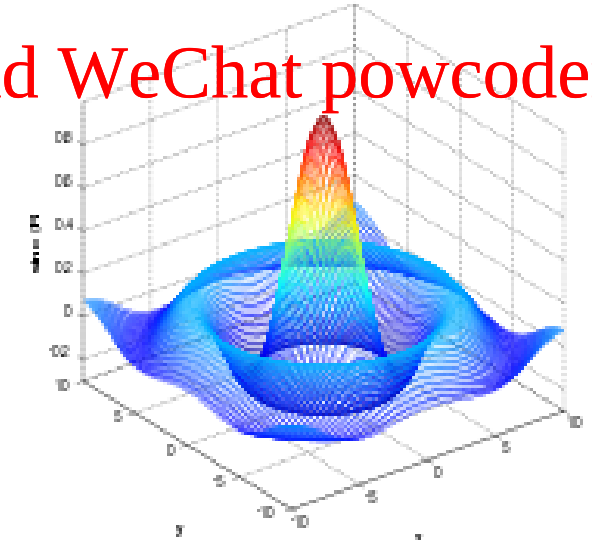
- MATLAB is amazingly powerful, you will be able to use this software throughout your course to help you solve problems, interpret, visualise and present data.
- You will be able to use MATLAB as a tool throughout every technical module you study on your course.

Why Use Matlab?

- Computers don't make mistakes

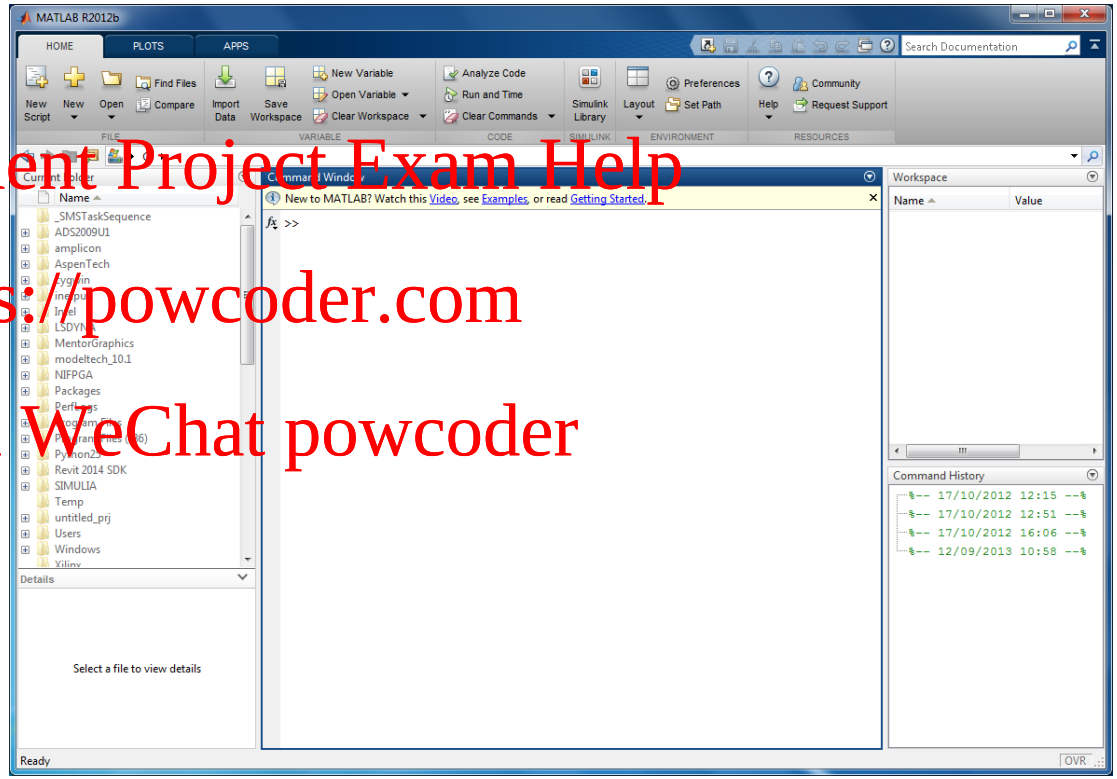
$$x + 3y - 2z = 5$$
$$3x + 5y + 6z = 7$$
$$2x + 4y + 3z = 8$$

- Visualization

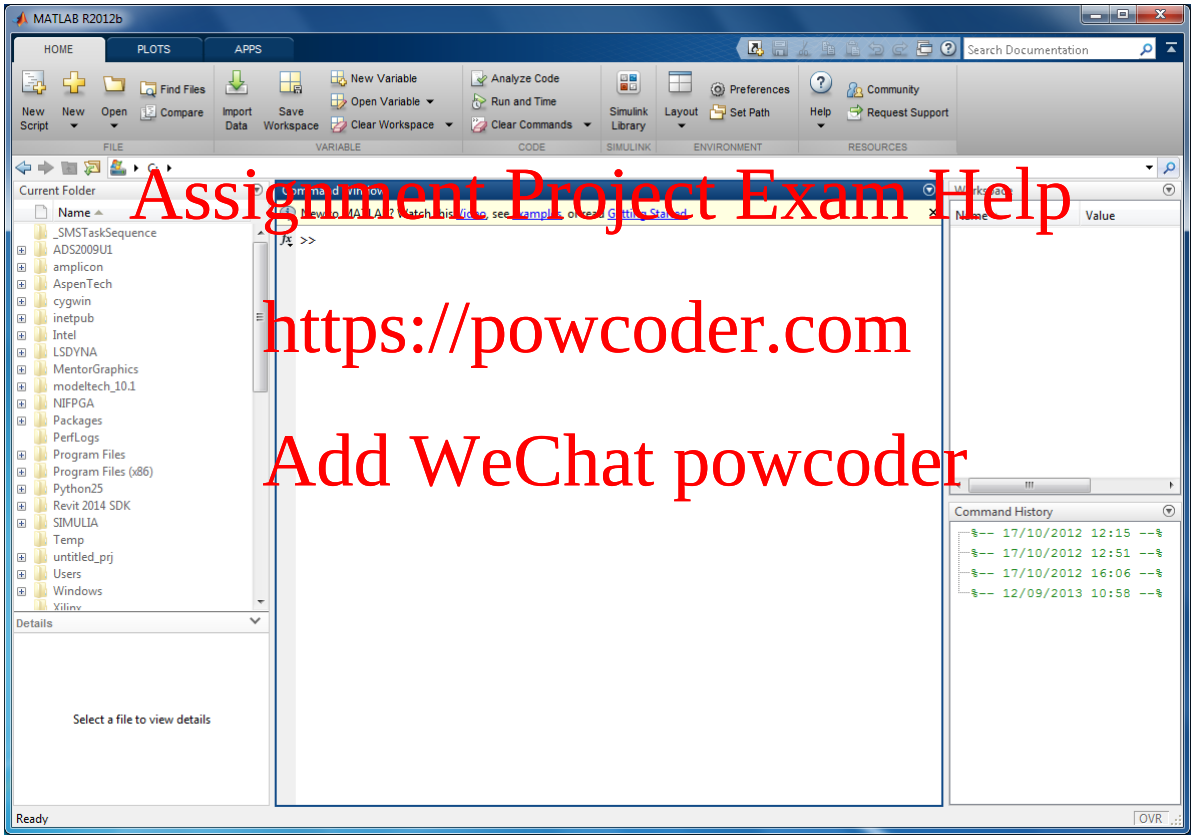


The Matlab Desktop

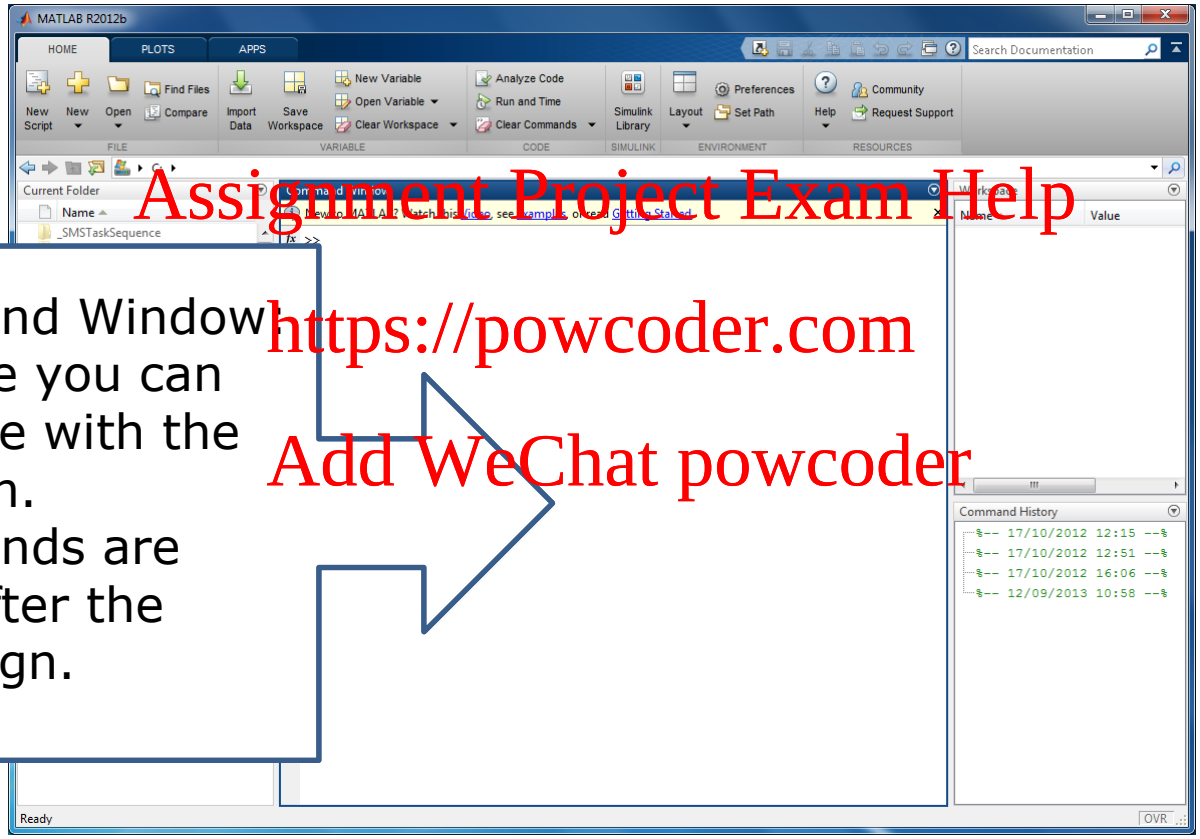
- Start Matlab. May take a while to start!
- The Matlab desktop should appear
- For new users explore links at the top of the main panel
  - Videos
  - Demos
  - Getting started



The Matlab Desktop

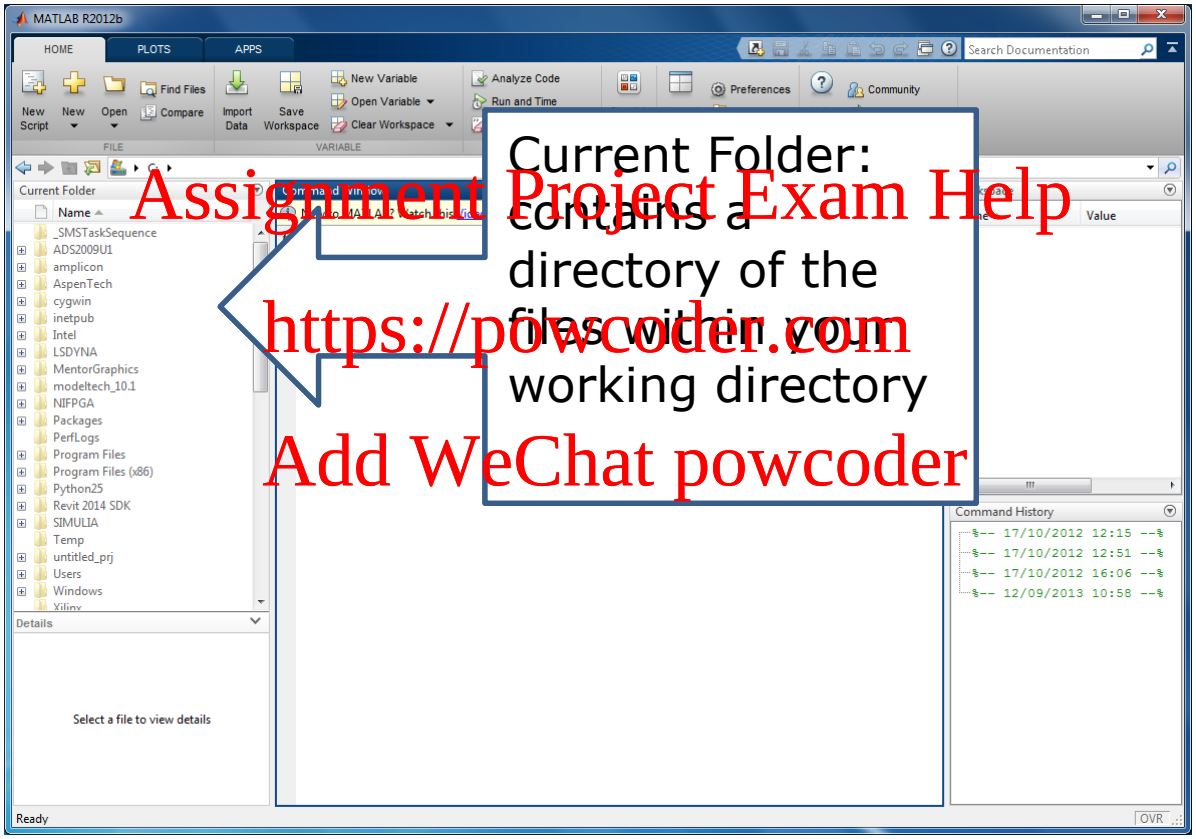


The Matlab Desktop

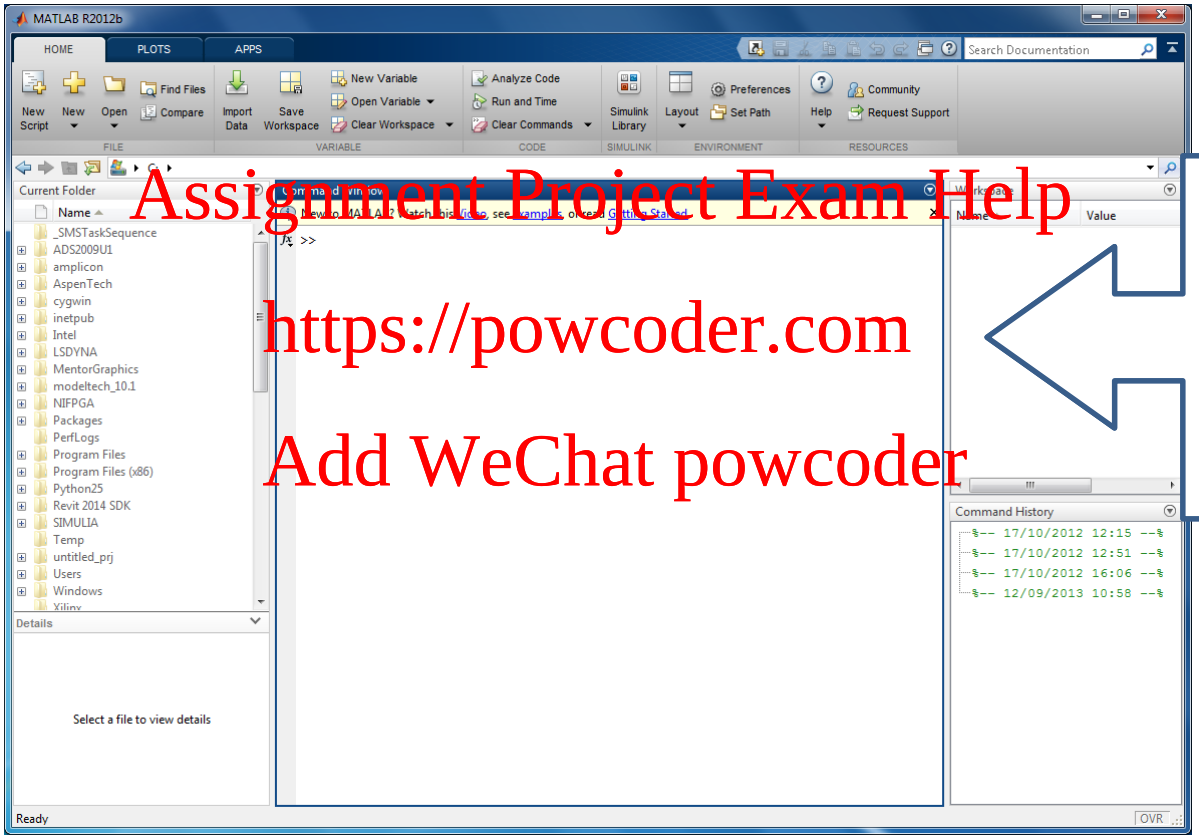


Command Window is where you can interface with the program. Commands are input after the "**>>**" sign.

The Matlab Desktop

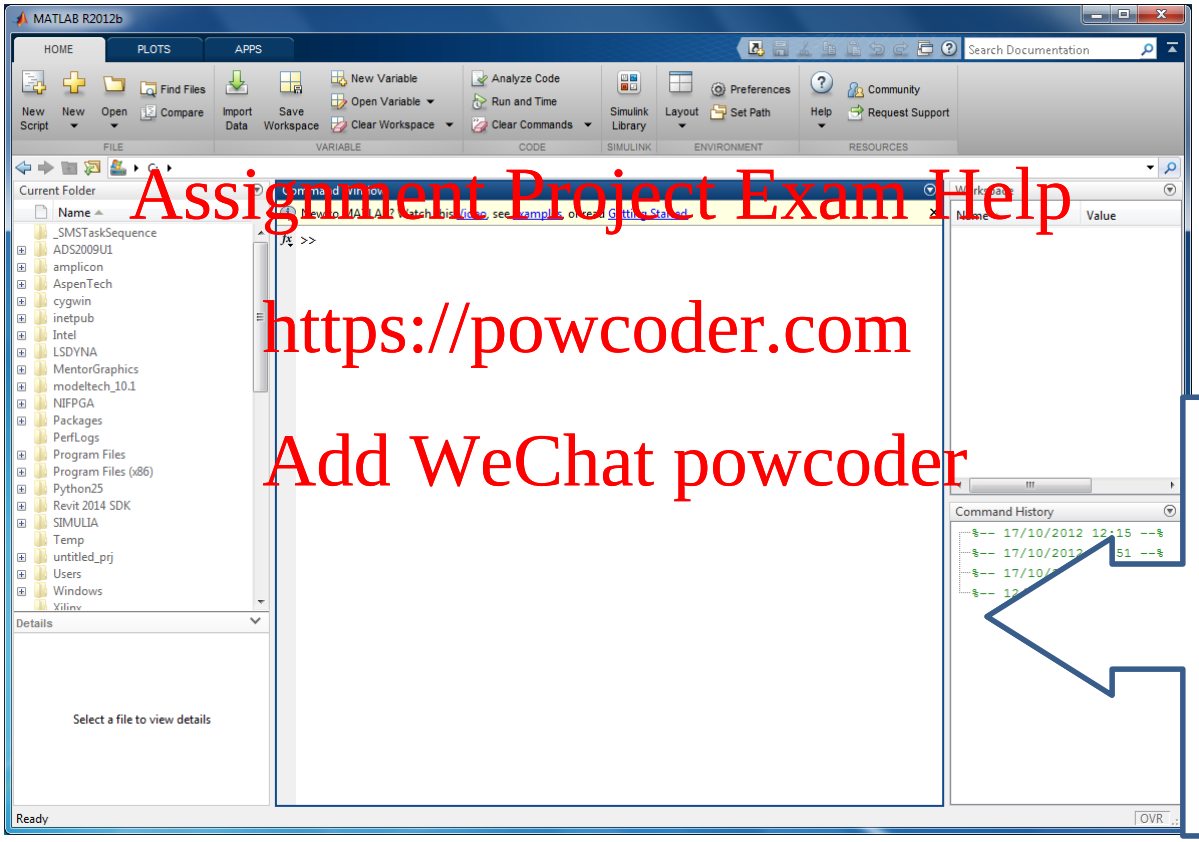


The Matlab Desktop



Workspace: contains list of all current variables held in memory.

The Matlab Desktop



Command history: lists commands typed during the current & previous session

A Brief Introduction

Matlab

A brief introduction

The University of Nottingham

- The Matlab command prompt >>
  - Matlab ready to accept command
  - Command executed when you press return
- Use the Matlab diary
  - Stores a readable text file of your activity
  - >> diary ('c:\myactivity.txt')
  - Copies all user input and output to myactivity.txt
  - Suspend with diary off, resume with diary on

Matlab

A brief introduction

The University of Nottingham

Some Useful Commands:

>> exit	This exits from MATLAB
>> CTRL c	Pressing the "Ctrl" key and "c" simultaneously jumps out of whatever was running and returns you to the previous set of commands
>> up arrow	This scrolls you BACKWARDS through the list of commands used in the session
>> Down arrow	This scrolls you FORWARDS through the list of commands used in the session
>> help	Help on its own gives a list of the available subdirectories and the files they contain
>> **Help plot**	This gives help on the function "plot"

Matlab

A brief introduction

- In its most basic form, MATLAB can act as a calculator, it has all the functions that your pocket calculator has.

- **Echoing commands**
  - ‘Echoing’ simply means that MATLAB displays the result after a calculation.
  - When performing numerous calculations you don’t want to be swamped with line after line of echoed results. The echoing can be suppressed (stopped) by using a semicolon ‘;’ after the command.

Matlab

A brief introduction

The University of Nottingham

- Variables
  - Format: *name of variable = value*
  - e.g. to assign the value 1.1 to x

```
>> x = 1.1

    x =

        1.1
```


- To suppress immediate output end the line with ';'

Matlab

A brief introduction

The University of Nottingham

- Content of variables
  - Can be obtained simply by typing name of variable and return
- To format output:
  - >> format short x = 1.1000
  - >> format long x = 1.10000000000
  - >> format short e x = 1.1000 e000

Matlab

A brief introduction

The University of Nottingham

- Matrices
  - Can be; a single value, a row or column vector, multiple rows and columns
  - Defined row by row
  - To define m=[]

    >> m = [1 2 3; 4 5 6; 7 8 9] or >> m = [1 2 3
    4 5 6
    7 8 9]

    m = 1 2 3
    4 5 6
    7 8 9

Matlab

A brief introduction

The University of Nottingham

- Selecting elements of a matrix
  - m(i,j) = selects value of i^{th} row and j^{th} column
  - m(i,:) = selects the entire i^{th} row
  - m(:,j) = selects the entire j^{th} column
  - m(a:b,x:z) = selects the sub matrix (a…b, x…z)
- Matrix transpose
  - >> n = m’

Matlab

A brief introduction

The University of Nottingham

- Calculations in Matlab
  - + addition
  - – subtraction
  - * multiplication
  - / right division (a/b means a * inv(b))
  - \ left division (a\b means inv(a) * b)
  - ^ raise to the power (a^b means a^b)
- Operators apply to numbers and matrices
- Matlab obeys rules of linear algebra…

Missing multiplication operators and parenthesis are common errors

Matlab

A brief introduction

The University of Nottingham

- Built in constants
  - pi 3.14159265358979323846
  - j (or i) sqrt(-1)
- Built-in functions
  - cos, sin, tan — input and output in radians
  - cosd, sind, tand — input and output in degrees
  - exp, log, log10, sqrt

Matlab

A brief introduction

The University of Nottingham

Matrix operations:

- inv(a) computes the inverse of a
- det(a) computes the determinant of a
- eye(n) gives the n-dimensional unit matrix
- diag(d) returns a diagonal matrix with elements d(i) on its diagonal
- rand(n, m) generates a random n-by-m matrix
- size(a) returns number of rows and columns of a
- rank(a) computes the rank of a
- cond(a) computes the condition number of a
- norm(a) computes the norm of a vector or a matrix
- orth(a) returns an orthonormal basis for the range of a
- lu(a) returns the factors of the LU-decomposition of a

Matlab

A brief introduction

The University of Nottingham

- Sometimes full matrix multiplication is not what you want MATLAB to perform.
- For example, if you have the matrix [A] defined as:

[A] = [2 3 5; 2 3 5]

.. and you want to multiply EACH ELEMENT by 2, you write this in MATLAB as:

A*2; *… this would work*

Matlab

A brief introduction

The University of Nottingham

[A] = [2 3 5 ; -2 3 5]

[B] = [1 4 10; 3 2 7]

- However, if you have two separate matrices, [A] and [B] as above, and you want to multiply the first term in [A] by the first term in [B], the second term in [A] by the second term in [B], etc… Simply using (in MATLAB):

  A*B … gives an error

Matlab

A brief introduction

The University of Nottingham

Use of the 'dot' operator ensures that a point-by-point calculation is performed rather than applying the normal rules of matrices

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \end{bmatrix}$$

$$A.*B = \begin{bmatrix} A_{11}B_{11} & A_{12}B_{12} & A_{13}B_{13} \\ A_{21}B_{21} & A_{22}B_{22} & A_{23}B_{23} \end{bmatrix}$$

$$A./B = \begin{bmatrix} A_{11}/B_{11} & A_{12}/B_{12} & A_{13}/B_{13} \\ A_{21}/B_{21} & A_{22}/B_{22} & A_{23}/B_{23} \end{bmatrix}$$

$$A.\hat{}n = \begin{bmatrix} (A_{11})^n & (A_{12})^n & (A_{13})^n \\ (A_{21})^n & (A_{22})^n & (A_{23})^n \end{bmatrix}$$

Matlab

A brief introduction

The University of Nottingham

- **Suppose we wanted to find the solution to the following equation for a range of x values:**

$$c = (a - b) + 40 - x - 10\frac{q}{b}$$

where x = 1 to 10 in steps of 0.5.


```
>>x = 0:0.5:10;
```

```
0  0.5000  1.0000  1.5000   ..... 9.0000  9.5000  10.0000
```

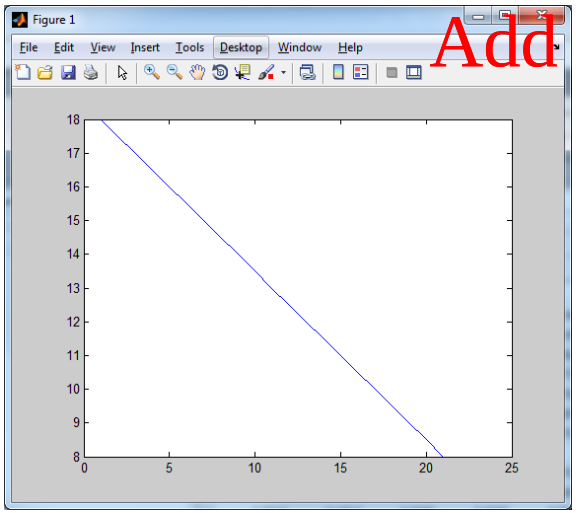
Matlab

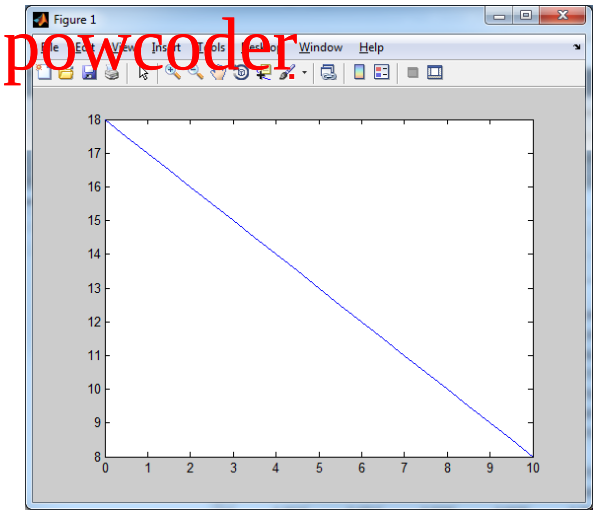
A brief introduction

The University of Nottingham

```
>> a = 12;
>> b = 4;
>> x = 0:0.5:10;
>> c = a - b + 40 - x – 10*a/b;
>> plot(c);
```

or `>> plot(x,c);`

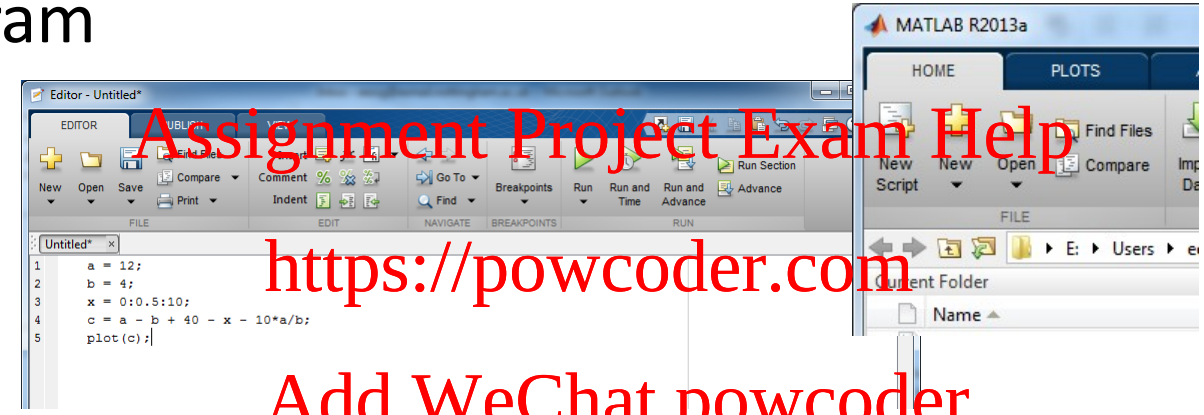




Matlab

A brief introduction

- A series of Matlab commands may be saved as a program



- Program saved in a file with the extension '.m'
- Executed when the name of the file is typed at the command prompt*

*Pre version 2013: To ensure Matlab can find your '.m' file, change to the directory where the file is located or add your directory to the Matlab Path e.g

>> path (path, 'C:\my matlab files')

Plotting with Matlab

Plotting with Matlab

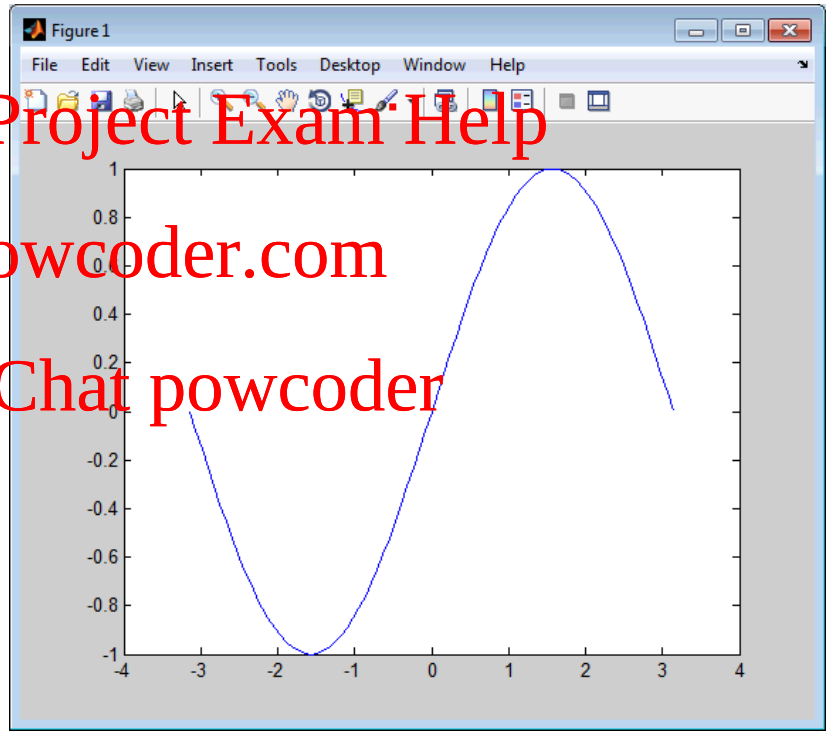
The University of Nottingham

- The basics
- Step 1:
  - Sample the function
- Step 2:
  - Render the plot

e.g
```
>> x = -pi:pi/20:pi;
>> s = sin (x);
>> plot (s,x);
```

Plotting functions within Matlab

```
>> x = -pi:pi/20:pi;
>> s = sin(x);
>> plot (x,s);
```

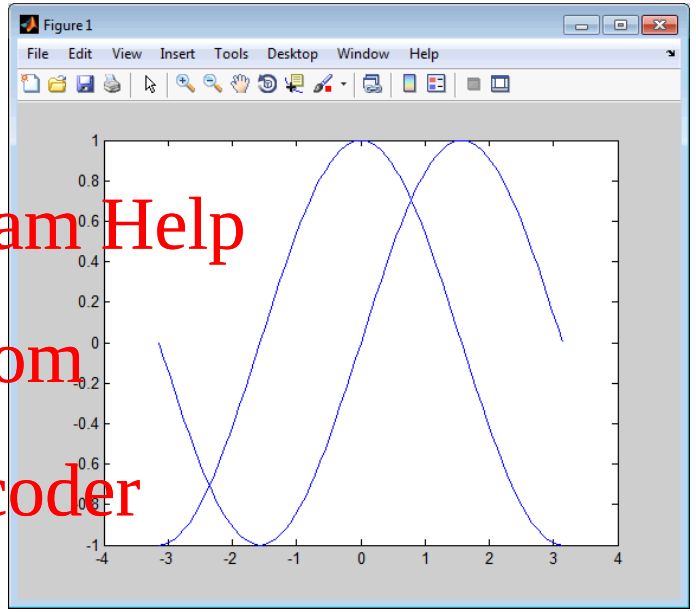


Plotting functions within Matlab

The University of Nottingham

- Use ‘hold on’ to plot another function on the same figure:

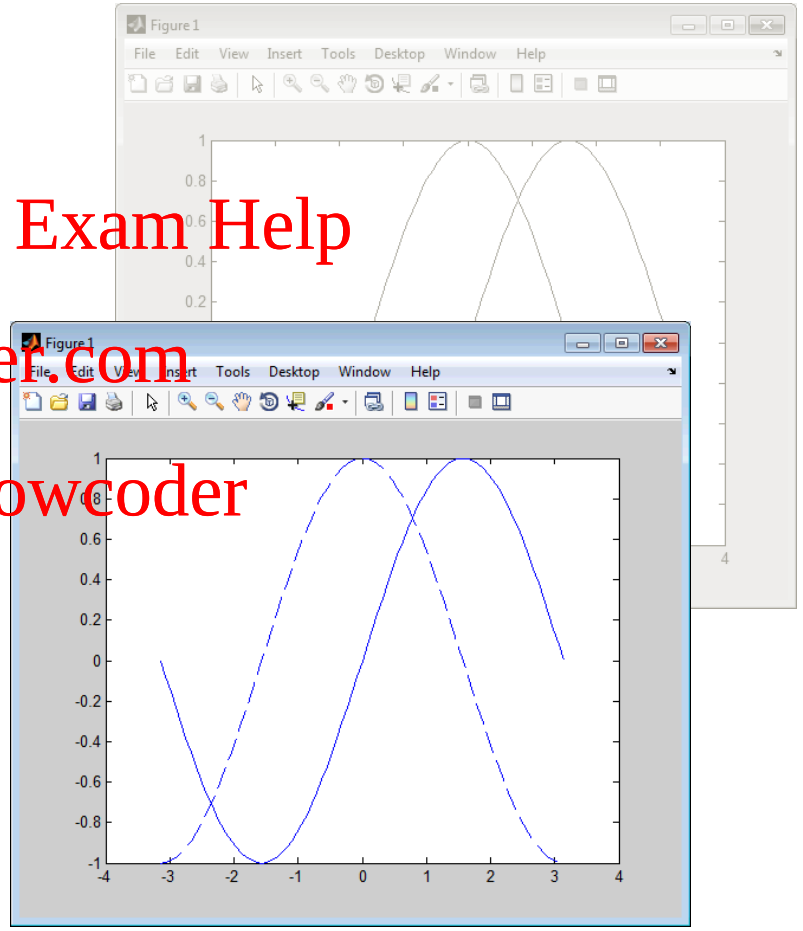

```
>> hold on;
>> c = cos(x);
>> plot (x,c);
```



Plotting functions within Matlab

- Change the line type by adding the LineSpec '--'

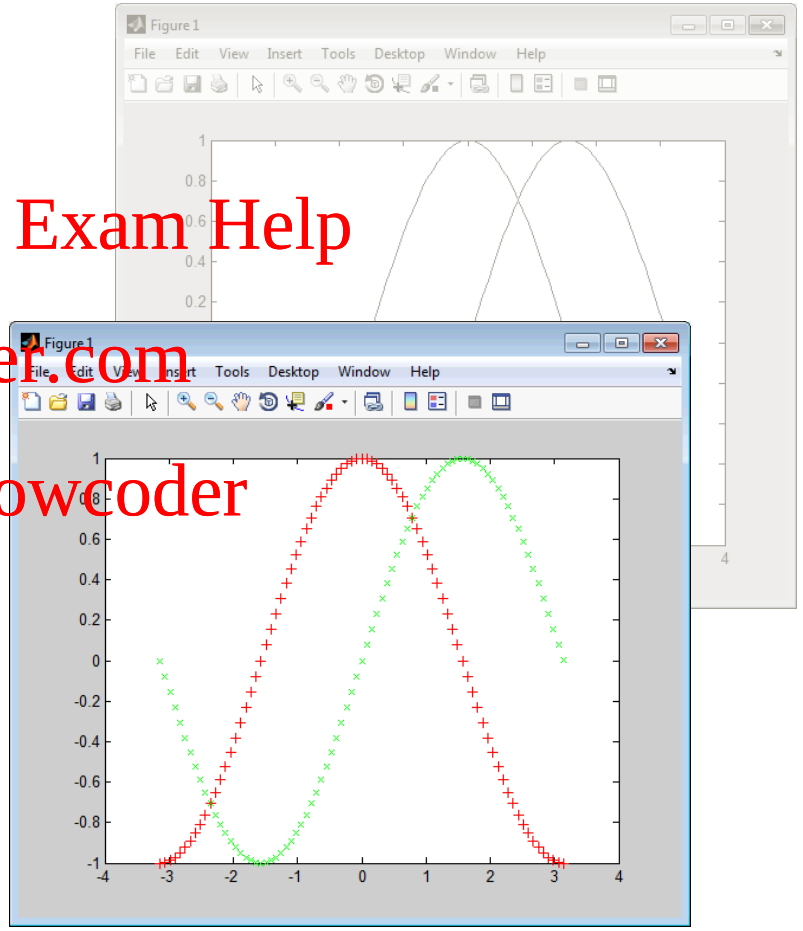

```
>> hold on;
>> c = cos(x);
>> plot (x,c, '--');
```



Plotting functions within Matlab

- Other LineSpec's

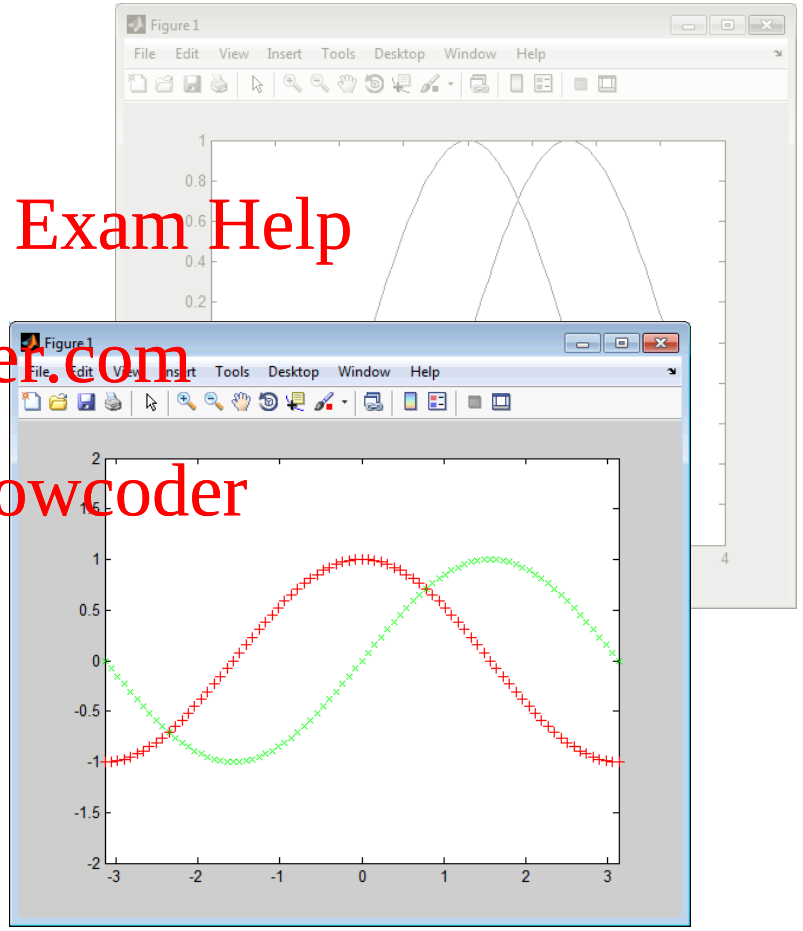
```
>> hold off;
>> plot (x,c, 'r+');
>> hold on
>> plot (x,s, 'gx');
```

Plotting functions within Matlab

- Changing the axis scaling

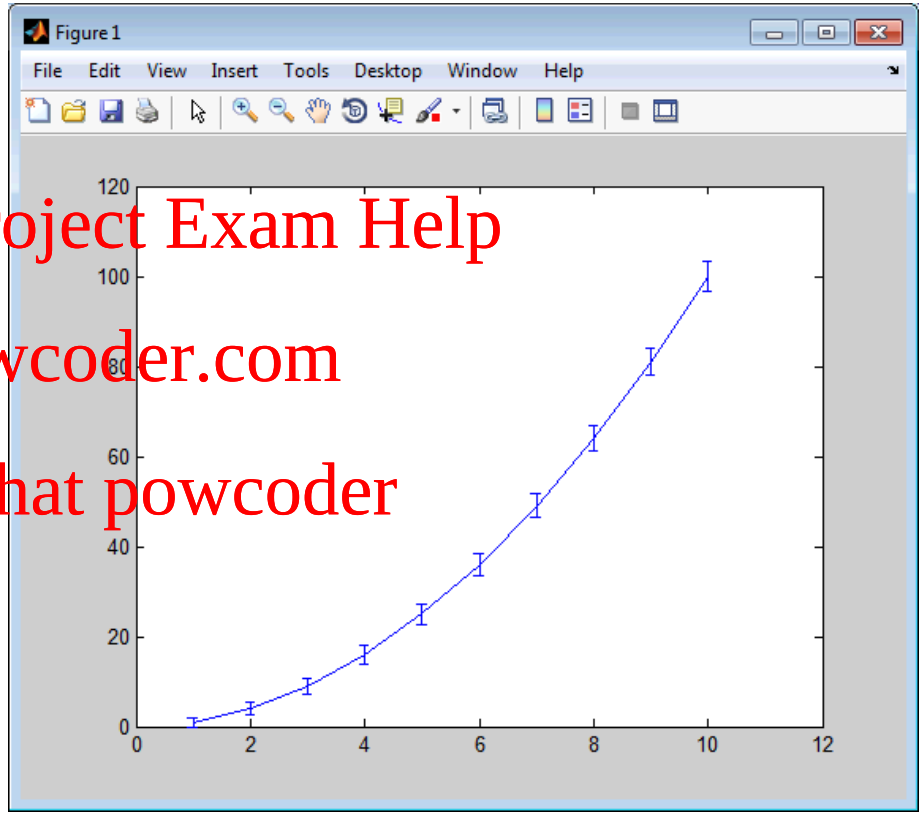

```
>> axis ([-pi,pi,-2,2]);
```

Plotting functions within Matlab

- Error bars

```
>> x = 1:10 ;
>> y = x.^2 ;
>> e = sqrt(x);
>> errorbar (x,y,e)
```

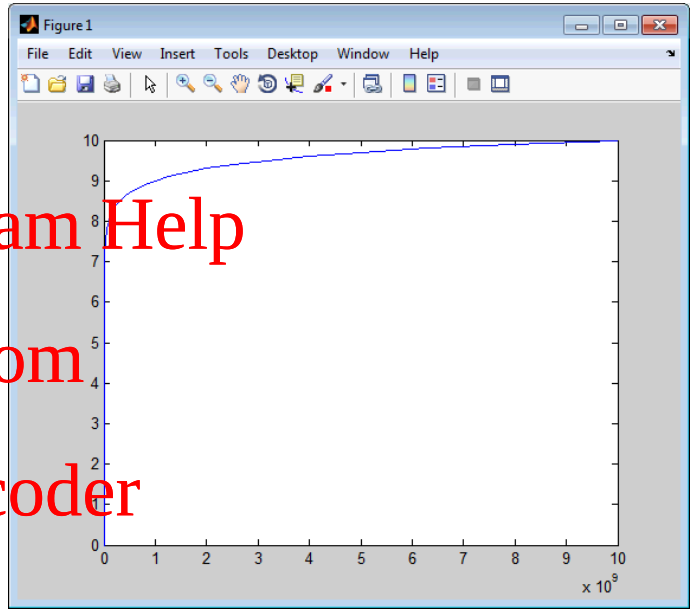


Note: Error bars extend above and below plot by a distance e i.e. they are 2e in length.

Plotting functions within Matlab

- Plotting on a logarithmic scale: $f(x) = 10^x$


```
>> x = 0:0.1:10
>> plot (10.^x,x);
```

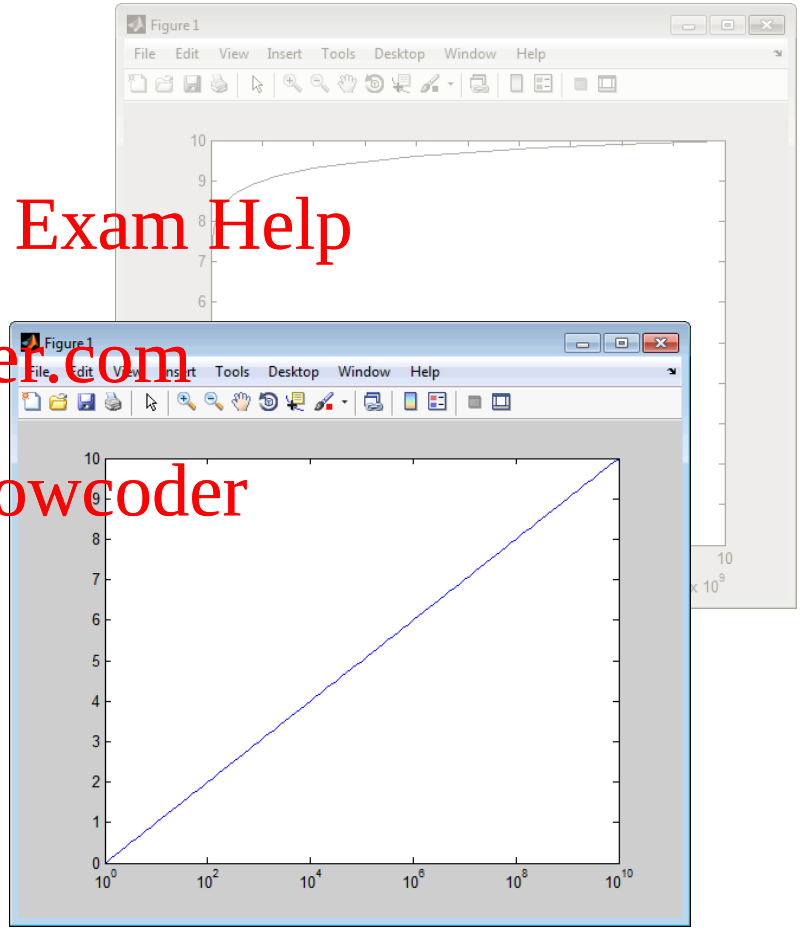


Plotting functions within Matlab

- Plotting on a logarithmic scale: $f(x) = 10^x$


```
>> x = 0:0.1:10
>> semilogx (10.^x,x);
```

Also semilogy and loglog

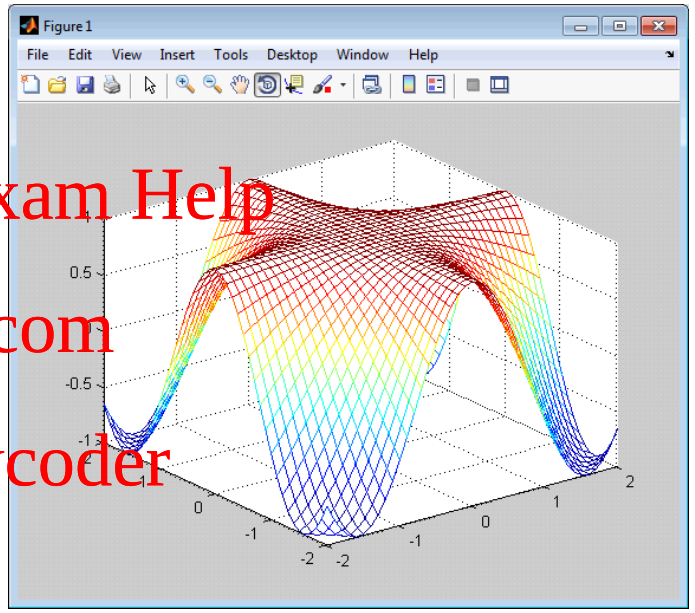


Plotting functions within Matlab

- 3 Dimensional Plots:

$$f(xy) = cos\ (xy)$$

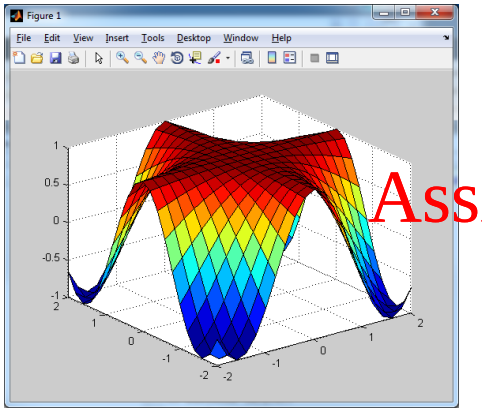

```
>> xrange = -2:0.2:2;
>> yrange = -2:0.2:2;
>> [x,y] = meshgrid(xrange,yrange);
>> z = cos (x .* y);
>> mesh (x,y,z);
```



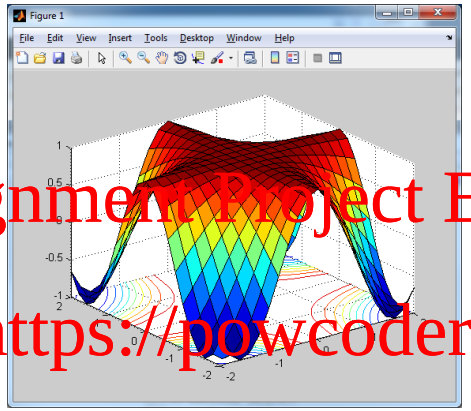
Plotting functions within Matlab

The University of Nottingham

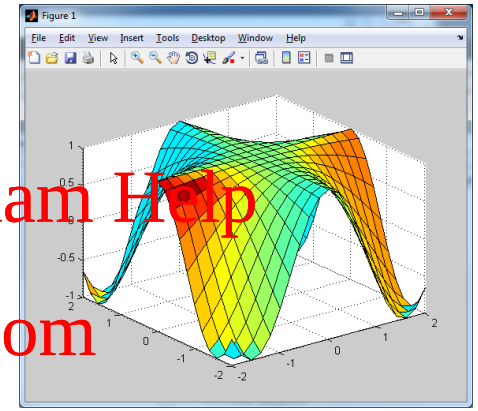
>> surf (x,y,z);



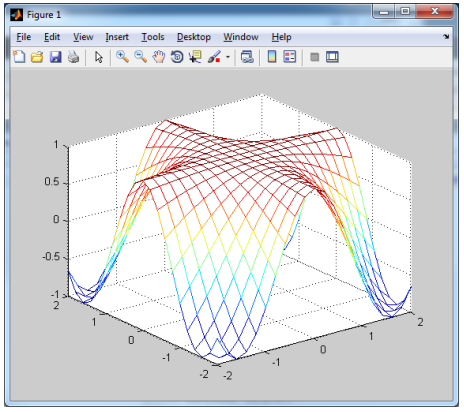
>> surfc (x,y,z);



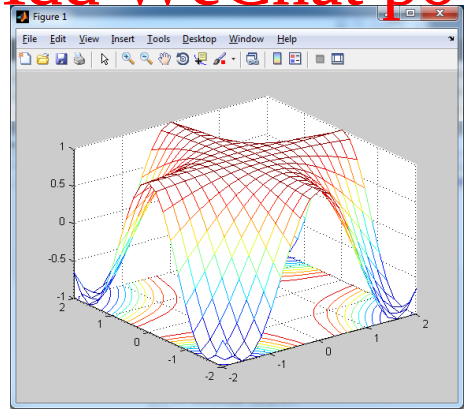
>> surfl (x,y,z);



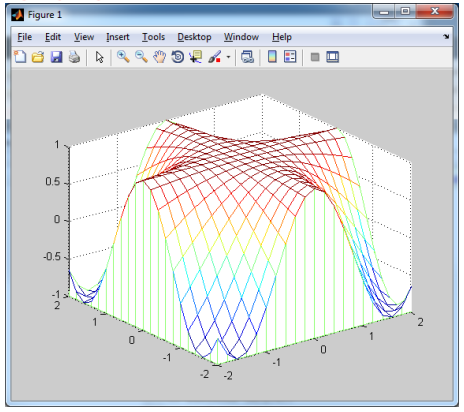
>> mesh (x,y,z);



>> meshc (x,y,z);



>> meshz (x,y,z);



Plotting from External Data

- Use ‘load’ command
- Data required to be a rectangular array
- File cannot contain text e.g. column titles


```
>> load graphdata.dat;
```

Copies data in graphdata.dat into Matlab matrix called graphdata

Plotting from External Data

- If data set is delimited e.g. from excel
- Use ‘dlmread’


```
>> x = dlmread(‘graphdata.txt’, ‘\t’);
```

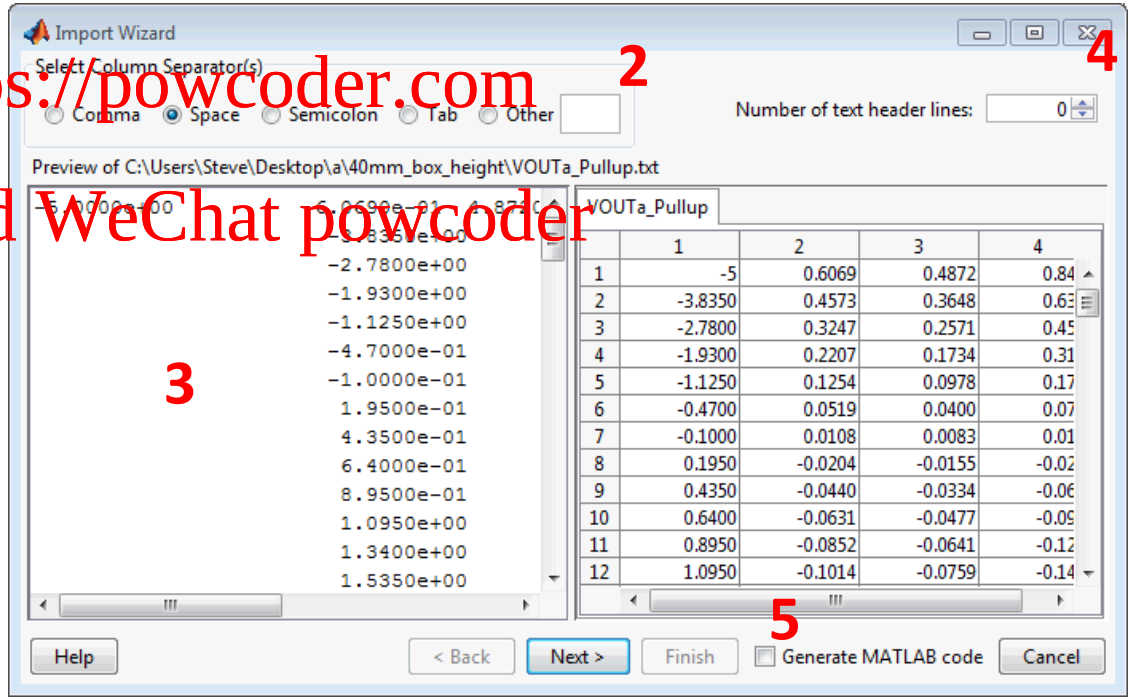

Read in tab separated data from graphdata.txt and store in matrix x.

Common delimeters; ‘\t’ ‘ ‘ ‘,’

Plotting from External Data

- Use the 'Import Data' wizard from the file menu
- Cannot use from within matlab program file – use 'load' or dlmread or use wizard to generate code

1. Choose file
2. Select delimeter
3. Preview file
4. Header lines?
5. Generate code?
6. Finish



Exporting Data to a File

- Save – writes matrix to file in Matlab binary format

  >>save ('filename', 'variable');

  Will write contents of variable to file called filename

  >>save ('filename');

  Will write all current variables to a file

- dlmwrite – writes matrix to a file in delimited ascii format

  >>dlmwrite ('filename', variable, 'delimiter');

Each have their associated options

Further Information

- Matlab help
- http://www.mathworks.com
  - Both excellent sources of information and examples
- http://www.mathworks.com/matlabcenteral/fileexchange
  - User contributions & solutions

Questions

Exercises

- Work through the examples in these notes to perform basic calculations and matrix operations

- Download the file ‘sombrero.txt’ from moodle
  - Import the data
  - Plot 3D mesh and surface plots

Exercises

- The data in sombrero.txt was generated with:

```
>> x=-10:.5:10;
>> y=x;
>> [X, Y]=meshgrid(x,y);
>> R=sqrt(X.^2+Y.^2) + eps;
>> Z=sin(R)./R;
>> mesh(Z);
```


- Re-generate Z data for the range -25 to 25 in steps of 0.1. Save data using ‘save’ and ‘dlmwrite’