MM3110 - Expt 3 - Numerical Integration

Vishal S
MM14B048

K Harish
MM14B040

1 Question 1

$$f(x) = |x^5 - 15x^4 + 85x^3 - 225x^2 + 274x - 120| \tag{1}$$

The function in equation 1 is plotted in figure 1 and the extreme roots are found to be 1 and 5.

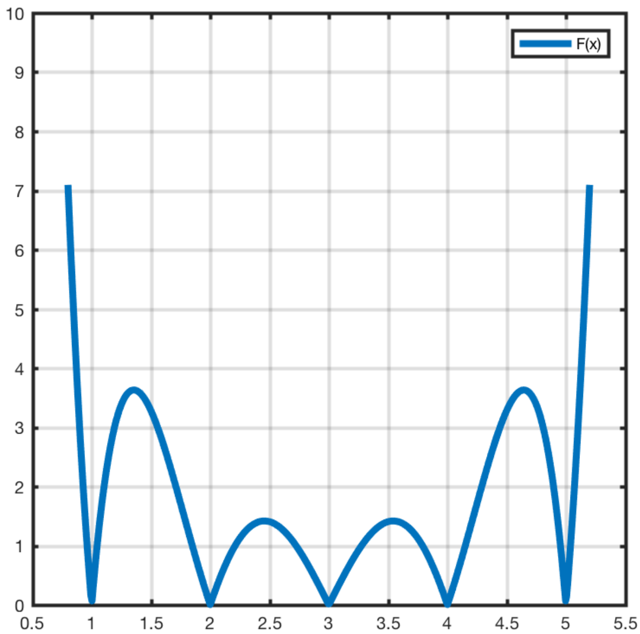

Figure 1: Plot of f(x) as given in equation 1

The function in equation 1 is integrated from 1 to 5 using different pre-defined functions available in MATLAB. The code used is given below.

```
%Defining the function
fun = @(x) abs( x.^5 - 15.*(x.^4) + 85.*(x.^3) - 225.*(x.^2) + 274.*x - 120 );
%Performing Integration using integral()
integ1 = integral(fun,1,5);
integ1
%Performing Integration using int()
syms x
integ2 = int(fun,x,1,5);
integ2
%Performing Integration using trapz()
X = 1:0.01:5;
Y = fun(X);
integ3 = trapz(X,Y);
integ3
```

The output obtained on executing the code is as given below.

```
 ------------------
integ1 =

    6.3333
 ------------------

 ------------------
 integ2 =

19/3
 ------------------

 ------------------
integ3 =

    6.3327
 ------------------
```

A separate code is written in MATLAB to look into the dependence of the calculated value using `trapz` on the step size. The code is given below.

```
%Defining the function
fun =  @(x) abs(x.^5 - 15.*x.^4 + 85.*x.^3 - 225.*x.^2 + 274.*x - 120);
%Integrating for different values of h
h= [0.1,0.05,0.01,0.005,0.001,0.0005];
integ = zeros(6,1)
for i = 1:6
    x=1:h(i):5;
    y = fun(x);
    integ(i) = trapz(x,y);
end
%Calculating Error
error = zeros(6,1);
for i = 1:6
    error(i) = abs((19/3) - integ(i)) /(19/3);
end
%Plotting
loglog(h,error,'-o','linewidth',4.0);
xlim([0.0003 0.11]);
xlabel('Step size');
ylabel('Relative error (in %)')
ax = gca ;
set(ax, 'linewidth',2.0);
axis('square');
grid on;
```

The relative error is then calculated and plotted in fig 2.

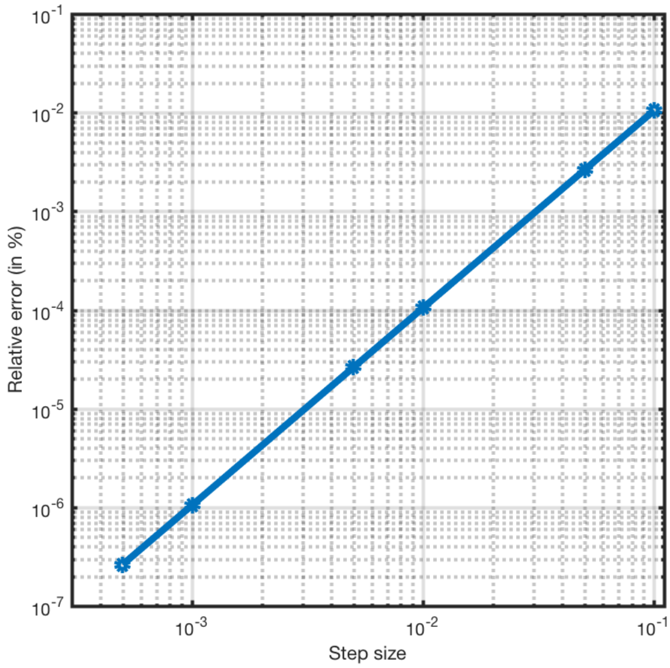

Figure 2: Log-Log plot of relative error for various `h` values

2 Question 2

The velocity of the object falling under the effect of gravity and air resistance is given by,

$$v(t) = \sqrt{\frac{gm}{c_d}} tanh(\sqrt{\frac{gc_d}{m}}t),$$
$$\frac{dx}{dt} = \sqrt{\frac{gm}{c_d}} tanh(\sqrt{\frac{gc_d}{m}}t),$$

$$dx = \sqrt{\frac{gm}{c_d}} tanh(\sqrt{\frac{gc_d}{m}}t)dt \tag{2}$$

To calculate how far the body falls in first 10 s , equation 2 should be integrated from 0 to 10s .

$$x = \int_0^{10} \sqrt{\frac{gm}{c_d}} tanh(\sqrt{\frac{gc_d}{m}}t)dt \tag{3}$$

Thus, equation 3 can be integrated using *Trapezoidal Method.* The values of the constants are given as $g = 9.8ms^{-2}$, $m = 70kg$, and $c_d = 0.25\ kg\ m^{-1}$.

This integration can be implemented in a MATLAB code as shown below,

```
% variables in SI units

g = 9.8 ;
cd = 0.25;
m = 70;
t0 = 0;
tend = 10;
h = 0.01;

% Definiton of function to be integrated
fun = @(t) sqrt(g*m/cd)*tanh(sqrt(g*cd/m)*t);
time = t0:h:tend;
i = 1;
integ = 0;
% Loop to implement Trapezoidal method
while time(i) < tend
    integ = integ + 0.5*(fun(time(i)+h) + fun(time(i)) )*h;
    i = i+1;
end

integ
```

On executing the above code, the value of `integ`, which is the distance it has travelled in first 10 s, is 336.3134 m.

The variable `h` is used to define the step size and it takes the value of 0.01 in this code.

3 Question 3

Solving analytically ,

$$y = x^2 cosx,$$

Differentiating both sides with respect to x. We get,

$$\frac{dy}{dx} = 2xcosx - x^2 sinx \tag{4}$$

Substituting $x = 0.4$ in equation 4, gives the derivative at that value.

$$\left(\frac{\mathrm{d}y}{\mathrm{d}x}\right)_{x=0.4} = 2 \times (0.4)\cos(0.4) - (0.4)^2 \times \sin(0.4)$$

Thus, the value of $\left(\frac{\mathrm{d}y}{\mathrm{d}x}\right)_{x=0.4}$ calculated analytically is 0.67454.

The above calculation can also be performed numerically by approximating,

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}, \tag{5}$$

Since evaluating the above expression at h = 0 is not computationally feasible, the calculation is performed for smaller and smaller values of `h` and the relative error is plotted in figure 3. The code that performs the above calculation is given below,

```

% Initialisation of variables
h= [0.1,0.005,0.001,0.0005,0.0001,0.00001];
x = 0.4;
fdash_analytically = 0.6745418604
%Calculating Fdash for different values of h
for i = 1:6
fdash(i) = ( f(x+h(i)) - f(x) )/h(i);
end
fdash
%Calcualting the relative error in Fdash for different values of h
error = zeros(6,1);
for i = 1:6
    error(i) = (fdash(i) - fdash_analytically)*100/fdash_analytically;
end
%Plotting

loglog (h,error,'linewidth',4.0);
xlabel('h');
ylabel('Relative error (in %)')
ax = gca ;
set(ax, 'linewidth',2.0);
axis('square');
grid on;
function y = f(x)
y = x^2*cos(x) ;
end
```

The plot of the calculated value versus the step size is given in figure 4.

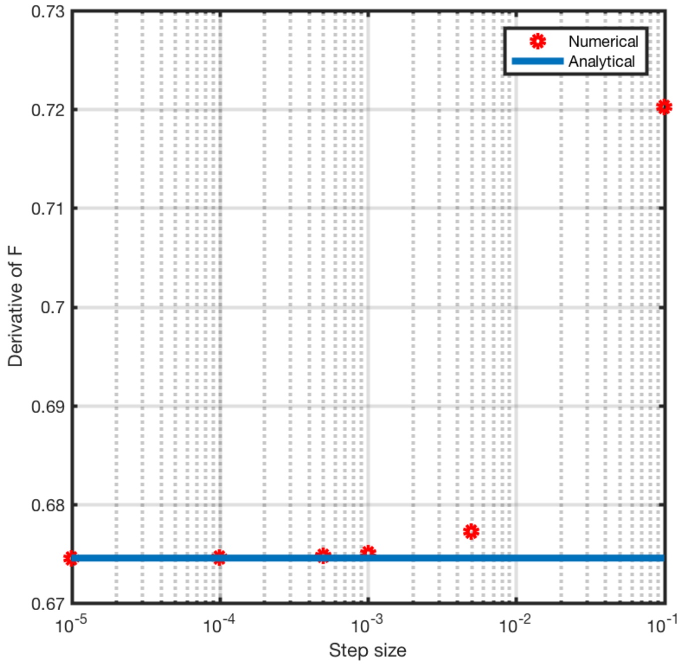

Figure 3: Semi-log plot of $\left(\dfrac{\mathrm{d}y}{\mathrm{d}x}\right)_{x=0.4}$ as a function of step size

The relative error in the computation is also computed and plotted below,

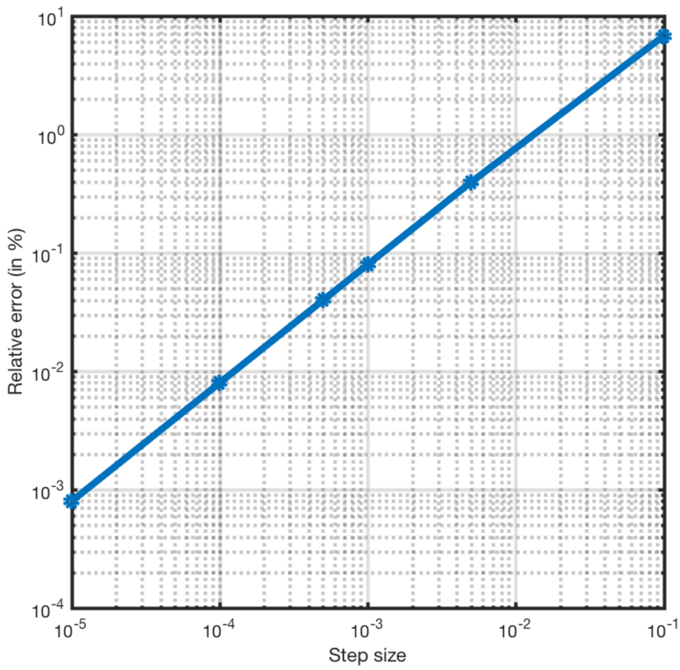

Figure 4: log-log plot of relative error in $\left(\dfrac{\mathrm{d}y}{\mathrm{d}x}\right)_{x=0.4}$ as a function of step size

The error is expected to decrease with smaller and smaller values of `h`. This is because the forward difference is obtained by assuming Taylor series approximation and hence there is an error term of $O(h)$ associated with it.

4 Question 4

$$\int_{-2}^{2}\int_{0}^{4}(x^2+-3y^2+xy^3)\,dx\,dy \tag{6}$$

Integrating equation 6 with respect to x first we get ,

$$\int(x^2+-3y^2+xy^3)\,dx=\frac{x^3}{3}-3y^2x+\frac{x^2y^3}{2}$$

Applying appropriate limits the equation 6 reduces

$$\int_{-2}^{2}(\frac{64}{3}-12y^2+8y^3)\,dy \tag{7}$$

Integrating the above equation with respect to y and then applying the limits we obtain the answer to be $\frac{64}{3}\approx 21.333$. The above calculation can also be performed numerically using codes written in MATLAB. In MATLAB a pre-defined function called `integral2` is available which can numerically evaluvate double integrals. The code given below is used to evaluate equation 6

```
%Definig the function
fun = @(x,y) x.^2 + - 3.*(y.^2) + x .* ( y.^3) ;
%Defining the limits
xmin = 0;
xmax = 4;
ymin = -2;
ymax = 2;
%Performing Integration
integ = integral2 (fun , xmin,xmax,ymin,ymax);
%Printing the output
integ
```

On executing the code the value of `integ` obtained is 21.3333 which is as expected.