Matlab Basics

CS 111

Introduction to Computing in Engineering and Science

Pinar Duygulu

Bilkent University, Spring 2007

Slides are taken from Selim Aksoy and Ozlem Ozgu

MATLAB: MATrix LABoratory

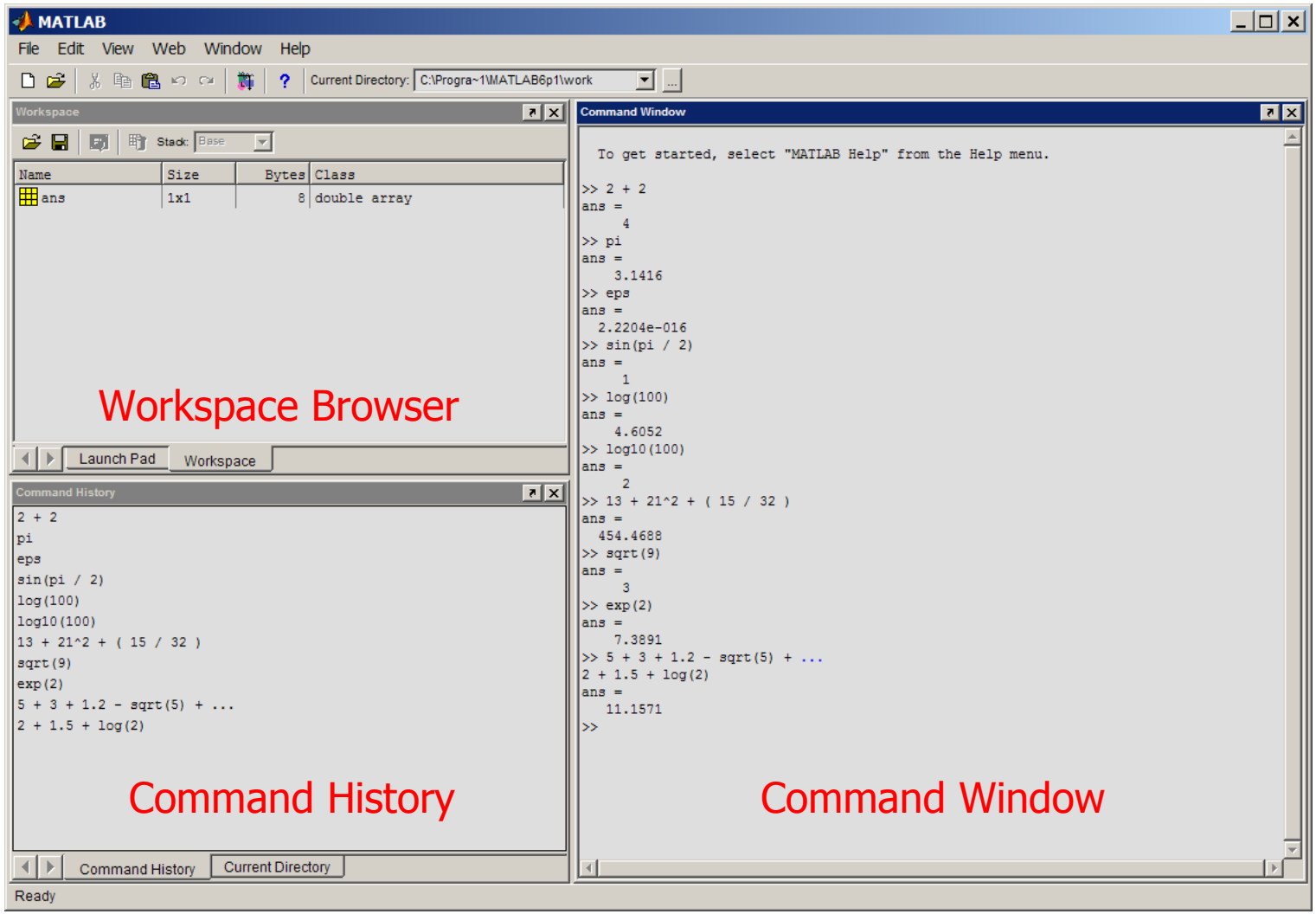
- www.mathworks.com

- References:
- Mathworks' Getting Started with Matlab
  - http://www.mathworks.com/access/helpdesk/help/techdoc/learn_matlab/
- Matlab Primer by Kermit Sigmon
  - http://ise.stanford.edu/Matlab/matlab-primer.pdf

MATLAB

- Advantages of MATLAB
  - Ease of use
  - Platform independence
  - Predefined functions
  - Plotting
- Disadvantages of MATLAB
  - Can be slow
  - Expensive

MATLAB Desktop



MATLAB Basics

- A program can be input
  - command by command using the command line (lines starting with "»" on the MATLAB desktop)
  - as a series of commands using a file (a special file called M-file)
- If a command is followed by a semicolon (;), result of the computation is not shown on the command window

MATLAB Basics: Getting Help

- help
  - help *toolbox* → e.g., help elfun
  - help *command* → e.g., help sin
- helpdesk, helpwin, "?" button
- lookfor
  - lookfor *keyword* → e.g., lookfor cotangent
- which
  - which *name* → e.g., which log
- demo

MATLAB Basics: Variables

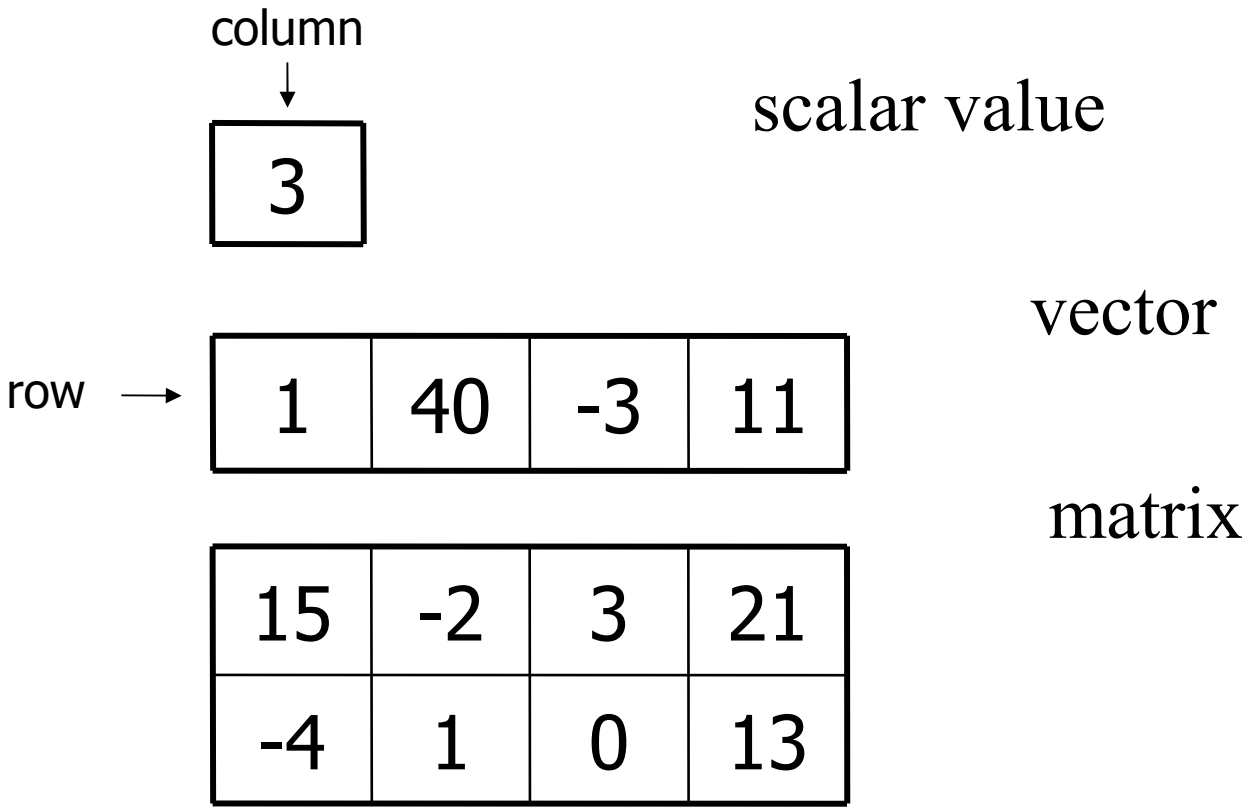
- Variable is a name given to a reserved location in memory
  - class_code = 111;
  - number_of_students = 65;
  - name = 'Bilkent University';
  - radius = 5;
  - area = pi * radius^2;

MATLAB Basics: Variables

- Use meaningful names for variables
- MATLAB variable names
  - must begin with a letter
  - can contain any combination of letters, numbers and underscore (_)
  - must be unique in the first 31 characters
- MATLAB is case sensitive: “name”, “Name” and “NAME” are considered different variables
- Never use a variable with the same name as a MATLAB command
- Naming convention: use lowercase letters

MATLAB Basics: Arrays

- The fundamental unit of data is array



MATLAB Basics: Arrays

Initializing Variables in Assignment Statements

An assignment statement has the general form

var = expression

Examples:

```
>> var = 40 * i;
>> var2 = var / 5;
>> array = [1 2 3 4];
>> x = 1;  y = 2;
>> a = [3.4];
>> b = [1.0 2.0 3.0 4.0];
>> c = [1.0; 2.0; 3.0];
>> d = [1, 2, 3; 4, 5, 6];
>> e = [1, 2, 3
        4, 5, 6];
```

```
>> a2 = [0 1+8];
   >> b2 = [a2(2) 7 a];
>> c2(2,3) = 5;
   >> d2 = [1 2];
>> d2(4) = 4;
```

‘;’ semicolon suppresses the automatic echoing of values but it slows down the execution.

MATLAB Basics: Variables

- x = 5

```
x =
     5
```

- y = x + 1

```
y =
     6
```

- vector = [1 2 3 4]

```
vector =
     1     2     3     4
```

MATLAB Basics: Variables

- matrix = [1 2 3; 4 5 6]

```
matrix =
     1     2     3
     4     5     6
```

- matrix = [1 2 3; 4 5]

??? Error

- a = [5 (2+4)]

```
a =
     5     6
```

MATLAB Basics: Variables

- Initialization using shortcut statements
  - colon operator → first:increment:last
    - x = 1:2:10

      x =

      1 3 5 7 9
    - y = 0:0.1:0.5

      y =

      0 0.1 0.2 0.3 0.4 0.5

MATLAB Basics: Variables

- transpose operator → '
  - u = [1:3]'

    u =

    1
    2
    3
  - v = [u u]

    v =

    1 1
    2 2
    3 3

- v = [u'; u']

  v =

  1 2 3
  1 2 3

MATLAB Basics: Variables

- Initialization using built-in functions
  - zeros()
    - x = zeros(2)

      x =

      0 0

      0 0
    - z = zeros(2,3)

      z =

      0 0 0

      0 0 0
  - ones(), size(), length()

- y = zeros(1,4)

  y =

  0 0 0 0

- t = zeros(size(z))

  t =

  0 0 0

  0 0 0

MATLAB Basics: Variables

Initializing with Built-in Functions

- zeros(n)
- zeros(n,m)
- zeros(size(arr))
- ones(n)
- ones(n,m)
- ones(size(arr))
- eye(n)
- eye(n,m)

- length(arr)
- size(arr)

MATLAB Basics: Variables

- Initialization using keyboard input
- The **input** function displays a prompt string in the Command Window and then waits for the user to respond.
  - input()
    - value = input('Enter an input value: ')

      Enter an input value: 1.25

      value =
          1.2500
    - name = input('What is your name: ', 's')

      What is your name: Pinar

      name =
      Pinar

MATLAB Basics: Subarrays

Multidimensional Arrays

- A two dimensional array with m rows and n columns will occupy mxn successive locations in the computer's memory. MATLAB always allocates array elements in **column major order.**

  a= [1 2 3; 4 5 6; 7 8 9; 10 11 12];
  a(5) = a(1,2) = 2

- A 2x3x2 array of three dimensions

  c(:, :, 1) = [1 2 3; 4 5 6];
  c(:, :, 2) = [7 8 9; 10 11 12];

1	2	3
4	5	6
7	8	9
10	11	12

1
4
7
10
2
5
8
11

MATLAB Basics: Subarrays

Subarrays

- It is possible to select and use subsets of MATLAB arrays.

  arr1 = [1.1 -2.2 3.3 -4.4 5.5];

  arr1(3) is 3.3

  arr1([1 4]) is the array [1.1 -4.4]

  arr1(1 : 2 : 5) is the array [1.1 3.3 5.5]

- For two-dimensional arrays, a colon can be used in a subscript to select all of the values of that subscript.

  arr2 = [1 2 3; -2 -3 -4; 3 4 5];

  arr2(1, :)

  arr2(:, 1:2:3)

MATLAB Basics: Subarrays

Subarrays

- The **end** function: When used in an array subscript, it returns the highest value taken on by that subscript.

  arr3 = [1 2 3 4 5 6 7 8];

  arr3(5:end) is the array [5 6 7 8]

  arr4 = [1 2 3 4; 5 6 7 8; 9 10 11 12];

  arr4(2:end, 2:end)

- Using subarrays on the left hand-side of an assignment statement:

  arr4(1:2, [1 4]) = [20 21; 22 23];

  (1,1) (1,4) (2,1) and (2,4) are updated.

  arr4 = [20 21; 22 23]; all of the array is changed.

MATLAB Basics: Subarrays

Subarrays

- Assigning a Scalar to a Subarray: A scalar value on the right-hand side of an assignment statement is copied into every element specified on the left-hand side.

```
>> arr4 = [1 2 3 4; 5 6 7 8; 9 10 11 12];
>> arr4(1:2, 1:2) = 1
arr4 =
     1     1     3     4
     1     1     7     8
     9    10    11    12
```

MATLAB Basics: Subarrays

- Array indices start from 1
- x = [-2 0 9 1 4];
  - x(2)

    ans =

    0
  - x(4)

    ans =

    1

- x(8)

  ??? Error

- x(-1)

  ??? Error

MATLAB Basics: Subarrays

- y = [1 2 3; 4 5 6];
  - y(1,2)

    ans =

    2
  - y(2,1)

    ans =

    4
  - y(2)

    ans =

    4 (column major order)

MATLAB Basics: Subarrays

- y = [1 2 3; 4 5 6];
  - y(1,:)

    ans =

    1 2 3
  - y(:,2)

    ans =

    2

    5
  - y(2,1:2)

    ans =

    4 5

- y(1,2:end)

  ans =

  2 3
- y(:,2:end)

  ans =

  2 3

  5 6

MATLAB Basics: Subarrays

- x = [-2 0 9 1 4];
  - x(2) = 5

    x =

    -2 5 9 1 4
  - x(4) = x(1)

    x =

    -2 5 9 -2 4
  - x(8) = -1

    x =

    -2 5 9 -2 4 0 0 -1

MATLAB Basics: Subarrays

- y = [1 2 3; 4 5 6];
  - y(1,2) = -5

    y =

    ```
        1    -5     3
        4     5     6
    ```
  - y(2,1) = 0

    y =

    ```
        1    -5     3
        0     5     6
    ```
  - y(1,2:end) = [-1 9]

    y =

    ```
        1    -1     9
        0     5     6
    ```

MATLAB Basics: Subarrays

- y = [1 2 3; 4 5 6; 7 8 9];
  - y(2:end,2:end) = 0

    y =

    ```
        1     2     3
        4     0     0
        7     0     0
    ```

  - y(2:end,2:end) = [-1 5]

    ??? Error

  - y(2,[1 3]) = -2

    y =

    ```
        1     2     3
       -2     0    -2
        7     0     0
    ```

MATLAB Basics: Special values

Special Values

- MATLAB includes a number of predefined special values. These values can be used at any time without initializing them.
- These predefined values are stored in ordinary variables. They can be overwritten or modified by a user.
- If a new value is assigned to one of these variables, then that new value will replace the default one in all later calculations.

```
>> circ1 = 2 * pi * 10;
>> pi = 3;
>> circ2 = 2 * pi * 10;
```

Never change the values of predefined variables.

MATLAB Basics: Special Values

- pi: π value up to 15 significant digits
- i, j: sqrt(-1)
- Inf: infinity (such as division by 0)
- NaN: Not-a-Number (such as division of zero by zero)
- clock: current date and time as a vector
- date: current date as a string (e.g. 16-Feb-2004)
- eps: epsilon
- ans: default variable for answers

MATLAB Basics: Displaying Data

- Changing the data format
  - value = 12.345678901234567

format short → 12.3457

format long → 12.34567890123457

format short e → 1.2346e+001

format long e → 1.234567890123457e+001

format short g → 12.346

format long g → 12.3456789012346

format rat → 1000/81

MATLAB Basics: Displaying Data

- The disp(*array*) function
  - disp('Hello');
    Hello
  - disp(5);
    5
  - disp(['Bilkent ' 'University']);
    Bilkent University
  - name = 'Pinar'; disp(['Hello ' name]);
    Hello Pinar

MATLAB Basics: Displaying Data

- The num2str() and int2str() functions
  - d = [num2str(16) '-Feb-' num2str(2004)];
  - disp(d);

    16-Feb-2004
  - x = 23.11;
  - disp(['answer = ' num2str(x)]);

    answer = 23.11
  - disp(['answer = ' int2str(x)]);

    answer = 23

MATLAB Basics: Displaying Data

- The fprintf(*format*, *data*) function
  - %d integer
  - %f floating point format
  - %e exponential format
  - \n new line character
  - \t tab character

MATLAB Basics: Displaying Data

- fprintf('Result is %d', 3);
  Result is 3
- fprintf('Area of a circle with radius %d is %f', 3, pi*3^2);
  Area of a circle with radius 3 is 28.274334
- x = 5;
- fprintf('x = %3d', x);
  x = 5
- x = pi;
- fprintf('x = %0.2f', x);
  x = 3.14
- fprintf('x = %6.2f', x);
  x = 3.14
- fprintf('x = %d\ny = %d\n', 3, 13);
  x = 3
  y = 13

MATLAB Basics: Data Files

- save *filename var1 var2 …*
  - save homework.mat x y → binary
  - save x.dat x –ascii → ascii
- load *filename*
  - load filename.mat → binary
  - load x.dat –ascii → ascii

MATLAB Basics: Scalar Operations

- *variable_name* = *expression*;
  - addition a + b → a + b
  - subtraction a - b → a - b
  - multiplication a x b → a * b
  - division a / b → a / b
  - exponent a^b → a ^ b

MATLAB Basics: Scalar Operations

- x = 3 * 2 + 6 / 2
  - x = ?
- Processing order of operations is important
  - parenthesis (starting from the innermost)
  - exponentials (left to right)
  - multiplications and divisions (left to right)
  - additions and subtractions (left to right)
- x = 3 * 2 + 6 / 2
  - x = 9

MATLAB Basics: Built-in Functions

- *result = function_name(input)*;
  - abs, sign
  - log, log10, log2
  - exp
  - sqrt
  - sin, cos, tan
  - asin, acos, atan
  - max, min
  - round, floor, ceil, fix
  - mod, rem
- help elfun

MATLAB Basics: Debugging

- Syntax errors
  - Check spelling and punctuation
- Run-time errors
  - Check input data
  - Can remove “;” or add “disp” statements
- Logical errors
  - Use shorter statements
  - Check typos
  - Check units
  - Ask your friends, TAs, instructor, parents, …

MATLAB Basics: Useful Commands

- help *command* → Online help
- lookfor *keyword* → Lists related commands
- which → Version and location info
- clear → Clears the workspace
- clc → Clears the command window
- diary *filename* → Sends output to file
- diary on/off → Turns diary on/off
- who, whos → Lists content of the workspace
- more on/off → Enables/disables paged output
- Ctrl+c → Aborts operation
- … → Continuation
- % → Comments