

# Introduction to MATLAB

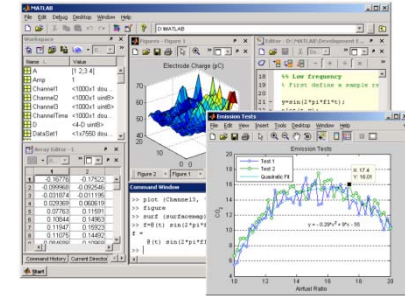
**Todd Atkins**

[tatkins@mathworks.com](mailto:tatkins@mathworks.com)

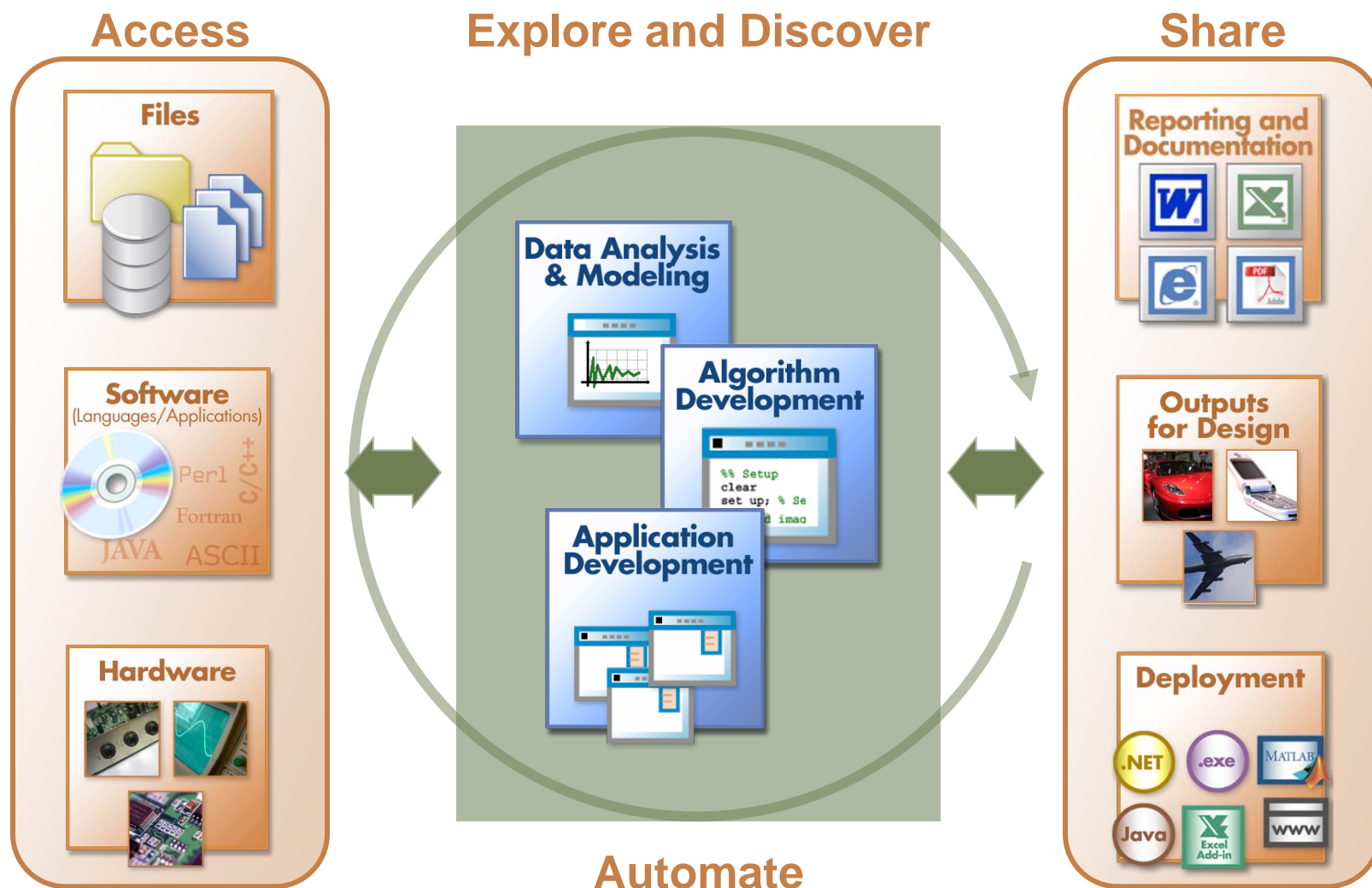
# MATLAB

## The Language for Technical Computing

- Key Features
  - High-level language of technical computing
  - Development environment for engineers, scientists
  - Interactive tools for design, problem solving
  - Mathematical function libraries
  - Graphics and data visualization tools
  - Custom GUIs
  - External Interfaces: C, C++, Fortran, Java, COM, Excel, .NET



# Technical Computing Workflow



# Outline

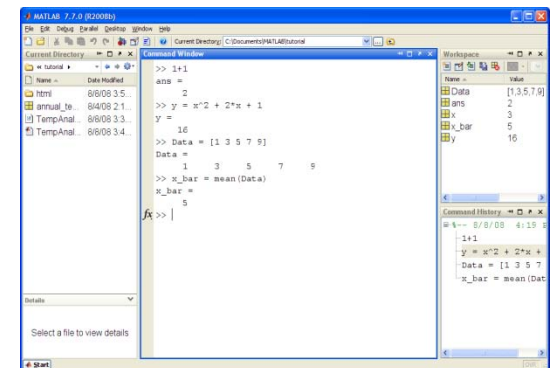
**MATLAB Desktop**

Computing in MATLAB

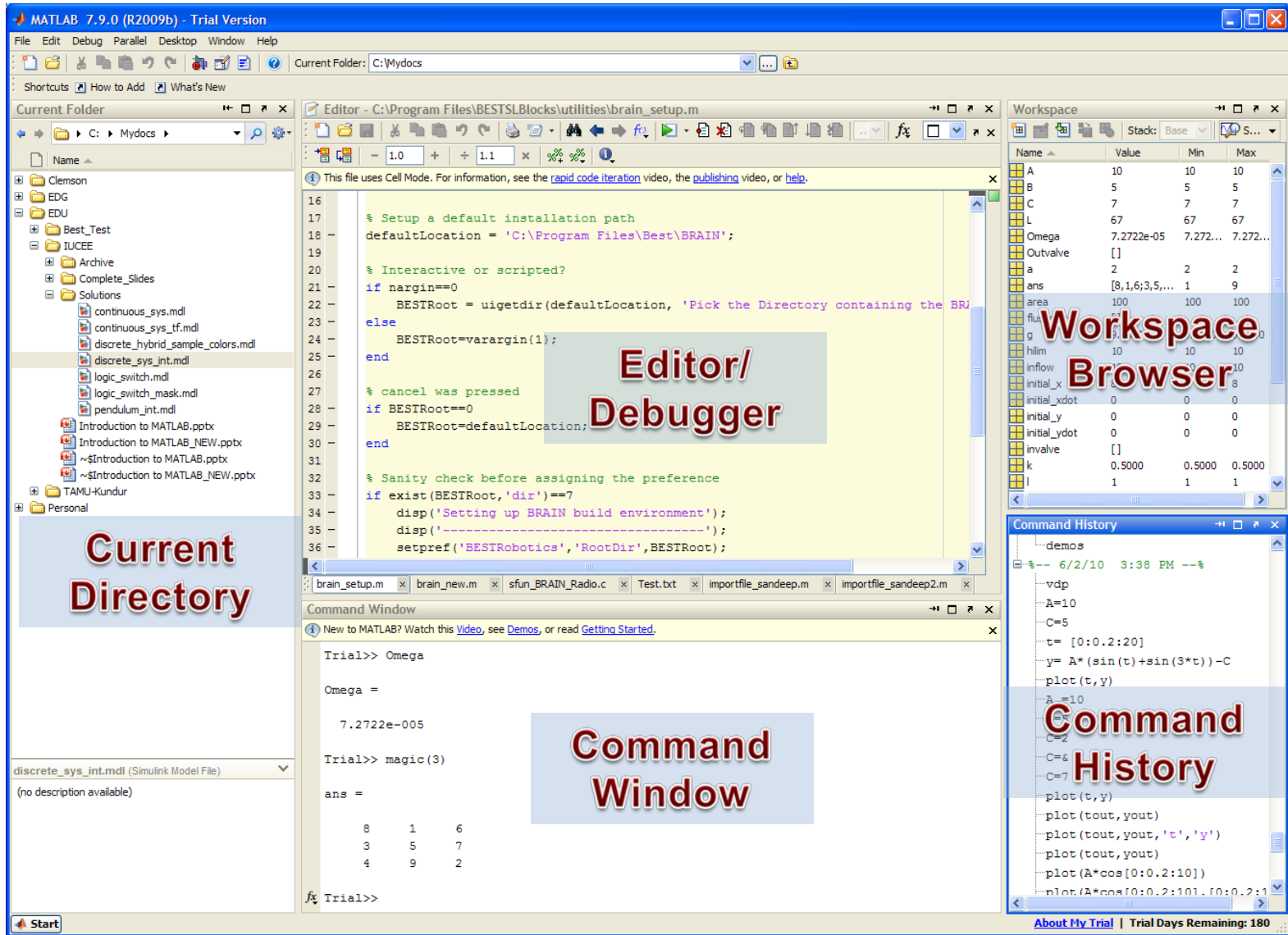
Problem Solving with MATLAB

Working with data in MATLAB

Programming in MATLAB



# MATLAB Desktop



The screenshot shows the MATLAB 7.9.0 (R2009b) - Trial Version desktop environment. The interface includes a menu bar (File, Edit, Debug, Parallel, Desktop, Window, Help), a toolbar, and several panes:

- Current Folder:** Displays the file explorer for the current directory (C:\Mydocs).
- Editor:** Shows the code editor for the file `C:\Program Files\BESTSLBlocks\utilities\brain_setup.m`. The code is in Cell Mode. A red box labeled "Editor/Debugger" highlights the editor area.
- Workspace:** Displays the current workspace variables. A red box labeled "Workspace Browser" highlights this pane. The variables and their values are:
 

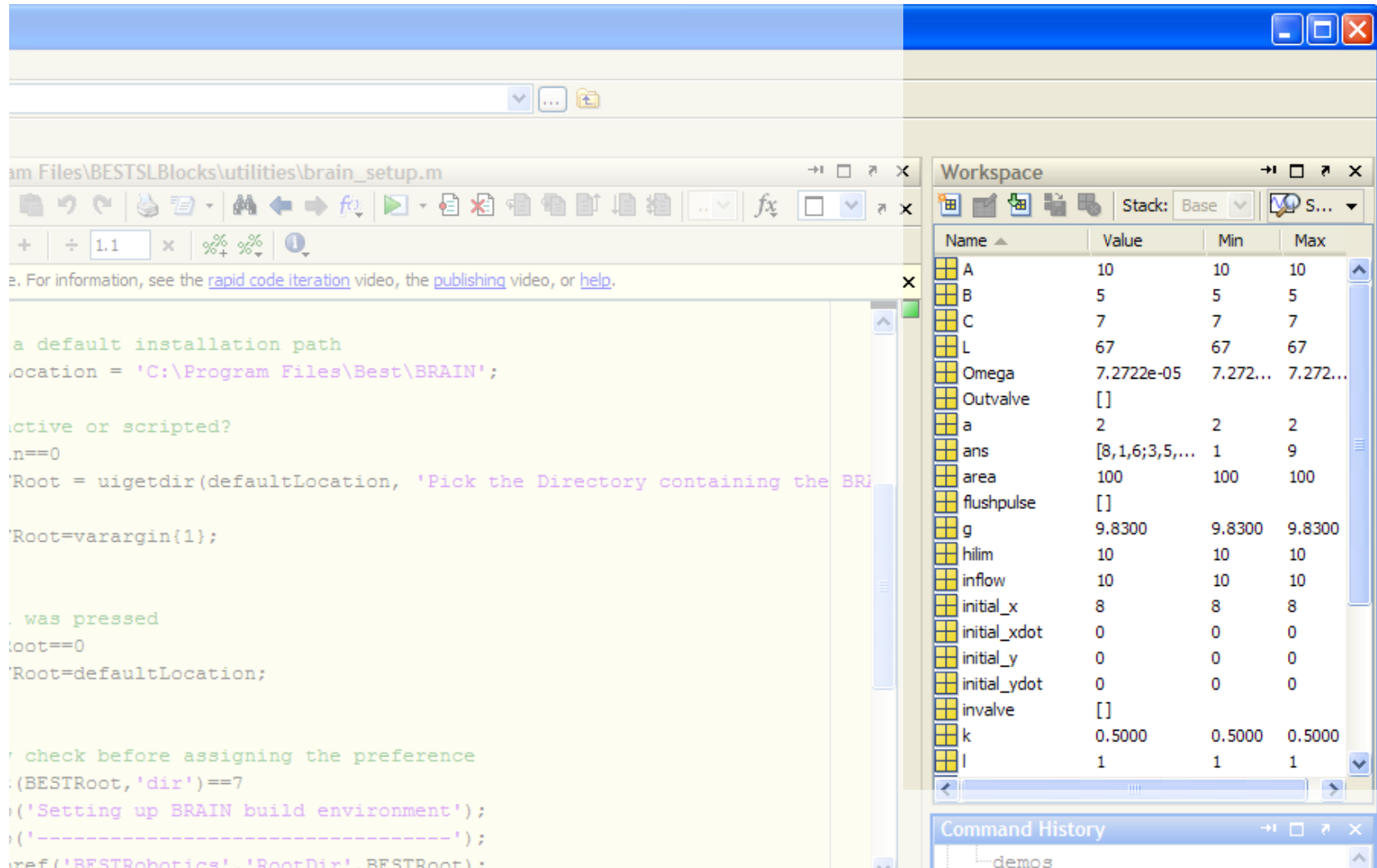
Name	Value	Min	Max
A	10	10	10
B	5	5	5
C	7	7	7
L	67	67	67
Omega	7.2722e-05	7.272...	7.272...
Outvalve	[]		
a	2	2	2
ans	[8,1,6;3,5,...]	1	9
area	100	100	100
flu			
g			
hlim	10	10	10
inflow			
initial_x			
initial_xdot	0	0	0
initial_y	0	0	0
initial_ydot	0	0	0
invalve	[]		
k	0.5000	0.5000	0.5000
l	1	1	1
- Command Window:** Shows the command history and the current command prompt. A red box labeled "Command Window" highlights this pane. The commands entered are:
 

```
Trial>> Omega
Omega =
7.2722e-005
Trial>> magic(3)
ans =
8 1 6
3 5 7
4 9 2
fx Trial>>
```
- Command History:** Displays the list of commands entered in the Command Window. A red box labeled "Command History" highlights this pane. The commands are:
 

```
demos
vdp
A=10
C=5
t= [0:0.2:20]
y= A*(sin(t)+sin(3*t))-C
plot(t,y)
A=10
C=2
C=6
C=7
plot(t,y)
plot(tout,yout)
plot(tout,yout,'t','y')
plot(tout,yout)
plot(A*cos[0:0.2:10])
plot(A*cos[0:0.2:10].-[0:0.2:1
```

At the bottom of the window, there is a status bar with the text "About My Trial | Trial Days Remaining: 180".

# Workspace Browser



The screenshot shows the MATLAB Workspace Browser window. The left pane displays a script editor with the following code:

```

a default installation path
location = 'C:\Program Files\Best\BRAIN';

active or scripted?
n==0
Root = uigetdir(defaultLocation, 'Pick the Directory containing the BRAI

Root=varargin{1};

. was pressed
oot==0
Root=defaultLocation;

check before assigning the preference
(BESTRoot, 'dir')==7
('Setting up BRAIN build environment');
('-----');
ref('BESTRobotics'. 'RootDir'. BESTRoot):

```

The right pane shows the Workspace browser with the following variables:

Name	Value	Min	Max
A	10	10	10
B	5	5	5
C	7	7	7
L	67	67	67
Omega	7.2722e-05	7.272...	7.272...
Outvalve	[]		
a	2	2	2
ans	[8,1,6;3,5,...	1	9
area	100	100	100
flushpulse	[]		
g	9.8300	9.8300	9.8300
hilim	10	10	10
inflow	10	10	10
initial_x	8	8	8
initial_xdot	0	0	0
initial_y	0	0	0
initial_ydot	0	0	0
invalve	[]		
k	0.5000	0.5000	0.5000
l	1	1	1

The Command History window at the bottom shows the command 'demos'.

# Getting Help

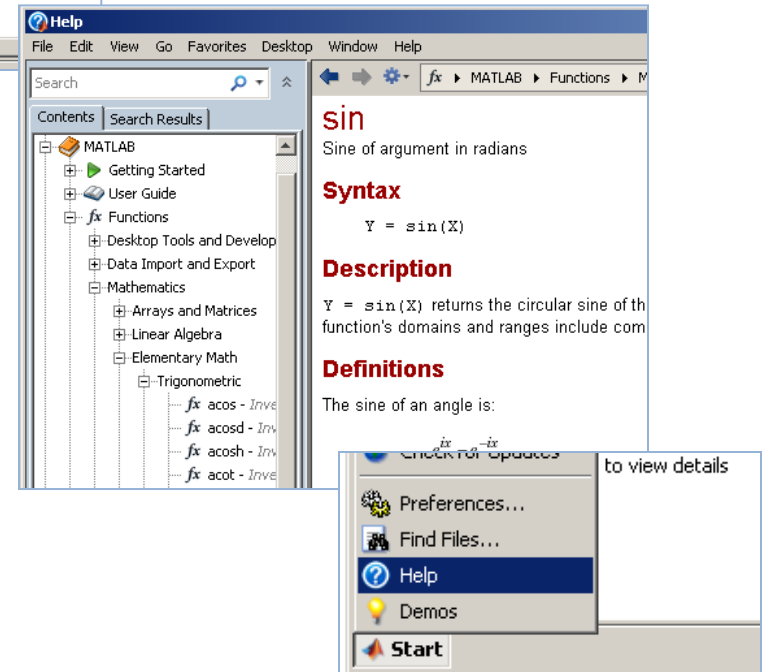
- `doc`
- Start -> Help
- <http://mathworks.in> -> Support -> Product Documentation
- `help`
- Search the web

```
>> help sin
SIN    Sine of argument in radians.
       SIN(X) is the sine of the elements of X.

       See also asin, sind.

       Overloaded methods:
           codistributed/sin

       Reference page in Help browser
           doc sin
```



# Outline

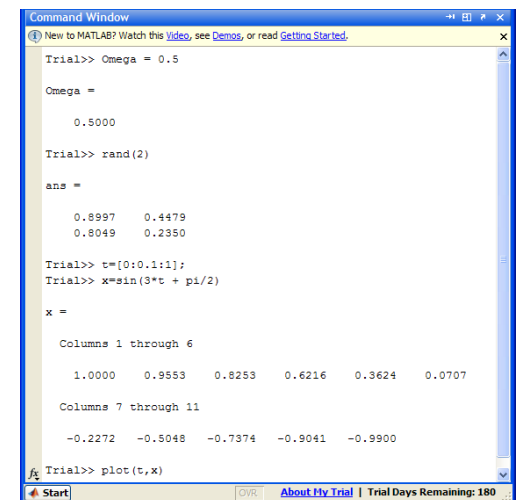
MATLAB Desktop

**Computing in MATLAB**

Problem Solving with MATLAB

Working with data in MATLAB

Programming in MATLAB



```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

Trial>> Omega = 0.5

Omega =

    0.5000

Trial>> rand(2)

ans =

    0.8997    0.4479
    0.8049    0.2350

Trial>> t=[0:0.1:1];
Trial>> x=sin(3*t + pi/2)

x =

Columns 1 through 6

    1.0000    0.9553    0.8253    0.6216    0.3624    0.0707

Columns 7 through 11

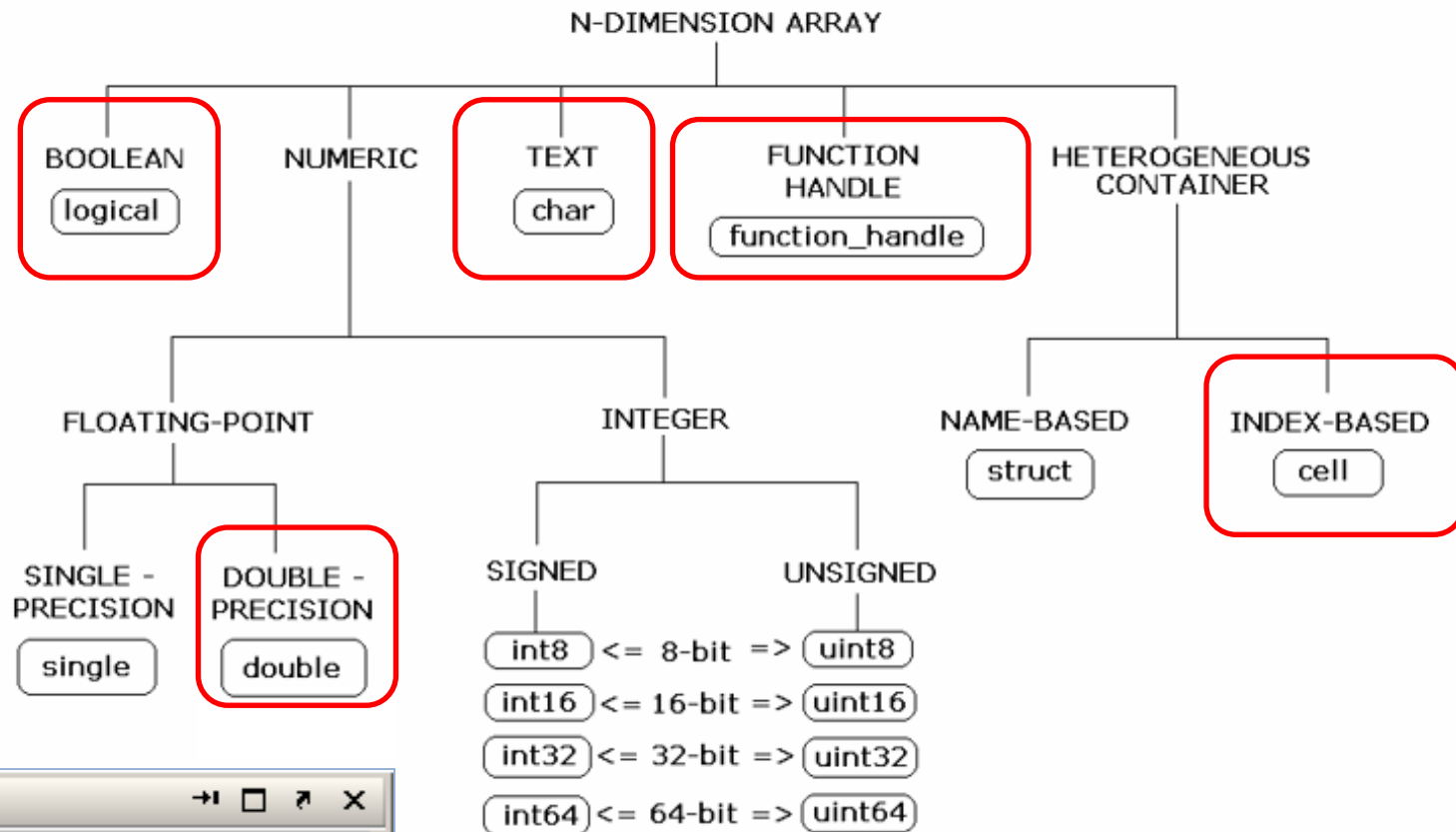
   -0.2272   -0.5048   -0.7374   -0.9041   -0.9900

f Trial>> plot(t,x)
  
```



# Basics: Data and Variables

- Class
- Size
- Value



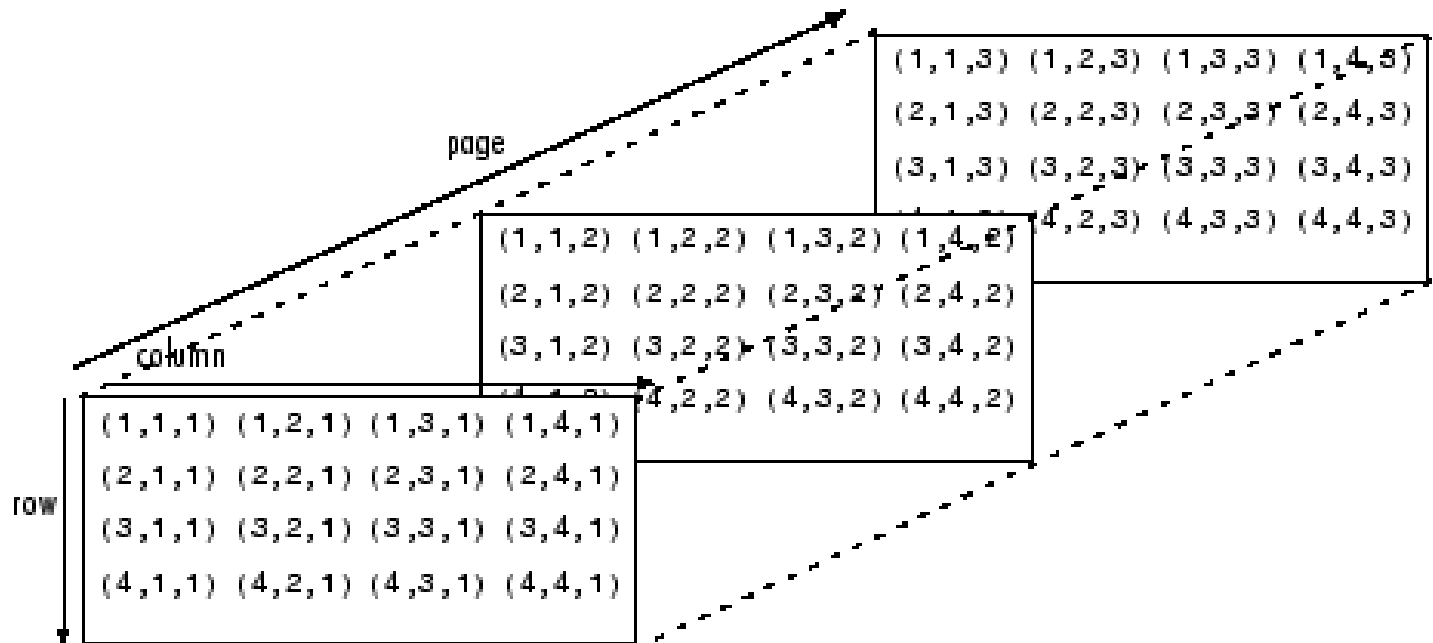
Workspace

Select data to plot

Name	Value	Size	Class
a	10	1x1	double
b	[0.8147,0,...	3x3	double
c	<1x2 cell>	1x2	cell
d	<1x5 struct>	1x5	struct

# Basics: Data and Variables

- Class
- Size
- Value



Workspace			
Name	Value	Size	Class
a	10	1x1	double
b	[0.8147,0....	3x3	double
c	<1x2 cell>	1x2	cell
d	<1x5 struct>	1x5	struct

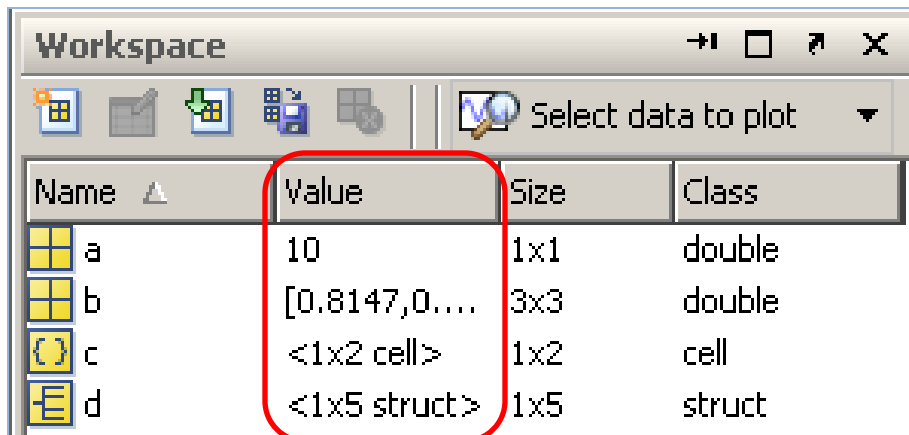
 $m \times n$ 
 $m \times n \times \dots \times z$

# Basics: Data and Variables

- Class
- Size
- Value

```
>> magic(4)  
ans =  
    16     2     3    13  
     5    11    10     8  
     9     7     6    12  
     4    14    15     1
```

```
fx >> |
```



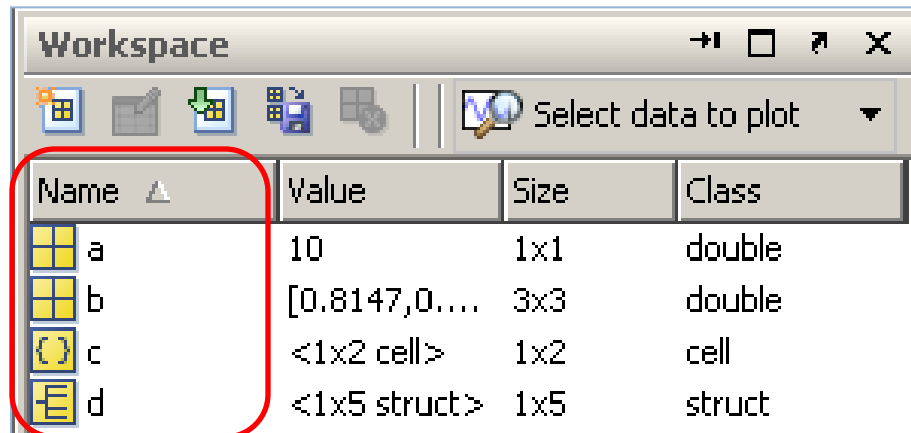
Name	Value	Size	Class
a	10	1x1	double
b	[0.8147,0....	3x3	double
c	<1x2 cell>	1x2	cell
d	<1x5 struct>	1x5	struct

# Basics: Data and Variables

- Class
- Size
- Value
- Name (“variable”)

foo

16	2	3	13
5	11	10	8
9	7	6	12
4	14	15	1



Name	Value	Size	Class
a	10	1x1	double
b	[0.8147,0....	3x3	double
c	<1x2 cell>	1x2	cell
d	<1x5 struct>	1x5	struct

# Basics: Manipulating Data

- Creation
- Extraction (subset)
- Union (merge)
- Deletion

```
>> a = rand(2,4)
a =
    0.957506835434298    0.157613081677548    0.957166948242946    0.800280468888800
    0.964888535199277    0.970592781760616    0.485375648722841    0.141886338627215
>> a([1,3,5])
ans =
    0.957506835434298    0.157613081677548    0.957166948242946
>> a(1,3)
ans =
    0.957166948242946
>> a(3)
ans =
    0.157613081677548
fx >>
```

# Basics: Manipulating Data

- Creation

- Extraction (subset)

- Union (merge)

- Deletion

```
:  
rand  
zeros  
ones  
diag  
magic  
;  
end  
linspace  
logspace  
...
```

# Basics: Manipulating Data

- Creation
- Extraction (subset)
- Union (merge)
- Deletion

Subscript  
Linear  
Logical  
( )  
sub2ind  
ind2sub  
...

# Basics: Manipulating Data

## Subscript

- Creation
- Extraction (subset)
- Union (merge)
- Deletion

```
<name>(row, col)  
<name>(row, col, ..., z)
```

Indices may themselves be arrays.



# Basics: Manipulating Data

## Linear

- Creation
- Extraction (subset) `<name>(location)`
- Union (merge)
- Deletion

Where `<location>` is:

`row+(col-1)*num_rows`

Location may be an array.

# Basics: Manipulating Data

## Logical

- Creation
- **Extraction (subset)**      `<name> (<logical_array> )`
- Union (merge)
- Deletion

Where the dimensions of `<name>` and `<logical_array>` are the same and `<logical_array>` is of type logical.

# Basics: Manipulating Data

- Creation
- Extraction (subset)
- **Union (merge)**
- Deletion

[ ]  
repmat  
strvcat  
**Expansion**  
cat  
horzcat  
vertcat  
...

# Basics: Manipulating Data

- Creation
- Extraction (subset)
- Union (merge)
- **Deletion**

`clear`  
`clearvars`  
Assign to empty

# Basics: Manipulating Data

**() versus []**

Indexing  
Order of operations  
Argument list

Matrix/Vector creation  
Concatenation  
Multiple outputs

# Basics: Manipulating Data

■

■

`1 : 5``0 : 5 : 25``25 : -3 : 2``25 : 5 : 0``0 : .5 : 4`

# Basics: Manipulating Data

end

```
a(1:end)  
b(end, end)
```

## Basics: Math

- Matrix operations

```
>> a = [1 2; 3 4]
```

```
>> b = [5 6; 7 8]
```

```
>> c = a*b
```

- Element operations (dot)

```
>> a = [1 2; 3 4]
```

```
>> b = [5 6; 7 8]
```

```
>> c = a.*b
```

```
>> d = a.^3 + b^2
```

- Others

^ - Power

\ - Left divide

' - Transpose



## Basics: Math

- Scalar expansion

```
>> a = [1 2; 3 4]
```

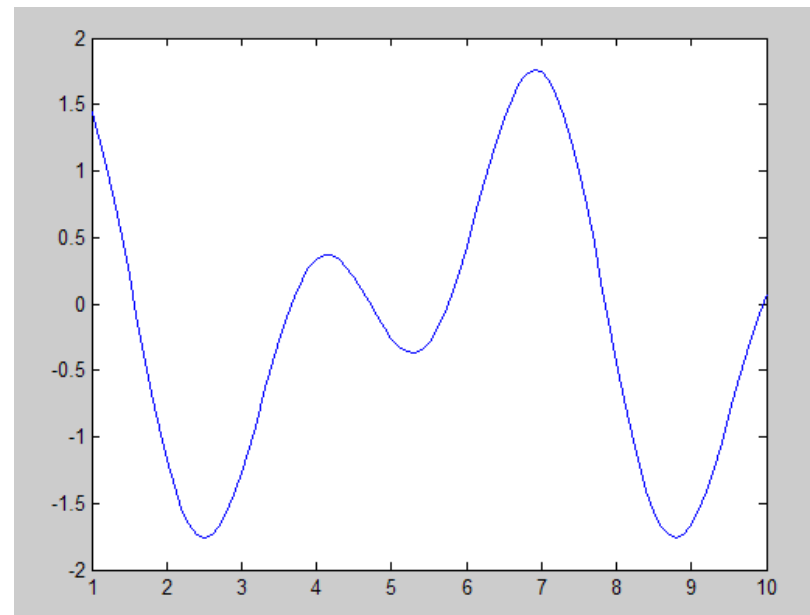
```
>> a+[1 2]
```

```
>> a+1
```

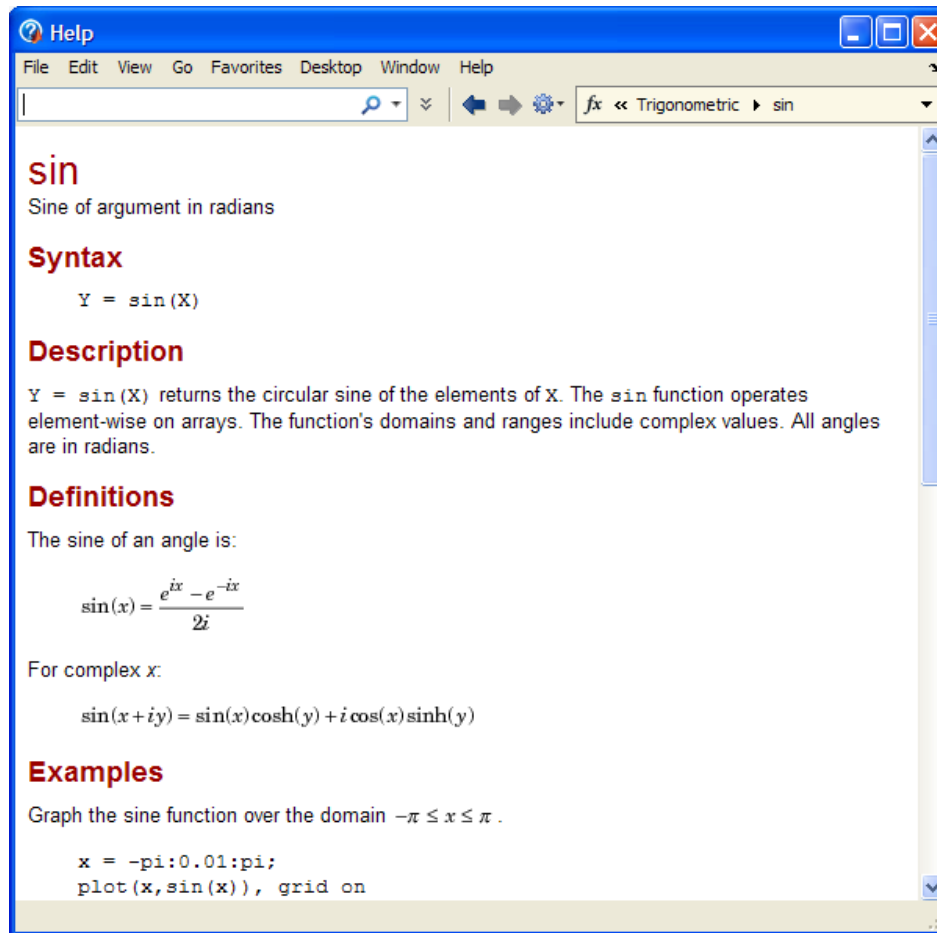
- What happens?

# Demo

- Compute  $y = \sin(2t) + \cos(t)$  where  $t$  is from 1 to 10 seconds.
- MATLAB Functions
  - `sin`
  - `cos`
  - `plot`



# MATLAB Functions



The screenshot shows the MATLAB Help window for the `sin` function. The window has a blue title bar with "Help" and standard window controls. Below the title bar is a menu bar with "File", "Edit", "View", "Go", "Favorites", "Desktop", "Window", and "Help". A search bar is located below the menu bar. The breadcrumb trail at the top right reads "fx << Trigonometric >> sin". The main content area is divided into sections: "sin" (Sine of argument in radians), "Syntax" (Y = sin(X)), "Description" (Y = sin(X) returns the circular sine of the elements of X), "Definitions" (The sine of an angle is:  $\sin(x) = \frac{e^{ix} - e^{-ix}}{2i}$ ), and "Examples" (Graph the sine function over the domain  $-\pi \leq x \leq \pi$ ). The examples section includes the code: `x = -pi:0.01:pi;` and `plot(x,sin(x)), grid on`.

**sin**  
Sine of argument in radians

**Syntax**  
`Y = sin(X)`

**Description**  
`Y = sin(X)` returns the circular sine of the elements of `X`. The `sin` function operates element-wise on arrays. The function's domains and ranges include complex values. All angles are in radians.

**Definitions**  
The sine of an angle is:

$$\sin(x) = \frac{e^{ix} - e^{-ix}}{2i}$$

For complex `x`:

$$\sin(x + iy) = \sin(x) \cosh(y) + i \cos(x) \sinh(y)$$

**Examples**  
Graph the sine function over the domain  $-\pi \leq x \leq \pi$ .

```
x = -pi:0.01:pi;  
plot(x,sin(x)), grid on
```

# Calling Functions

Function calling syntax:

```
[out1, out2, ..., outN] = functionname(in1, in2, ..., inN)
```

Aside - Command syntax:

```
functionname string1 string2 string3 ... stringN
```

Revisit the Help.

# Outline

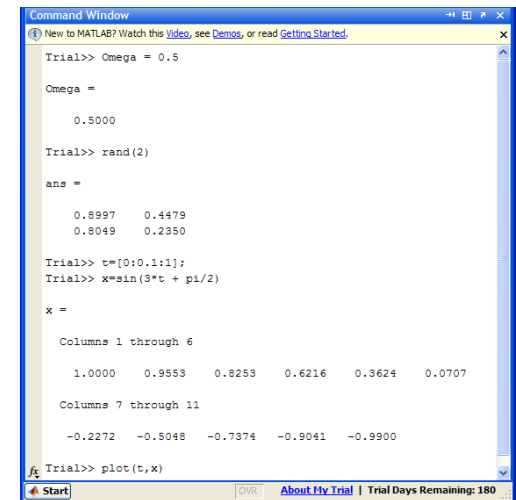
MATLAB Desktop

Computing in MATLAB

**Problem Solving with MATLAB**

Working with data in MATLAB

Programming in MATLAB



The screenshot shows the MATLAB Command Window interface. At the top, there is a title bar and a menu bar. Below the menu bar, there is a toolbar with icons for file operations and a search icon. The main area of the window displays the following text:

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

Trial>> Omega = 0.5

Omega =

    0.5000

Trial>> rand(2)

ans =

    0.8997    0.4479
    0.8049    0.2350

Trial>> t=[0:0.1:1];
Trial>> x=sin(3*t + pi/2)

x =

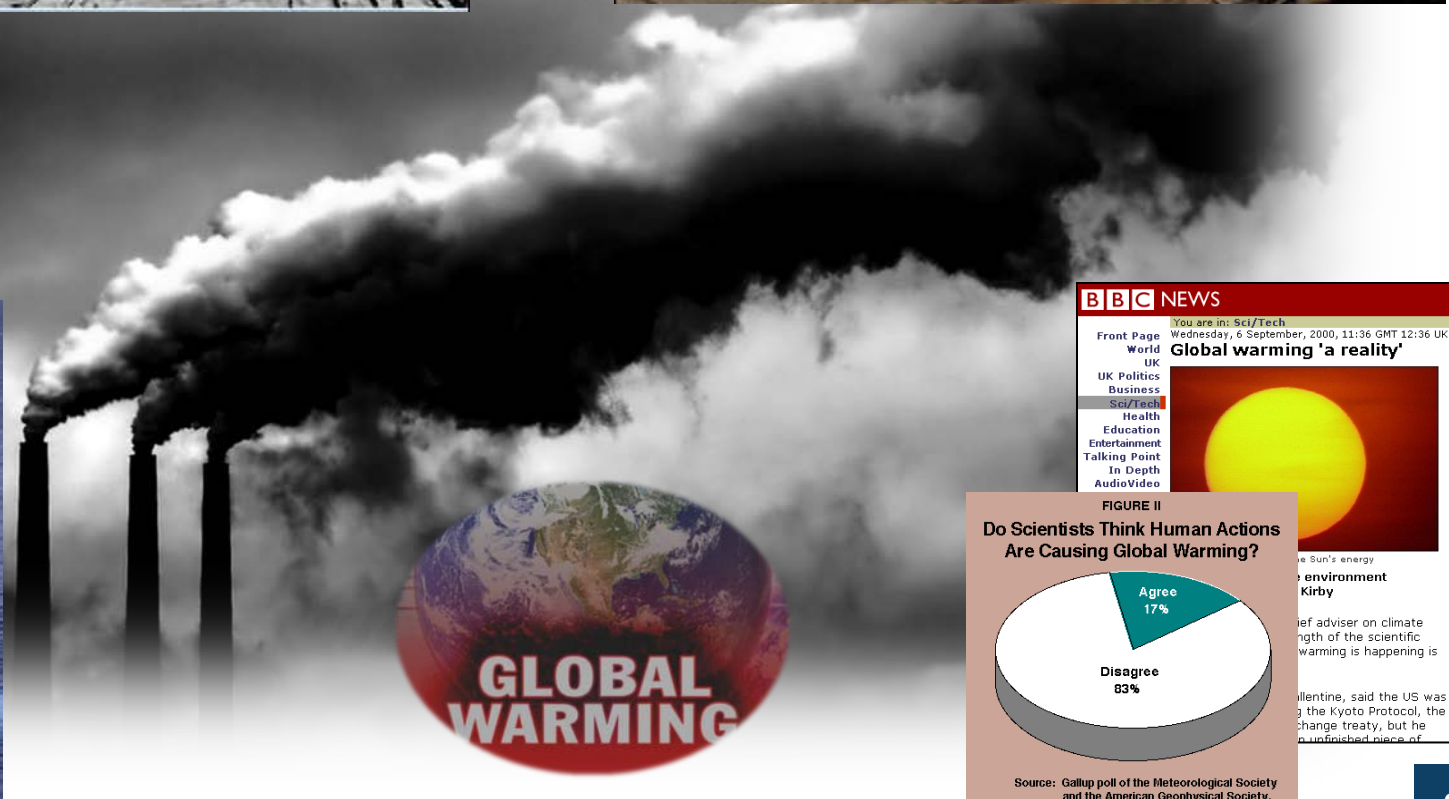
Columns 1 through 6
    1.0000    0.9553    0.8253    0.6216    0.3624    0.0707

Columns 7 through 11
   -0.2272   -0.5048   -0.7374   -0.9041   -0.9900

Trial>> plot(t,x)
```

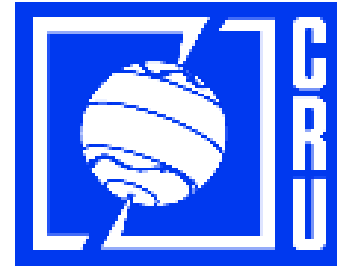
At the bottom of the window, there is a status bar with a "Start" button, a "Stop" button, and a link to "About My Trial". The status bar also indicates "Trial Days Remaining: 180".

# Consider one – Global Warming!



# Is the temperature rising?

- Climatic Research Unit
  - School of Environmental Sciences at University of East Anglia, Norwich, UK
  - <http://www.cru.uea.ac.uk/>
- Importing data
  - Data set for land/sea combined temperature anomalies on a 5 by 5 grid-box basis (HadCRUT3.mat)
- Visualizing data
  - Different methods to plot and observe data
- Analyzing data
  - Computational analysis to check if temperature is rising?



# Outline

MATLAB Desktop

Computing in MATLAB

Problem Solving with MATLAB

**Working with data in MATLAB**

Programming in MATLAB



# Outline

MATLAB Desktop

Computing in MATLAB

Problem Solving with MATLAB

**Working with data in MATLAB**

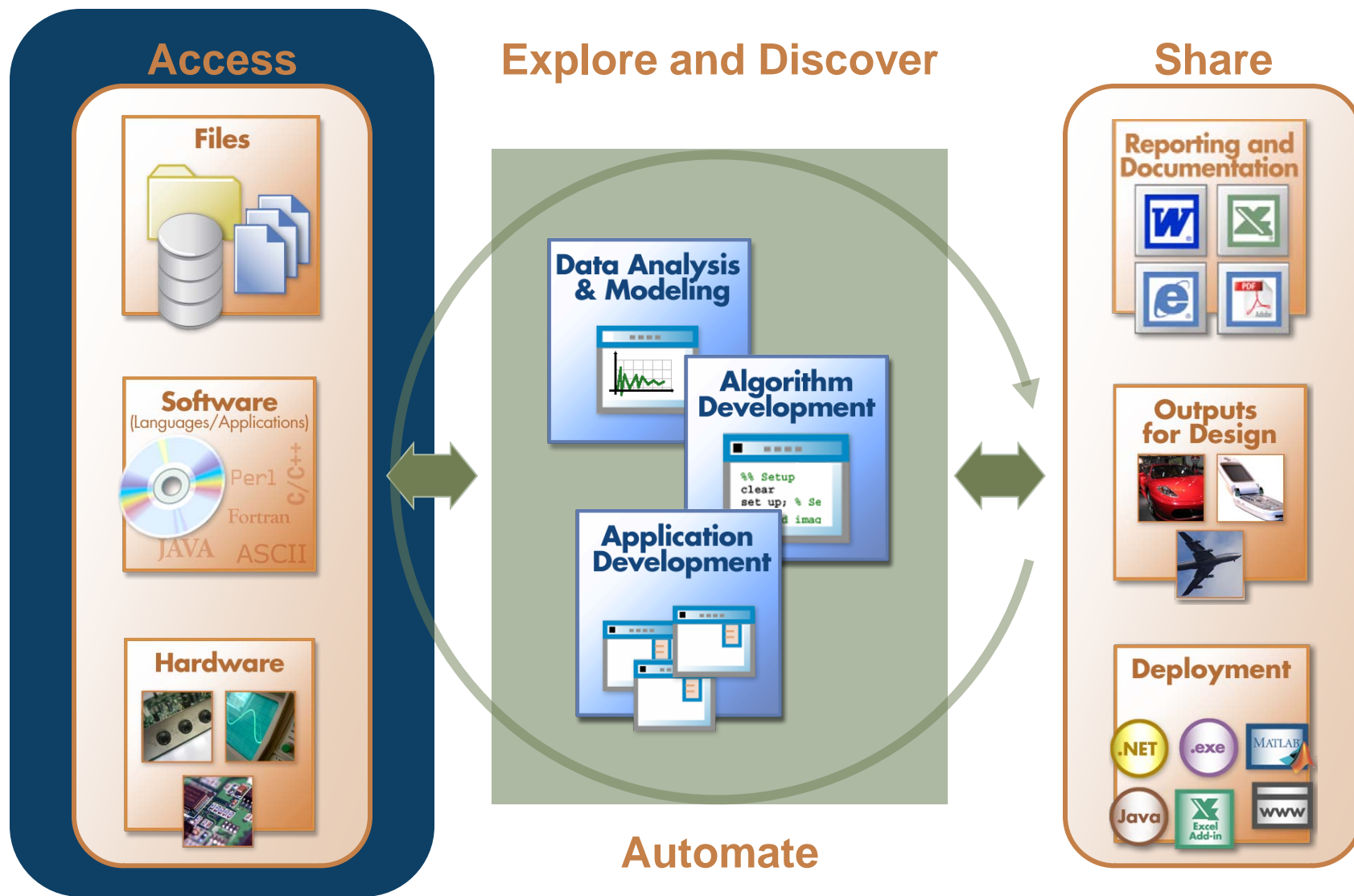
Programming in MATLAB

Import/Extract/Export

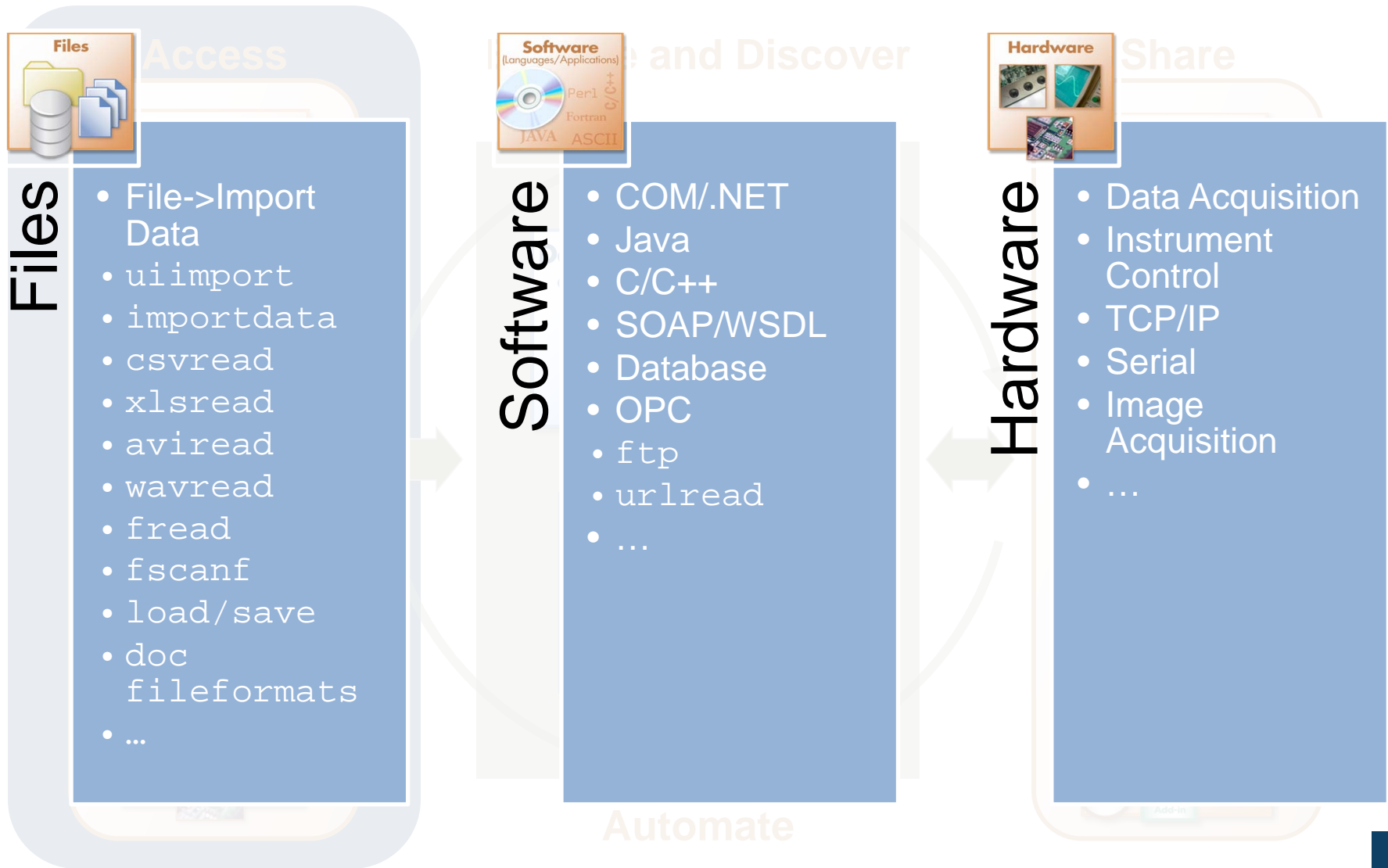
Visualize

Analyze

# Technical Computing Workflow



# Technical Computing Workflow: Access



# Outline

MATLAB Desktop

Computing in MATLAB

Problem Solving with MATLAB

Working with data in MATLAB

**Programming in MATLAB**

**Writing MATLAB  
programs**

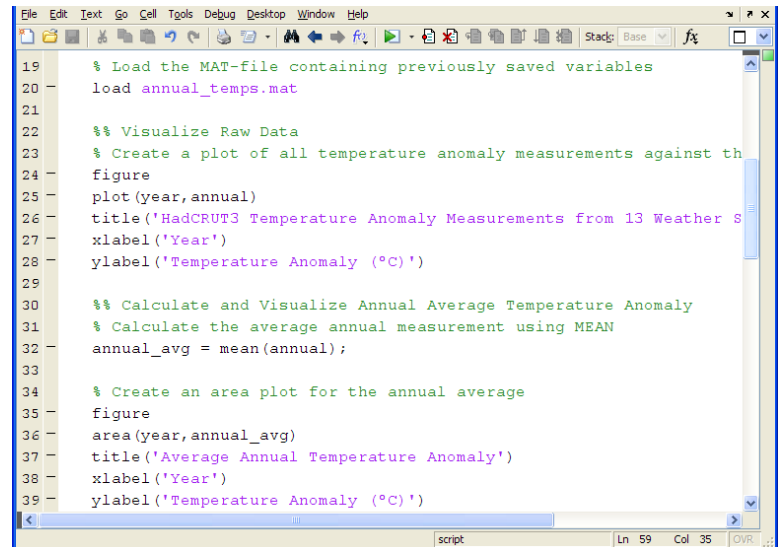
Programming logic  
control

# Writing MATLAB Programs

Using MATLAB Editor

Executing MATLAB script

Reusing MATLAB programs

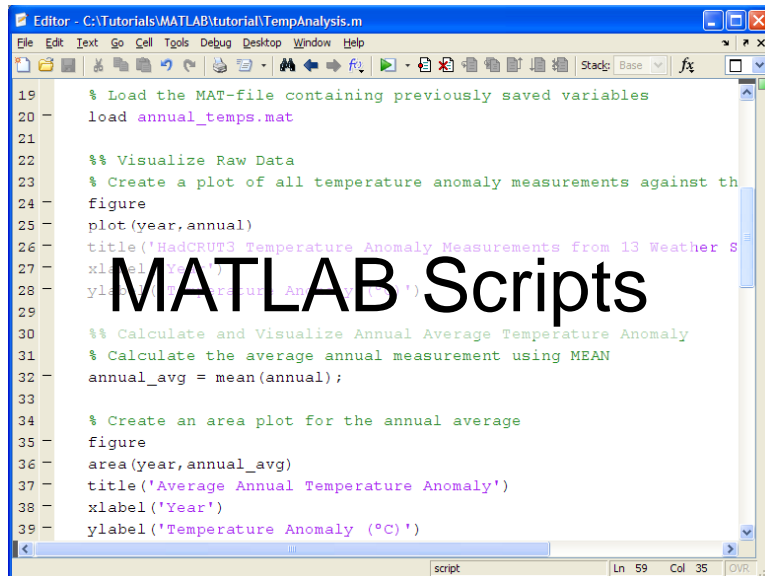


```

19 % Load the MAT-file containing previously saved variables
20 load annual_temps.mat
21
22 %% Visualize Raw Data
23 % Create a plot of all temperature anomaly measurements against th
24 figure
25 plot(year,annual)
26 title('HadCRUT3 Temperature Anomaly Measurements from 13 Weather S
27 xlabel('Year')
28 ylabel('Temperature Anomaly (°C)')
29
30 %% Calculate and Visualize Annual Average Temperature Anomaly
31 % Calculate the average annual measurement using MEAN
32 annual_avg = mean(annual);
33
34 % Create an area plot for the annual average
35 figure
36 area(year,annual_avg)
37 title('Average Annual Temperature Anomaly')
38 xlabel('Year')
39 ylabel('Temperature Anomaly (°C)')
  
```

# MATLAB Program Files

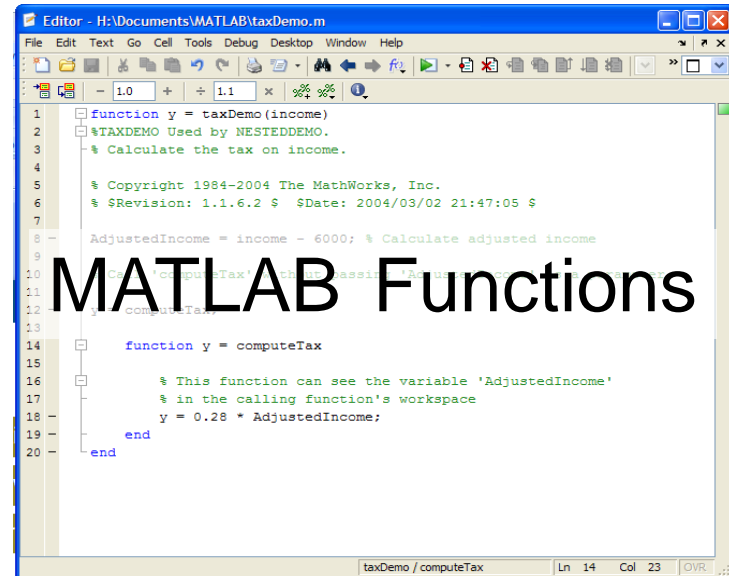
- Why?
  - Automating
  - Editing/Debugging
  - Deploying as applications



The image shows a MATLAB Editor window titled 'Editor - C:\Tutorials\MATLAB\tutorial\TempAnalysis.m'. The script contains the following code:

```
19 % Load the MAT-file containing previously saved variables
20 load annual_temps.mat
21
22 %% Visualize Raw Data
23 % Create a plot of all temperature anomaly measurements against th
24 figure
25 plot(year,annual)
26 title('HadCRUT3 Temperature Anomaly Measurements from 13 Weather S
27 xlabel('Year')
28 ylabel('Temperature Anomaly (°C)')
29
30 %% Calculate and Visualize Annual Average Temperature Anomaly
31 % Calculate the average annual measurement using MEAN
32 annual_avg = mean(annual);
33
34 % Create an area plot for the annual average
35 figure
36 area(year,annual_avg)
37 title('Average Annual Temperature Anomaly')
38 xlabel('Year')
39 ylabel('Temperature Anomaly (°C)')
```

## MATLAB Scripts



The image shows a MATLAB Editor window titled 'Editor - H:\Documents\MATLAB\taxDemo.m'. The function contains the following code:

```
1 function y = taxDemo(income)
2 %TAXDEMO Used by NESTEDDEMO.
3 % Calculate the tax on income.
4
5 % Copyright 1984-2004 The MathWorks, Inc.
6 % $Revision: 1.1.6.2 $ $Date: 2004/03/02 21:47:05 $
7
8 AdjustedIncome = income - 6000; % Calculate adjusted income
9
10 % Calculate the tax on adjusted income
11 y = computeTax(AdjustedIncome);
12
13
14 function y = computeTax
15
16 % This function can see the variable 'AdjustedIncome'
17 % in the calling function's workspace
18 y = 0.28 * AdjustedIncome;
19
20 end
```

## MATLAB Functions

# Basics of a MATLAB Program File

```
function f = fact(n)
% Compute a factorial value.
% FACT(N) returns the factorial of N,
% usually denoted by N!

% Put simply, FACT(N) is PROD(1:N).
f = prod(1:n);
```

```
function [y1, y2] = functionName(x1,x2,...)
```

```
>> help fact
```

```
Compute a factorial value.           H1 line
FACT(N) returns the factorial of N,  Help text
usually denoted by N!
```

```
>> fact(3)
```

```
ans =
```

```
6
```





# Types of Functions

- Primary MATLAB-file Functions

```
function [avg, med] = newstats(u) % Primary function
% NEWSTATS Find mean and median with internal functions.
n = length(u);
avg = mean(u, n);
med = median(u, n);
```

# Types of Functions

- Primary MATLAB-file Functions
- Subfunctions

```
function [avg, med] = newstats(u) % Primary function
% NEWSTATS Find mean and median with internal functions.
n = length(u);
avg = mean(u, n);
med = median(u, n);
```

```
function a = mean(v, n) % Subfunction
% Calculate average.
a = sum(v)/n;
```

```
function m = median(v, n) % Subfunction
% Calculate median.
w = sort(v);
if rem(n, 2) == 1
    m = w((n+1) / 2);
else
    m = (w(n/2) + w(n/2+1)) / 2;
end
```

# Types of Functions

- Primary MATLAB-file Functions
- Subfunctions
- Nested Functions

```
function x = A(p1, p2)
...
    function y = B(p3)
        ...
    end
...
end
```

```
function x = A(p1, p2)
...
    function y = B(p3)
        ...
    end

    function z = C(p4)
        ...
    end
...
end
```

# Types of Functions

- Primary MATLAB-file Functions
- Subfunctions
- Nested Functions
  
- Anonymous Functions
- Overloaded Functions
- Private Functions

# Outline

MATLAB Desktop

Computing in MATLAB

Problem Solving with MATLAB

**Working with data in MATLAB**

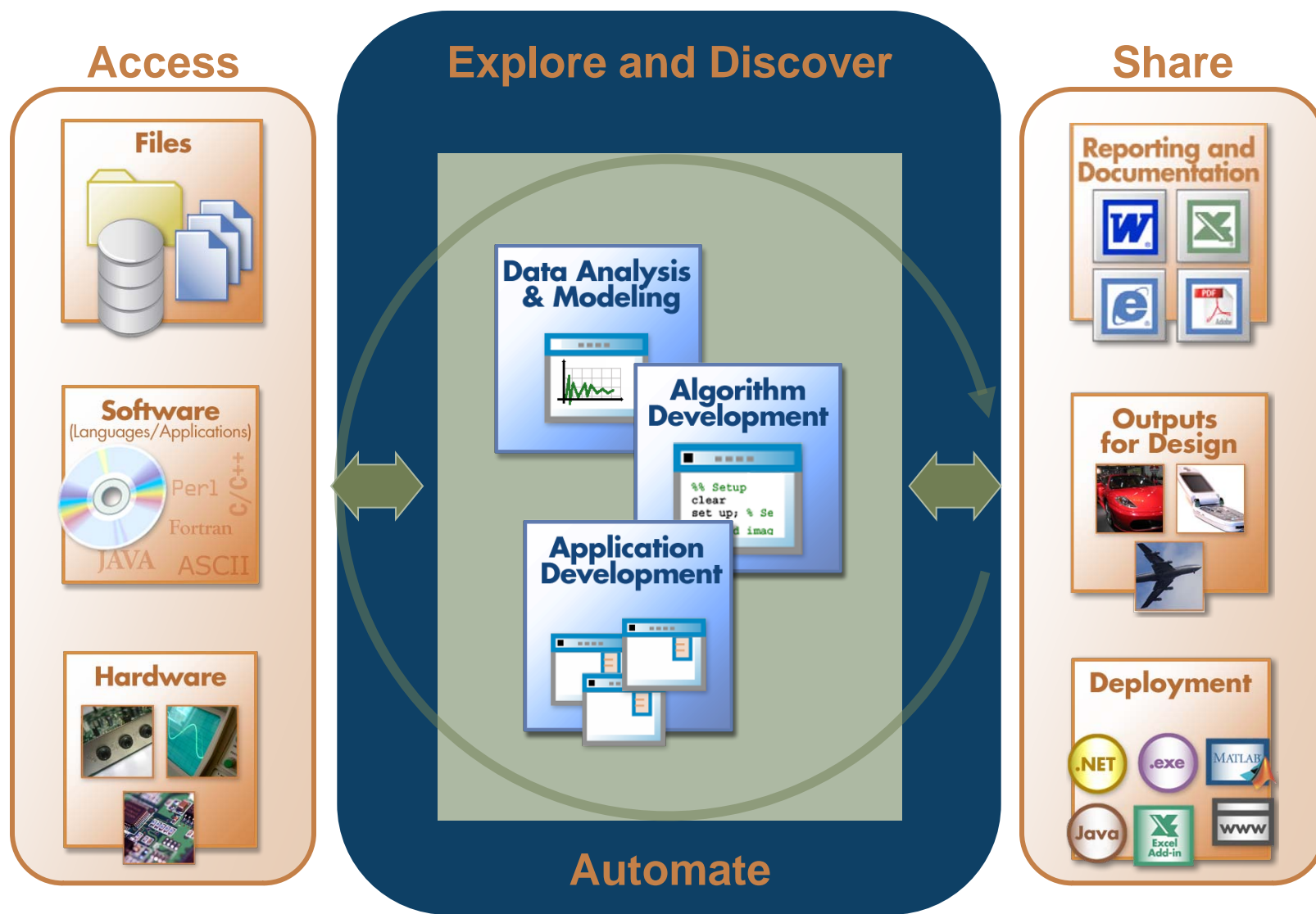
Programming in MATLAB

Import/Extract/Export

**Visualize**

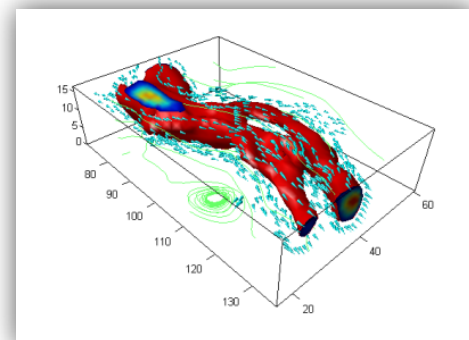
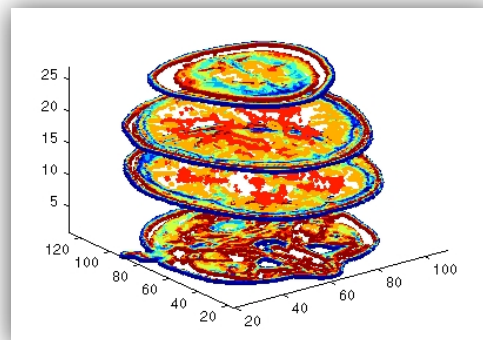
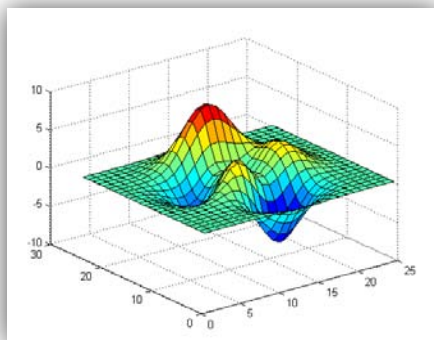
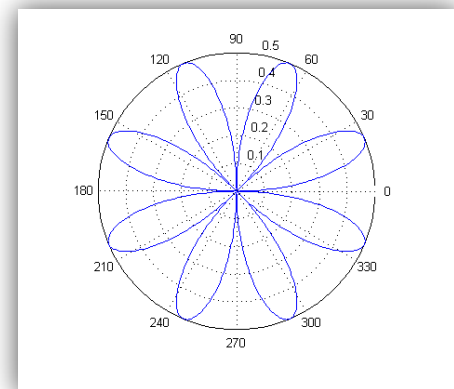
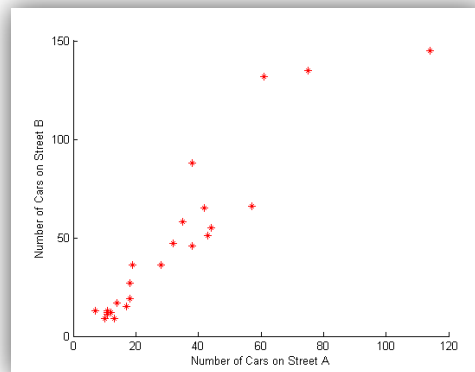
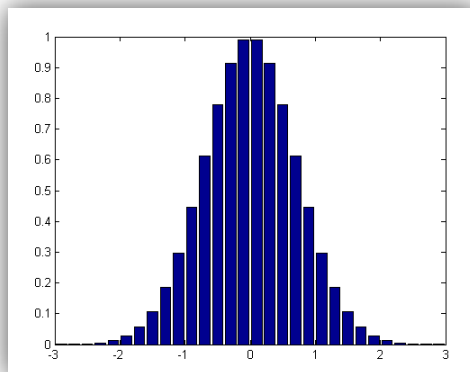
Analyze

# Technical Computing Workflow



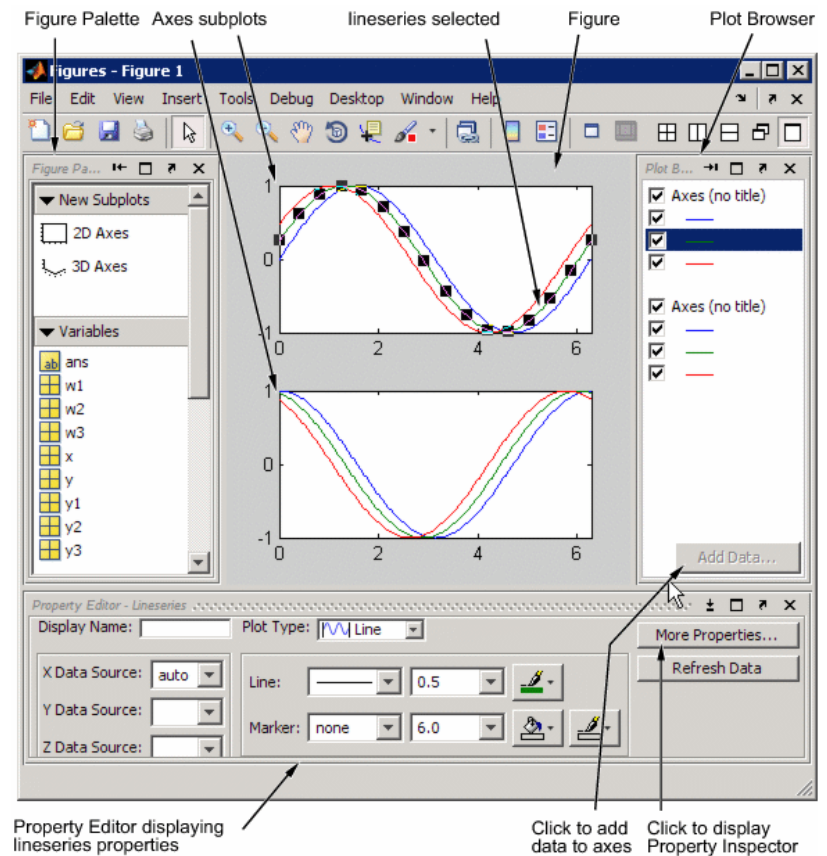
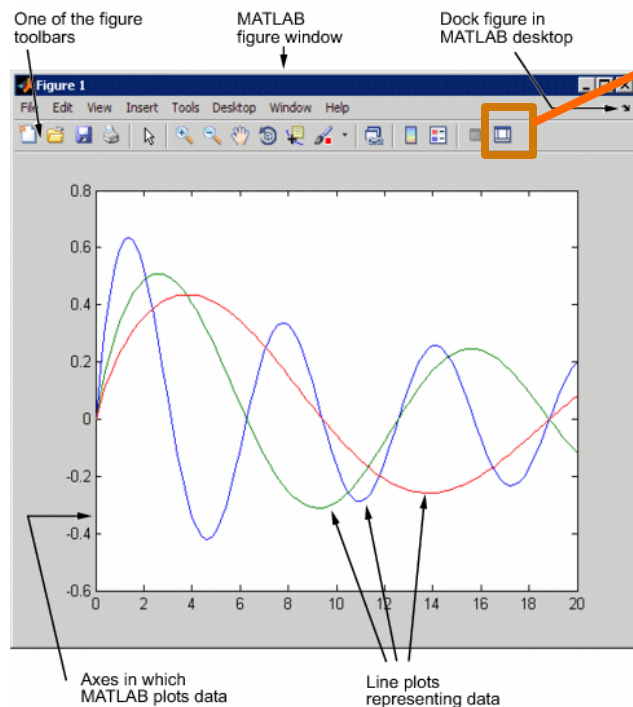
# Visualization Tools

- 2-d plots
- 3-d plots



# Plotting Tools

```
>> x = [0:0.2:20];
>> y = sin(x).sqrt(x+1);
>> y(2,:) = sin(x/2).sqrt(x+1);
>> y(3,:) = sin(x/3).sqrt(x+1);
>> plot(x,y);
```





# Outline

MATLAB Desktop

Computing in MATLAB

Problem Solving with MATLAB

**Working with data in MATLAB**

Programming in MATLAB

Import/Extract/Export

Visualize

**Analyze**

# Processing and Analyzing Data

- Data Analysis functions in MATLAB
  - Statistics
    - `cov` , `max`, `mean`, `median`, `std`
  - Filtering and Convolution
    - `conv`, `deconv`, `filter`, `filter2`
  - Interpolation and Regression
    - `interp1`, `interp`, `mldivide`, `polyfit`, `polyval`
  - Fourier Transforms
    - `fft`, `fftn`, `fftshift`, `ifft`, `unwrap`
  - Derivatives and Integrals
    - `del2`, `diff`, `gradient`, `polyint`, `trapz`



# Outline

MATLAB Desktop

Computing in MATLAB

Problem Solving with MATLAB

Working with data in MATLAB

**Programming in MATLAB**

Writing MATLAB programs

Programming logic control

# Outline

MATLAB Desktop

Computing in MATLAB

Problem Solving with MATLAB

Working with data in MATLAB

**Programming in MATLAB**

Writing MATLAB  
programs

**Programming  
logic control**

# Program Control Statements

- Conditional Control
  - `if/elseif/else, switch/case`
- Loop Control
  - `while, for, break`
- Error Control
  - `try, catch`

# Conditional Control Statements

- `if`, `elseif` and `else`

```
if n < 0                % If n negative, display error message.
    disp('Input must be positive');
elseif rem(n,2) == 0 % If n positive and even, divide by 2.
    A = n/2;
else
    A = (n+1)/2;        % If n positive and odd, increment and divide.
end
```

- `switch`, `case` and `otherwise`

```
switch input_num
    case -1
        disp('negative one');
    case 0
        disp('zero');
    case 1
        disp('positive one');
    otherwise
        disp('other value');
end
```

# Loop Control Statements

- while (Conditional Loop)

```
n = 1;
while prod(1:n) < 1e100
    n = n + 1;
end
```

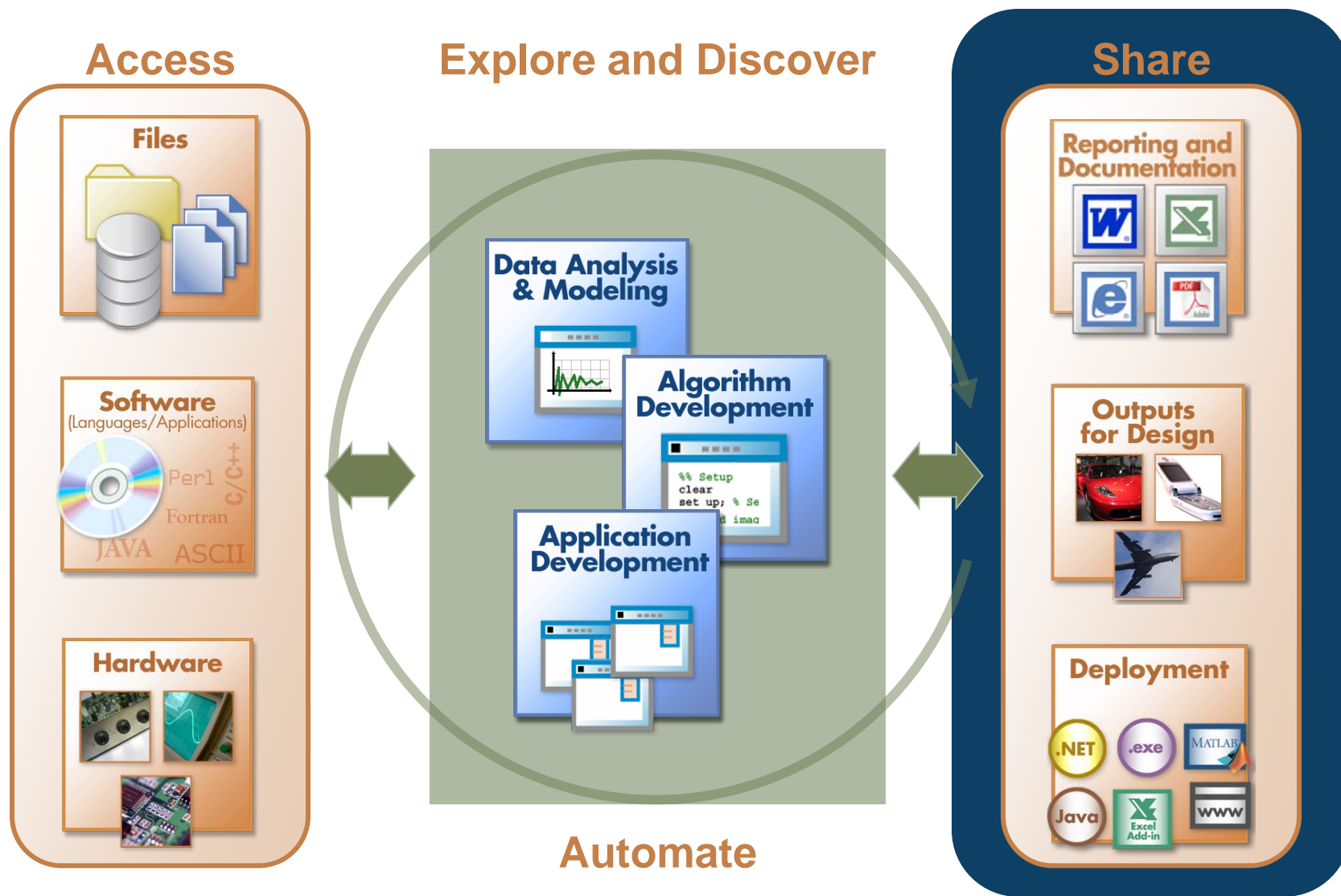
- for (Iterative Loop)

```
for index = start:increment:end
    statements
end
```

```
for m = 1:5
    for n = 1:100
        A(m, n) = 1/(m + n - 1);
    end
end
```

- continue, break

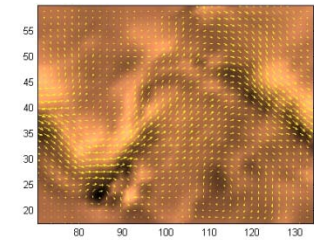
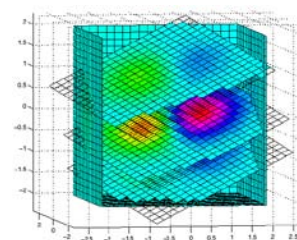
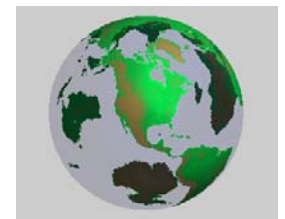
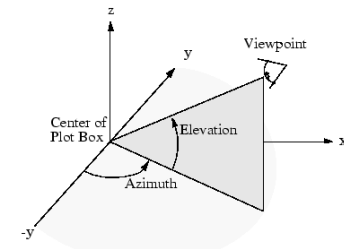
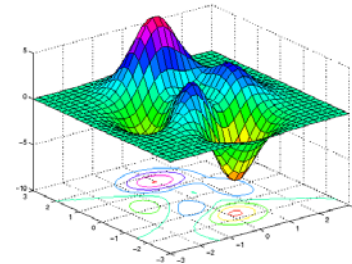
# Technical Computing Workflow





# 3-D Visualization Features

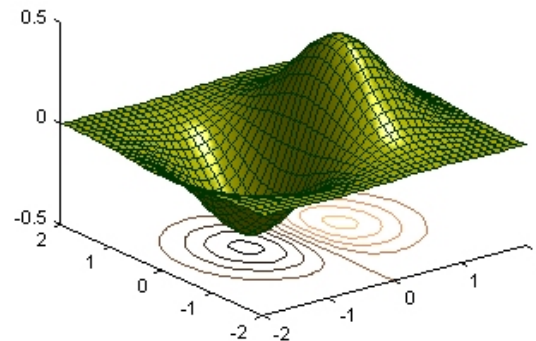
- **Surface and Mesh plots**
  - peaks, surf, mesh, meshgrid
  - colorbar, colormap, shading
- **View Control**
  - campos, view, daspect, rotate3d
- **Lighting and Transparency**
  - camlight, diffuse, alpha
- **Volume Visualization**
  - curl, isosurface, slice



# Object-Oriented Programming

```
Editor - C:\Program Files\MATLAB\R2009b\help\techdoc\matlab_oop\examples\@topo\...
File Edit Text Go Cell Tools Debug Desktop Window Help
1 classdef topo < handle % Plots function of 2 vars
2     properties
3         FigHandle      % Store figure handle
4         FofXY;          % Function handle to fcn being evaluated
5         Lm = [-2*pi 2*pi]; % Default range if not specified
6     end
7
8     properties (Dependent = true, SetAccess = private)
9         Data % Data property depends on current value of FofXY
10    end
11
12    methods
13        function obj = topo(fnc,limits)
14            obj.FofXY = fnc;
15            obj.Lm = limits;
16        end
17
18        function set.Lm(obj,lim)
19            % Lm property set after checking limits
20            if ~(lim(1) < lim(2))
21                disp('Bad limits, using [-2pi 2pi]')
```

```
tobj = topo(@(x,y) x.*exp(-x.^2-y.^2),[-2 2]);
a = tobj;
surflight(a) % Call class method to create a graph
```

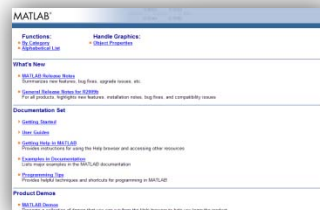
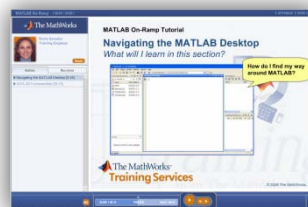


## External Interfaces

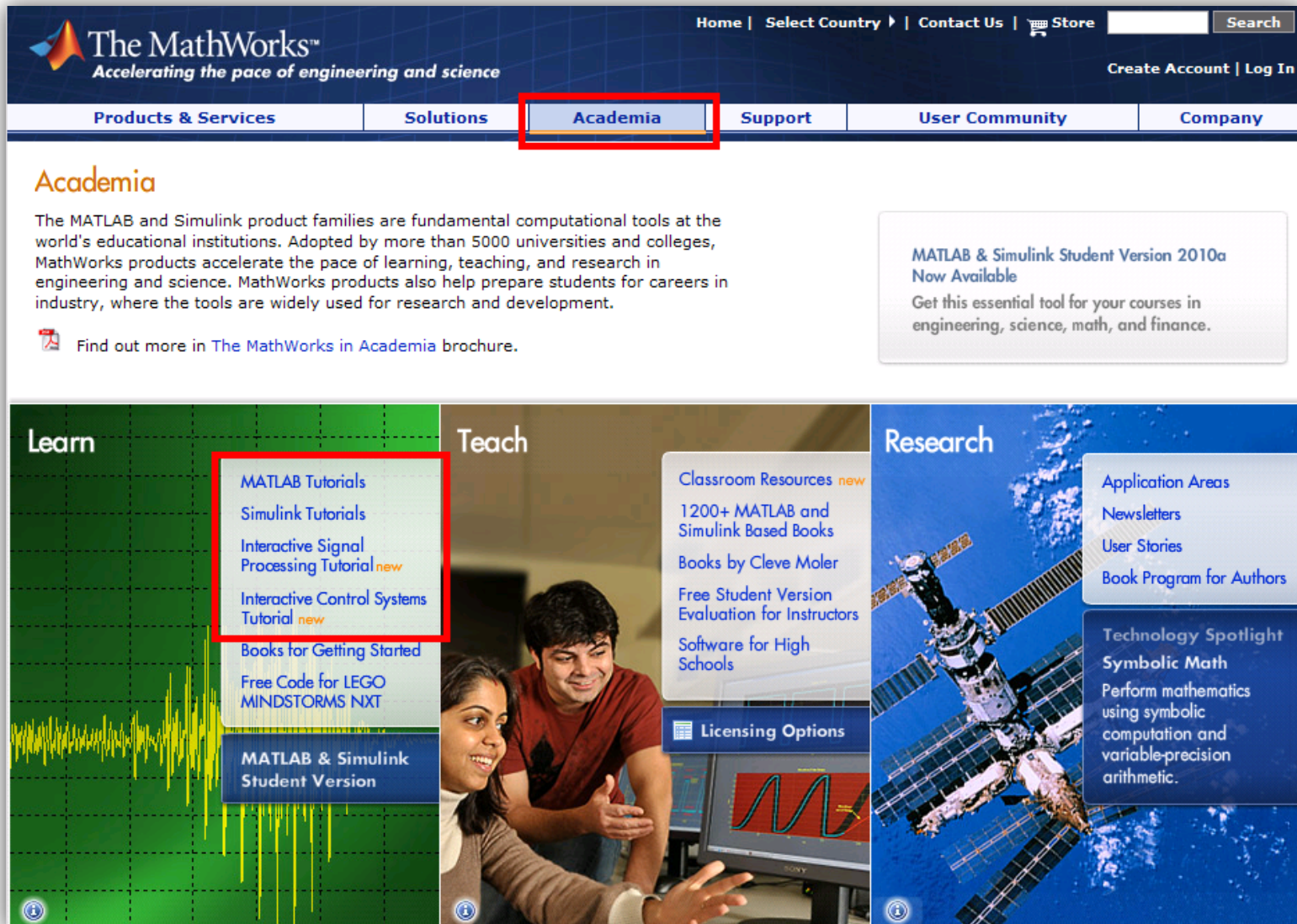
- Shared libraries (.dll, .so, .dylib)
- C, Fortran interface
- C, Fortran MEX-files (.mex)
- Sun Java classes
- COM/.NET support
- Web services
- Serial Port I/O

# More on MATLAB

- MATLAB Tutorials
- Demos and Webinars
- Documentation
- MATLAB Central  
(User Community)



# Training Tutorials



The screenshot shows the MathWorks Academia page. The top navigation bar includes links for Home, Select Country, Contact Us, Store, and a Search bar. Below this is a secondary navigation bar with links for Products & Services, Solutions, Academia (highlighted with a red box), Support, User Community, and Company. The main content area is titled "Academia" and features a paragraph about the educational use of MATLAB and Simulink. A red box highlights a list of resources under the "Learn" section, including MATLAB Tutorials, Simulink Tutorials, and various interactive tutorials. Other sections include "Teach" with classroom resources and "Research" with application areas and technology spotlights.

**The MathWorks™**  
Accelerating the pace of engineering and science


Home | Select Country | Contact Us | Store | Search

Create Account | Log In

Products & Services | Solutions | **Academia** | Support | User Community | Company

## Academia

The MATLAB and Simulink product families are fundamental computational tools at the world's educational institutions. Adopted by more than 5000 universities and colleges, MathWorks products accelerate the pace of learning, teaching, and research in engineering and science. MathWorks products also help prepare students for careers in industry, where the tools are widely used for research and development.

 Find out more in [The MathWorks in Academia brochure](#).

### Learn

- MATLAB Tutorials
- Simulink Tutorials
- Interactive Signal Processing Tutorial new
- Interactive Control Systems Tutorial new
- Books for Getting Started
- Free Code for LEGO MINDSTORMS NXT
- MATLAB & Simulink Student Version

### Teach

- Classroom Resources new
- 1200+ MATLAB and Simulink Based Books
- Books by Cleve Moler
- Free Student Version Evaluation for Instructors
- Software for High Schools
- Licensing Options

### Research


- Application Areas
- Newsletters
- User Stories
- Book Program for Authors
- Technology Spotlight
- Symbolic Math**
- Perform mathematics using symbolic computation and variable-precision arithmetic.




# Training Tutorials

[http://www.mathworks.com/academia/student\\_center/tutorials/ml\\_onramp/player.html](http://www.mathworks.com/academia/student_center/tutorials/ml_onramp/player.html)

MATLAB On-Ramp ( 00:09 / 34:03 )
APPENDIX | SEND LINK





Rocio Saracho  
Training Engineer

[Email](#)

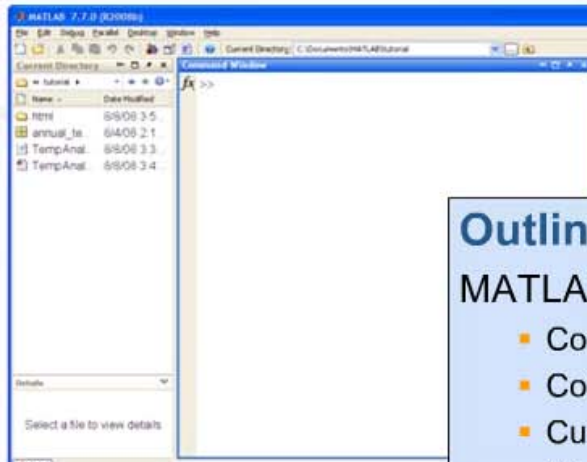
Outline
Narration

▶ Navigating the MATLAB Desktop [5:45]  
▼ MATLAB Fundamentals [28:15]  
    Section Outline  
    ▶ MATLAB as a Calculator  
        Creating Variables  
        Q&A – The Truth About Variables in MATLAB  
        Q&A – Creating a Variable  
    ▶ Accessing Data in MATLAB  
        Creating Vectors  
        Creating Matrices  
        Try It – Creating Arrays  
        Manipulating Arrays  
        Q&A – The Colon Operator (:)  
    ▶ Matrix Operations  
        Array Operations  
        Q&A – Matrix vs Array Operations  
        Visualizing of Mathematical Functions  
        Calling a MATLAB Function  
        Using plot to Visualize Computed Data

## MATLAB On-Ramp Tutorial

### Navigating the MATLAB Desktop

*What will I learn in this section?*




How do I find my way around MATLAB?

### Outline

#### MATLAB Desktop Environment

- Command Window
- Command History
- Current Directory Browser
- Workspace Browser


The MathWorks™  
**Training Services**

© 2008 The MathWorks, Inc.

SLIDE 1 OF 41
PAUSED
00:09 / 00:13

Transferring data from www.mathworks.com...

# Training Tutorials

MATLAB

Simulink

Control Systems

Signal Processing

# Demos

## Products &

### MATLAB Overview

- Description
- Function List
- Demos and Webinars**
- System Requirements
- Latest Features

### Support & Training

- Product Support
- Documentation
- Downloads & Trials
- Training
- Consulting

### Other Resources

- Technical Literature
- User Stories
- Related Books
- News and Events

## MATLAB 7.9

Learn more about MATLAB application examples.

### Demos

### Recorded

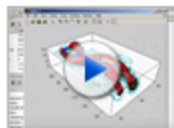
### MATLAB Overview



Develop  
overview  
to deter



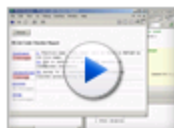
Analyz  
of data



Visualiz  
of data



Export  
the diff  
applica



Optimiz  
overview

## MATLAB 7.9

### Graphical Approach to Solving Inequalities

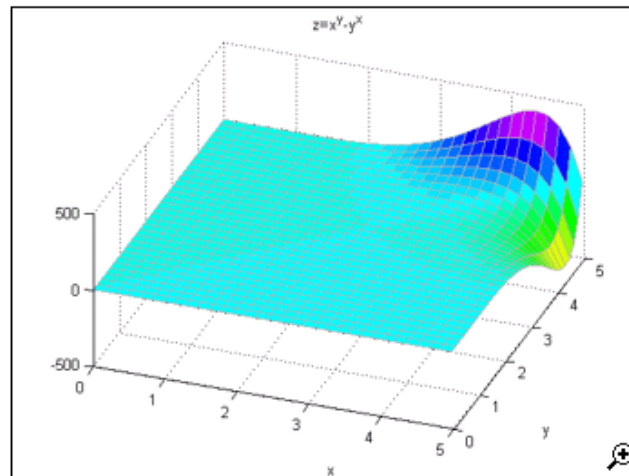
Here is an interesting graphical approach to find out whether  $e^{\pi}$  is greater than  $\pi^e$  or not.

The question is: which is greater,  $e^{\pi}$  or  $\pi^e$ ? The easy way to find out is to type it directly at the MATLAB® command prompt. But it motivates a more interesting question. What is the shape of the function  $z=x^y-y^x$ ? Here is a plot of  $z$ .

```
%Define the mesh
x=0:0.16:5;
y=0:0.16:5;
[xx,yy]=meshgrid(x,y);

%The plot
zz=xx.^yy-yy.^xx;
h=surf(x,y,zz);

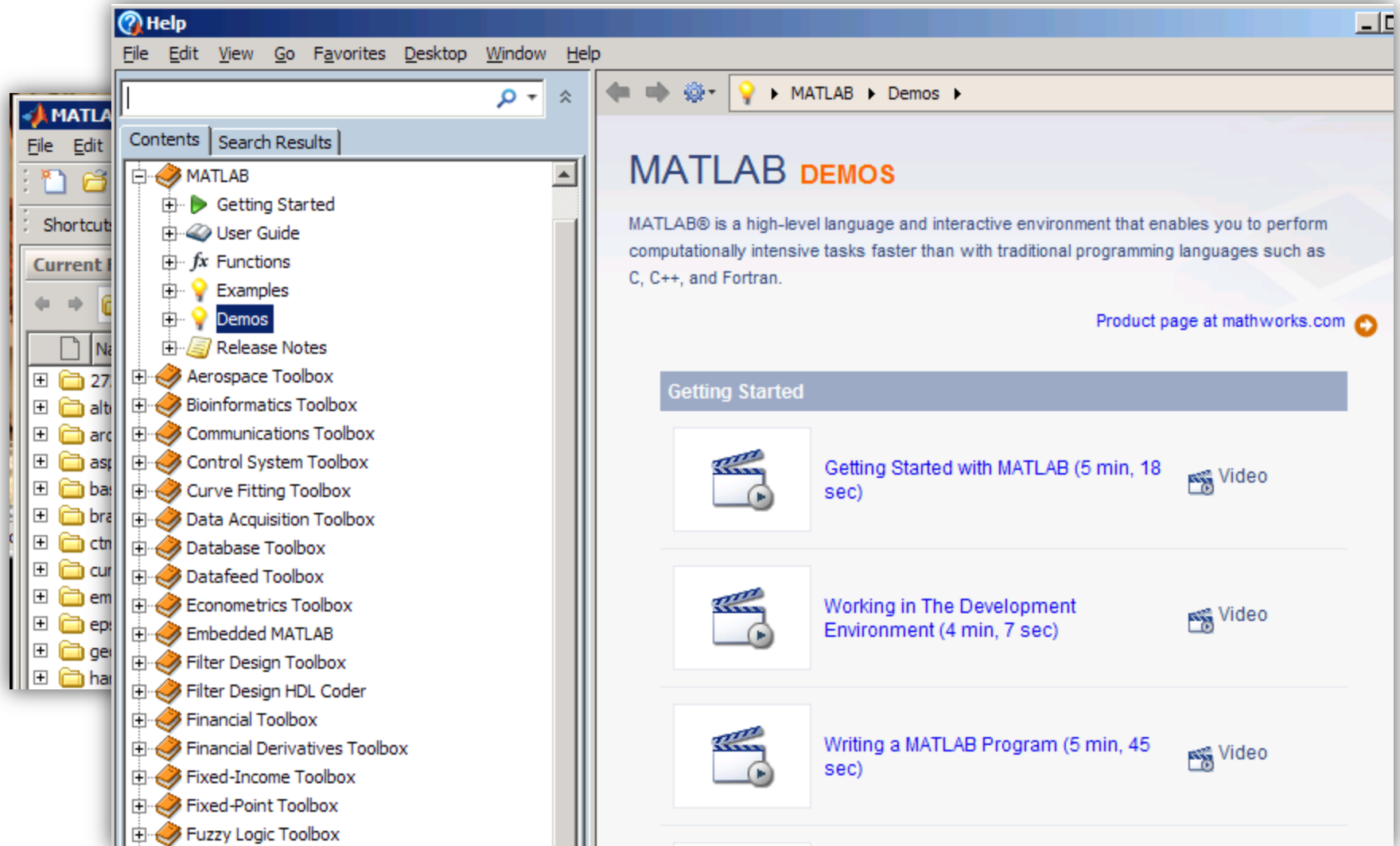
%Set the properties of the plot
set(h,'EdgeColor',[0.7 0.7 0.7]);
view(20,50);
colormap(hsv);
title('z=x^y-y^x'); xlabel('x'); ylabel('y');
hold on;
```



It turns out that the solution of the equation  $x^y - y^x = 0$  has a very interesting shape. Because



# Demos



The screenshot shows the MATLAB Help window with the 'Demos' section selected in the left-hand navigation pane. The main content area displays the 'MATLAB DEMOS' page, which includes a description of MATLAB as a high-level language and a list of video tutorials under the 'Getting Started' section.

**Contents** | Search Results




- MATLAB
  - Getting Started
  - User Guide
  - Functions
  - Examples
  - Demos**
  - Release Notes
- Aerospace Toolbox
- Bioinformatics Toolbox
- Communications Toolbox
- Control System Toolbox
- Curve Fitting Toolbox
- Data Acquisition Toolbox
- Database Toolbox
- Datafeed Toolbox
- Econometrics Toolbox
- Embedded MATLAB
- Filter Design Toolbox
- Filter Design HDL Coder
- Financial Toolbox
- Financial Derivatives Toolbox
- Fixed-Income Toolbox
- Fixed-Point Toolbox
- Fuzzy Logic Toolbox

**MATLAB DEMOS**

MATLAB® is a high-level language and interactive environment that enables you to perform computationally intensive tasks faster than with traditional programming languages such as C, C++, and Fortran.

[Product page at mathworks.com](http://mathworks.com)

**Getting Started**

- Getting Started with MATLAB (5 min, 18 sec)  Video
- Working in The Development Environment (4 min, 7 sec)  Video
- Writing a MATLAB Program (5 min, 45 sec)  Video

# Summary

- MATLAB is a **high level-language** for **technical computing**
- Interactive tool with **mathematical and graphical** functions
- MATLAB provides features to **access, compute, analyze and visualize data**
- MATLAB also provides capabilities to **interface with external languages**