Emile Goulard
uID: u1244855
CS3200 - Assignment 3

Question 1

To find the [wi, xi] pairs for the 3 Newton-Cotes methods, open “A3_Question1And2.m” and follow the instructions provided in the README file and Header Comment.

The methods used to make these work simply involved making arrays for N, wi, and xi values to store for later. Within each individual function lies that particular Composite Rule computations suhc as solving for DeltaX and how to iterate for N values in a for-loop. Inside the for-loop for these functions, I simply store the values of wi and xi using their respective formulas. Afterwards, I print out the values (iterating by 2) to a table.

a) In the parameter, specify the a, b, N values. **figureNum MUST BE SET TO 1.**

For the following outputs, I inserted a=1, b=4, N=1024 into the parameters

Command Window

Composite Midpoint Rule Table

N	wi	xi
1	0.0029296875	1.00146484375
2	0.0029296875	1.00439453125
0	0	0
4	0.0029296875	1.01025390625
0	0	0
6	0.0029296875	1.01611328125
0	0	0
8	0.0029296875	1.02197265625
0	0	0
10	0.0029296875	1.02783203125
0	0	0
12	0.0029296875	1.03369140625
0	0	0
14	0.0029296875	1.03955078125
0	0	0
16	0.0029296875	1.04541015625
0	0	0
18	0.0029296875	1.05126953125
0	0	0
20	0.0029296875	1.05712890625
0	0	0
22	0.0029296875	1.06298828125
0	0	0
24	0.0029296875	1.06884765625
0	0	0
26	0.0029296875	1.07470703125
0	0	0
28	0.0029296875	1.08056640625
0	0	0
30	0.0029296875	1.08642578125
0	0	0
32	0.0029296875	1.09228515625
0	0	0
34	0.0029296875	1.09814453125
0	0	0
36	0.0029296875	1.10400390625
0	0	0
38	0.0029296875	1.10986328125
0	0	0

Command Window

0	0	0
984	0.0029296875	3.88134765625
0	0	0
986	0.0029296875	3.88720703125
0	0	0
988	0.0029296875	3.89306640625
0	0	0
990	0.0029296875	3.89892578125
0	0	0
992	0.0029296875	3.90478515625
0	0	0
994	0.0029296875	3.91064453125
0	0	0
996	0.0029296875	3.91650390625
0	0	0
998	0.0029296875	3.92236328125
0	0	0
1000	0.0029296875	3.92822265625
0	0	0
1002	0.0029296875	3.93408203125
0	0	0
1004	0.0029296875	3.93994140625
0	0	0
1006	0.0029296875	3.94580078125
0	0	0
1008	0.0029296875	3.95166015625
0	0	0
1010	0.0029296875	3.95751953125
0	0	0
1012	0.0029296875	3.96337890625
0	0	0
1014	0.0029296875	3.96923828125
0	0	0
1016	0.0029296875	3.97509765625
0	0	0
1018	0.0029296875	3.98095703125
0	0	0
1020	0.0029296875	3.98681640625
0	0	0
1022	0.0029296875	3.99267578125
0	0	0
1024	0.0029296875	3.99853515625

b) In the parameter, specify the a, b, N values. **figureNum MUST BE SET TO 2.**

For the following outputs, I inserted a=1, b=4, N=1024 into the parameters

Command Window

Composite Trapezoidal Rule Table

N	wi	xi
1	0.00146627565982405	1
2	0.00293255131964809	1.00293255131965
0	0	0
4	0.00293255131964809	1.00879765395894
0	0	0
6	0.00293255131964809	1.01466275659824
0	0	0
8	0.00293255131964809	1.02052785923754
0	0	0
10	0.00293255131964809	1.02639296187683
0	0	0
12	0.00293255131964809	1.03225806451613
0	0	0
14	0.00293255131964809	1.03812316715543
0	0	0
16	0.00293255131964809	1.04398826979472
0	0	0
18	0.00293255131964809	1.04985337243402
0	0	0
20	0.00293255131964809	1.05571847507331
0	0	0
22	0.00293255131964809	1.06158357771261
0	0	0
24	0.00293255131964809	1.06744868035191
0	0	0
26	0.00293255131964809	1.0733137829912
0	0	0
28	0.00293255131964809	1.0791788856305
0	0	0
30	0.00293255131964809	1.08504398826979
0	0	0
32	0.00293255131964809	1.09090909090909
0	0	0
34	0.00293255131964809	1.09677419354839
0	0	0
36	0.00293255131964809	1.10263929618768
0	0	0
38	0.00293255131964809	1.10850439882698
0	0	0

fx

Command Window

0	0	0
984	0.00293255131964809	3.88269794721408
0	0	0
986	0.00293255131964809	3.88856304985337
0	0	0
988	0.00293255131964809	3.89442815249267
0	0	0
990	0.00293255131964809	3.90029325513196
0	0	0
992	0.00293255131964809	3.90615835777126
0	0	0
994	0.00293255131964809	3.91202346041056
0	0	0
996	0.00293255131964809	3.91788856304985
0	0	0
998	0.00293255131964809	3.92375366568915
0	0	0
1000	0.00293255131964809	3.92961876832845
0	0	0
1002	0.00293255131964809	3.93548387096774
0	0	0
1004	0.00293255131964809	3.94134897360704
0	0	0
1006	0.00293255131964809	3.94721407624633
0	0	0
1008	0.00293255131964809	3.95307917888563
0	0	0
1010	0.00293255131964809	3.95894428152493
0	0	0
1012	0.00293255131964809	3.96480938416422
0	0	0
1014	0.00293255131964809	3.97067448680352
0	0	0
1016	0.00293255131964809	3.97653958944282
0	0	0
1018	0.00293255131964809	3.98240469208211
0	0	0
1020	0.00293255131964809	3.98826979472141
0	0	0
1022	0.00293255131964809	3.9941348973607
0	0	0
1024	0.00146627565982405	4

fx >>

c) In the parameter, specify the a, b, N values. **figureNum MUST BE SET TO 3.**

For the following outputs, I inserted a=1, b=4, N=1024 into the parameters

Command Window

Composite Simpsons Rule Table

N	wi	xi
1	0.00048828125	1
2	0.001953125	1.00146484375
0	0	0
4	0.001953125	1.00439453125
0	0	0
6	0.001953125	1.00732421875
0	0	0
8	0.001953125	1.01025390625
0	0	0
10	0.001953125	1.01318359375
0	0	0
12	0.001953125	1.01611328125
0	0	0
14	0.001953125	1.01904296875
0	0	0
16	0.001953125	1.02197265625
0	0	0
18	0.001953125	1.02490234375
0	0	0
20	0.001953125	1.02783203125
0	0	0
22	0.001953125	1.03076171875
0	0	0
24	0.001953125	1.03369140625
0	0	0
26	0.001953125	1.03662109375
0	0	0
28	0.001953125	1.03955078125
0	0	0
30	0.001953125	1.04248046875
0	0	0
32	0.001953125	1.04541015625
0	0	0
34	0.001953125	1.04833984375
0	0	0
36	0.001953125	1.05126953125
0	0	0
38	0.001953125	1.05419921875
0	0	0

fx

Command Window

2008	0.001953125	3.93994140625
0	0	0
2010	0.001953125	3.94287109375
0	0	0
2012	0.001953125	3.94580078125
0	0	0
2014	0.001953125	3.94873046875
0	0	0
2016	0.001953125	3.95166015625
0	0	0
2018	0.001953125	3.95458984375
0	0	0
2020	0.001953125	3.95751953125
0	0	0
2022	0.001953125	3.96044921875
0	0	0
2024	0.001953125	3.96337890625
0	0	0
2026	0.001953125	3.96630859375
0	0	0
2028	0.001953125	3.96923828125
0	0	0
2030	0.001953125	3.97216796875
0	0	0
2032	0.001953125	3.97509765625
0	0	0
2034	0.001953125	3.97802734375
0	0	0
2036	0.001953125	3.98095703125
0	0	0
2038	0.001953125	3.98388671875
0	0	0
2040	0.001953125	3.98681640625
0	0	0
2042	0.001953125	3.98974609375
0	0	0
2044	0.001953125	3.99267578125
0	0	0
2046	0.001953125	3.99560546875
0	0	0
2048	0.001953125	3.99853515625
2049	0.00048828125	4

Question 2

To find the [wi, xi] pairs for theGaussian Quadrature, open “A3_Question1And2.m” and follow the instructions provided in the README file and Header Comment.

The methods used wsa simply checking the value of N between 2 and 5 in a for loop. If a certain N value was met, I would plug in the values provided by the Gaussian Quadrature table into arrays (N, wi, xi). This would then print out a table showing the [wi, xi] pairs into a command window.

- In the parameter, a, b and N values won’t be needed (but still specify values for the program to run).
- **figureNum MUST BE SET TO 4.**

The following will be printed into the command window:

```
Command Window
>> A3_Question1And2(1, 4, 2, 4)
Gaussian Quadrature for finding [wi,xi] pairs (N=2,3,4,5)

Gaussian Quadrature Rule [wi, xi] pairs for N = 2

    N    wi           xi
    _    __    _________________

    2    1      0.577350269189626
    2    1     -0.577350269189626

Gaussian Quadrature Rule [wi, xi] pairs for N = 3

    N           wi                    xi
    _    _________________    __________________

    3    0.888888888888889                     0
    3    0.555555555555556     0.774596669241483
    3    0.555555555555556    -0.774596669241483

Gaussian Quadrature Rule [wi, xi] pairs for N = 4

    N           wi                    xi
    _    _________________    __________________

    4    0.652145154862546     0.339981043584856
    4    0.652145154862546    -0.339981043584856
    4    0.347854845137454     0.861136311594053
    4    0.347854845137454    -0.861136311594053

Gaussian Quadrature Rule [wi, xi] pairs for N = 5

    N           wi                    xi
    _    _________________    __________________

    5    0.568888888888889                     0
    5    0.478628670499366     0.538469310105683
    5    0.478628670499366    -0.538469310105683
    5    0.236926885056189     0.906179845938664
    5    0.236926885056189    -0.906179845938664

fx >>
```

<u>Question 3</u>

Part 1) To find the solutions/errors for the 3 Newton-Cotes methods, open “A3_Question3.m” and follow the instructions provided in the README file and Header Comment.

The methods used for each Newton-Cotes formulas involved the same process for storing the values in Question 1. However, I also added a summation array that would store the current value of the integral approximation. Afterwards, I would plot the summation-actual solution over N for each Newton-Cote formula.

- Specify the N values **and set “isNewtonCotes” to true** in the parameter.

Here are the following figures:

Figure A - Graph at N = 1024

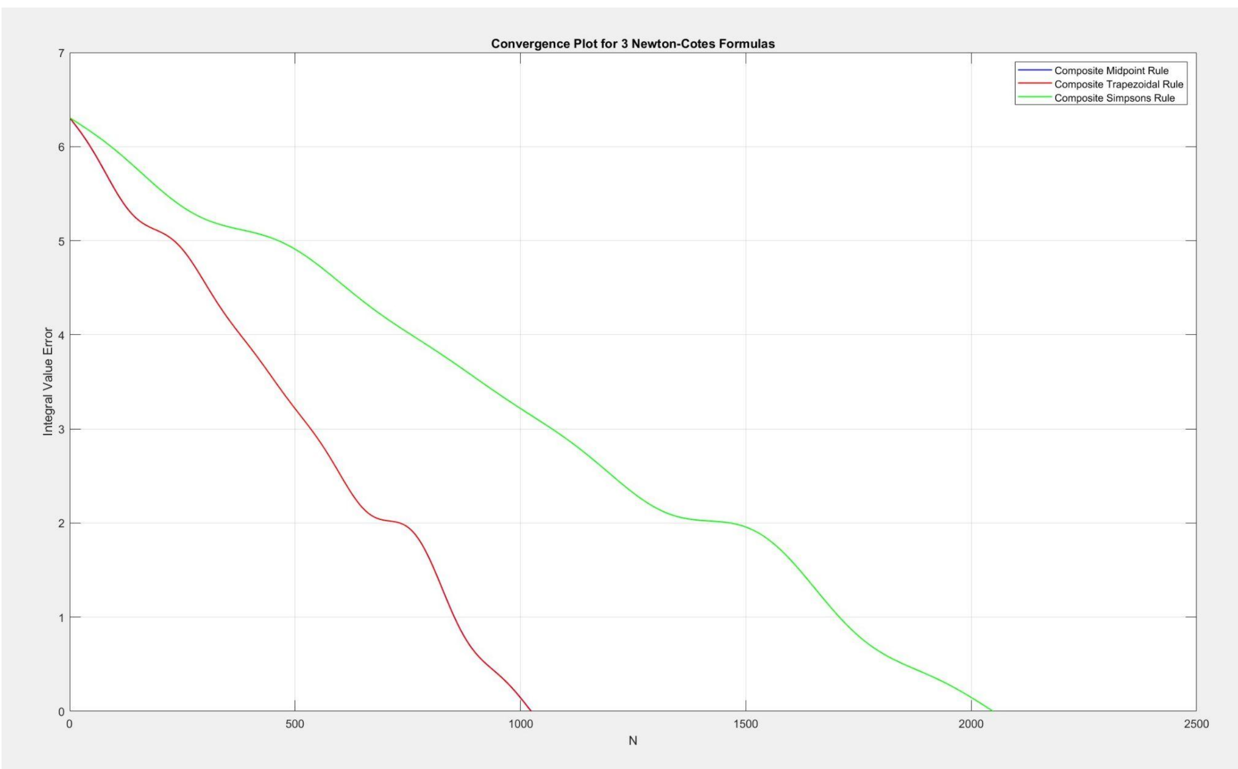


Figure B - Composite Midpoint and Trapezoidal Error

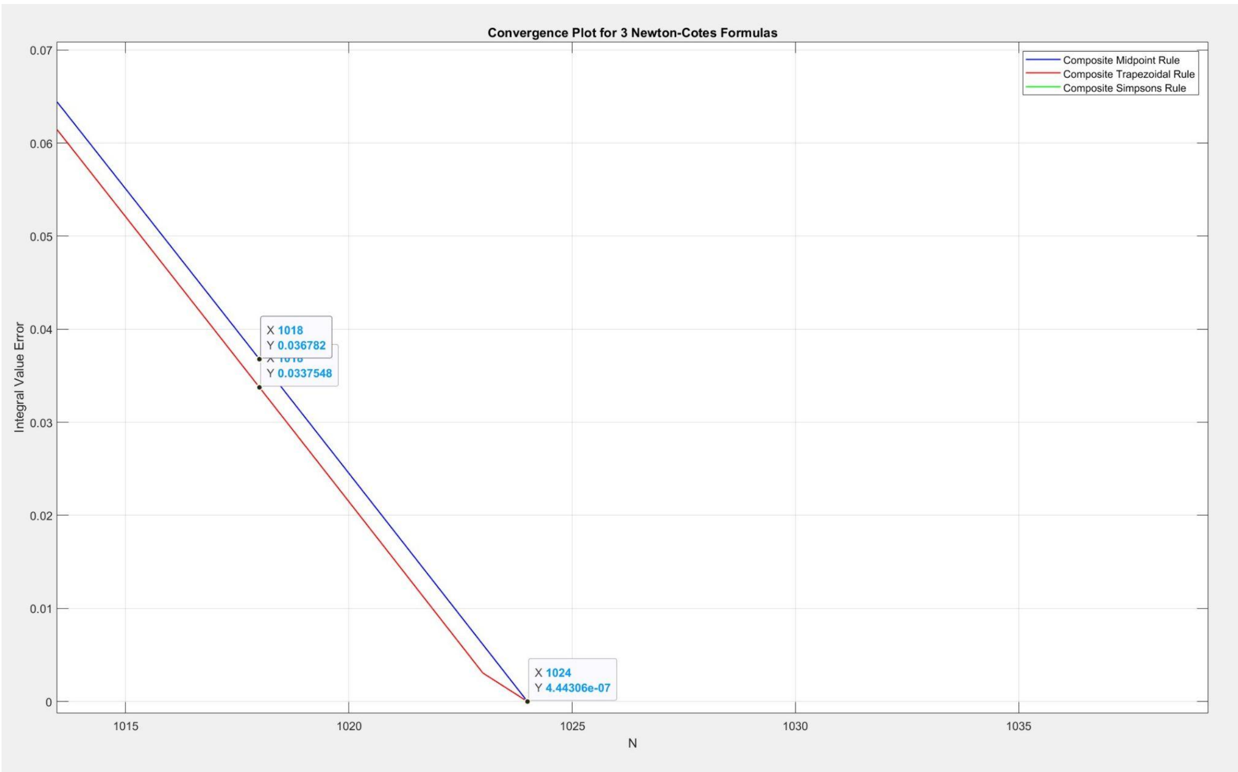


Figure C - Composite Simpson's Error

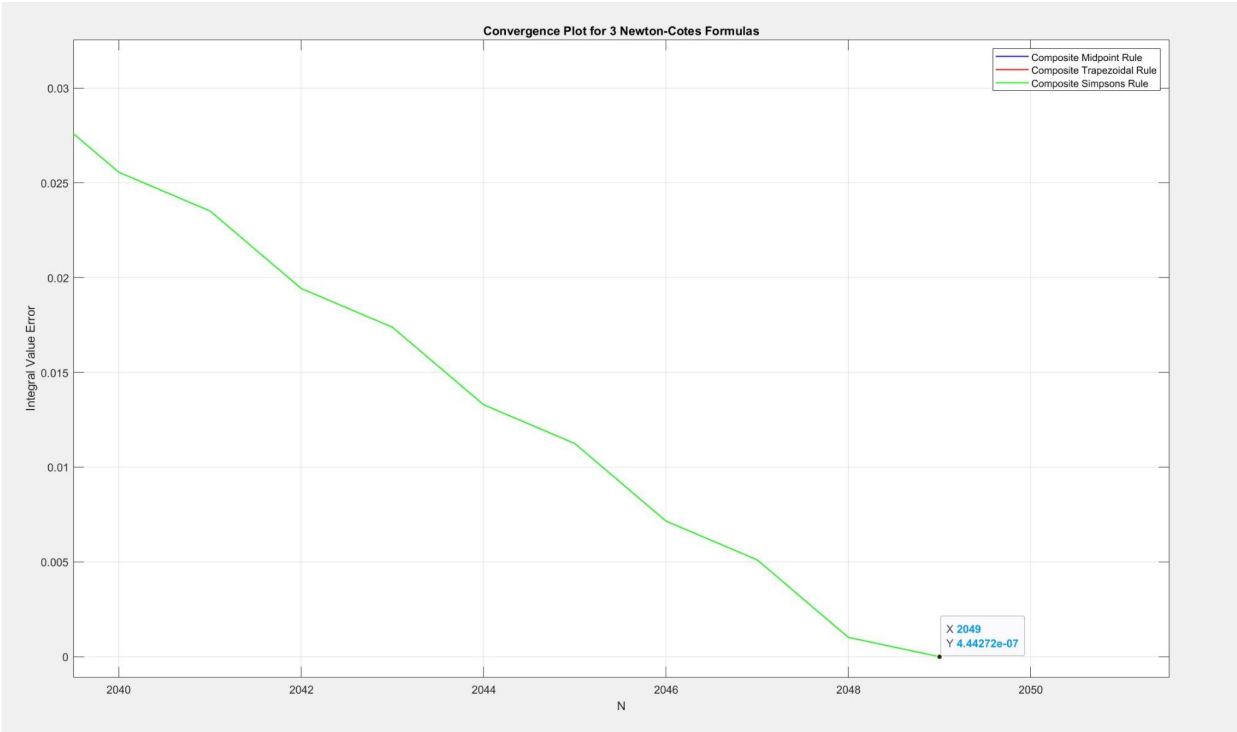
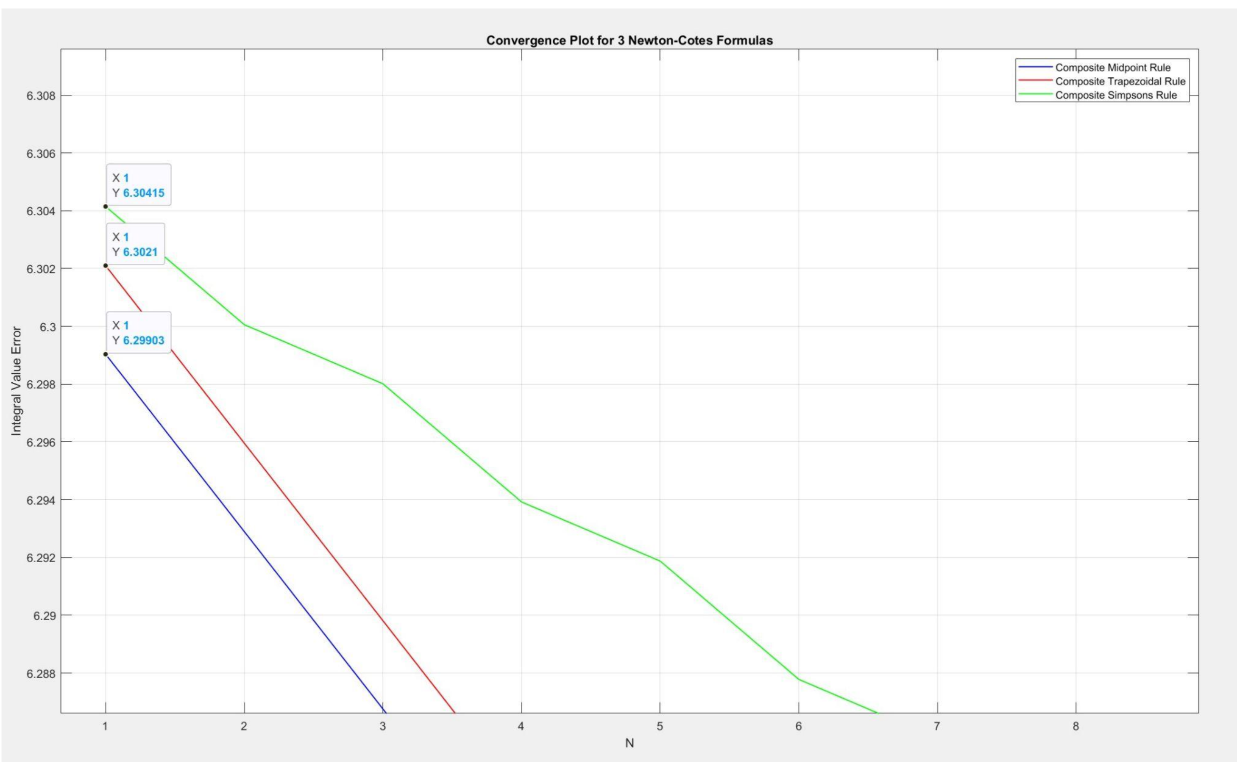


Figure D - Integral Values



Report: From the data shown in Figures B and C, Composite Midpoint Rule converges the fastest, despite Composite Trapezoidal Rule being a close second. This is because the theoretical errors, O(DeltaX^2), represent both functions properly since they are practically overlapping (as seen in Figure A). This also means that the Composite Simpson Rule's error, O(DeltaX^4), is also accurate because it converges the slowest among increasing N values since it has a worse Big-O computation.

Because of this information, it proves that the Composite Simpson's Rule is the most accurate because it has to iterate over double the amount the input N value to get to a closer approximation. The actual solution of the integral is: 6.305171. As shown in Figure D, the corresponding values for Composite Midpoint, Trapezoidal, and Simpson's Rule are: 6.29903, 6.3021, 6.30415 respectively.

Again, this proves that Composite Simpson's Rule is slower than the other computations, however, it's the most frequently accurate approximation of the actual solution to the integral.

Part 2) To find the results of the Gaussian Quadrature for N = 2,3,4,5, open "A3_Question3.m" and follow the instructions provided in the README file and Header Comment.

The method used to find the Gaussian Quadrature for N = 2,3,4,5 is the same process I used to store the [wi, xi] pairs but, again, I added a summation array that stores the integral approximation at some value N.

- No need to specify a proper N value. **Set "isNewtonCotes" to false** in the parameter.

This is the output:

Command Window

```
>> A3_Question3(1024, true)
>> A3_Question3(1024, true)
>> A3_Question3(1024, false)
Gaussian Quadrature
N == 2:    5.071054755798431

N == 3:    6.807774066063649

N == 4:    5.542404407274256

N == 5:    6.498331091711171
```

Report: As shown by the output, the Gaussian Quadrature struggles to find a proper approximation as it essentially “zig-zags” from farther to closer approximate values. To reiterate, the actual solution of the integral is: 6.305171. The closest it got to the actual solution was at N = 5, but it’s still off by approximately .20 values.

Even though it’s a high-order function, it doesn’t do well approximating this integral because it integrates over N = 5 points which means we have 10 unknowns. Therefore, we can only exactly integrate 9th degree polynomial functions; with this function being a 1st degree (linear) polynomial, it makes attaining that exact approximation very difficult.

To make it have better results, we could adjust the N size to only approximate to N = 1 instead of having to go up to N = 5. This is because, for this linear degree polynomial, we will only have 2 unknowns which is exactly the approximation we’ll need for linear polynomials. So, N = 1 integration will only be necessary.

Sources

https://www.mathworks.com/matlabcentral/answers/285761-how-to-reset-variables-before-each-iteration

https://www.mathworks.com/help/matlab/learn_matlab/matrices-and-arrays.html

https://www.mathworks.com/help/matlab/ref/sin.html

https://mathworld.wolfram.com/Legendre-GaussQuadrature.html

https://keisan.casio.com/exec/system/1280883022