Matlab Programming

Yunong Zhang (张雨浓)

Emails: zhynong@mail.sysu.edu.cn， jallonzyn@sina.com

Introduction

- What is Matlab?
- Why do we learn Matlab?
- How do we learn Matlab?

What have you learned from Matlab?

A Brief History of Matlab

- Engineering and scientific applications involve a lot of "number crunching".
- For many years, the main language for this was FORTRAN -- first "high level" programming language, and especially designed for numerical computing.
- Here's a Fortran code to solve $ax^2 + b x + c = 0$:

```
C     Solve a quadratic equation (this is a comment).
      DESC = B*B - 4*A*C
      IF ( DESC .LT. 0.0 ) GOTO 10
        DESC = SQRT(DESC)
        X1 = (-B + DESC)/(2.0*A)
        X2 = (-B - DESC)/(2.0*A)
        WRITE(6,*) "SOLUTIONS ARE ",X1," AND ", X2
        RETURN
   10 WRITE(6,*) "EQUATION HAS COMPLEX ROOTS"
      RETURN
```

Open topic 1.1: find all roots of an *n*th-degree polynomial (mathematically)

Problems using FORTRAN

"Number crunching" on a computer can be tricky.

Problems that occur are:

- loss of precision and inaccurate results:

  ```
  X = 0.1
  Y = 1.0 - 10*X
  ```

  Y "should" equal 0, but probably does not!
- underflow and overflow: `X = 1.0E20, X*X` --> too big!
- efficient coding of algorithms not always obvious

  ```
      DO 10 N=1,100000
   10 Y(N) = SQRT(2.0)*X(N)
  ```

  <-- less efficient! cf. inefficient
- programming errors!

Solving a Linear System in Fortran

Here's a Fortran code to solve a linear system A*x = b for x. It does not check for degeneracy or zeros.

```
C Solve B = A*X for X.
C N is dimension of vectors and matrix
C Does not use row interchange, scaling.
     SUBROUTINE LINSYS(N, A, X, B, TMP)
     INTEGER N
     DOUBLE PRECISION A(N,N), X(N), B(N)
     DOUBLE PRECISION TMP(N), RATIO
C... Forward elimination
     DO 13 J=1,N-1
       DO 12 I=J+1,N
         RATIO = -A(I,J)/A(J,J)
         A(I,*) = A(I,*) +RATIO*ROW(J,*)
         DO 11 K=J+1,N
  11     A(I,K) = A(I,K) + RATIO*A(J,K)
         A(I,J) = 0.0
         X(I) = X(I) + RATIO*X(J)
  12   CONTINUE
  11 CONTINUE
          Continued...
```

```
C... Backwards substitution
     X(N) = X(N)/A(N,N)
     DO 21 I=N-1,1,-1
       TMP = X(I)
       DO 20 J=I+1,N
  20   TMP = TMP - A(I,J)*X(J)
       X(I) = TMP/A(I,I)
  21 CONTINUE
     RETURN
     END
```

This is just a small example.

A full program may be thousands of lines long.

Need for Numerical Libraries

- The U.S. government recognized these problems, and the inefficiency of many engineers all writing the *same* algorithms... again and again.
- So, they commissioned *numerical analysts* to write good quality algorithms for common tasks.
- Make the results freely available as "libraries" of subroutines so that anyone can use in their programs.
- Libraries are available at: www.netlib.org

Examples of Numerical Libraries

- BLAS (Basic Linear Algebra Subroutines): operations on vectors, like adding to vectors, dot product, norm.
- LINPACK: linear algebra subroutines for vector-matrix operations, solving linear systems, factoring a matrix, inverting a matrix.   Later replaced by LAPACK.
- EISPACK: compute eigenvalues and eigenvectors of matrices.
- Example: solve A*x = b using LINPACK

```
C.... factor the A matrix
      CALL SGEFA(A, N, N, IPVT, INFO)
C.... copy B vector into X vector
      CALL SCOPY(N, B, 1, X, 1)
C.... solve the system of equations
      CALL SGESL(A, N, N, IPVT, X, 0)
```

Still Not Easy Enough!

When? 1984

- Cleve Moler, mathematician, C.S. Professor, and co-author of LINPACK, thought this is still too much work:
  - write FORTRAN, compile, debug, compile, run...
- He wanted to give students easy access to LINPACK.
- So, he wrote MATLAB ("Matrix Laboratory").
  - interactive
  - easy input, easy output
  - operations on a whole vector or matrix at once
- Example: solve b = A*x in Matlab...

```
x = A \ b
```

Immediate Popularity!

- MATLAB quickly became quite popular and used for both teaching and research. It was also *free*.
- An engineer, Jack Little, saw Matlab during a lecture by Cleve Moler at Stanford University.
- He saw the commercial potential and (with permission)
  - rewrote Matlab in C
  - added "M-files" (stored programs)
  - added many new features and libraries
  - co-founded *The Mathworks* to market it.

Software principles...

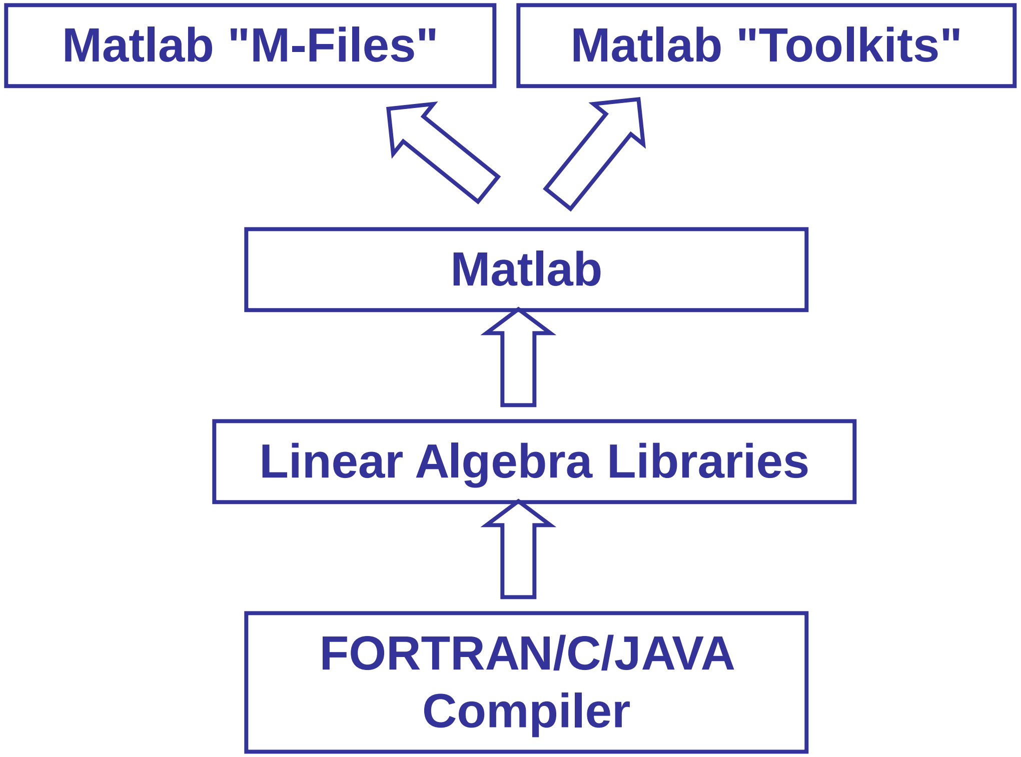
- Matlab illustrates some useful design concepts for software.

Extensible using "Toolkits" or user-contributed programs called M-files.

Interactive user interface; hides boring details

Modular, reusable software components

Standard base platform



Matlab "M-Files"

Matlab "Toolkits"

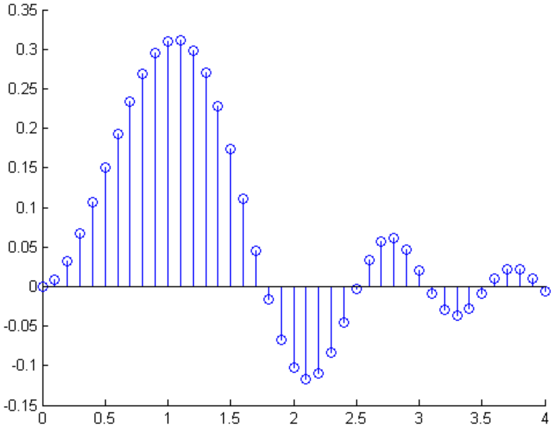
Matlab

Linear Algebra Libraries

FORTRAN/C/JAVA Compiler

Matlab Today

- Millions of users!
- A standard tool in both professional and academic use
- "Toolboxes" providing functions for many applications:
  - control systems
  - identification
  - neural networks
  - bio-informatics
  - statistics and time-series analysis



- Can do symbolic mathematics, too.
- Simulink: GUI based simulation tool

Summary

- Matrix Laboratory
- High-performance language for technical computing
- Computation, visualization, and programming in an easy-to-use environment

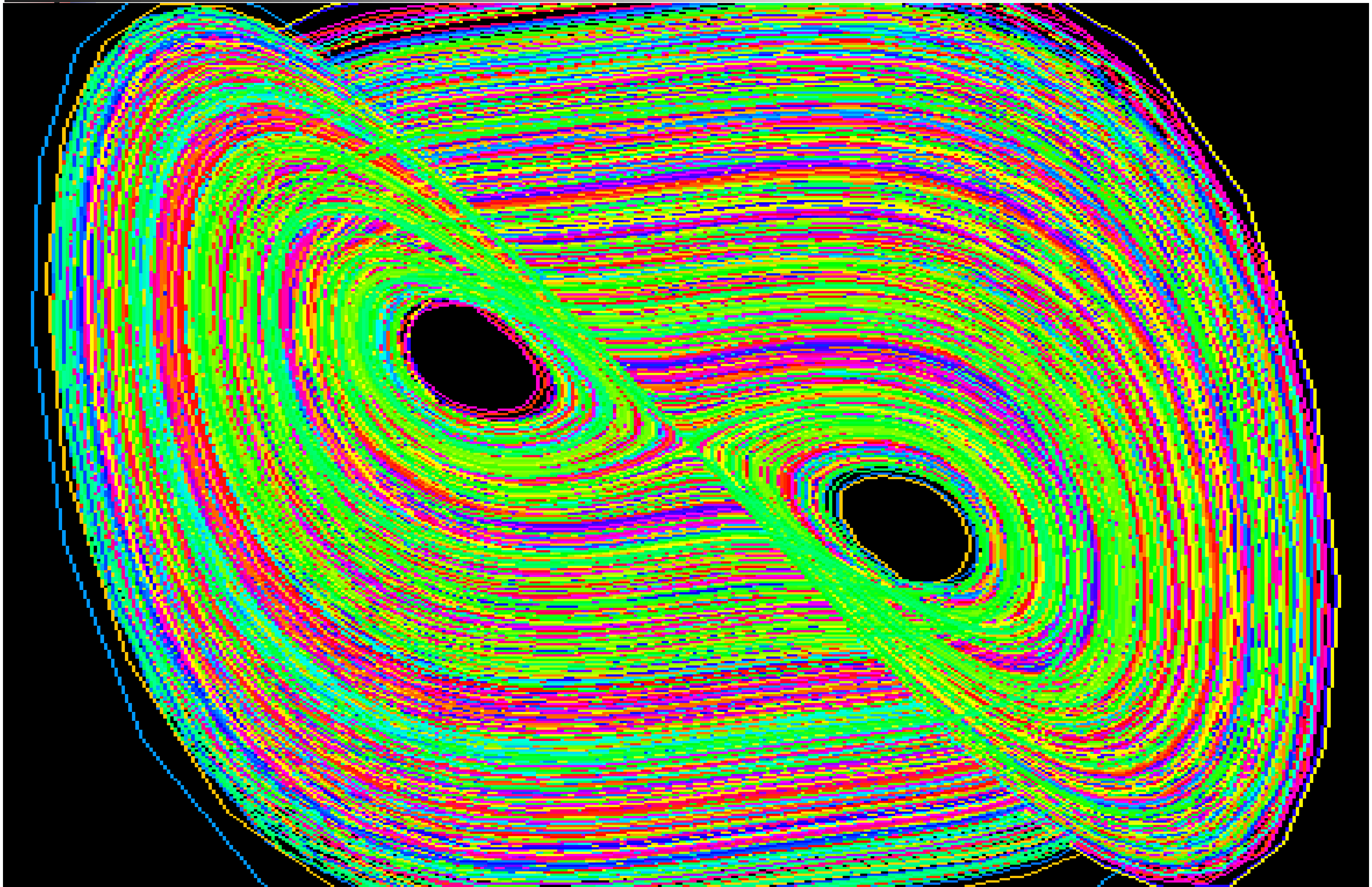
University-level computer-programming language

cf. JAVA/C/Fortran/VB/… and machine-level CPA

Typical Applications

- Math and computation
- Algorithm development
- Modelling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including Graphical User Interface building

Applications: Example 1



Applications: Example 1 (cont.)

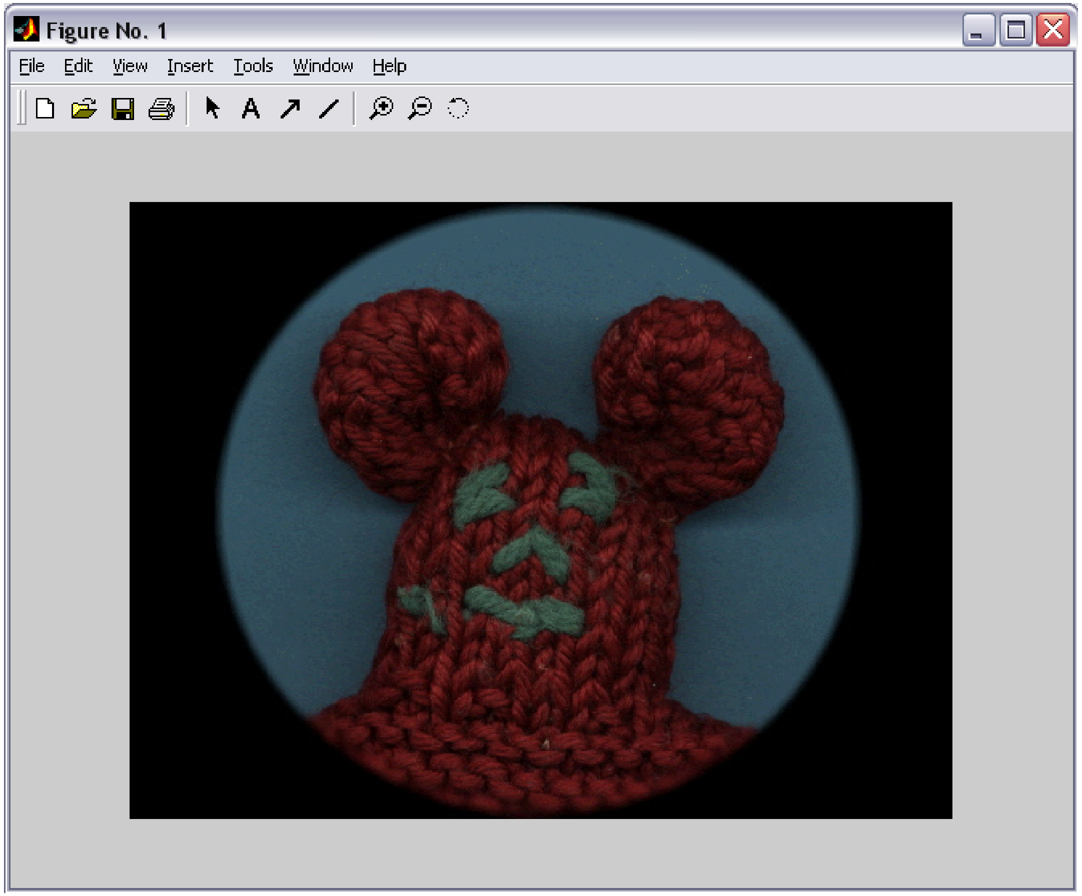
- Butterfly Effect

- Solution of ODE

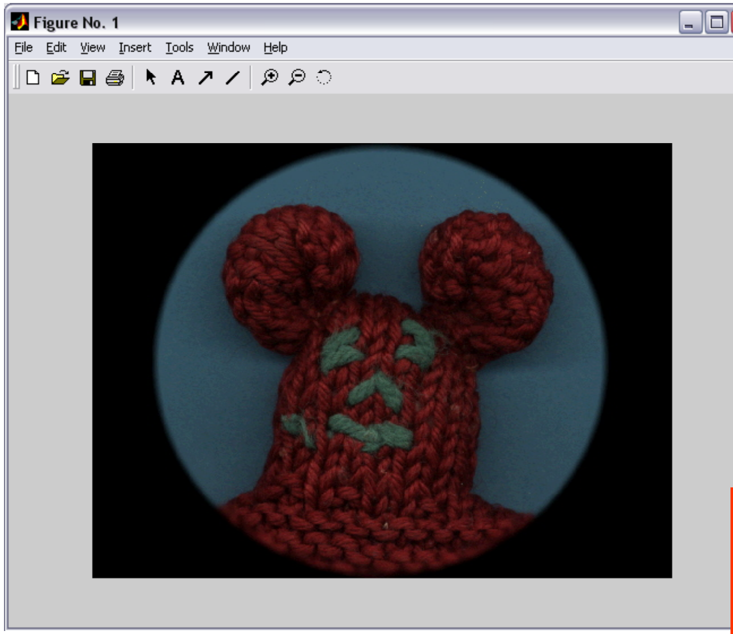
$$x' = 3(y - x)$$
$$y' = -xz + 26.5x - y$$
$$z' = xy - z$$

- ODE45 function in Matlab

Applications: Example 2

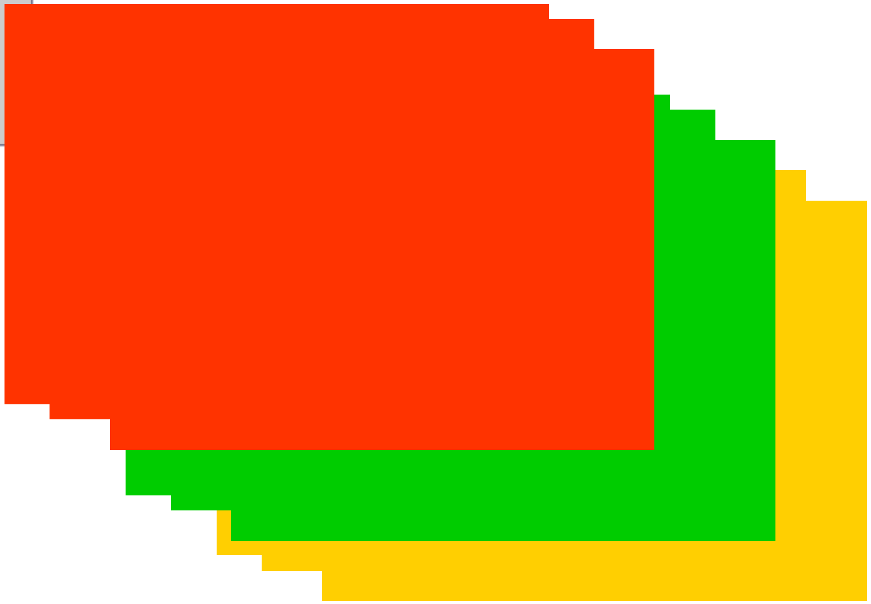


Applications: Example 2 (Cont.)

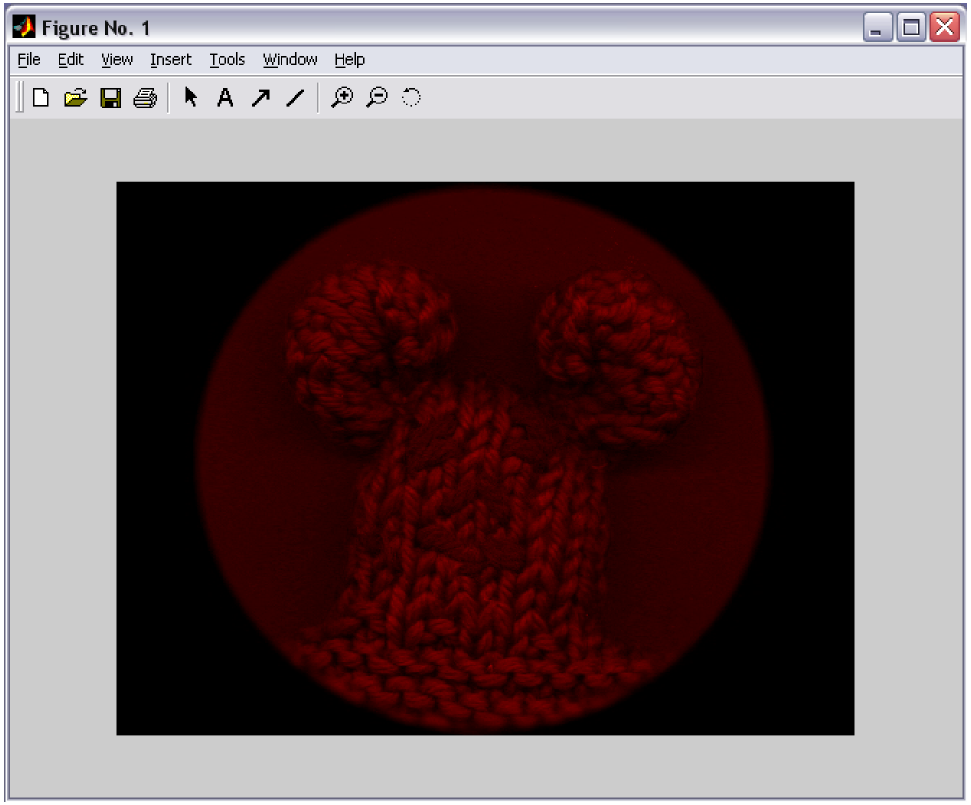


Loading an image:

```
a = imread(‘picture.jpg’);
imshow(a);
```



Applications: Example 2 (Cont.)



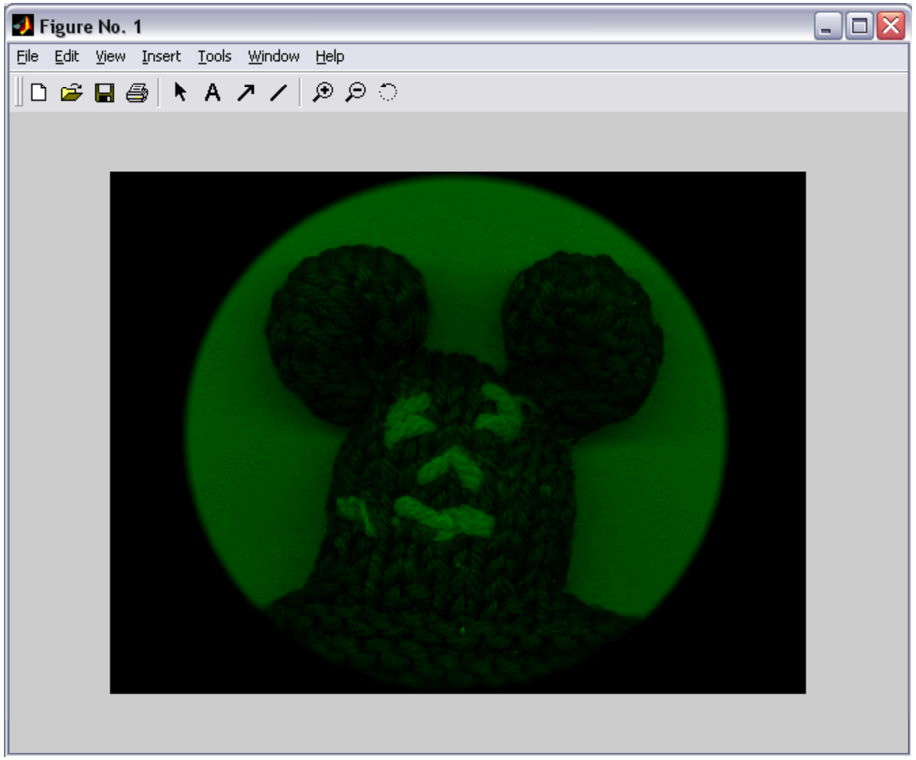
Loading an image:

```
a = imread(‘picture.jpg’);
imshow(a);
```

Show RED plane:

```
a(:,:,2:3) = 0;
imshow(a);
```

Applications: Example 2 (Cont.)

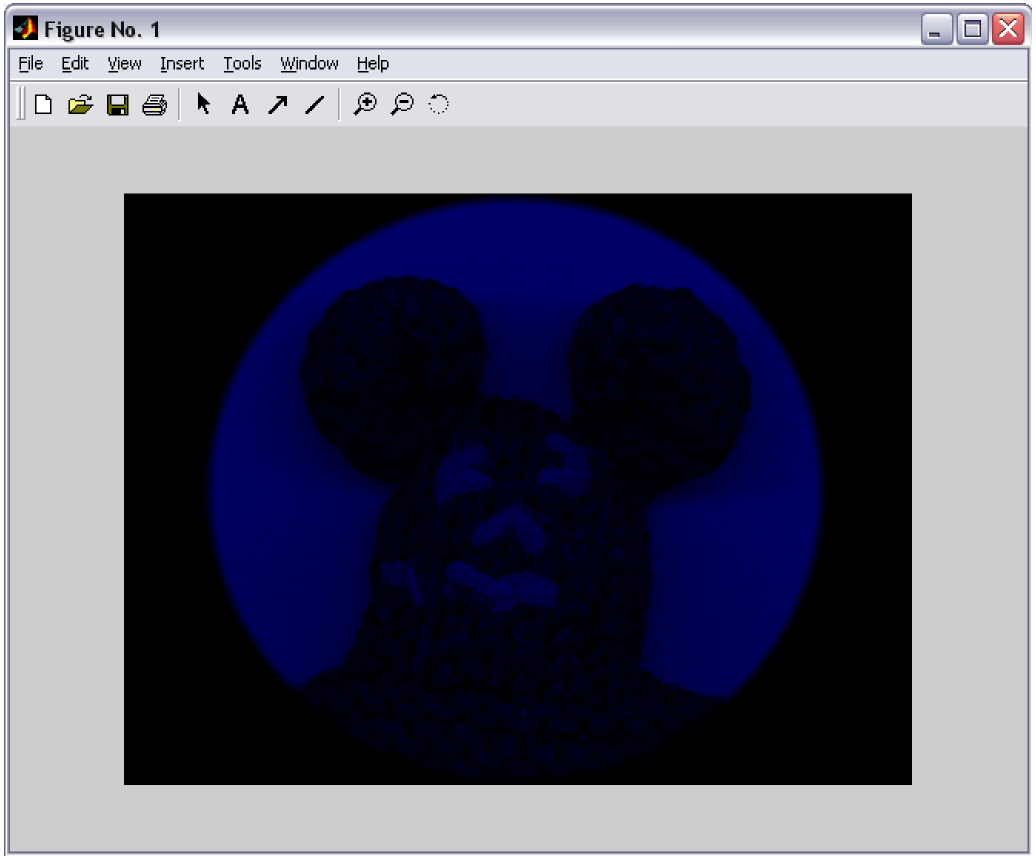


Show GREEN plane:

```
a(:,:,[1 3]) = 0;
imshow(a);
```

Applications: Example 2 (Cont.)

Why?



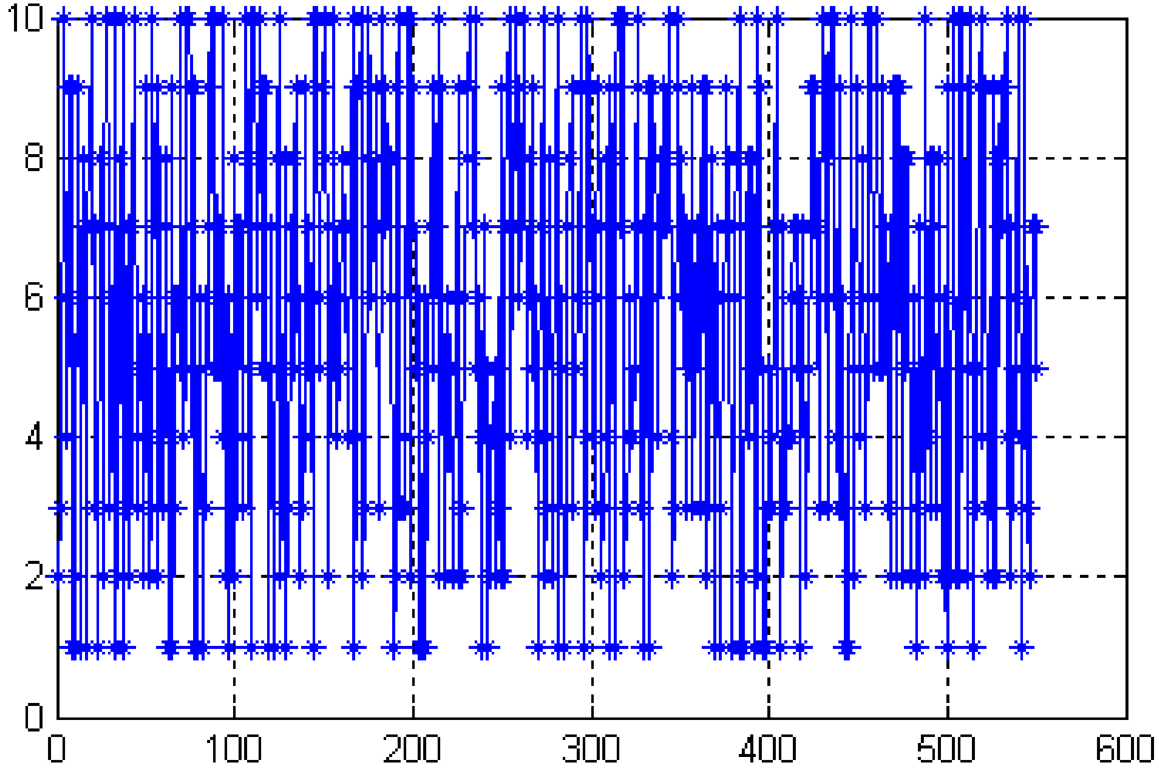
Show BLUE plane:

```
a(:,:,1:2) = 0;
imshow(a);
```

Open topic 1.2: color picture ←→ black-and-white / grey picture?

Applications: Example 3

- Data Analysis



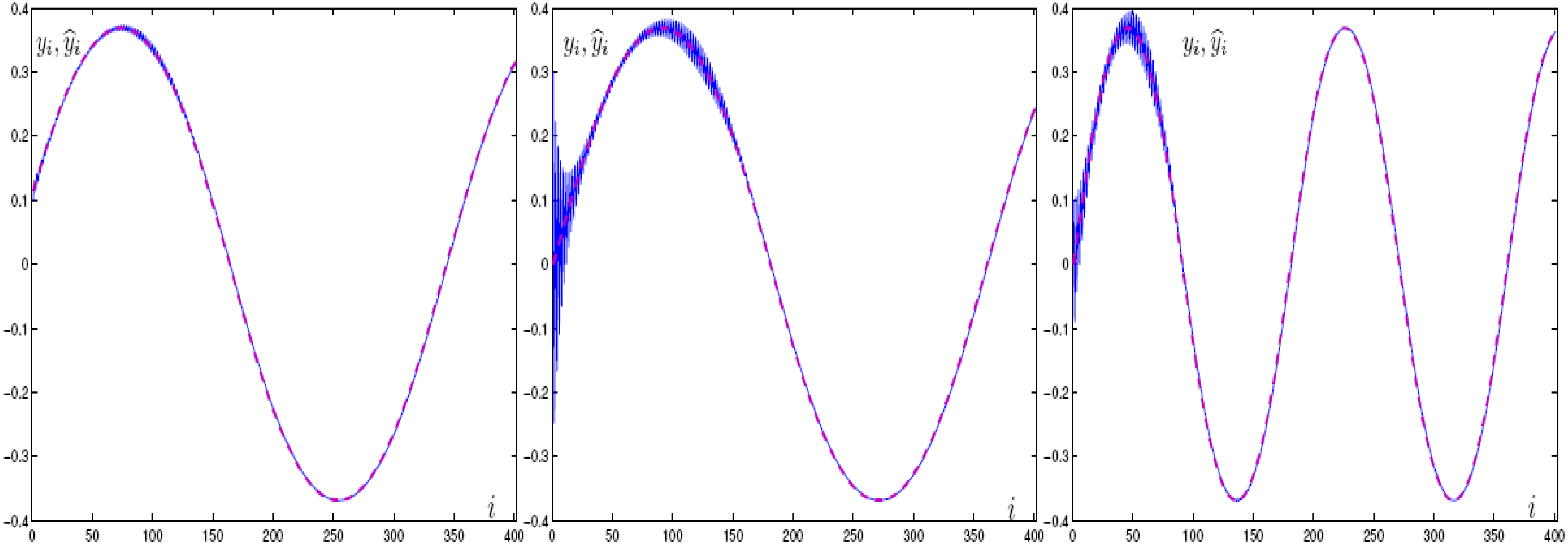
Stock market

Applications: Example 3 (cont.)

- Predicted by Neural Network Toolbox in Matlab
- BP, RBF Neural Network
- Training the network: *train* function

Applications: Example 3 (cont.)

Predicting example



(a) different-phase testing via (7)

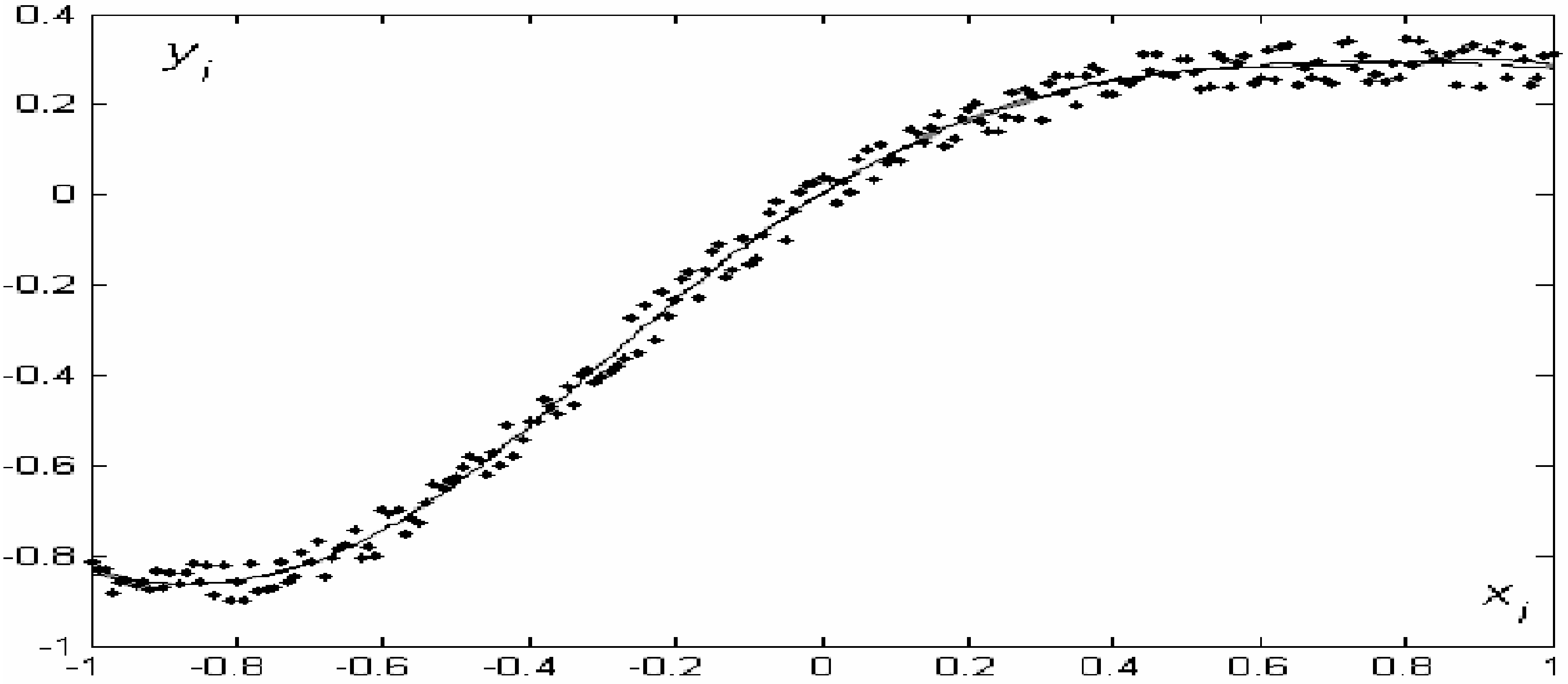
(b) different-initial-value testing via (7)

(c) different-frequency testing via (7)

Fig. 6. Comparison on neural-prediction and system-response (further testing situation by using weights-direct-determination method)

Applications: Example 3 (cont.)

De-noise example



（c） 直接确定法生成网络的去噪声情况

图 3 三种权值方法的网络去噪声逼近情况

Advantages of Matlab

- Ease to Use

  Interpreted Language, like Basic;

  Integrated development environment;

  Online documentation, manuals, and demos, etc.

- Platform Independence

  Independence of operation systems and computers: Windows 95/98/ME/NT/2000/XP, Unix, Linux, and super-computers

Advantages of Matlab (Cont.)

How to define?

- Predefined Functions

  Extensive library of predefined functions; such as arithmetic mean, standard deviation, median, etc;

  Toolboxes such as Communications, Control systems, Signal Processing,…

- Graphical User Interface

  Interactive;

  Easier to use for inexperienced users

Disadvantages of Matlab

- Slow for some kinds of processes

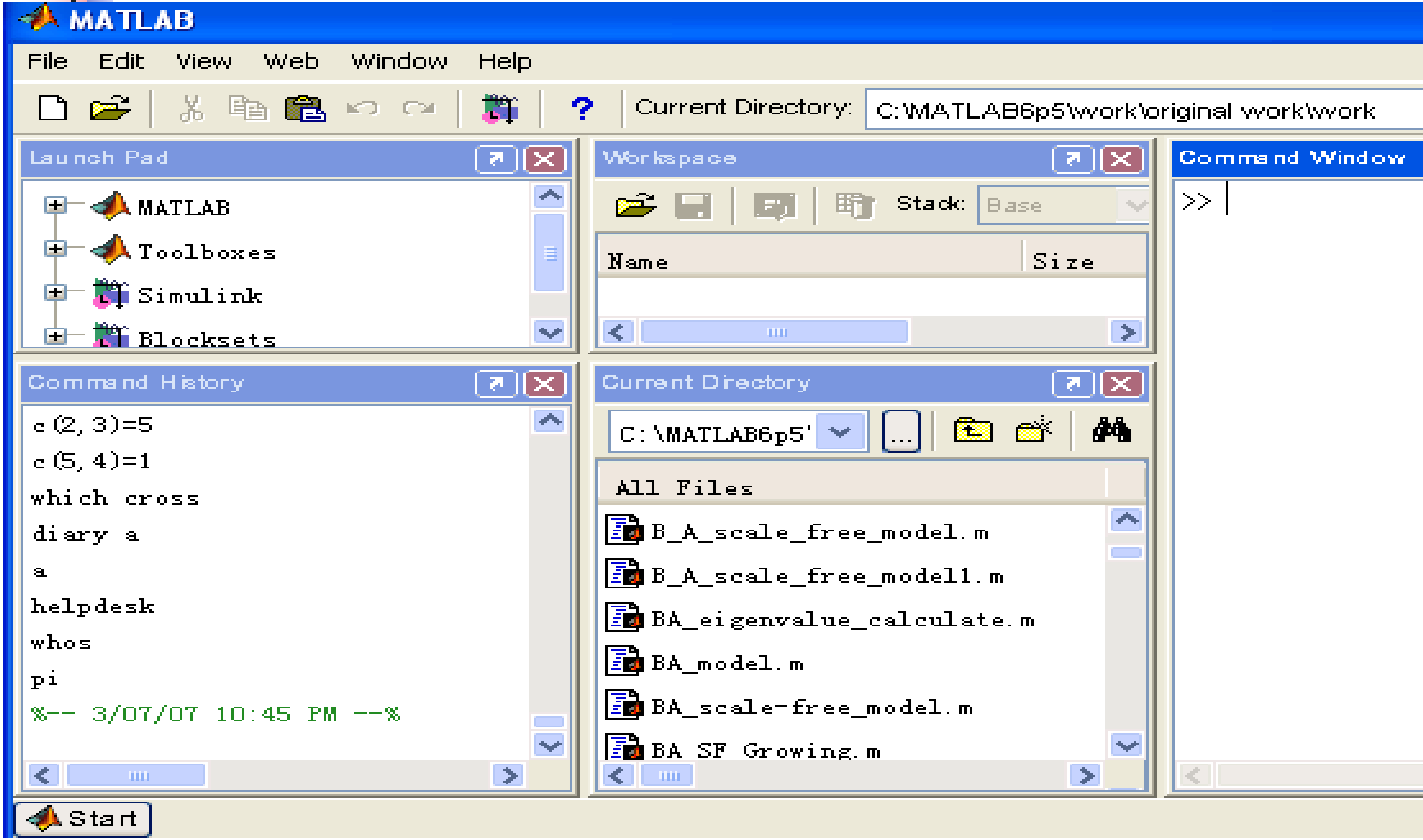
  Interpreted Language

  Not designed for large-scale system development

- High cost

  Expensive for individuals

Matlab Environment



```
>>
```

```
c(2,3)=5
c(5,4)=1
which cross
diary a
a
helpdesk
whos
pi
%-- 3/07/07 10:45 PM --%
```

```
B_A_scale_free_model.m
B_A_scale_free_model1.m
BA_eigenvalue_calculate.m
BA_model.m
BA_scale-free_model.m
BA_SF_Growing.m
```

Matlab Environment (Cont.)

Matlab desktop includes the following windows:

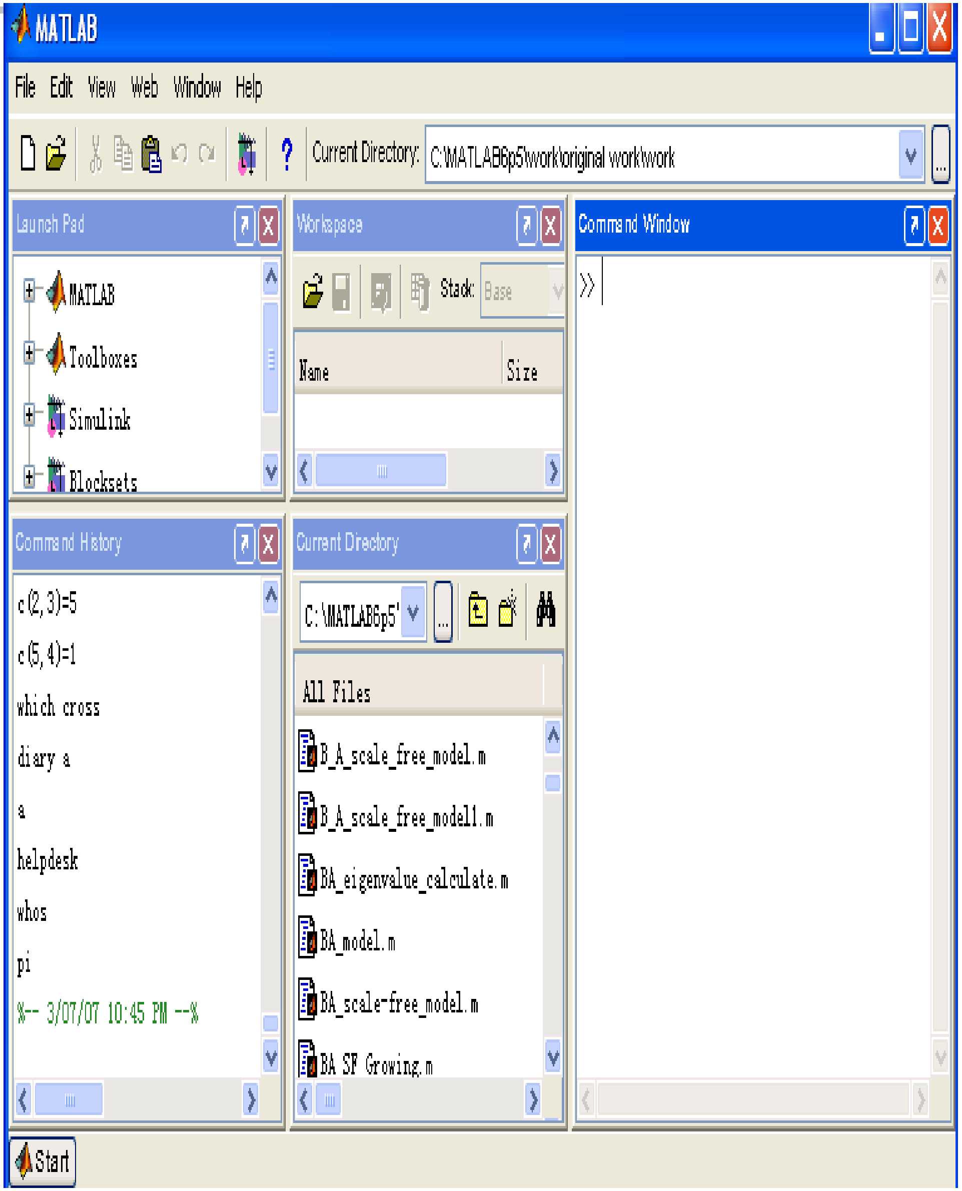
The Command Window;

The Command History Window;

Launch Pad;

Workspace;

Current Directory



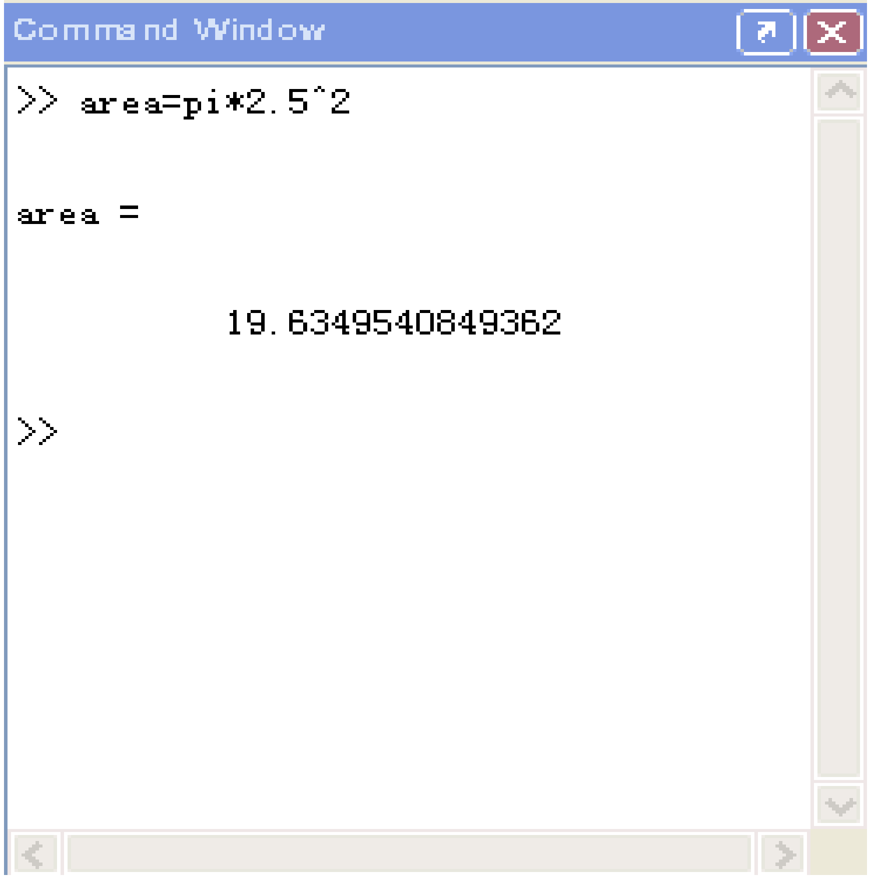
```
c(2,3)=5
c(5,4)=1
which cross
diary a
a
helpdesk
whos
pi
%-- 3/07/07 10:45 PM --%
```

The Command Window

- Allow users to enter commands at/under the command prompt
- Matlab computes the answer once the Enter key is pressed
- **Ellipsis** (…) is used if a statement is too long.

cf. ellipse!

```
x1=1+1/2+1/3+1/4+1/5+1/6

 x1=1+1/2+1/3+1/4…
      +1/5+1/6
```

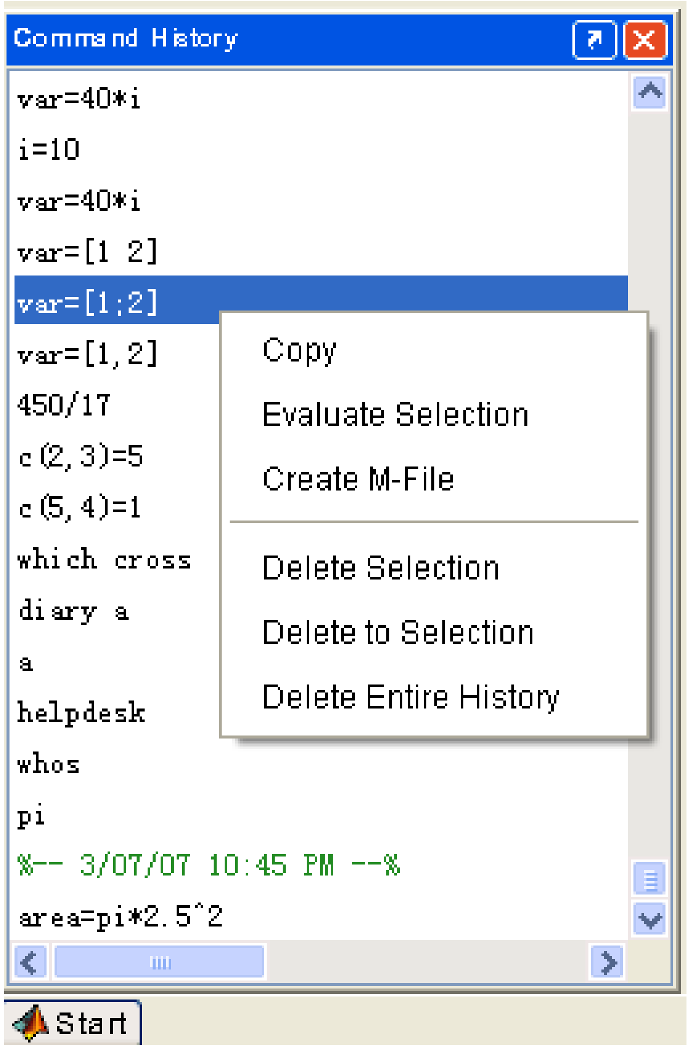


```
>> area=pi*2.5^2

area =

          19.6349540849362

>>
```

The Command History Window

- Display a list of the commands typed before
- Re-execute a command by double-clicking it
- Delete a command by right clicking it and selecting the item ``Delete Section” from the popup menu

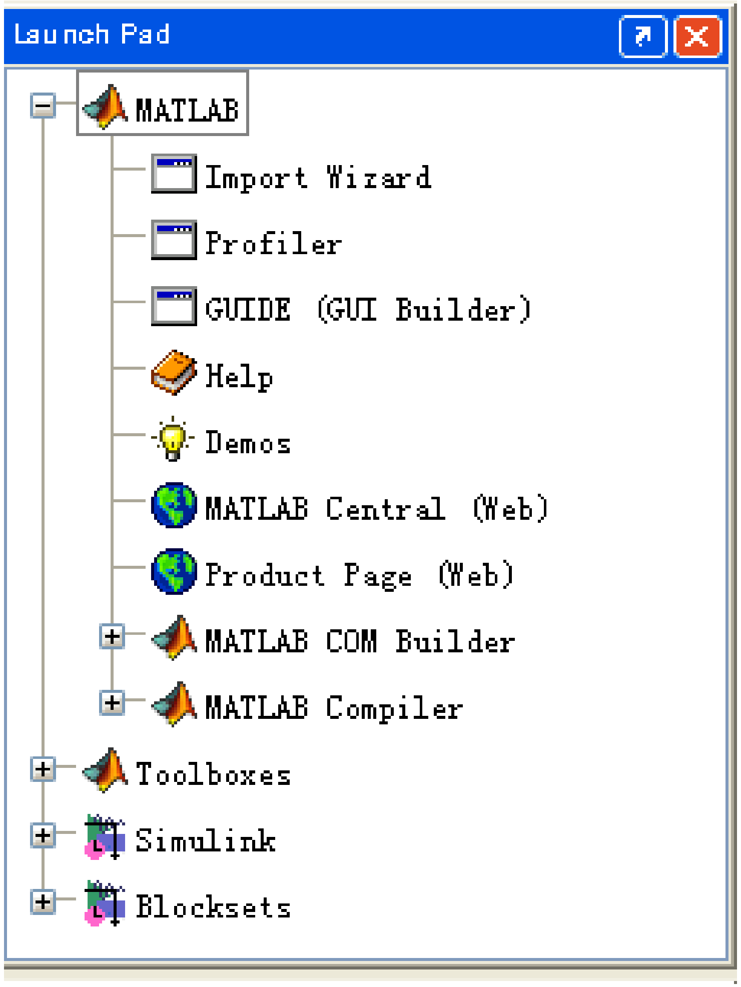
What if?



```
var=40*i
i=10
var=40*i
var=[1 2]
var=[1;2]
var=[1,2]
450/17
c(2,3)=5
c(5,4)=1
which cross
diary a
a
helpdesk
whos
pi
%-- 3/07/07 10:45 PM --%
area=pi*2.5^2
```

The Launch Pad

- A tree of documentation, demos, and related tools installed here



Launch Pad

MATLAB
Import Wizard
Profiler
GUIDE (GUI Builder)
Help
Demos
MATLAB Central (Web)
Product Page (Web)
MATLAB COM Builder
MATLAB Compiler
Toolboxes
Simulink
Blocksets

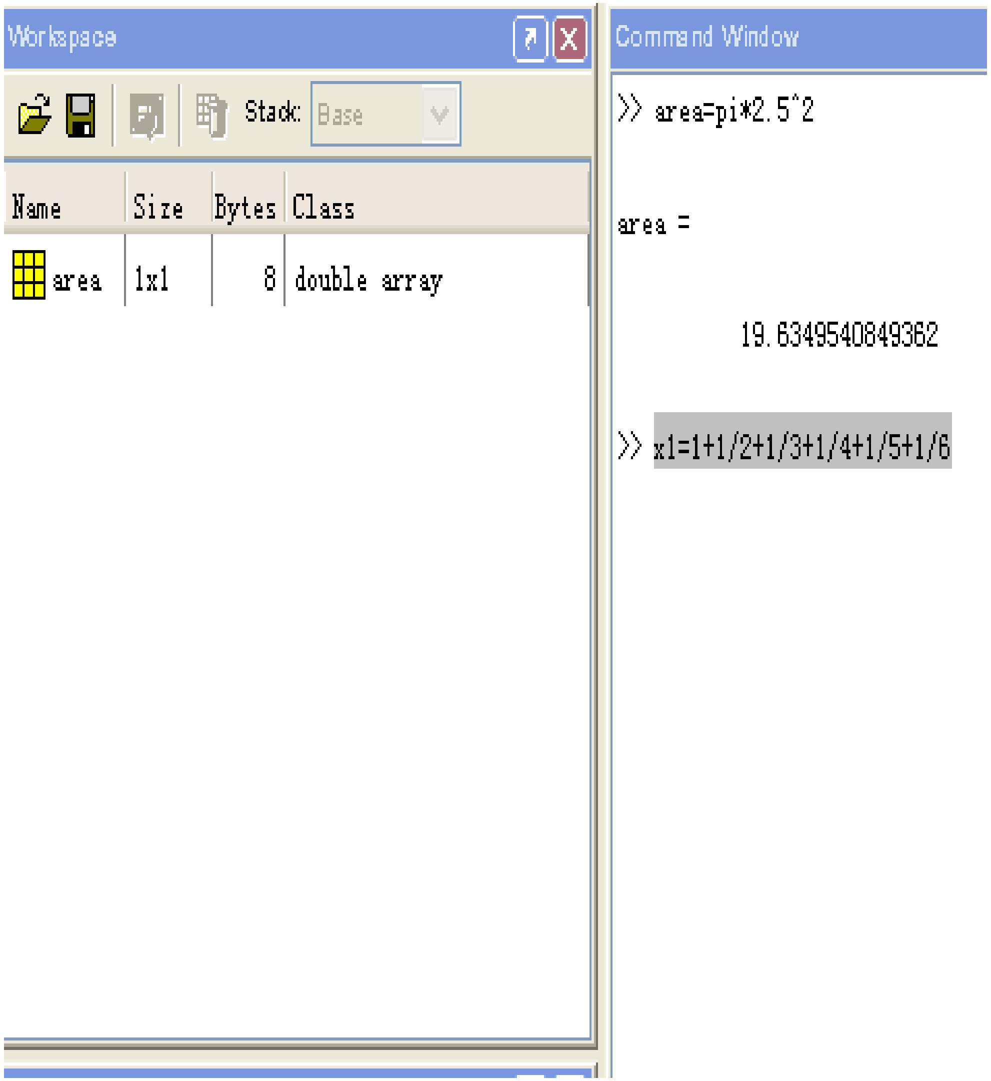
COM: component object module

The Workspace

- Computer Memory occupied by variables and arrays used by Matlab

- *whos*

  List variables and arrays in the current workspace



Stack: Base

Name	Size	Bytes	Class
area	1x1	8	double array

```
>> area=pi*2.5^2

area =

          19.6349540849362

>> x1=1+1/2+1/3+1/4+1/5+1/6
```

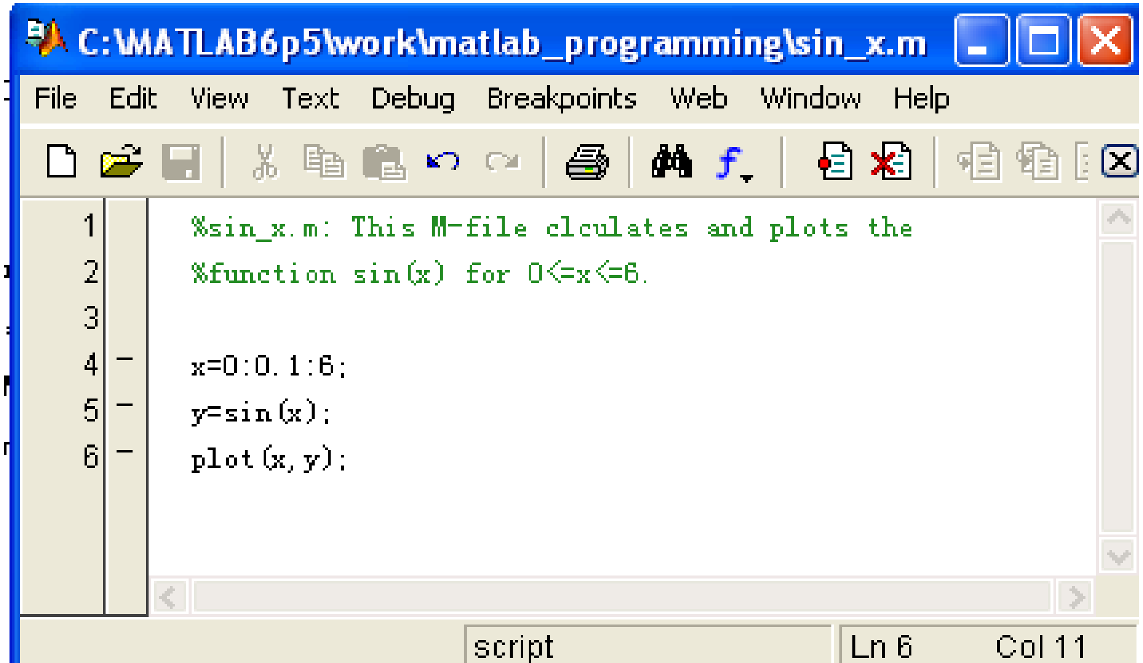
The Edit/Debug and Figure Windows

- An Edit/Debug window is used to create a new .m file or to modify an existing one.

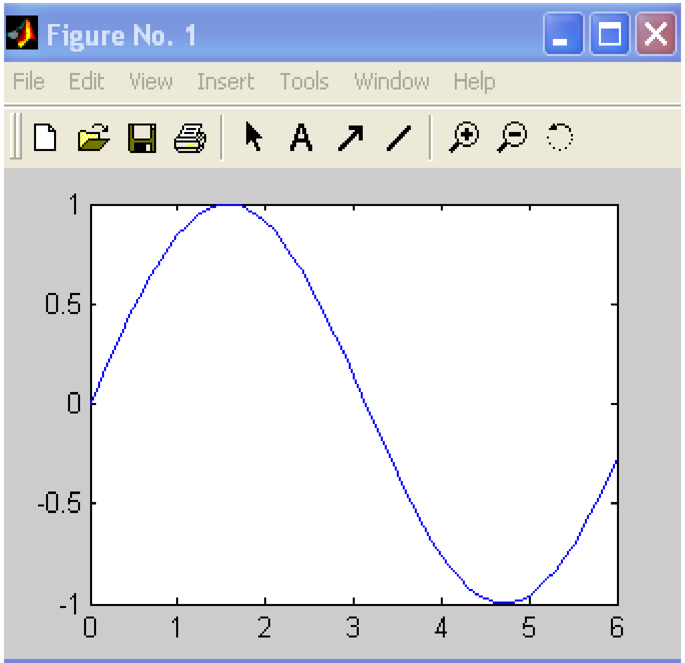
  Click 🗋 Toolbar icon to create a new one

  Click 📂 Toolbar icon to modify an existing one

- A Figure Window is used to display Matlab graphics.


C:\MATLAB6p5\work\matlab_programming\sin_x.m
File Edit View Text Debug Breakpoints Web Window Help
script
Ln 6
Col 11

```
%sin_x.m: This M-file clculates and plots the
%function sin(x) for 0<=x<=6.

x=0:0.1:6;
y=sin(x);
plot(x,y);
```



Important Commands

- *demo*: To run Matlab’s built-in demonstrations
- *clc*: To clear the contents of the Command Window
- *clf:* To clear the contents of the current figure window
- *clear:* To clear the variables in the workspace
- ^c (ctrl+c): abort command
- !: To execute commands of computer’s operating system

  !copy jzsf.m jzsftest.m
- diary: to copy all input and output displayed in the Command Window

  diary off: to suspend input into the diary file

  diary on: to resume input again

Searching and locating files

1) It looks for the name as a variable. If it is a variable, Matlab displays the current contents of the variable.

2) It checks to see if the name is a built-in function or command. If it is, Matlab executes that function or command.

3) It checks to see if the name is an M-file in the current directory. If it is, Matlab executes that function or command.

4) It checks to see if the name is an M-file in any directory in the search path. If it is, Matlab executes that function or command.

Searching and locating files (cont.)

- Remember in mind

  Never use a variable with the same name as a Matlab function or command. Otherwise, that function or command will become inaccessible.

  Never create an m-file with the same name as a Matlab function or command.

  - sin=5; sin(1)=?
  - Answer: 5 Wrong!!!

Searching and locating files (cont.)

- *which*

  To find out which version of the file, and where it is located.

```
>> which cross
C:\MATLAB6p5\toolbox\matlab\specfun\cross.m
```

Getting Help

- Get help in three ways in Matlab
  - Click ? Toolbar icon to start the Help Browser
  - Type *help specificfunction* in Command Window

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

  - Type *lookfor specificfunction* in Command Window

Getting Help (cont.)

- Help Browser: Allow full access to the entire Matlab documentation set
- *help functionname*: You must know the name of the function to get help about it
- *lookfor functionname*: Search for a given string in the first comment line of every Matlab function, and display all matches

```
>> help a
     a.m not found.
```

```
>> lookfor a
ADDPATH Add directory to search path.
BINPATCH Patch binary file.
CD      Change current working directory.
CLEAR  Clear variables and functions from memory.
DATATIPINFO Produce a short description of a variable.
DBCLEAR Remove breakpoint.
DBDOWN Change local workspace context.
DBSTACK Display function call stack.
DBSTATUS List all the breakpoints.
……..
```

ZYN related research papers

[B13] 张雨浓, 杨逸文, 李巍, 神经网络权值直接确定法, 中山大学出版社, 2010年11月

[B12] 张雨浓, 蔡炳煌（编）, 人工神经网络研究进展及论文发表过程论辩, 电子工业出版社, 2010年6月

[B7] 邹阿金, 张雨浓, 基函数神经网络及应用, 中山大学出版社, 2009年4月

[B1] 张雨浓, 人工神经网络的面向对象软件实现, 硕士毕业论文, 华南理工大学, 1999

[J65] 张雨浓, 杨逸文, 陈轲, 蔡炳煌, 梯度神经网络求解Sylvester方程之MATLAB仿真, 系统仿真学报, 2009年7月, 第21卷第13期, 4028-4031/4037

[J61] Yunong Zhang, Chenfu Yi, and Weimu Ma, Simulation and verification of Zhang neural network for online time-varying matrix inversion, Simulation Modelling Practice and Theory 17 (2009) 1603-1617

[J54] Yunong Zhang, Weimu Ma, Xiao-Dong Li, Hong-Zhou Tan, and Ke Chen, MATLAB Simulink modeling and simulation of LVI-based primal–dual neural network for solving linear and quadratic programs, Neurocomputing 72 (2009) 1679-1687

[J38] 张雨浓, 张禹珩, 陈轲, 蔡炳煌, 马伟木, 线性矩阵方程的梯度法神经网络求解及其仿真验证, 中山大学学报(自然科学版), 2008年5月, 第47卷第3期, 26-32

ZYN related research papers

[C67] Yunong Zhang, Xuezhong Li, Zhan Li, Modeling and Verification of Zhang Neural Networks for Online Solution of Time-Varying Quadratic Minimization and Programming, ISICA 2009, LNCS 5821, pp. 101–110, 2009

[C62] Ning Tan, Ke Chen, Yanyan Shi, Yunong Zhang, Modeling, Verification and Comparison of Zhang Neural Net and Gradient Neural Net for Online Solution of Time-Varying Linear Matrix Equation, ICIEA 2009, pp. 3698-3703

[C57] Yunong Zhang, Shuai Yue, Ke Chen, Chenfu Yi, MATLAB Simulation and Comparison of Zhang Neural Network and Gradient Neural Network for Time-Varying Lyapunov Equation Solving, ISNN 2008, Part I, LNCS 5263, pp. 117–127, 2008

[C53] Weimu Ma, Yunong Zhang, Jiahai Wang, MATLAB Simulink Modeling and Simulation of Zhang Neural Networks for Online Time-Varying Sylvester Equation Solving, 2008 International Joint Conference on Neural Networks (IJCNN 2008)

[C51] Yunong Zhang, Ning Tan, Binghuang Cai, Zenghai Chen, MATLAB Simulink Modeling of Zhang Neural Network Solving for Time-Varying Pseudoinverse in Comparison with Gradient Neural Network, the Second International Symposium on Intelligent Information Technology Application, 2008

ZYN related research papers

[C50] Yunong Zhang, Yiwen Yang, Simulation and Comparison of Zhang Neural Network and Gradient Neural Network Solving for Time-Varying Matrix Square Roots, the Second International Symposium on Intelligent Information Technology Application, 2008

[C47] Yunong Zhang, Xiaojiao Guo, Weimu Ma, Ke Chen, Binghuang Cai, MATLAB Simulink Modeling and Simulation of Zhang Neural Network for Online Time-Varying Matrix Inversion, ICNSC 2008, pp. 1480-1485

[C46] Yunong Zhang, Ke Chen, Xuezhong Li, Chengfu Yi, Hong Zhu, Simulink Modeling and Comparison of Zhang Neural Networks and Gradient Neural Networks for Time-Varying Lyapunov Equation Solving, the Fourth International Conference on Natural Computation, 2008

[C44] Yu-Nong Zhang, Xiao-Jiao Guo, Wei-Mu Ma, Modeling and Simulation of Zhang Neural Network for Online Linear Time-Varying Equations Solving Based on Matlab Simulink, Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming, 12-15 July 2008

[C39] Ke Chen, Shuai Yue, Yunong Zhang, MATLAB Simulation and Comparison of Zhang Neural Network and Gradient Neural Network for Online Solution of Linear Time-Varying Matrix Equation AXB-C=0, ICIC 2008, LNAI 5227, pp. 68–75, 2008

ZYN related research papers

[C24] Yunong Zhang, Weimu Ma, Ke Chen, Peng Li, Matlab Simulation of Zhang Neural Networks for Time-Varying Sylvester Equation Solving, Proceedings of the International Conference on Information Computing and Automation, 20-22 December 2007

[C23] Yunong Zhang, Ke Chen, Weimu Ma, Xiao-Dong Li, MATLAB Simulation of Gradient-Based Neural Network for Online Matrix Inversion, ICIC 2007, LNAI 4682, pp. 98-109, 2007

[C20] Yunong Zhang, Ke Chen, Weimu Ma, MATLAB Simulation and Comparison of Zhang Neural Network and Gradient Neural Network for Online Solution of Linear Time-Varying Equations, DCDIS Proceedings of 2007 International Conference on Life System Modeling and Simulation (LSMS2007), pp. 450-454

Sincere Thanks!

- Using this group of PPTs, please read

- [1] Yunong Zhang, Weimu Ma, Xiao-Dong Li, Hong-Zhou Tan, Ke Chen, MATLAB Simulink modeling and simulation of LVI-based primal-dual neural network for solving linear and quadratic programs, Neurocomputing 72 (2009) 1679-1687

- [2] Yunong Zhang, Chenfu Yi, Weimu Ma, Simulation and verification of Zhang neural network for online time-varying matrix inversion, Simulation Modelling Practice and Theory 17 (2009) 1603-1617