

Matlab Tutorial

Based on Matlab tutorial by
IPLab@SUT

Motivation: powerful *and* easy!

- Matlab: a high-performance software and a programming language
- Powerful
 - Mathematics and visualization
 - Toolboxes in statistics, signal processing, vision...
 - Widely used in AI-related fields
- Easy
 - Prototyping, testing, debugging, finding helps...

Motivation: the 80/20 rule

- "Remember the 80/20 rule. Hence 80% of your code should be in some high level language like Matlab... Your time is more valuable than the computer's time."
 - Prof. Kevin P. Murphy

Outline

- Environment
- Matrices
- Elementary math
- Data types
 - String
 - Cell arrays
 - Multidimensional arrays
- Programming part next week!
 - Plot, flow control, functions, debug, ...

Calculations at the Command Line

MATLAB as a calculator

```

> -5/(8-9+5.32)^2
ans =
   -0.0488
> (2+4i)^(3-4i)
ans =
    25
> cos(pi/2)
ans =
   6.1230e-317
> exp(logs(1.31))
ans =
    3.9470
  
```

Assigning Variables

```

> a = 2;
> b = 5;
> a^b
ans =
    32
> x = 5/2*pi;
> y = atan(x)
y =
    1
> z = atan(y)
z =
    1.1070
  
```

Semicolon suppresses screen output

Results assigned to "ans" if name not specified

() parentheses for function inputs

A Note about Workspace:
Numbers stored in double-precision floating point format

9

General Functions

- **whos**: List current variables
- **clear**: Clear variables and functions from memory
- **cd**: Change current working directory
- **ls**: List files in directory

10

Getting help

- help* command (`>>help`)
- lookfor* command (`>>lookfor`)
- Printable Documents
 - "Matlabroot\help\pdf_doc1"

Matrices

- Entering and Generating Matrices
- Subscripts
- Scalar Expansion
- Concatenation
- Deleting Rows and Columns
- Array Extraction
- Matrix and Array Multiplication

Entering Numeric Arrays

Row separator:
semicolon (;)

Column separator:
space / comma (,)

```

>> a=[1 2;3 4]
a =
     1     2
     3     4
>> b=[-2.5, sqrt(-7), (3+5i)^3/4]
b =
 -2.5000    0 + 2.6458i    13.5000
>> k(2:5) = 23
k =
 -2.5000    0 + 2.6458i    10.5000    0
         0
>>
    
```

Use square brackets []

- Any MATLAB expression can be entered as a matrix element
- Matrices must be rectangular. (Set undefined elements to zero)

The Matrix in MATLAB

		Columns (c)				
		1	2	3	4	5
Rows (m)	1	4	10	1	6	2
	2	8	1.2	9	4	25
	3	7.2	5	7	1	11
	4	0	0.5	4	5	56
	5	23	83	13	0	10

A(2,4) points to the value 4 in row 2, column 4.

A(17) points to the value 4 in row 4, column 4.

Rectangular Matrix:
 Scalar: 1-by-1 array
 Vector: m-by-1 array
 1-by-n array
 Matrix: m-by-n array

Entering Numeric Arrays

Scalar expansion

Creating sequences:
colon operator (:)

Utility functions for creating matrices.

```

>> w=[1 2;3 4] + 5
w =
     6     7
     8     9
>> x = 1:5
x =
     1     2     3     4     5
>> y = 2:-0.5:0
y =
     2.0000    1.5000    1.0000    0.5000    0
>> z = rand(2,4)
z =
     0.9501    0.4909    0.8913    0.4505
     0.2311    0.4860    0.7421    0.0185
    
```

Numerical Array Concatenation

Use [] to combine existing arrays as matrix "elements"

Row separator:
semicolon (;)

Column separator:
space / comma (,)

```

>> a=[1 2;3 4]
a =
     1     2
     3     4
>> cat_m(a, 2*a, 3*a, 4*a, 5*a, 6*a)
cat_m =
     1     2     2     4
     3     4     6     8
     9    12    12    16
    15    20    18    24
    
```

Use square brackets []

*4*a* points to the value 12 in row 2, column 4.

Note:
The resulting matrix must be rectangular

Deleting Rows and Columns

```

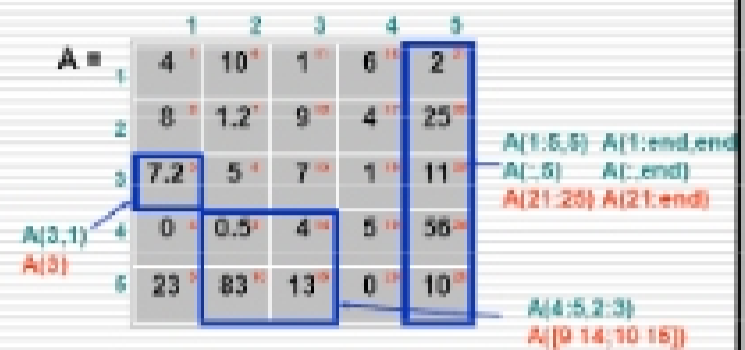
> A=[1 5 9;4 3 2.5; 8.1 10 34+2]
A =
    1.0000    5.0000    9.0000
    4.0000    3.0000    2.5000
    0.1000   10.0000   1.0000+3.0000i

> A(1,2)=[]
A =
    1.0000    9.0000
    4.0000    2.5000
    0.1000    1.0000 + 3.0000i

> A(2,2)=[]
??? Indexing empty matrix assignment is not allowed.
    
```

17

Array Subscripting / Indexing



18

Matrix Multiplication

```

> a = [1 2 3 4; 5 6 7 8];           [2x4]
> b = ones(4,3);                   [4x3]
> c = a*b;                           [2x4][4x3] -> [2x3]
c =
    10    10    10
    26    26    26
    
```

Array Multiplication

```

> a = [1 2 3 4; 5 6 7 8];
> b = [1,4; 1,4];
> c = a.*b;
c =
    1     4     9    16
    5    12    21    32
    
```

19

Matrix Manipulation Functions

- **zeros**: Create an array of all zeros
- **ones**: Create an array of all ones
- **eye**: Identity Matrix
- **rand**: Uniformly distributed random numbers
- **diag**: Diagonal matrices and diagonal of a matrix
- **size**: Return array dimensions
- **repmat**: Replicate and tile a matrix

20

Matrix Manipulation Functions

- **det**: Matrix determinant
- **inv**: Matrix inverse
- **eig**: Evaluate eigenvalues and eigenvectors
- **rank**: Rank of matrix

21

Elementary Math

- Logical Operators
- Math Functions
- Polynomial and Interpolation

25