Lesson 13 Numerical Methods

Exercise 1 Solving a non linear algebraic equation (Transcendental equations)

1. Solve the following transcendental equation:

    $sin\ x = e^x - 5$

    Solution

    Step 1: Write the equation in standard form

    $$sin(x) - e^x + 5 = 0$$

    $$\text{i.e, } f(x) = sin(x) - ex + 5$$

    Step 2: Write the function that computes $f(x)$

    ```
    function f=transf(x);
    %TRANSF: computes f(x) = sin(x)-exp(x)+5.
    %call syntax: ftransf(x);
    f=sin(x)-exp(x)+5;
    ```

    save the above function file as transf.m

    Step 3: Use fzero to find the solution in the command window

    ```
    >> x=fzero('transf',1) `%initial guess x0=1

    x =

          1.7878
    ```

2. Solve the same problem exercise 1 as mentioned above using MATLAB and show the number of iterations required to find the roots of the equation.

    ```
    >> x=fzero('transf',1,optimset('Display','iter'))%inital gues x0=1
    ```

    Search for an interval around 1 containing a sign change:

Func-count	a	f(a)	b	f(b)	Procedure
1	1	3.12319	1	3.12319	initial interval
3	0.971716	3.18338	1.02828	3.06015	search
5	0.96	3.2075	1.04	3.03319	search
7	0.943431	3.2408	1.05657	2.99419	search
9	0.92	3.28631	1.08	2.93728	search
11	0.886863	3.34759	1.11314	2.8532	search
13	0.84	3.42828	1.16	2.72687	search
15	0.773726	3.53098	1.22627	2.53273	search
17	0.68	3.65492	1.32	2.22529	search
19	0.547452	3.79167	1.45255	1.71902	search
21	0.36	3.91894	1.64	0.842437	search
23	0.0949033	3.99521	1.9051	-0.775417	search

```
Search for a zero in the interval [0.0949033, 1.9051]:
Func-count     x          f(x)             Procedure
   23           1.9051    -0.775417         initial
   24           1.9051    -0.775417         interpolation
   25          1.77601     0.0727553        interpolation
   26          1.78709     0.00467758       interpolation
   27          1.78784    -2.94363e-06      interpolation
   28          1.78784     1.24935e-09      interpolation
   29          1.78784     1.77636e-15      interpolation
   30          1.78784     1.77636e-15      interpolation


Zero found in the interval [0.0949033, 1.9051]


x =

    1.7878
```

LESSON 14   Curve Fitting

Exercise 1: Polynomial curve fitting – Linear fit

Curve fitting is a technique of finding an algebraic relationship that "best"(in a least square sense) fits a given set of data. MATLAB helps us to explore the best possible fit. MATLAB includes **Basic Fitting** in its figure window's **Tools** menu that lets you fit a polynomial curve (up to the tenth order) to your data on the fly. It also gives you options of displaying the residual at the data points and computing and computing the norms of the residuals.

Let us say that we have the following data for *x* and *y* and we want to get the best linear (straight-line) fit through this data.

| x	5	10	20	50	100
y	15	33	53	140	301

Step 1: Plot raw data: Enter the data and plot it as a scatter plot using some marker, say circles.

```
>> x = [5 10 20 50 100];      % x-data
>> y = [15 33 53 140 301];    % y - data
>> plot(x,y,'o');             % plot x vs y using circles
>> xlabel('x');
>> ylabel('y');
```

Step 2: Use built-in Basic fitting from the pull-down menu as shown in the below figure:
    A separate window appears with Basic Fitting options.

Step 3: Fit a linear cure and display the equation: Check the boxes for linear and show equations from the Basic Fitting window options. The best-fitted line as well as its equation appears in the figure window.
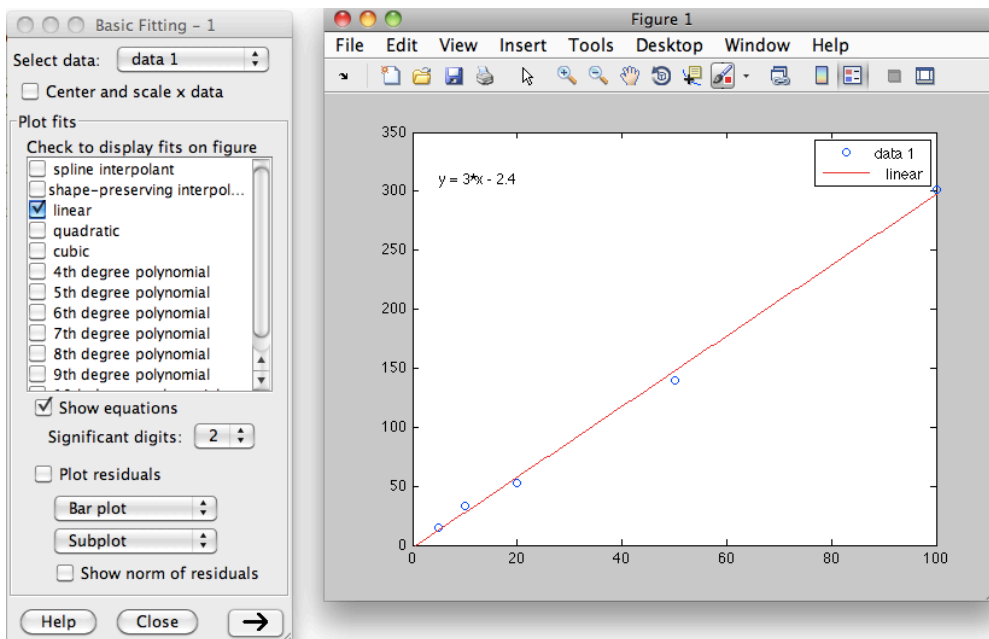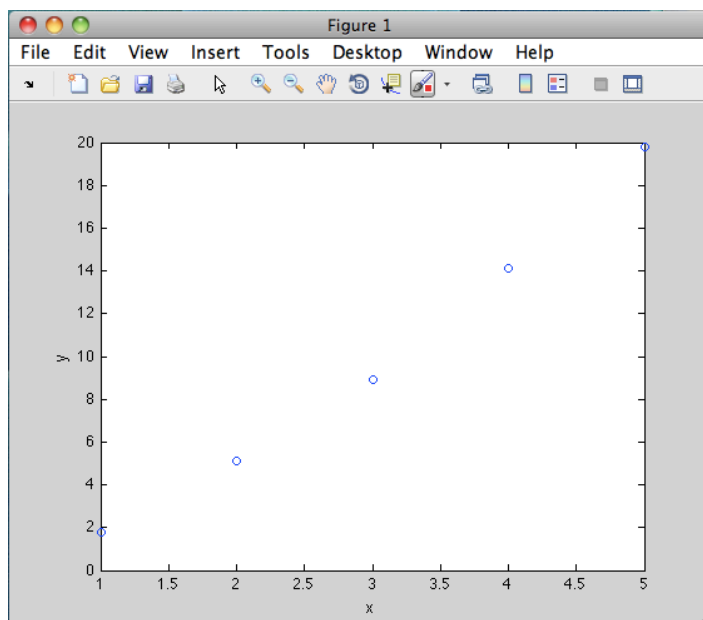
Exercise 2: Computing different fits - Quadratic and Cubic

Step 1: Plot raw data: Use the $x - y$ data

x	1	2	3	4	5
y	1.8	5.1	8.9	14.1	19.8

So, go ahead and plot the raw data as shown below:

```
>> x=[1 2 3 4 5];    % x-data
>> y=[1.8 5.1 8.9 14.1 19.8]; % y-data
>> plot(x,y,'o');
>> xlabel('x');
>> ylabel('y');
```

Step 2:   Use Basic Fitting to do quadratic and a cubic fit: Go to figure window click on **Tools** and select **Basic Fitting** from the pull-down menu (as in the above example 1). In the **Basic Fitting** window, check **quadratic** and **cubic** boxes. In addition, check the box for the show equation:
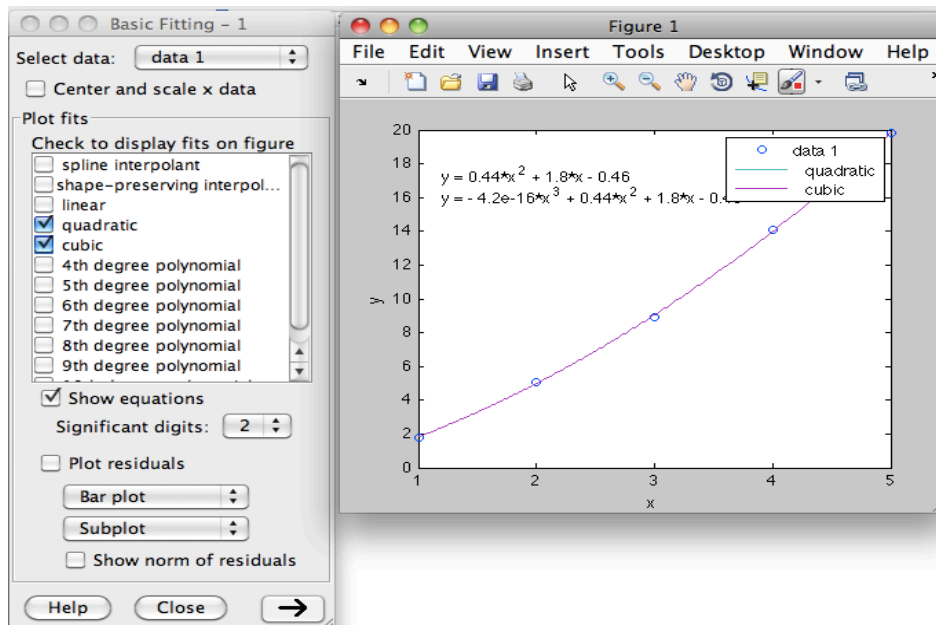
Lesson 15   Numerical Integration

1. Trapezoidal rule
2. Simpson's Rule

Exercise 1: Trapezoidal rule

The MATLAB function **trapz(x,y,n)** where **y** is the integral with respect to **x**, approximates the integral of a function $y = f(x)$ using the trapezoidal rule, and **n** (optional) performs integration along dimension n.

Use the MATLAB function **trapz(x,y)** to approximate the value of the integral

$$\int_{1}^{2} \frac{1}{x}\,dx$$

and by comparison with the exact value, compute the percent error when $n = 5$ and $n = 10$

Solution:

The exact value is found from

$$\int_{1}^{2} \frac{1}{x}\,dx \;=\; \ln x\Big|_{1}^{2} \;=\; \ln 2 - \ln 1 \;=\; 0.6931 - 0.0000 \;=\; 0.6931$$

For the approximation using the trapezoidal rule, we let x_5 represent the row vector with $n = 5$, and x_{10} the vector with $n = 10$, that is, $\Delta x = 1/5$ and $\Delta x = 1/10$ respectively. The corre- sponding values of y are denoted as y_5 and y_{10}, and the areas under the curve as $area5$ and $area10$ respectively. We use the following MATLAB code in command window.

```
>> x5=linspace(1,2,5);
>> x10=linspace(1,2,10);
>> y5=1./x5;  y10=1./x10;
>> area5=trapz(x5,y5);
>> area10=trapz(x10,y10)

     area5 = 0.6970
     area10 = 0.6939
```

Exercise 2: Use the trapezoidal approximation to compute the values of the following definite integral and compare your results with the analytical values. Verify your answers with the MATLAB **trapz(x,y,n)** function.

a. $\int_0^2 x\,dx \quad n = 4$

b. $\int_0^2 x^3\,dx \quad n = 4$

    a. The exact value is

$$\int_0^2 x\,dx = \frac{x^2}{2}\Big|_0^2 = 2$$

    For the trapezoidal rule approximation we have

$$T = \left(\frac{1}{2}y_0 + y_1 + y_2 + \dots + y_{n-1} + \frac{1}{2}y_n\right)\Delta x$$

$x_0 = a = 0$

$x_n = b = 2$

$n = 4$

$\Delta x = \dfrac{b-a}{n} = \dfrac{2-0}{4} = \dfrac{1}{2}$

$y = f(x) = x$

$x_0 = a = 0 \qquad\qquad y_0 = f(x_0) = 0$

$x_1 = a + \Delta x = \dfrac{1}{2} \qquad y_1 = f(x_1) = \dfrac{1}{2}$

$x_2 = a + 2\Delta x = 1 \qquad y_2 = f(x_2) = 1$

$x_3 = a + 3\Delta x = \dfrac{3}{2} \qquad y_3 = f(x_3) = \dfrac{3}{2}$

$x_4 = b = 2 \qquad\qquad y_4 = f(x_4) = 2$

$$T = \left(\frac{1}{2}\times 0 + \frac{1}{2} + 1 + \frac{3}{2} + \frac{1}{2}\times 2\right)\times\frac{1}{2} = 4\times\frac{1}{2} = 2$$

a.
```
>> x=linspace(0,2,4);
>> y=x; area=trapz(x,y)
   area
       = 2
```

```
    b. >> x=linspace(0,2,4);
       >> y=x.^3;
       >> area=trapz(x,y)

          area =

            4.4444
```

2. Simpsons rule

Exercise 1: Using Simpson's rule with 4 subdivisions ($n = 4$) , compute the approximate value of

$$\int_{1}^{2} \frac{1}{x} dx$$

We can find the analytical value with MATLAB's **int(f,a,b)** function where **f** is a symbolic expression, and **a** and **b** are the lower and upper limits of integration respectively. For this example,

```
>> syms x
>> Area=int(1/x,1,2)

Area =

log(2)
```

Exercise 2: Use Simpson's rule to approximate the following definite integral using MATLAB

$$y = f(x) = \int_{0}^{2} e^{-x^2} dx$$

Solution:

```
>> syms x;
>> y=int(exp(x.^2),0,2)  % Define symbolic variable x and integrate

y =

-(pi^(1/2)*erf(2*i)*i)/2

>> pretty(y)

         1/2
      pi     erf(2 i) i
    - ----------------
            2
or

>> syms x;
>> area=int(exp(x^2),0,2)

area =

-(pi^(1/2)*erf(2*i)*i)/2

>> pretty(area)

         1/2
      pi     erf(2 i) i
    - ----------------
            2
```