

MatLab Primer - CS4210/MATH4250

Matt Hin

Cornell University

26 August 2016

Overview

- Introduction to MatLab
- Matrices and Arrays in MatLab
- Matrix Operations
- Built-in Functions
- Ifs, Loops, and Relations
- Scripts vs. Functions
- Printing to Console
- Graphics
- Coding Fundamentals

Introduction to MatLab

What is MatLab?

MatLab is a computational engine created by the people of MathWorks intended for computational use by scientists and engineers. This engine is available for many different operating systems and has a large documentation center as well.

Introduction to MatLab

What can I use it for?

MatLab can be used as the computational language with which to perform numeric studies of any problem. Its a strong computational engine.

Introduction to MatLab

What should I use it?

MatLab excels at computation when data is structured as an array or matrix. There are built-in routines deep in MatLab's code that are heavily optimized for fast matrix operations. MatLab is also a higher level programming language which allows for quick implementation of code with easy to interpret commands.

Introduction to MatLab

How do I access MatLab?

You can run MatLab by either opening the application from its icon on the GUI or you can run MatLab via terminal with the command `matlab`.

Introduction to MatLab

How do I get MatLab?

Cornell's IT has got you covered:

http://www.it.cornell.edu/services/software_licensing/available/matlab.cfm

Initializing Matrices

```
>> A = [ 1 , 4 , 7 ; 2 , 5 , 8 ; 3 , 6 , 9 ] ;  
>> B = 1:5;
```

The above line creates the 3-by-3 matrix:

$$A = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

and the row vector

$$B = [1 \ 2 \ 3 \ 4 \ 5].$$

Initializing Matrices

```
>> A = [ 1 , 4 , 7 ; 2 , 5 , 8 ; 3 , 6 , 9 ]  
>> B = 1:5;
```

- Square brackets begin and end a matrix.
- Semi-colons end a row in a matrix.
- Commas or whitespace demarcates an element in the matrix.
- Colons denote a range of integers.

Built-in Functions for Special Matrices

- `eye(n)` generates an identity matrix of dimension n .
- `zeros(n,m)` generates a zero matrix of n rows and m columns.
- `ones(n,m)` generates a matrix of ones with n rows and m columns.
- `rand(n,m)` generates a matrix of n rows and m columns with elements that are uniformly randomly selected between 0 and 1.
- `magic(n)` generates a matrix of dimension n that is a magic square (columns and rows sum to the same value).
- `linspace(a,b,n)` generates a vector with n elements that are equally spaced between a and b .

Accessing Matrices

Given

$$A = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}.$$

```
>> A(2,1)
```

```
ans = 2
```

```
>> A([1,3],2:3)
```

```
ans = [ 4 , 7 ; 6 , 9 ]
```

```
>> A(end-1)
```

```
ans = 8
```

Other Array Types

- Cell Arrays: Like matrices but elements can be anything.
- Structs: More traditional form of objects as seen in C++ and Python.

Common Operations

- $+$, Addition
- $-$, Subtraction
- $*$, Matrix Multiplication
- $'$, Conjugate Transpose
- $/$, Left Divide
- \backslash , Right Divide

Other Common Operations

- `.*`, Elementwise Multiplication
- `.\`, Elementwise Division
- `.'`, Transpose

Elementwise Built-in Functions

- Trigonometric Functions: `sin`, `cos`, `tan`, `asin`, `acos`, `atan`
- Exponents and Logarithms: `exp`, `log`
- Rounding: `round`, `floor`, `ceil`
- Other: `abs`, `sqrt`, `sign`

Vector Built-in Functions

- max, min, sort,
- sum, prod,
- median, mean, std,
- any, all, length

Matrix Built-in Functions

- eig, svd
- inv, det
- size, norm, cond

If Statements

```
if (x < 0)
    fprintf('x is negative \n')
elseif (x > 0)
    fprintf('x is positive \n')
else
    fprintf('x is zero \n')
end
```

For Loops

```
n = 10;  
f = 1;  
for i = 1:n  
    f = f * i  
end  
disp(['n! = ', num2str(f)])
```

While Loops

```
n = 10;
f = n;
while (n > 1)
    n = n-1;
    f = f * n;
end
disp(['n! = ', num2str(f)])
```

Relations

- $<$, less than
- $<=$, less than or equal to
- $>$, greater than
- $>=$, greater than or equal to
- $==$, equal to
- $\sim=$, not equal to
- $\&$, logical AND
- $|$, logical OR
- \sim , logical NOT

M-files

MatLab primarily saves code with the `.m` extension. There are two types of M-files: scripts and functions.

- Scripts are just compilations of MatLab expressions and code.
- Functions are user defined routines that expand the capabilities of MatLab.

Functions

Example

```
function [ m , s ] = stat(x)
n = length(x);
m = sum(x)/n;
s = sqrt( sum( (x-m).^2 / n ) );
end
```

Function Details

- Semi-colons at the end of statements suppress output to console.
- `return` can be used to terminate a function before the function end is reached.
- Special cell arrays `nargin`, `nargout`, `varargin`, and `varargout` can be used to handle inconsistent inputs and outputs.

Print Commands

Print Commands

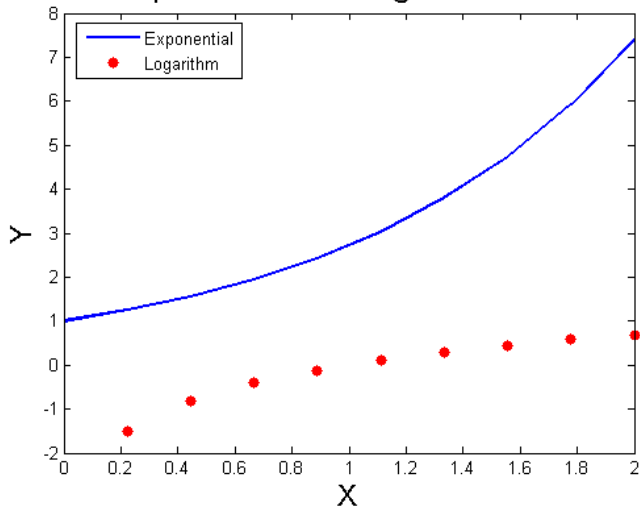
There are 2 main ways to print to console: `disp` and `fprintf`.
Learn more about them from the MatLab documentation available
at the MathWorks webpage.

Output Format

These change the format of the console output whenever something is printed to the console. Say we want to print out π :

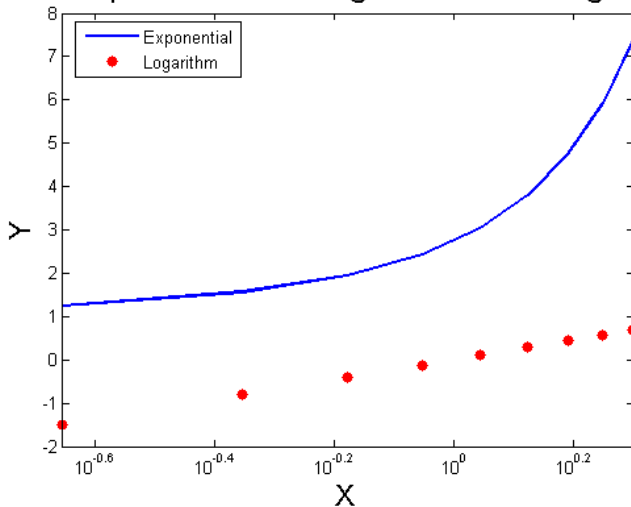
- (default) format short : 3.1416
- format long : 3.141592653589793
- format shortE : 3.1416E + 00
- format longE : 3.141592653589793E + 00

Exponential vs Logarithm Plot

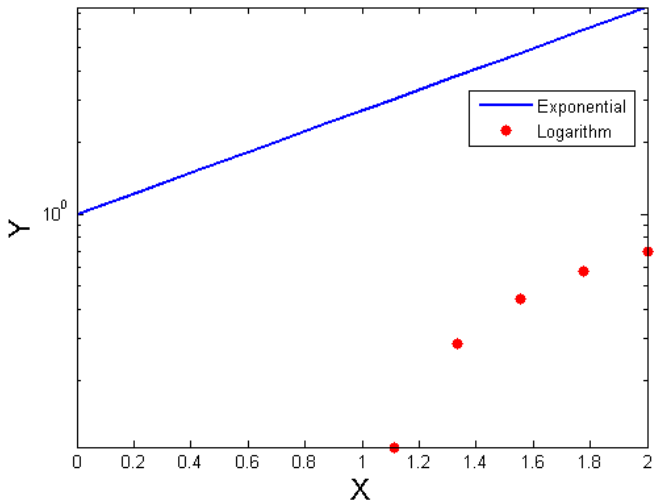


```
1 x = linspace(0,2,10);
2 y1 = exp(x);
3 y2 = log(x);
4
5 fig1 = figure;
6 plot(x,y1,'b-','LineWidth',2)
7 hold on
8 plot(x,y2,'r.','MarkerSize',20)
9 hold off
10 xlabel('X','FontSize',16)
11 ylabel('Y','FontSize',16)
12 title('Exponential vs Logarithm Plot','FontSize',20)
13 legend('Exponential','Logarithm','FontSize',14,'Location','Best')
14 box on
```

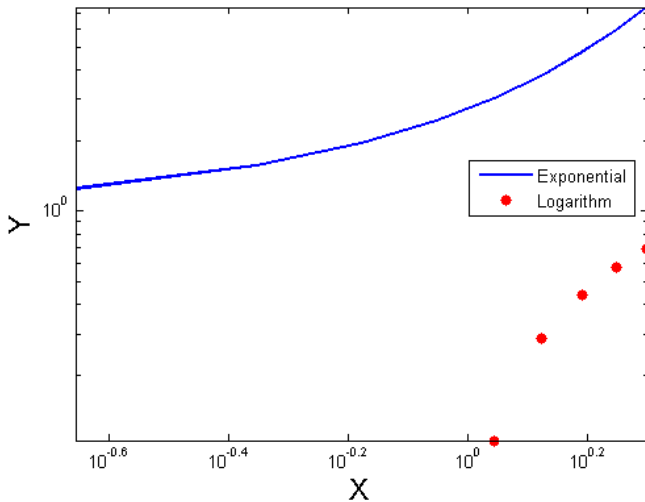
Exponential vs Logarithm Semilogx



Exponential vs Logarithm Semilogy



Exponential vs Logarithm Loglog



Coding Basics

- Never have magic numbers in your code.
- Try not to use global variables.
- Always thoroughly comment your code (%)
- Always have legible features on figures and console outputs.
- MatLab specific: For minimal runtime, vectorize code if possible.

Sources and References

- MatLab Primer:
<http://www.math.toronto.edu/mpugh/primer.pdf>
- MatLab Documentation:
<http://www.mathworks.com/help/matlab/>