Mazen Alotaibi

Table of Contents

Problem 1

```
clc
fprintf('---Trapezoidal Rule--- \n');
% a.
fprintf('\n1(i)\n');
fprintf('Function: exp(-x.^2) \n');
a=0;
b=2;
n0=2;
index_f=1; % exp(-x.^2)
[integral,difference,ratio]=trapezoidal(a,b,n0,index_f);
fprintf('n \tApproximation \tDifference \tRatio\n');
for i=1:length(integral),
   fprintf('%d\t%0.12f\t%0.5e\t%g
\n',n0*2^(i-1),integral(i),difference(i),ratio(i))
end
% This function (exp(-x.^2)) isn't a periodic function, the difference
% converges linearly, and the ratio converges to 4. (Unexpected
 behavior:
% the ratio started as 31.1208, but then it decreased in a linear
% convergence rate)
fprintf('\n1(ii)\n');
fprintf('Function: 1 ./(1+x.^2) \n');
a=0;
b=4;
n0=2;
index_f=2; % 1 ./(1+x.^2)
[integral,difference,ratio]=trapezoidal(a,b,n0,index_f);
fprintf('n \tApproximation \tDifference \tRatio\n');
for i=1:length(integral),
   fprintf('%d\t%0.12f\t%0.5e\t%g
\n',n0*2^(i-1),integral(i),difference(i),ratio(i))
end
% This function (1 ./(1+x.^2)) isn't a periodic function, the
 difference
% converges linearly, and the ratio converges to 4. Also, there was a
% weird behavior as the ratio bounced from 0 to 31.1208, then to
 -9.89664,
% then it started converging to 4. (As expected)
```

```
fprintf('\n1(iii)\n');
fprintf('Function: 1 ./(2+sin(x)) \n');
a=0;
b=(2*pi);
n0=2;
index_f=3; % 1 ./(2+sin(x))
[integral,difference,ratio]=trapezoidal(a,b,n0,index_f);
 fprintf('n \tApproximation \tDifference \tRatio\n');
 for i=1:length(integral),
    fprintf('%d\t%0.12f\t%0.5e\t%g
\n',n0*2^(i-1),integral(i),difference(i),ratio(i))
 end
% This function (1 ./(2+sin(x))) is a periodic function, which means
 the
% function converges faster. (As expected)
fprintf('\n1(iv)\n');
fprintf('Function: sqrt(x) \n');
a=0;
b=1;
n0=2;
index_f=4; % sqrt(x)
[integral,difference,ratio]=trapezoidal(a,b,n0,index_f);
 fprintf('n \tApproximation \tDifference \tRatio\n');
 for i=1:length(integral),
    fprintf('%d\t%0.12f\t%0.5e\t%g
\n',n0*2^(i-1),integral(i),difference(i),ratio(i))
 end
% This function (sqrt(x)) isn't a periodic function, the difference
% converges linearly, and the ratio converges to 2.8. (As expected)


% b. Trapezoidal Rule (degree of accuracy of one):
% When the function is a periodic function, then the Trapezoidal Rule
 will
% converge rapidly as when the number of section of the graph that are
% concaved up are similar to the number of section of the graph that
 are
% concaved down, which means the difference close to zero.
```

```
---Trapezoidal Rule---

1(i)
Function: exp(-x.^2)
n  Approximation  Difference  Ratio
2 0.877037260616 0.00000e+00 0
4 0.880618634125 3.58137e-03 0
8 0.881703791332 1.08516e-03 3.30033
16 0.881986245266 2.82454e-04 3.84189
32 0.882057557801 7.13125e-05 3.96079
64 0.882075429611 1.78718e-05 3.99022
128 0.882079900293 4.47068e-06 3.99756
256 0.882081018134 1.11784e-06 3.99939
512 0.882081297605 2.79471e-07 3.99985
```

```
1(ii)
Function: 1 ./(1+x.^2)
n  Approximation  Difference  Ratio
2 1.458823529412 0.00000e+00 0
4 1.329411764706 -1.29412e-01 0
8 1.325253402497 -4.15836e-03 31.1208
16 1.325673581733 4.20179e-04 -9.89664
32 1.325781625682 1.08044e-04 3.88897
64 1.325808653076 2.70274e-05 3.99757
128 1.325815410952 6.75788e-06 3.99939
256 1.325817100485 1.68953e-06 3.99985
512 1.325817522872 4.22387e-07 3.99996

1(iii)
Function: 1 ./(2+sin(x))
n  Approximation  Difference  Ratio
2 3.141592653590 0.00000e+00 0
4 3.665191429188 5.23599e-01 0
8 3.627791516645 -3.73999e-02 -14
16 3.627598733591 -1.92783e-04 194
32 3.627598728468 -5.12258e-09 37634
64 3.627598728468 0.00000e+00 -Inf
128 3.627598728468 -8.88178e-16 -0
256 3.627598728468 8.88178e-16 -1
512 3.627598728468 2.22045e-15 0.4

1(iv)
Function: sqrt(x)
n  Approximation  Difference  Ratio
2 0.603553390593 0.00000e+00 0
4 0.643283046243 3.97297e-02 0
8 0.658130221624 1.48472e-02 2.67591
16 0.663581196877 5.45098e-03 2.72376
32 0.665558936279 1.97774e-03 2.75616
64 0.666270811379 7.11875e-04 2.77821
128 0.666525657297 2.54846e-04 2.79335
256 0.666616548977 9.08917e-05 2.80384
512 0.666648881550 3.23326e-05 2.81115
```

Problem 2

```
clc
fprintf('---Simpsons Rule---\n');
% a.
fprintf('\n2(i)\n');
fprintf('Function: exp(-x.^2) \n');
a=0;
b=2;
n0=2;
index_f=1; % exp(-x.^2)
[integral,difference,ratio]=simpson(a,b,n0,index_f);
 fprintf('n \tApproximation \tDifference \tRatio\n');
 for i=1:length(integral),
```

```
    fprintf('%d\t%0.12f\t%0.5e\t%g
\n',n0*2^(i-1),integral(i),difference(i),ratio(i))
 end

% When comparing the difference of the Trapezoidal Rule and the
 Simpson
% Rule at n = 512, it seems that Trapezoidal Rule's difference was
% 2.79471e-07 compared to Simpsons Rule's difference was 1.42157e-11,
 which
% means that Simpson Rule is more accurate than Trapezoidal Rule. In
% addition, when comparing the ratio of both, it seems that
 Trapezoidal
% Rule's ratio converged to 4 compared to Simpson Rule's ratio
 converged
% to 16.
fprintf('\n2(ii)\n');
fprintf('Function: 1 ./(1+x.^2) \n');
a=0;
b=4;
n0=2;
index_f=2; % 1 ./(1+x.^2)
[integral,difference,ratio]=simpson(a,b,n0,index_f);
 fprintf('n \tApproximation \tDifference \tRatio\n');
 for i=1:length(integral),
    fprintf('%d\t%0.12f\t%0.5e\t%g
\n',n0*2^(i-1),integral(i),difference(i),ratio(i))
 end

% When comparing the difference of the Trapezoidal Rule and the
 Simpson
% Rule at n = 512, it seems that Trapezoidal Rule's difference was
% 4.22387e-07 compared to Simpson Rule's difference was 5.35216e-12,
 which
% means that Simpson Rule is more accurate than Trapezoidal Rule. In
% addition, when comparing the ratio of both, it seems that
 Trapezoidal
% Rule's ratio converged to 4 compared to Simpson Rule's ratio
 converged
% to 16.
fprintf('\n2(iii)\n');
fprintf('Function: 1 ./(2+sin(x)) \n');
a=0;
b=(2*pi);
n0=2;
index_f=3; % 1 ./(2+sin(x))
[integral,difference,ratio]=simpson(a,b,n0,index_f);
 fprintf('n \tApproximation \tDifference \tRatio\n');
 for i=1:length(integral),
    fprintf('%d\t%0.12f\t%0.5e\t%g
\n',n0*2^(i-1),integral(i),difference(i),ratio(i))
 end

% When comparing the difference of the Trapezoidal Rule and the
 Simpson
```

```
% Rule at n = 512, it seems that Trapezoidal Rule's difference was
% 2.22045e-15 compared to Simpsons Rule's difference was -2.66454e-15,
 which
% means that both of their behaviors are similar. In addition, when
% comparing the ratio of both, it seems they both also behaved
 similarly as
% both of their ratios coverge to 0.
fprintf('\n2(iv)\n');
fprintf('Function: sqrt(x) \n');
a=0;
b=1;
n0=2;
index_f=4; % sqrt(x)
[integral,difference,ratio]=simpson(a,b,n0,index_f);
 fprintf('n \tApproximation \tDifference \tRatio\n');
 for i=1:length(integral),
    fprintf('%d\t%0.12f\t%0.5e\t%g
\n',n0*2^(i-1),integral(i),difference(i),ratio(i))
 end

% When comparing the difference of the Trapezoidal Rule and the
 Simpson
% Rule at n = 512, it seems that Trapezoidal Rule's difference was
% 3.23326e-05 compared to Simpsons Rule's difference was 1.28129e-05,
 which
% means that both of their behaviors are similar In addition, when
% comparing the ratio of both, it seems that both of their ratio
 converged
% to 2.8.

% b. Simpsons Rule (degree of accuracy of three)
```

---Simpsons Rule---

2(i)
Function: exp(-x.^2)
n Approximation Difference Ratio
2 0.829944467858 0.00000e+00 0
4 0.881812425294 5.18680e-02 0
8 0.882065510401 2.53085e-04 204.943
16 0.882080396577 1.48862e-05 17.0014
32 0.882081328646 9.32069e-07 15.9711
64 0.882081386881 5.82343e-08 16.0055
128 0.882081390520 3.63916e-09 16.0021
256 0.882081390747 2.27440e-10 16.0006
512 0.882081390761 1.42157e-11 15.9992

2(ii)
Function: 1 ./(1+x.^2)
n Approximation Difference Ratio
2 1.239215686275 0.00000e+00 0
4 1.286274509804 4.70588e-02 0
8 1.323867281761 3.75928e-02 1.25181
16 1.325813641478 1.94636e-03 19.3144

```
32 1.325817640332 3.99885e-06 486.729
64 1.325817662207 2.18759e-08 182.797
128 1.325817663577 1.36926e-09 15.9764
256 1.325817663662 8.56231e-11 15.9918
512 1.325817663668 5.35216e-12 15.9978

2(iii)
Function: 1 ./(2+sin(x))
n  Approximation  Difference  Ratio
2 3.141592653590 0.00000e+00 0
4 3.839724354388 6.98132e-01 0
8 3.615324879131 -2.24399e-01 -3.11111
16 3.627534472573 1.22096e-02 -18.3789
32 3.627598726761 6.42542e-05 190.02
64 3.627598728468 1.70753e-09 37630
128 3.627598728468 8.88178e-16 1.9225e+06
256 3.627598728468 0.00000e+00 Inf
512 3.627598728468 -2.66454e-15 -0

2(iv)
Function: sqrt(x)
n  Approximation  Difference  Ratio
2 0.638071187458 0.00000e+00 0
4 0.656526264793 1.84551e-02 0
8 0.663079280085 6.55302e-03 2.81627
16 0.665398188628 2.31891e-03 2.82591
32 0.666218182746 8.19994e-04 2.82796
64 0.666508103078 2.89920e-04 2.82834
128 0.666610605936 1.02503e-04 2.82841
256 0.666646846203 3.62403e-05 2.82842
512 0.666659659074 1.28129e-05 2.82843
```

Problem 3

```
clc

% a. the statement, the number of subdivisions reuired to achieve
% an accuracy of 10e-10 is at least n =160, is right as it has been
 proven
% in my results as when n = 256, the difference is equal to
 2.27440e-10,
% which is less than 10e-10. Hence, my results agree with this
 prediction.

% b. the statement, the number of subdivisions reuired to achieve
% an accuracy of 10e-12 is at least n =396, is right as it has been
 proven
% in my results as when n = 512, the difference is equal to
 5.35216e-12,
% which is less than 10e-12. Hence, my results agree with this
 prediction.
```

Problem 4

```
clc
fprintf('---Gaussian Quadrature Rule--- \n');
% a.
fprintf('\n3(i)\n');
fprintf('Function: exp(-x.^2) \n');
a=0;
b=2;
n0=2;
index_f=1; % exp(-x.^2)
[integral,difference,ratio]=gausstable(a,b,n0,index_f);
 fprintf('n \tApproximation \tDifference \tRatio\n');
 for i=1:length(integral),
    fprintf('%d\t%0.12f\t%0.5e\t%g
\n',n0*2^(i-1),integral(i),difference(i),ratio(i))
 end

% When comparing the difference of the Trapezoidal Rule, the Simpson
% Rule, and  Gaussian Quadrature Rule at n = 512, it seems that
% Trapezoidal Rule's difference was 2.79471e-07, Simpson Rule's
% difference was 1.42157e-11, and Gaussian Quadrature Rule's
 difference
% was -2.22045e-16, which means that Gaussian Quadrature Rule was more
% accurate than Simpson Rule and Trapezoidal Rule. In addition, when
% comparing the ratio of all of them, it seems that Trapezoidal
% Rule's ratio converged to 4, Simpson Rule's ratio converged
% to 16, and Gaussian Quadrature Rule's ratio converged to 0.
fprintf('\n3(ii)\n');
fprintf('Function: 1 ./(1+x.^2) \n');
a=0;
b=4;
n0=2;
index_f=2; % 1 ./(1+x.^2)
[integral,difference,ratio]=gausstable(a,b,n0,index_f);
 fprintf('n \tApproximation \tDifference \tRatio\n');
 for i=1:length(integral),
    fprintf('%d\t%0.12f\t%0.5e\t%g
\n',n0*2^(i-1),integral(i),difference(i),ratio(i))
 end


% When comparing the difference of the Trapezoidal Rule, the Simpson
% Rule, and  Gaussian Quadrature Rule at n = 512, it seems that
% Trapezoidal Rule's difference was 4.22387e-07, Simpson Rule's
% difference was 5.35216e-12, and Gaussian Quadrature Rule's
 difference
% was -4.44089e-16, which means that Gaussian Quadrature Rule was more
% accurate than Simpson Rule and Trapezoidal Rule. In addition, when
% comparing the ratio of all of them, it seems that Trapezoidal
% Rule's ratio converged to 4, Simpson Rule's ratio converged
% to 16, and Gaussian Quadrature Rule's ratio converged to 0.
fprintf('\n3(iii)\n');
```

```
fprintf('Function: 1 ./(2+sin(x)) \n');
a=0;
b=(2*pi);
n0=2;
index_f=3; % 1 ./(2+sin(x))
[integral,difference,ratio]=gausstable(a,b,n0,index_f);
 fprintf('n \tApproximation \tDifference \tRatio\n');
 for i=1:length(integral),
    fprintf('%d\t%0.12f\t%0.5e\t%g
\n',n0*2^(i-1),integral(i),difference(i),ratio(i))
 end

% When comparing the difference of the Trapezoidal Rule, the Simpson
% Rule, and  Gaussian Quadrature Rule at n = 512, it seems that
% Trapezoidal Rule's difference was 2.22045e-15, Simpson Rule's
% difference was -2.66454e-15, and Gaussian Quadrature Rule's
 difference
% was -1.33227e-15, which means that all of their behaviors are
 similar.
% In addition, when comparing the ratio of all of them, it seems that
% all for their ratios also behaved similarly as their ratio coverge
 to 0.
fprintf('\n3(iv)\n');
fprintf('Function: sqrt(x) \n');
a=0;
b=1;
n0=2;
index_f=4; % sqrt(x)
[integral,difference,ratio]=gausstable(a,b,n0,index_f);
 fprintf('n \tApproximation \tDifference \tRatio\n');
 for i=1:length(integral),
    fprintf('%d\t%0.12f\t%0.5e\t%g
\n',n0*2^(i-1),integral(i),difference(i),ratio(i))
 end

% When comparing the difference of the Trapezoidal Rule, the Simpson
% Rule, and  Gaussian Quadrature Rule at n = 512, it seems that
% Trapezoidal Rule's difference was 3.23326e-05, Simpson Rule's
% difference was 1.28129e-05, and Gaussian Quadrature Rule's
 difference
% was -5.33603e-09, which means that Gaussian Quadrature Rule was more
% accurate than Simpson Rule and Trapezoidal Rule. In addition, when
% comparing the ratio of all of them, it seems that Trapezoidal
% Rule's ratio converged to 2.8, Simpson Rule's ratio converged
% to 2.8, and Gaussian Quadrature Rule's ratio converged to 7.9.

% b. Gaussian Quadrature Rule (degree of accuracy is the largest
 degree
% possible).

---Gaussian Quadrature Rule---

3(i)
Function: exp(-x.^2)
```

```
n  Approximation  Difference  Ratio
2 0.919486116641 0.00000e+00 0
4 0.882229095933 -3.72570e-02 0
8 0.882081390420 -1.47706e-04 252.239
16 0.882081390762 3.42522e-10 -431229
32 0.882081390762 -6.66134e-16 -514194
64 0.882081390762 -4.44089e-16 1.5
128 0.882081390762 0.00000e+00 -Inf
256 0.882081390762 0.00000e+00 NaN
512 0.882081390762 -2.22045e-16 -0

3(ii)
Function: 1 ./(1+x.^2)
n  Approximation  Difference  Ratio
2 1.349112426036 0.00000e+00 0
4 1.327713222795 -2.13992e-02 0
8 1.325838869084 -1.87435e-03 11.4168
16 1.325817663720 -2.12054e-05 88.3905
32 1.325817663668 -5.23941e-11 404728
64 1.325817663668 -8.88178e-16 58990.5
128 1.325817663668 0.00000e+00 -Inf
256 1.325817663668 0.00000e+00 NaN
512 1.325817663668 -4.44089e-16 -0

3(iii)
Function: 1 ./(2+sin(x))
n  Approximation  Difference  Ratio
2 4.109480962483 0.00000e+00 0
4 3.679381279962 -4.30100e-01 0
8 3.628039679738 -5.13416e-02 8.37722
16 3.627600902211 -4.38778e-04 117.011
32 3.627598728468 -2.17374e-06 201.853
64 3.627598728468 6.79012e-13 -3.20133e+06
128 3.627598728468 4.44089e-16 1529
256 3.627598728468 0.00000e+00 Inf
512 3.627598728468 -1.33227e-15 -0

3(iv)
Function: sqrt(x)
n  Approximation  Difference  Ratio
2 0.673887338679 0.00000e+00 0
4 0.667827645375 -6.05969e-03 0
8 0.666835580100 -9.92065e-04 6.10816
16 0.666689631499 -1.45949e-04 6.79736
32 0.666669667368 -1.99641e-05 7.31054
64 0.666667050398 -2.61697e-06 7.62872
128 0.666666715190 -3.35208e-07 7.80701
256 0.666666672768 -4.24229e-08 7.90157
512 0.666666667432 -5.33603e-09 7.95028
```

Published with MATLAB® R2017b