Lab5

Table of Contents

Problem 1

Trapizoidal

a.1 'exp(-x^2)'

```
a=0;
b=2;
n0=2;
f='exp(-x^2)';
[inT,diT,raT]=trapezoidal(a,b,n0,f);
fprintf(strcat('\nFunction: ', f))
fprintf('\n \tIntegral \tDifference \tRatio\n')
for i=1:length(inT),
fprintf('%d\t%0.12f\t%0.5e\t%g\n', n0*2^(i-1),inT(i),diT(i),raT(i))
end
```

```
Function:exp(-x^2)
  Integral  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
```

b.1 Using Trapizoidal on this integral works reasonably well because on the interval being integrated e^(-x^2) is relativly linear. Trapizoidal does a very good job of estimation the area under the curve of a near linear functions.

a.2 '1/(1 + x^2)'

```
a=0;
b=4;
n0=2;
f='1/(1 + x^2)';
[inT,diT,raT]=trapezoidal(a,b,n0,f);
```

```
fprintf(strcat('\nFunction: ', f))
fprintf('\n \tIntegral \tDifference \tRatio\n')
for i=1:length(inT),
fprintf('%d\t%0.12f\t%0.5e\t%g\n', n0*2^(i-1),inT(i),diT(i),raT(i))
end
```

```
Function:1/(1 + x^2)
  Integral  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
```

b.2 The function 1/(1 + x^2) has a second derivative very near zero on the interval [0, 4]. This is why Trapizoidal does very well when approximating this function's integral.

a.3 '1/(2 + sin(x))'

```
a=0;
b=2*pi;
n0=2;
f='1/(2 + sin(x))';
fprintf(strcat('\nFunction: ', f))
[inT,diT,raT]=trapezoidal(a,b,n0,f);
fprintf('\n \tIntegral \tDifference \tRatio\n')
for i=1:length(inT),
fprintf('%d\t%0.12f\t%0.5e\t%g\n', n0*2^(i-1),inT(i),diT(i),raT(i))
end
```

```
Function:1/(2 + sin(x))
  Integral  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
```

b.3 The function 1/(2 + sin(x)) does very poorly when it's approximated using Trapizoidal rule. This poor behavior is because on the interval [0, 2pi] the function is nearly quadratic in some points this causes a lot of error at those points. On other points it's curve the function is nearly linear and trapizoidal does okay.

a.4 'x^(1/2)'

```
a=0;
```

```
b=1;
n0=2;
f='x^(1/2)';
[inT,diT,raT]=trapezoidal(a,b,n0,f);
fprintf(strcat('\nFunction: ', f))
fprintf('\n \tIntegral \tDifference \tRatio\n')
for i=1:length(inT),
fprintf('%d\t%0.12f\t%0.5e\t%g\n', n0*2^(i-1),inT(i),diT(i),raT(i))
end
```

```
Function:x^(1/2)
  Integral  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
```

b.4 The function x^(1/2) is consistently not linear and so doesn't have the irradic behavior of the function of a.3, but still has a lower ratio of errors than the function in a.2

Problem 2

Simpson

a.1 'exp(-x^2)'

```
fprintf('\n\n******************************\n* Approximation with Simpson *\n*****

a=0;
b=2;
n0=2;
f='exp(-x^2)';
[inS,diS,raS]=simpson(a,b,n0,f);
fprintf(strcat('\nFunction: ', f))
fprintf('\n \tIntegral \tDifference \tRatio\n')
for i=1:length(inS),
fprintf('%d\t%0.12f\t%0.5e\t%g\n', n0*2^(i-1),inS(i),diS(i),raS(i))
end
```

```
******************************
* Approximation with Simpson *
******************************

Function:exp(-x^2)
  Integral  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
```

b.1 f is nearly a degree 2 polynomial on each interval so Simpson does a really good job of approximating it. It seems that only after n >=8 does simpson get more accurate results than trapizoidal.

a.2 '1/(1 + x^2)'

```
a=0;
b=4;
n0=2;
f='1/(1 + x^2)';
[inS,diS,raS]=simpson(a,b,n0,f);
fprintf(strcat('\nFunction: ', f))
fprintf('\n \tIntegral \tDifference \tRatio\n')
for i=1:length(inS),
fprintf('%d\t%0.12f\t%0.5e\t%g\n', n0*2^(i-1),inS(i),diS(i),raS(i))
end
```

```
Function:1/(1 + x^2)
  Integral  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
```

b.2 The function '1/(1 + x^2)' is nearly linear on the interval [0, 4] and so simpson does a very good job at approximating it and is more accurate compared to Trapizoidal for all calculated values of n.

a.3 '1/(2 + sin(x))'

```
a=0;
b=2*pi;
n0=2;
f='1/(2 + sin(x))';
fprintf(strcat('\nFunction: ', f))
[inS,diS,raS]=simpson(a,b,n0,f);
fprintf('\n \tIntegral \tDifference \tRatio\n')
for i=1:length(inS),
fprintf('%d\t%0.12f\t%0.5e\t%g\n', n0*2^(i-1),inS(i),diS(i),raS(i))
end
```

```
Function:1/(2 + sin(x))
  Integral  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
```

b.3 Trapazoidal does a better job at approximating this function than Simpson does. This could be happening because the Trapizoidal's linear estimates could guess the actual value better than Simpsons quadratic estimates.

a.4 'x^(1/2)'

```
a=0;
b=1;
n0=2;
f='x^(1/2)';
[inS,diS,raS]=simpson(a,b,n0,f);
fprintf(strcat('\nFunction: ', f))
fprintf('\n \tIntegral \tDifference \tRatio\n')
for i=1:length(inS),
fprintf('%d\t%0.12f\t%0.5e\t%g\n', n0*2^(i-1),inS(i),diS(i),raS(i))
end
```

```
Function:x^(1/2)
  Integral  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
```

b.4 Simpson can approximate quadratic functions with no error and since x^(1/2) is nearly quadratic on the interval [0, 1] it integrates the function much better than Trapizoidal did.

Problem 3

Just by looking at the generated table for the first integral, you can see that Simpson's Rule achieves ten digits of accuracy somewhere between n = 128 and n = 265. This suggests that the asymtotic error formula is correct. Similarly, the error shown in the table for the second integral is between twelve digits of accuracy between n = 256 and n = 515. This also agrees with the asymtotic error formula.

```
*******************************************
* Approximation with Gaussian Quadrature *
*******************************************
```

Problem 4

Gaussian Quadrature (GQ)

a.1 'exp(-x^2)'

```
fprintf('\n\n*******************************************\n* Approximation with Gaus

a=0;
b=2;
n0=2;
f='exp(-x^2)';
[inG,diG,raG]=gausstable(a,b,n0,f);
fprintf(strcat('\nFunction: ', f))
fprintf('\n \tIntegral \tDifference \tRatio\n')
for i=1:length(inG),
fprintf('%d\t%0.12f\t%0.5e\t%g\n', n0*2^(i-1),inG(i),diG(i),raG(i))
end
```

```
Function:exp(-x^2)
  Integral  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 1.55431e-15 220369
64 0.882081390762 8.88178e-16 1.75
128 0.882081390762 1.11022e-16 8
256 0.882081390762 1.11022e-15 0.1
512 0.882081390762 7.77156e-16 1.42857
```

b.1 GQ does a very good job at approximating this function but both Simpson and Trapizoidal rules also do a very good job. Simpson and Trapizoidal are cheaper to calculate. If you have to generate the GQ weights from scratch.

```
Function:1/(1 + x^2)
  Integral  Difference  Ratio
```

a.2 '1/(1 + x^2)'

```
a=0;
b=4;
n0=2;
f='1/(1 + x^2)';
[inG,diG,raG]=gausstable(a,b,n0,f);
fprintf(strcat('\nFunction: ', f))
fprintf('\n \tIntegral \tDifference \tRatio\n')
for i=1:length(inG),
```

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

```
        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 5.23950e-11 404721
        64 1.325817663668 8.88178e-16 58991.5
        128 1.325817663668 0.00000e+00 Inf
        256 1.325817663668 1.33227e-15 0
        512 1.325817663668 8.88178e-16 1.5
```

b.2 This function is also very well approximated by Simpson and Trapizoidal. Simpson does do a better job at approximating it's integral.

```
        Function:1/(2 + sin(x))
          Integral  Difference  Ratio
```

a.3 '1/(2 + sin(x))'

```
a=0;
b=2*pi;
n0=2;
f='1/(2 + sin(x))';
fprintf(strcat('\nFunction: ', f))
[inG,diG,raG]=gausstable(a,b,n0,f);
fprintf('\n \tIntegral \tDifference \tRatio\n')
for i=1:length(inG),
fprintf('%d\t%0.12f\t%0.5e\t%g\n', n0*2^(i-1),inG(i),diG(i),raG(i))
end
```

```
        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 6.84341e-13 3.1764e+06
        128 3.627598728468 3.10862e-15 220.143
        256 3.627598728468 2.22045e-15 1.4
        512 3.627598728468 1.28786e-14 0.172414
```

b.3 Surprisingly, this function is still best approximated by Trapazoidal. GQ does do a better job at approximation than Simpson.

```
        Function:x^(1/2)
          Integral  Difference  Ratio
```

a.4 'x^(1/2)'

```
a=0;
b=1;
n0=2;
f='x^(1/2)';
```

```
[inG,diG,raG]=gausstable(a,b,n0,f);
fprintf(strcat('\nFunction: ', f))
fprintf('\n \tIntegral \tDifference \tRatio\n')
for i=1:length(inG),
fprintf('%d\t%0.12f\t%0.5e\t%g\n', n0*2^(i-1),inG(i),diG(i),raG(i))
end
```

```
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 7.95029
```

b.4 GQ approximates this function the best out of Trapizoidal and Simpson. The GQ reduces it's error better than Trapazoidal and Simpson.

Published with MATLAB® 7.13