Ryan Peraza
862089227
Section: 021

EE105 Lab 1

Purpose

The Purpose of this Lab is to provide an introduction, or tutorial, to Matlab. We used the **'help'** assistance to guide us through the basic features of Matlab.

Lab Results

Section 2 Matlab Tutorial - 3. Matrices and Arrays: This code multiplies two matrices using matrix operations, where it Transposes Matrix $\boldsymbol{A}$ and then multiplies it to Matrix $\boldsymbol{B}$ to find Matrix $\boldsymbol{C}$.

```
>>  A = [pi;sqrt(2);exp(1)]; %Matrix A 3x1
>> B = [1;5;7]; %Matrix B 3x1
>> C = A'*B %Transposes A to be a 1x3 matrix and multiplies it by B a 3x1 matrix

C =

   29.2406
```

Section 2 - 5. Scripts:

a) Clears the memory

b) Defines the Matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ given above

c) Implements a **'for loop'** to compute $\boldsymbol{D} = \sum_{i=1}^{3} \boldsymbol{a_i}\,\boldsymbol{b_i}$

```
clear
A = [ pi;sqrt(2);exp(1)];   %Matrix A
B = [1;5;7];                %Matrix B
D = 0;                      %initializes D to zero
Current = 0;                %initializes Current to zero
for i = 1:3                 %runs through the 3x1 matrix of A and B

    Current = A(i)*B(i);     %stores value of product at each indice of the matrix
    D = D + Current;         %keeps a running total of each elements value
end

D
```

Answer should be identical to $\boldsymbol{C}$ from the previous step.

```
>> summation

D =

   29.2406
```

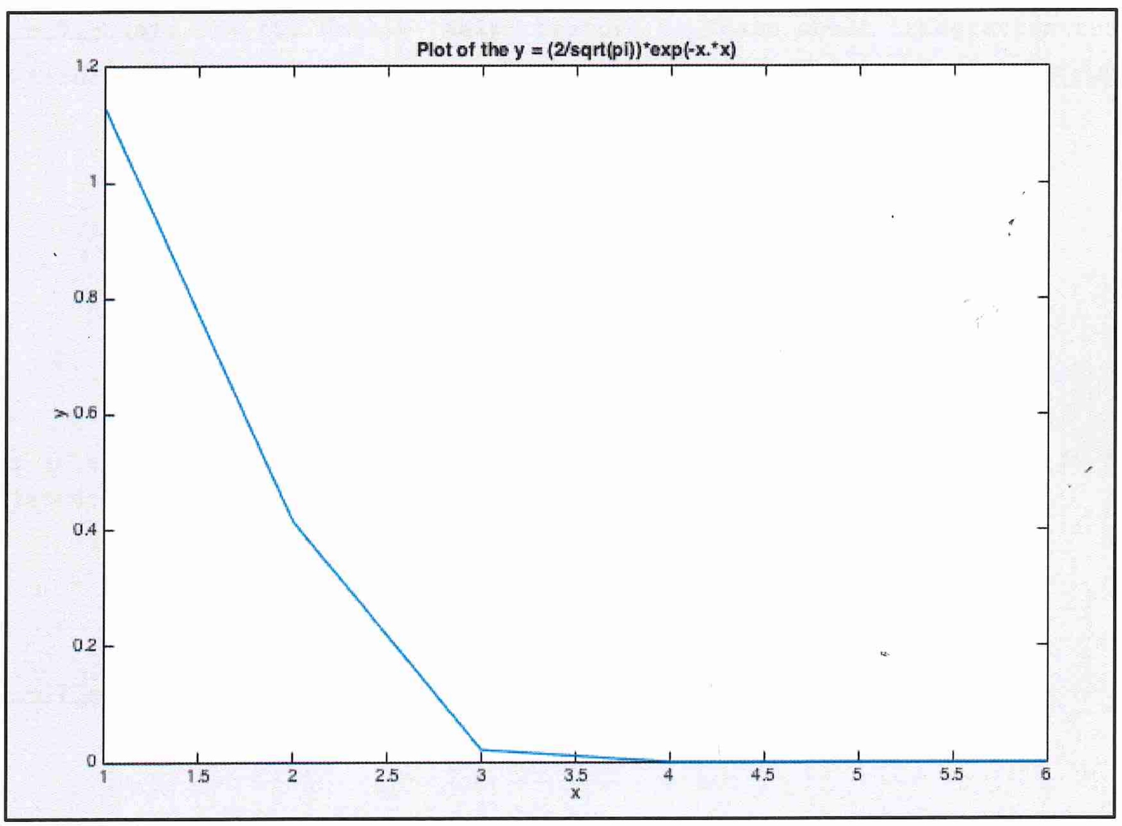
ion 2 - 6 - (d) i: The first m-file will have an input column vector $\boldsymbol{x}$ and an output umn vector $\boldsymbol{y}$ where the ***i-th*** element of the output vector is $y_i = f(x_i)$ and

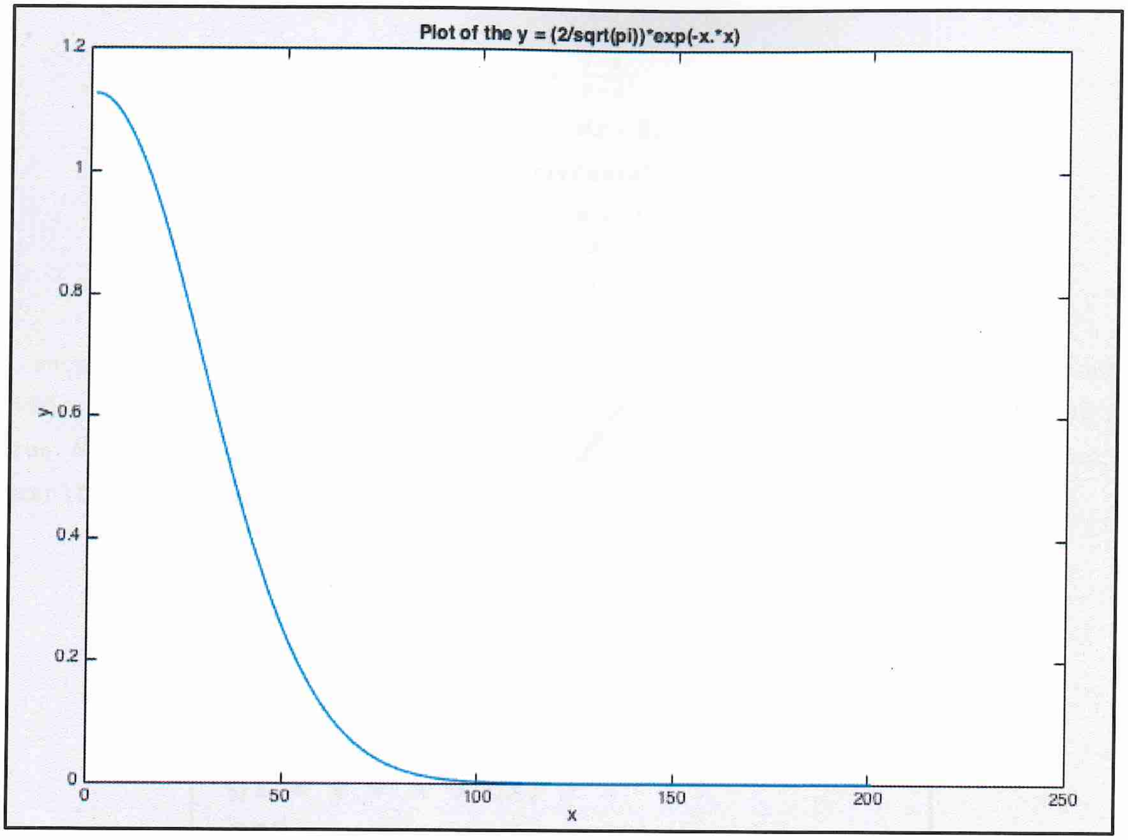
$$(x_i) = \frac{sin(x_i)}{1 + exp(2x_i)}.$$

```
function [ y ] = mfile1( x ) % takes x as the input and outputs y as function of x

y = (2/sqrt(pi))*exp(-x.*x); %output which computes the function for each column element of matrix x

end
```

Section 2 - 6 - (d) ii: The second m-file will have an input integer $\boldsymbol{N}$ and no outputs. It should define the vector $\boldsymbol{x}$ so that it contains $(\boldsymbol{N} + \mathbf{1})$ equally spaced points on the interval [0,5], call the previous m-file to calculate the vector $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$, and plot $\boldsymbol{y}$ as a function of $\boldsymbol{x}$.

```
function [  ] = mfile2( N ) %input N
x = linspace(0,5,N+1)          %creates row vector on interval [0,5], with N+1 evenly spaced elements
xnew = x';                     %transposes row vector to make column vector
plot(mfile1(xnew));            %passes xnew column vector through mfile1 which gives the y output
xlabel('x');                   %and plots it
ylabel('y');
title('Plot of the y = (2/sqrt(pi))*exp(-x.*x) ');
end
```





<u>Section 2 - 7 - (a):</u> Use the Matlab '**help**' feature to learn about integration routines and in particular the '**quad**' function. Then use '**quad**' to approximate $\boldsymbol{Q} = \int_0^5 \boldsymbol{f(x)dx}$.

```
>> Q = quad(@mfile1,0,5)

Q =

    1.0000
```

This takes an approximation of the integral from 0 to 5 of the function 'y'. It has an error of 1e^-6.

<u>Section 2 - 7 - (b):</u> Hand written response. See attached work.

<u>Section 2 - 7 - (c):</u> The summation in the calculation of $\boldsymbol{Q_1}$ can be written as the product of two vectors and calculated using vector arithmetic.

$$dx = \frac{5-0}{N-1};$$

$$x = 0 : dx : 5;$$

$$y = myfun(x);$$

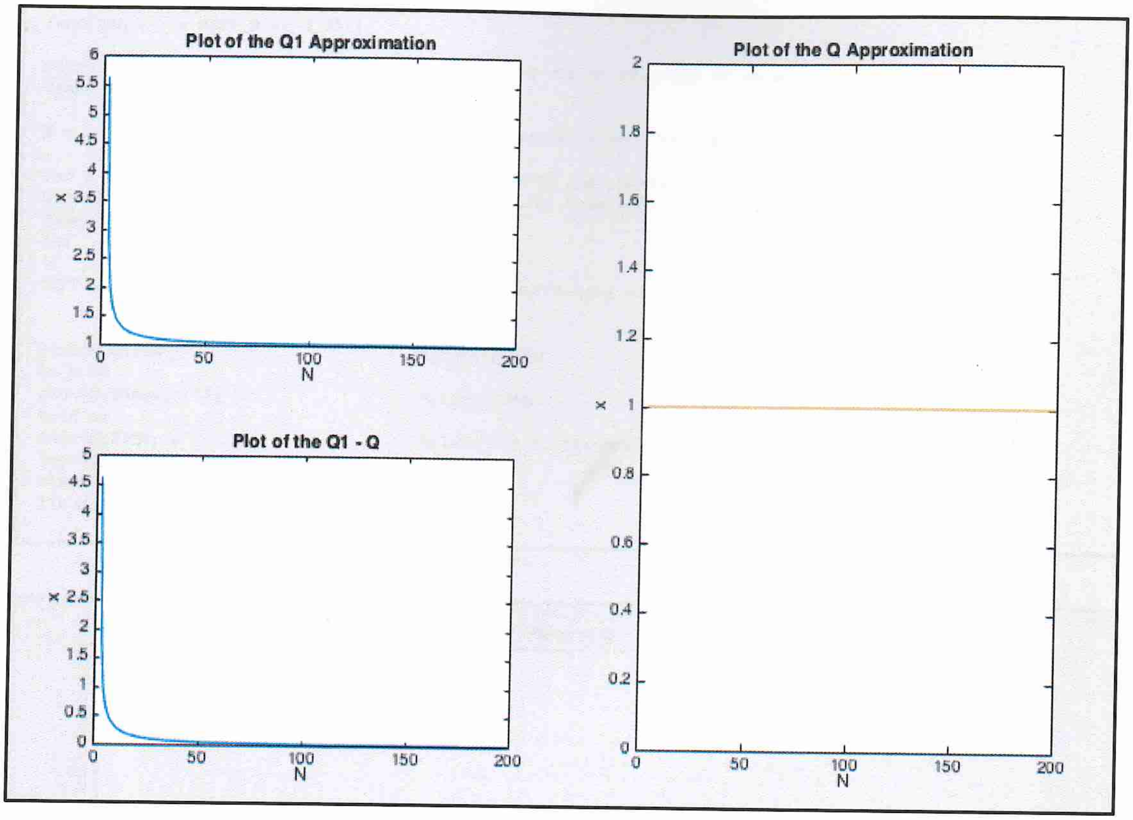
$$A = \begin{bmatrix} ones(N-1,1) \\ 0 \end{bmatrix};$$

$$Q_1 = y * A * dx;$$

The above should be working Matlab code. Implement the above code. Then Implement another function which computes and plots Q_1 for all integer values of N from 2 to 200. Plot $Q_1(N)$ versus N. Also plot the constant value of Q determined by the **'quad'** function. Compare results.

```
function [ Q1 ] = Compute_Q1( N )
dx = (5-0)/(N-1);
x = 0 : dx : 5;
y = mfile1(x);
A = [ones(N-1,1);0];
Q1 = y * A * dx;
end
```

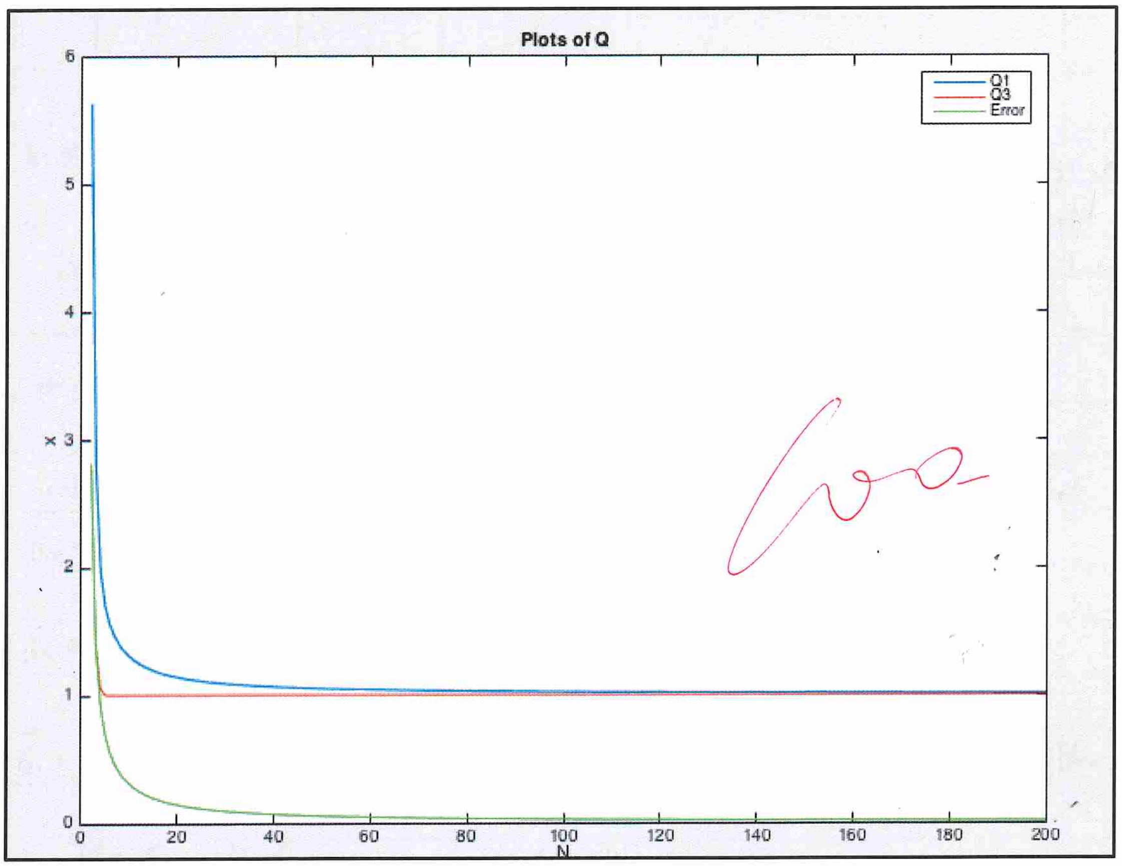
```
function [ ] = make_plot_Q_Q1()

Q1new = zeros(1,199);                  %creates a vector of 1x199 of zeros

N = 2:200;                             %uses N values from 2 to 200
Q = quad(@mfile1,0,5);                 %uses quad to approximate integral of y(x) from 0 to 5 - constant value
Q_plot = quad(@mfile1,0,5)*ones(length(N));           %plots Q, a vertical line y = 1 filling a matrix of 1's
Q_matrix = Q*ones(1,length(N));        %creates a 1x199 matrix to match Q1_new to find difference below

for i = 2:200                          %computes Q1 and shfits it into the i-1 place of Q1_new matrix
Q1new(i-1) = Compute_Q1(i);            %Computes Q1 from previous m file
end

DIFF = Q1new - Q_matrix;               %takes the difference of the two 1x199 functions

subplot(2,2,1);                        %divides plot window into a matrix of 2x2
plot(N,Q1new);                          %plots Q1new
xlabel('N'); ylabel('x');
title('Plot of the Q1 Approximation ');
subplot(1,2,2);
plot(N,Q_plot);                         %plots constant value Q
xlabel('N'); ylabel('x');
title('Plot of the Q Approximation ');
axis([0 200 0 2]);
subplot(2,2,3);
plot(N,DIFF);                           %plots the difference between the two functions
xlabel('N'); ylabel('x');
title('Plot of the Q1 - Q');
end
```



Section 2 - 7 - (d): Write your own code similar to the above to compute Q_3 directly; clearly state your definition of the Matrix A. Compute Q_3 for all integer values of N from 2 to 200. Plot $Q_1(N)$ versus N on the same graph as Q_1 versus N. Also plot the error in Q_3 versus N on the same graph as the error in Q_1.

```
function [ Q3 ] = Compute_Q3( N )
dx = (5-0)/(N-1);
x = 0 : dx : 5;
y = mfile1(x);
A = [0.5;ones(N-2,1);0.5]; %same as the funciton provided in lab except it now places 0.5 at beginning and end
Q3 = y * A * dx;            %of the A matrix with N-2 as the number of rows in middle as 1 column
end
```

```
function [ ] = make_plot_Q_Q3()

Q1new = zeros(1,199);                  %creates a vector of 1x199 of zeros
Q3new = zeros(1,199);

N = 2:200;                             %uses N values from 2 to 200

for i = 2:200                          %computes Q1 and shfits it into the i-1 place of Q1_new matrix
Q1new(i-1) = Compute_Q1(i);            %Computes Q1 from previous m file
Q3new(i-1) = Compute_Q3(i);
end

DIFF = Q1new - Q3new;              %takes the difference of the two 1x199 functions


plot(N,Q1new);                       %plots Q1new
hold on
plot(N,Q3new,'r');                   %plots Q3new
hold on
plot(N,DIFF,'g');                    %plots the difference between the two functions
legend('Q1','Q3','Error')
xlabel('N'); ylabel('x');
title('Plots of Q');
```



Conclusion

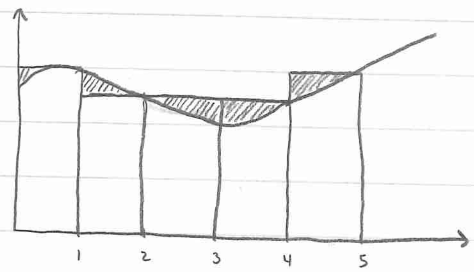
This Lab was an introduction to Matlab with assistance from the 'help' feature and also a great way to learn about how to use different functions of Matlab that I have never used before. This Lab taught me to visually represent Matrices and Functions with Matlab graphing tools. The new Concepts in this Lab proved to be very difficult for me. I struggled to get my m-files to work properly, but with the assistance I received from the Transfer Center I was able to figure it out. This Lab helped me tremendously with understanding how to use the new functions we will be utilizing for this class.

Ryan Peraza
1-24-19
EE105
Lab #1

EE105 - Lab #1

Section 2 - 7 (b) i - Draw a picture and explain the equation for the computation of Q_2. Label the figure appropriately and explain why the formula is valid for small dx_i.

$$Q_2 = \sum_{i=2}^{N} f(x_i)\,dx_{i-1}$$



* The formula is valid for a small dx_i, because having a large dx_i will have parts that are not under the curve, so parts that should be under the curve will not be accounted for in the equation. So having a smaller dx_i fixes this problem so that the approximation will be a much more accurate representation.

Section 2 - 7 (b) ii - A) Derive the above equation for Q_3. B) Show that $Q_3 = (Q_1 + Q_2)/2$.

$$Q_3 = f(x_1)\frac{dx_1}{2} + \sum_{i=2}^{N-1} f(x_i)\,dx_i + f(x_N)\frac{dx_{N-1}}{2} \quad \ldots \text{ from } [0,5]$$

$$Q_3 = f(0)\frac{dx_0}{2} + \left(\frac{f(1)+f(0)}{2}\right)(dx_1 - dx_0) + \left(\frac{f(2)+f(1)}{2}\right)(dx_2 - dx_1) + \left(\frac{f(3)+f(2)}{2}\right)(dx_3 - dx_2) + \left(\frac{f(4)+f(3)}{2}\right)(dx_4 - dx_3) + \left(\frac{f(5)+f(4)}{2}\right)(dx_5 - dx_4)$$

$$Q_3 = \cancel{\tfrac{1}{2}}\left(\cancel{f(0)dx_0}\right) + \tfrac{1}{2}\left(\cancel{f(1)dx_1} - f(1)dx_0 + f(0)dx_1 - \cancel{f(0)dx_0}\right) + \tfrac{1}{2}\left(\cancel{f(2)dx_2} - f(2)dx_1 + f(1)dx_2 - \cancel{f(1)dx_1}\right) + \tfrac{1}{2}\left(\cancel{f(3)dx_3} - f(3)dx_2 + f(2)dx_3 - \cancel{f(2)dx_2}\right)$$
$$+ \tfrac{1}{2}\left(\cancel{f(4)dx_4} - f(4)dx_3 + f(3)dx_4 - \cancel{f(3)dx_3}\right) + \tfrac{1}{2}\left(f(5)dx_5 - f(5)dx_4 + f(4)dx_5 - \cancel{f(4)dx_4}\right)$$

$$Q_3 = \tfrac{1}{2}\left(f(0)dx_1 - f(1)dx_0\right) + \tfrac{1}{2}\left(f(1)dx_2 - f(2)dx_1\right) + \tfrac{1}{2}\left(f(2)dx_3 - f(3)dx_2\right) + \tfrac{1}{2}\left(f(3)dx_4 - f(4)dx_3\right) + \tfrac{1}{2}\left(f(4)dx_5 - f(5)dx_4\right)$$

$$Q_3 = \tfrac{1}{2}\left[-\left(\cancel{f(1)dx_0}\right) + \left(f(0) - \cancel{f(2)}\right)dx_1 + \left(f(1) - \cancel{f(3)}\right)dx_2 + \left(f(2) - \cancel{f(4)}\right)dx_3 + \left(f(3) - \cancel{f(5)}\right)dx_4 + \left(f(4)\,dx_5\right)\right] =$$

$$= \frac{Q_1 + Q_2}{2} = \tfrac{1}{2}\left[f(0)dx_0 + \cancel{f(1)dx_1} + f(2)dx_2 + f(3)dx_3 + f(4)dx_4 + f(5)dx_5 + \cancel{f(1)dx_0} + \cancel{f(2)dx_1} + \cancel{f(3)dx_2} + \cancel{f(4)dx_3} + \cancel{f(5)dx_4}\right]$$

$$\boxed{* \quad Q_3 = \frac{Q_1 + Q_2}{2}}$$