Introduction to MATLAB

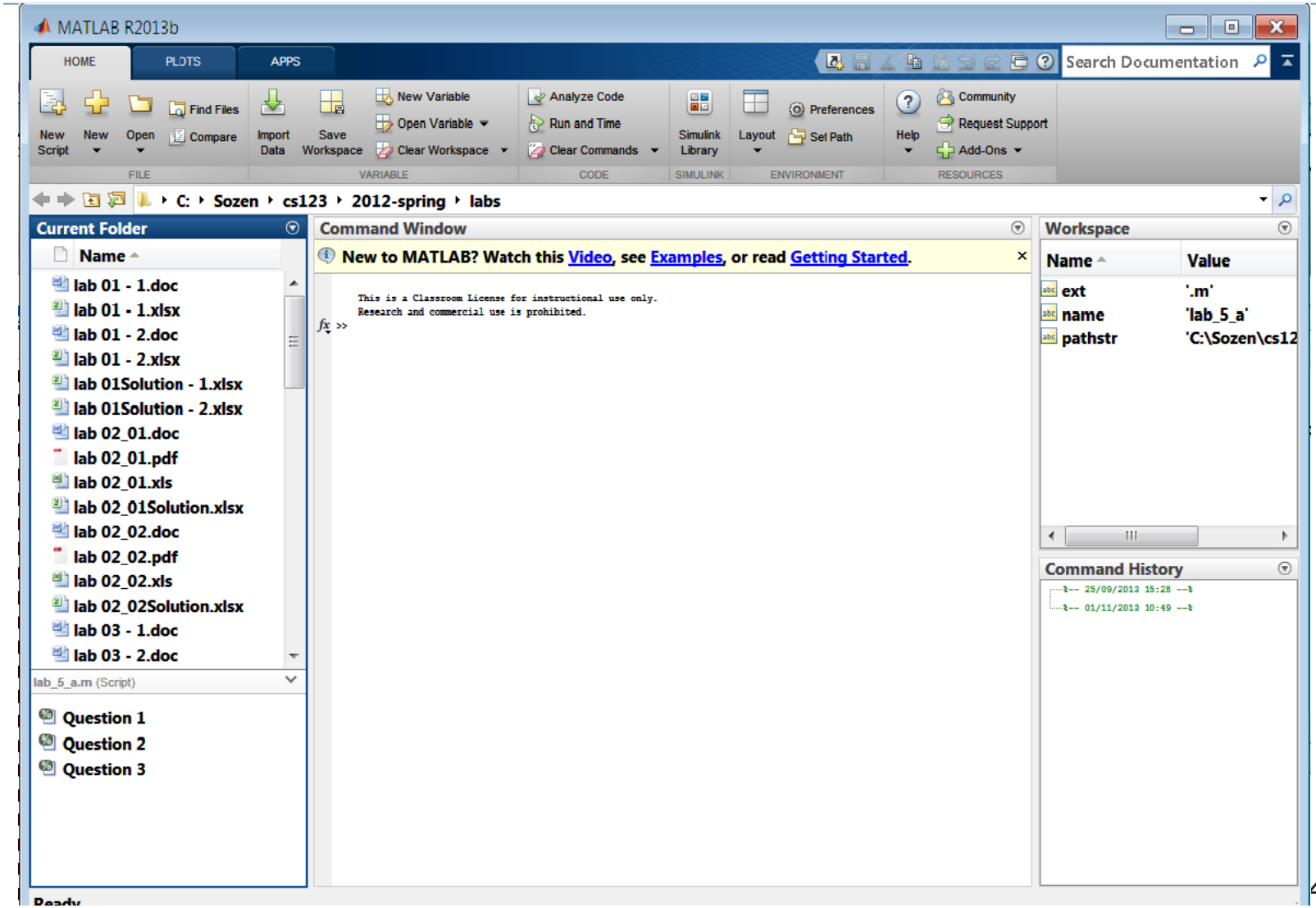
MATrix LABoratory

- First introduced at Stanford University in 1979
- Initially an interactive shell to FORTRAN routines
- MathWorks was formed to market Matlab
- Powerful engineering environment and language
- Useful for problem solving, data analysis, modeling and visualization
- Runs on nearly every operating system

MATrix LABoratory

- http://www.mathworks.com
- 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

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. 17-Nov-2008)
- eps: epsilon
- ans: default variable for answers

MATLAB Basics: Variables

Let's evaluate the following expression in matlab :

$$\cot(3)\sqrt{(\log(3))^3} + \cos(3) * \sin(\log(3))$$

Now let's do the following:

$$\cot(2.7)\sqrt{(\log(2.7))^3} + \cos(2.7) * \sin(\log(2.7))$$

What if we need to evaluate the same expression for 2.1, 2.2, 2.3, … and lots of other values?

MATLAB Basics: Variables

- Given: $\cot(3)\sqrt{(\log(3))^3} + \cos(3) * \sin(\log(3))$

Rather than :

```
>>cot(3)*sqrt(log(3)^3) + cos(3)*sin(log(3)),
>>cot(2.7)*sqrt(log(2.7)^3) + cos(2.7)*sin(log(2.7)), …
```

Use a variable:

```
>>x=3
>>cot(x)*sqrt(log(x)^3) + cos(x)*sin(log(x))
>>x=2.7
>>cot(x)*sqrt(log(x)^3) + cos(x)*sin(log(x))
```

MATLAB Basics: Variables

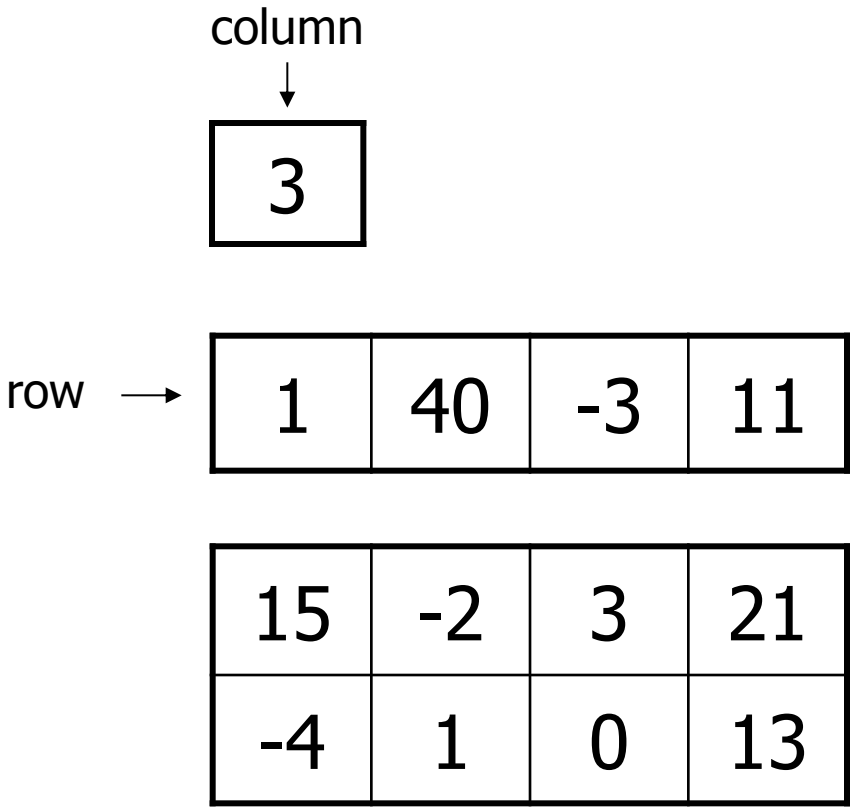
- Variable is a name given to a reserved location in memory
  - class_code = 111;
  - number_of_students = 65;
  - name = 'Bilkent University';
  - radius = 5;
  - area = pi * radius^2;

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

- The fundamental unit of data is array



MATLAB Basics: Arrays

- Initialization using assignment statements
- Assignment is always in form of variable equals expression or a constant
  - x = 5

    x =

    5
  - y = x + 1

    y =

    6
  - v = [1 2 3 4]

    v =

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

    m =

    1 2 3

    4 5 6
  - m2 = [1 2 3; 4 5]

    ??? Error
  - a = [5 (2+4)]

    a =

    5 6

MATLAB Basics: Arrays

- 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: Arrays

- 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
    - y = zeros(1,4)

      y =

      0 0 0 0
    - t = zeros(size(z))

      t =

      0 0 0

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

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(2,3)

    ans =

    6

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,:) = [4 -1 9]

    y =

4	-1	9
0	5	6

  - y(:,2) = [3; 2]

    y =

4	3	9
0	2	6

MATLAB Basics: Subarrays

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

    z =

1	2	3
4	5	6
0	0	0

  - z(:,1) = -2

    z =

-2	2	3
-2	5	6
-2	0	0

  - z(2,:) = [1 5]

    ??? Error

Subarray examples

- A(1:4,3) is the column vector consisting of the first four entries of the third column of A
- A(:,3) is the third column of A
- A(:,[2 4]) contains as columns, columns 2 and 4 of A
- Such subscripting can be used on both sides of an assignment statement
- A(:, [2 4 5])= B(:,1:3) replaces columns 2, 4 and 5 of A with the first three columns of B.

MATLAB Basics: Addressing Arrays by using a Colon

- Va(:) refers to all the elements of the vector Va
- Va(m:n) refers to elements m though n of the vector Va
- A(:,n) refers to the elements in all the rows of column n of the matrix A
- A(n,:) refers to the elements in all the columns of row n of the matrix A
- A(:,m:n) refers to the elements in all the rows between columns m through n of the matrix A
- A(m:n,:) refers to the elements in all the columns between rows m through n of the matrix A
- A(m:n,p:q) refers to the elements in rows m through n and columns m through n of the matrix A

Built-in Functions for Arrays

Function	Description	Example
mean(A)	Returns mean value of vector A	>>A=[5 9 4 2]; >>mean(A) Ans= 5
C=max(A) [d,n]=max(A)	C is largest element of vector A d is largest element of vector A, n is position of element	>>A=[5 9 2 4 11 6 7 11]; >>C=max(A) C= 11 >>[d,n]=max(A) d=11 n=5
C=min(A) [d,n]=min(A)	C is smallest element of vector A The same as [d,n]= max(A),butfor smallest element	>>A=[5 9 4 2]; >>min(A) Ans= 2
sum(A)	Returns sum of elements of vector A	>>A=[5 9 4 2]; >>sum(A) Ans= 20

Built-in Functions for Arrays (cont.)

Function	Description	Example
sort(A)	Arranges elements of vector in ascending order	>>A=[5 9 4 2]; >>sort(A) Ans= 2 4 5 9
det(A)	Returns determinant of square matrix A	>>A=[2 4 ; 3 5]; >>det(A) Ans= -2
inv(A)	Returns inverse of a square matrix	>>A=[2 –2 1; 3 2 –1; 2 –3 2] >>inv(A) Ans= 0.2000 0.2000 0 -1.6000 0.4000 1.000 -2.6000 0.4000 2.000
dot(a,b)	Calculates scalar product of two vectors a and b	>>a=[1 2 3]; >>b=[2 4 1]; >>cross(a,b) Ans= -5 3 2

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

Rounding functions

Function	Description	Example
round(x)	Round to nearest integer	>>round(17/5) Ans= 3
fix(x)	Round towards zero	>>fix(13/5) Ans= 2
ceil(x)	Round towards infinity	>>ceil(11/5) Ans= 3
floor(x)	Round towards minus infinity	>>floor(-9/4) Ans= -3
rem(x,y)	Returns remainder after x is divided by y	>>rem(13,5) Ans= 3
sign(x)	Signum function 1 if x>0, –1if x<0,0 if x=0	>>sign(5) Ans= 1

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