ChBE 303
Introduction to Matlab
Part 1

Robert A Pinnick

Fall 2006

Outline

- Essentials
- Philosophy of Data Storage
- Mathematical Operators
  - Elementary
  - 1-Dimensional Arrays (Vectors)
  - 2-Dimensional Arrays (Matrix)
  - Structure Operations
  - Cell Operations
- Example Code
  - Linear Regression
  - Numerical Integration
  - Numerical Solvers
  - Statistics

Essentials

- Syntax & Logic Help
  - >> help <function_name>
    - help fzero
  - >> lookfor <keyword>
    - lookfor bessel
  - www.mathworks.com
    - Support (Documentation) & Forum Sections
  - www.google.com
- Useful Tips
  - >> clear (variable)
    - Clears All Memory or Specified Variable
  - >> clc
    - Clears Command Window Screen

Philosophy of Data Storage

- Numerical Values (Ex. 5, 3.14159, 2+i)
  - Integers
  - Floating (Single & Double)
  - Complex
- Character Strings (Ex. ‘asdf jkl;’)
- Structures
- Cells
- Boolean (True / False)

Elementary Operators

- Assignment (=)
  - >> X = 1;
  - >> Y = 3.14159;
- Addition (+)
  - >> Z = X + Y
    - Z = 4.14159
- Subtraction (-)
  - >> Z = X – Y
    - Z = -2.14159
- Multiplication (*)
  - >> Z = 2 * Y
    - Z = 6.28318
- Division (/)
  - >> Z = 1 / 4
    - Z = 0.25
- Power (^)
  - >> Z = 5 ^ 3
    - Z = 125

Vector Operators
– Scalar Operations –

- Assignment
  - Column
    - >> X = [1 ; 2 ; 3];
  - Row
    - >> Y = [1 , 2 , 3];
    - >> Y = [1 2 3];
  - Unique Commands
    - linspace(Initial , Final , # Points)
      - >> Z = linspace(5 , 20 , 4)
        - Z = [5 , 10 , 15 , 20];
    - logspace(Initial , Final , # Points)
    - Initial:Step:Final
      - >> Z = 1:-0.25:0
        - Z = [1 , 0.75 , 0.5 , 0.25 , 0]
- Index
  - >> Z = X(3)
    - Z = 3
  - >> Z = X(4)

    –OR–

    >> Z = X(0)
    - ERROR !!
  - >> Z = X(1:2)
    - Z = [1 ; 2]
  - >> Z = X([1 3])
    - Z = [1 ; 3]

Vector Operators
– Scalar Operations –

- Addition (+)
  - >> Z = X + 2
    - Z = [3 ; 4 ; 5]
- Subtraction (-)
- Multiplication (*)
  - >> Z = 2 * X
    - Z = [2 ; 4 ; 6]
- Division (/)
  - >> Z = X / 2
    - Z = [0.5 ; 1 ; 1.5]

- Summation

$$\sum_{i=1}^{n} x_i$$

  - >> X = [1 , 2 , 3 , 4];
    >> Z = sum(X)
    - Z = 10
- Product

$$\prod_{i=1}^{n} x_i$$

  - >> Y = [1 ; 2 ; 3 ; 4];
    >> Z = prod(Y)
    - Z = 24

Vector Operators

- Inner Product

$$\vec{x} \cdot \vec{y} = \begin{bmatrix} x_1 & \cdots & x_i & \cdots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{bmatrix} = \sum_{i=1}^{n} x_i y_i$$

  - >> X = [1 , 2 , 3];
    >> Y = [1 ; 2 ; 3];
    >> Z = X * Y
    - Z = 14

- Transpose

$$\begin{bmatrix} x_1 & \cdots & x_i & \cdots & x_n \end{bmatrix}^T = \begin{bmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix}$$

  - >> Z = X'
    - Z = [1 ; 2 ; 3]

- Euclidean Norm

$$\left\| \vec{x} \right\| = \vec{x} \cdot \vec{x} = \sum_{i=1}^{n} x_i^2$$

  - >> X = [1 , 2 , 3];
    >> Z = X * X'
    - Z = 14

- Cross Product

$$\vec{x} \times \vec{y} = \begin{bmatrix} x_2 y_3 - y_2 x_3 & x_3 y_1 - y_3 x_1 & x_1 y_2 - y_1 x_2 \end{bmatrix}$$

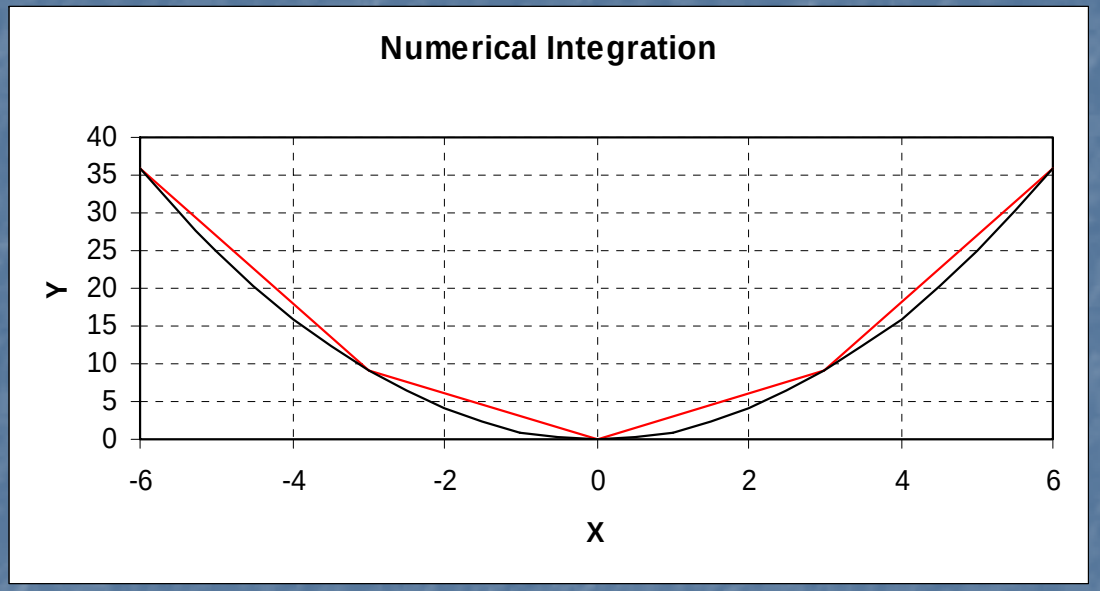
  - >> X = [1 , 2 , 3];
    >> Y = [3 , 2 , 1];
    >> Z = cross(X , Y)
    - Z =

Vector Operators
– Element Wise –

- Addition (+)
  - >> X = [1 , 2 , 3];
    >> Y = [2 , 4 , 6];
    >> Z = X + Y
    - Z = [3 , 6 , 9];
- Subtraction (-)
- Multiplication (.*)
  - >> X = [1 , 2 , 3];
    >> Y = [2 , 4 , 6];
    >> Z = X.*Y
    - Z = [2 , 8 , 18];
- Division (./)
- Power (.^)
  - >> Z = X.^2
    - Z = [1 , 4 , 9];

Numerical Integration

- Data (y = x^2)
  - (-6, 36)
  - (-3 , 9)
  - (0 , 0)
  - (3 , 9)
  - (6 , 36)
- Code
  - >> X = [-6 , -3 , 0 , 3 , 6];
    >> Y = [36 , 9 , 0 , 9 , 36];
    >> Z = sum(trapz(X , Y));
    - Z = 162
  - >> Z = QUAD(INLINE(‘x.^2‘) , -6 ... 6)
    - Z = 144



- Algorithms
  - Trapezoidal Rule
  - Simpson’s Rule
  - Etc.

Statistics Functions

- Mean
  - >> X = rand(10);
  - >> Y = mean(X)

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

- Median
  - >> Y = mean(X)
- Maximum
  - >> Y = max(X)
- Minimum
  - >> Y = min(X)

- Standard Deviation

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n}}$$

  - >> Y = std(X)
- Sort
  - >> Y = sort(X)
- Find
  - >> Y = X(find(X > 0.5))

Matrix Operators
– Scalar Operations –

- Assignment
  - >> X = [1 , 2 ; 3 , 4];
  - Concatenation
    - >> X = 1:3;
      >> Y = [X ; 2*X]
      - Y = [1 , 2 , 3 ; 2 , 4 , 6]
- Index
  - >> Z = X(1,2)
    –OR–
    >> Z = X(3)
    - Z = 3
  - >> Z = X(2,3)
    –OR–
    >> Z = X(5)
    - ERROR !!
- Unique Functions
  - Identity Matrix
    eye(Column , Row)
    - >> X = eye(2,2)
      - X = [1 , 0 ; 0 , 1]
  - Zeros & Ones Matrix
    - zeros(Column , Row)
    - ones(Column , Row)
  - Random # (0 – 1)
    rand(Column , Row)
    - >> X = rand(2 , 2)
      - X = [0.5234 , 0.9246 ; 0.2862 , 0.7378]

Matrix Operators
– Scalar / Element Wise –

- Scalar Operations
  - Addition (+)
    - >> Z = X + 2
      - Z = [3 , 4 ; 5 , 6]
  - Subtraction (-)
  - Multiplication (*)
    - >> Z = 2 * X
      - Z = [2 , 4 ; 6 , 8]
  - Division (/)
- Element Wise Operations
  - Addition (+)
    - >> X = [1 , 2 ; 3 , 4];
      >> Y = [4 , 3 ; 2 , 1];
      >> Z = X+Y
      - Z = [5 , 5 ; 5 , 5]
  - Subtraction (-)
  - Multiplication
    - >> Z = X.*Y
      - Z = [4 , 6 ; 6 , 4]
  - Division

Matrix Operations
– Linear Algebra ... A x = b –

- b = ?
  - >> A = [1 , 2 ; 3 , 4];
    >> x = [1 ; 2];
    >> b = A*x
    - b = [5 ; 11];

$$A \cdot \vec{x} = \begin{bmatrix} A_{1,1} & \cdots & A_{1,i} & \cdots & A_{1,n} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ A_{i,1} & \cdots & A_{i,i} & \cdots & A_{i,n} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ A_{n,1} & \cdots & A_{n,i} & \cdots & A_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} A_{1,i} x_i \\ \vdots \\ \sum_{i=1}^{n} A_{j,i} x_i \\ \vdots \\ \sum_{i=1}^{n} A_{n,i} x_i \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix} = \vec{b}$$

- x = ?
  - >> A = [1 , 2 ; 3 , 4];
    >> b = [5 ; 11];
    >> x = A\b
    - x = [1 ; 2];
- Algorithms ...
  - Gaussian Elimination
  - LU Factorization
  - Etc.

Matrix Operations

- Inverse
  - >> X = [1 2 3 ; 2 3 1 ; 3 1 2];
  - >> Y = inv(X);
  - >> Z = X*Y
    - Z = [1 0 0 ; 0 1 0 ; 0 0 1]
- Determinant
  - >> Z = det(X)
    - Z = -18
  - EXAMPLE (2 x 2)

$$|\mathbf{X}| = x_{1,1}x_{2,2} - x_{2,1}x_{1,2}$$

  - EXAMPLE (3 x 3)

$$|\mathbf{X}| = x_{1,1}x_{2,2}x_{3,3} - x_{1,1}x_{3,2}x_{2,3} + x_{1,2}x_{2,3}x_{3,1} - \cdots$$
$$x_{1,2}x_{2,1}x_{3,3} + x_{1,3}x_{3,2}x_{2,1} - x_{1,3}x_{2,2}x_{2,1}$$

- Eigenstates

$$\mathbf{Ax} = \alpha\mathbf{x}$$

  - >> [V , D] = eig(Z)
    - Eigenvalues ...
    - D = [-1.73, 1.73, 6.00]
    - Eigenvectors ...
    - V = [0.79 0.21 0.58 ; -0.21 -0.79 0.58 ; -0.58 0.58 0.58]

Linear Regression

- Data ...
  - (2, 3)
  - (4, 7)
  - (5, 10)
  - (6, 10)
  - (7, 12)

- QR Factorization
  - >> X = [2 ; 4 ; 5 ; 6 ; 7] ;
    >> Y = [3 ; 7 ; 10 ; 10 ; 12];
    >> A = [X , ones(size(X))];
    >> COEF = A \ Y
    - COEF = [1.7838 , -0.1622]
- Polyfit
  - >> COEF = polyfit(X , Y , 1)
    - COEF = [1.7838 , -0.1622]

Concept of Transform

- Instead of Nonlinear Regression ...
  - Linearize the Equation
  - Exponentials
    - Y = a*e^(b*X)
    - ln(Y) = ln(a) + b*X
  - Power Law
    - Y = a * X^b ...
    - ln(Y) = ln(a) + b*ln(X)

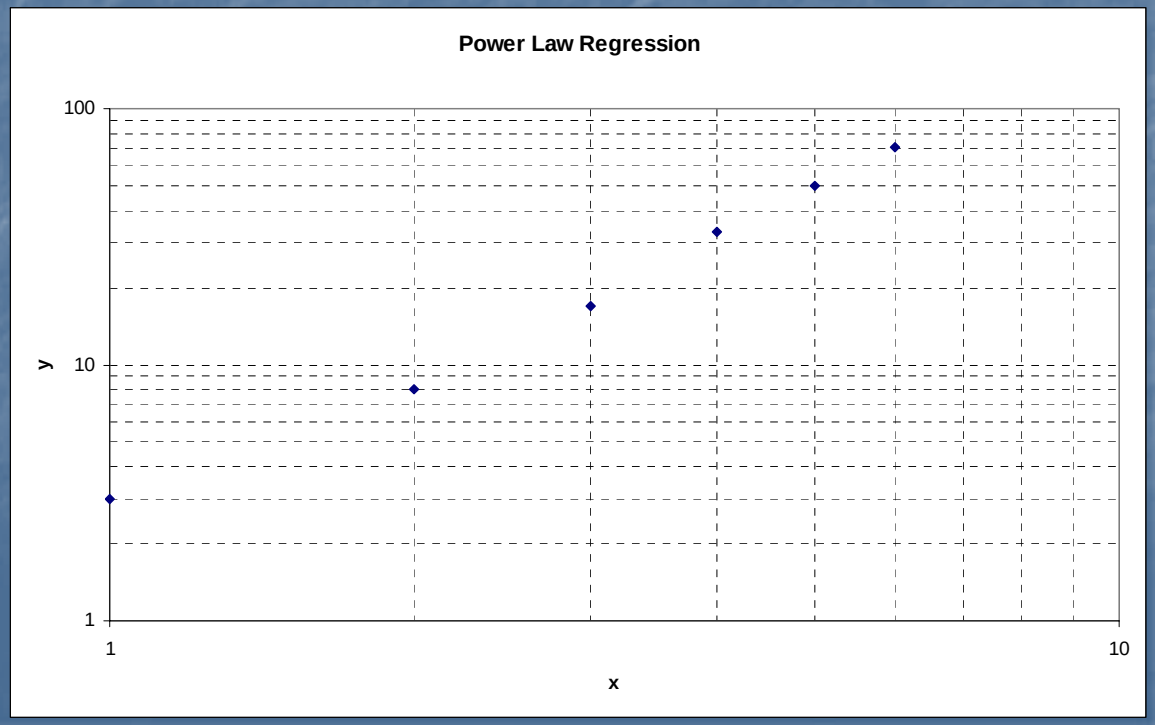
Power Law Regression
– Example –

- Regress Data
  (y = 2*x^2 + noise)
  - (1 , 3)
  - (2 , 8)
  - (3 , 17)
  - (4 , 33)
  - (5 , 50)
  - (6 , 72)
- POWERFIT

```
function COEF = powerfit( x , y )

ln_x = log( x );
ln_y = log( y );
COEF = polyfit( x , y , 1 );
COEF(2) = exp( COEF(2) );
```

- Calling Code
  - >> x = [1 , 2 , 3 , 4 , 5 , 6];
    >> y = [3 , 8 , 17 , 33 , 50 , 71];
    >> COEF = powerfit(x,y)
    - COEF = [1.7971 , 2.6552]



Character Strings

- Assignment
  - >> X = ‘HELLO’;
  - >> Y = 2.7183;
  - >> Z = num2str(Y)
    - Z = ‘2.7183’
- Comparison
  - >> X = ‘HI’
  - >> Y = ‘HE’
  - >> Z = strcmp(X, Y)
    - Z = 0
- Concatenation
  - >> X = strcat(‘HELLO’ , ‘ WORLD’)

    -OR-

    >> X = [‘HELLO’ ‘WORLD’]
    - X = ‘HELLOWORLD’

Structures

- Assignment (Variable.Property = XYZ)
  - >> Person(1).Name = ‘BOB’
  - >> Person(1).Addres = ’99 Sunset Blvd’
  - >> Person(1).Phone = 3334444
  - >> Person(2).Name = ‘JANE’
  - >> Person(2).Address = ‘1750 Kirby Dr’
  - … etc.

Cells

- Assignment
  - >> X = { 1:4 , ‘HI’ ; [1 , 0 ; 0 , 1] , 3.1459 };
- Index
  - >> Z = X{1,1}

    -OR-

    >> Z = X{1}
    - Z = [1 2 3 4]
  - >> Z = X{1,2}(1)
    - Z = ‘H’
  - >> Z = X{2,1}(1,2)
    - Z = 0

Nonlinear Root Finding

- FZERO

  (exp(1 / x) sin(x) / x^2)

  - >> FUN = inline('exp(1 / x) sin(x) / x^2');
  - >> X0 = 3;
  - >> X = fzero(FUN , X0)
    - X = 3.1416

- ROOTS – Polynomials

  (Y = -2X^3 + 4.7*X + 0.25)

  - >> X = [-2 , 0 , 4.7 , 0.25];
  - >> Z = roots(X);
    - Z = [1.56 -1.51 -0.05]

- Algorithms
  - Bisection Method
  - Newton's Method
  - Etc.