Chapter 4: Numerical Integration Homework

Joseph Sepich

Contents

1 Problem 1

Data table for $f(x) = e^{-x}, x \in [0, 0.8]$

x	0.0	0.2	0.4	0.6	0.8
f(x)	1	0.818731	0.670320	0.548812	0.449329

1.1 a. Write out the trapezoid rule and compute with 6 digits

Recall the trapezoid rule:

$$T(f;h) = h[\frac{1}{2}f(x_0) + \Sigma f(x_i) + \frac{1}{2}f(x_n)]$$
$$T(e^{-x};0.2) = 0.2[\frac{1}{2}(1) + \frac{1}{2}0.4493 + 0.8187 + 0.6703 + 0.5488]$$
$$T(e^{-x};0.2) = 0.552505$$

1.2 b. Write out Simpson's rule and compute with 6 digits

Recall Simpson's rule:

$$S(f;h) = \frac{h}{3}[f(x_0) + 4\Sigma f(x_{2i-1}) + 2\Sigma f(x_{2i}) + f(x_{2n})]$$
$$S(f;h) = 0.550676$$

1.3 c. What is exact value of integral? What is each absolute error? Which method is better?

Actual value is:

$$1 - e^{-\frac{4}{5}} = 0.550671$$

Abs error of Trapezoid is $0.552505 - 0.550671 = 0.001834$, while Simpson's rule had $0.550676 - 0.550671 = 0.00005$. Simpson's method clearly worked better in terms of error.

1.4 Given the error formula for each rule, how many points would each method need for an error bound of 10^-4?

Trapezoid Rule:

$$-\frac{0.8}{12}h^2e^{-0} \leq 10^{-4}$$
$$h^2 \leq 15 * 10^{-4}$$
$$h \leq 0.038730$$
$$n \geq 20.66$$

This requires at least 22 points. Simpson's Rule:

$$-\frac{0.8}{180}h^4 \leq 10^{-4}$$
$$h^4 \leq 225 * 10^{-4}$$
$$h \leq 0.38730$$
$$n \geq 1.033$$

This methods requires at least 5 points. Much less!

2 Problem 2, Simpson's Rule

2.1 a. Calculate the given function on [0,1]

Recall the rule:

$$S(f;h) = \frac{h}{3}[f(x_0) + 4\Sigma f(x_{2i-1}) + 2\Sigma f(x_{2i}) + f(x_{2n})]$$
$$S(f;h) = 0.48$$

2.2 b. Tolerance is 10^{-6}. How many points do you need?

$$-\frac{1}{180}h^4(-1.25) \leq 10^{-6}$$
$$h^4 \leq 144 * 10^{-6}$$
$$h \leq 0.1095445$$
$$n \geq 9.129$$

For an error tolerance of 10^{-6}, Simpson's rule would need at least 20 points.

3 Problem 3, Trapezoid Rule and Romberg

3.1 a. Compute the trapezoid rule for the integral for n = 1,2

$$f(x) = 3x^2, [a,b] = [-1,1]$$

Recall the rule:

$$T(f;h) = h[\frac{1}{2}f(x_0) + \Sigma f(x_i) + \frac{1}{2}f(x_n)]$$

n = 1 evaluates to 6

x	-1	1
f(x)	3	3

n = 2 evalutes to 3

x	-1	0	1
f(x)	3	0	3

3.2 b. Evaluate with Romberg until you get the exact value.

The exact value of x^3 evaluated at [-1,1] is 2.

With n = 4 we now evalute to 2.25.

x	-1	-0.5	0	0.5	1
f(x)	3	0.75	0	0.75	3

With n = 8 we then yield 2.0625. With n = 16 we then yield 2.015625. With n = 32 we then yield 2.003906, so we are getting closer, but we are not getting the exact value.

3.3 c.

w = 1 and a = 0.577350 (or $1/\sqrt{3}$) works for any polynomial of degree 3 or less. We derived this in lecture.

4 Problem 4

Trapezoid Function:

```
function v=trapezoid(fun,a,b,n)
%TRAPEZOID numerical integration by trapezoid rule
h=(b-a)/n;
xi=a:h:b;
v = h*(0.5*feval(fun,xi(1))+sum(feval(fun,xi(2:1:end-1)))+0.5*feval(fun,(xi(end))));
end
```

Script:

```
n = [4 8 16 32 64 128];
actual = 0.550676;
absErr = zeros(1,length(n));

for i = 1:length(n)
    % func is e^-x
   val = trapezoid('func',0,0.8,n(i));
   absErr(i) = val - actual;
   disp(val)
end

figure
loglog(n, absErr)
title('Absolute Error in trapezoid rule')
xlabel('Number of segments')
ylabel('Absolute Error')
```

And figure 1 is the plot of the error:

When n is doubled the error value appears to cut by a factor of 4 times.

5 Problem 5

Simpson Function:

```
function v=Simpson(fun,a,b,n)
%SIMPSON numerical integration using simpson's rule
    h=(b-a)/n; xi=a:h:b;
    v= h/3*(feval(fun,xi(1))+2*sum(feval(fun,xi(3:2:end-2)))+4*sum(feval(fun,xi(2:2:end)))+feval(fun,xi
end
```

Script:

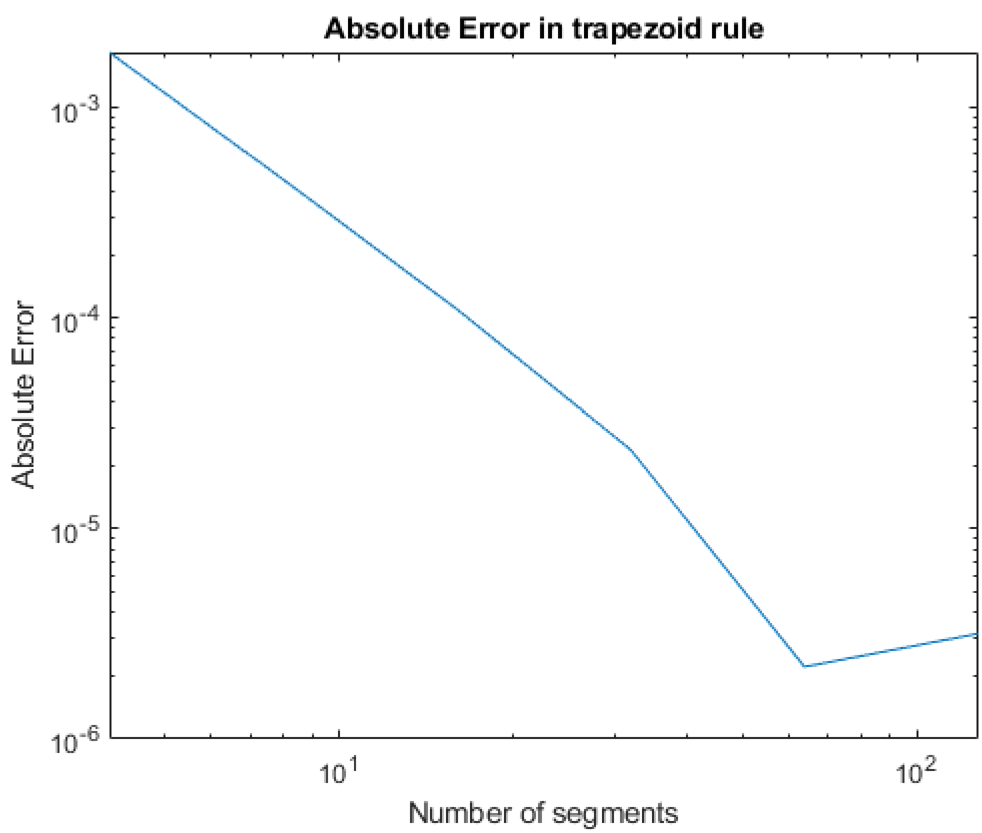
Absolute Error in trapezoid rule
Absolute Error
Number of segments
10^-3
10^-4
10^-5
10^-6
10^1
10^2

Figure 1:

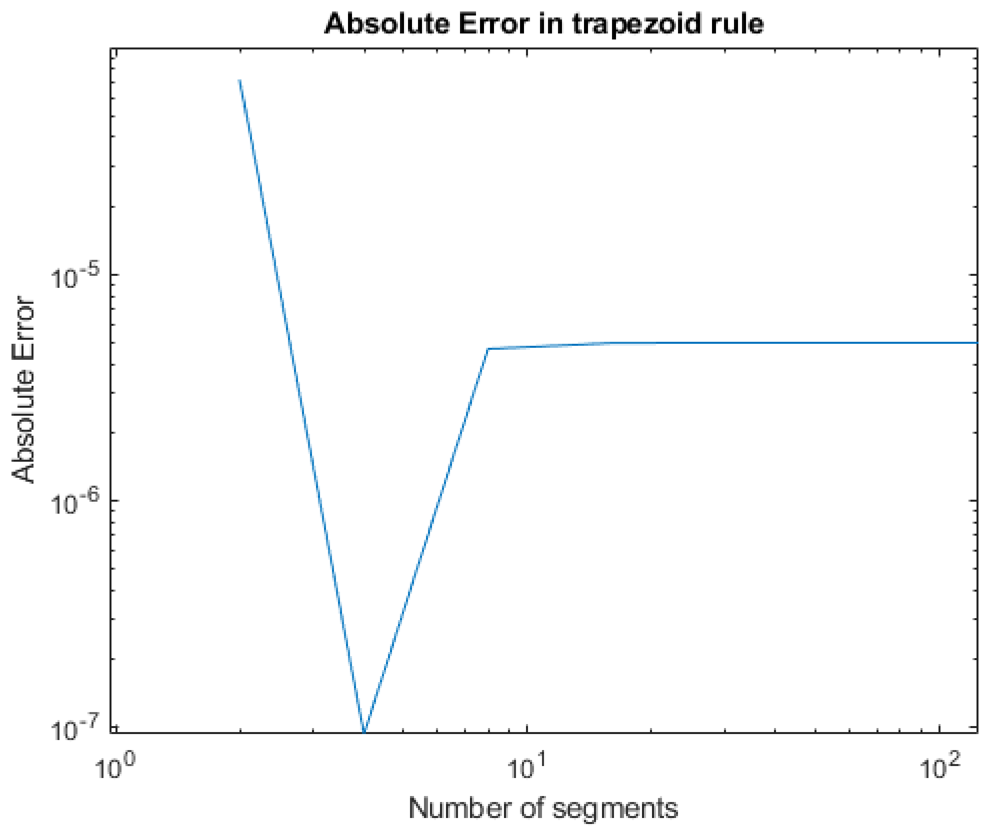

Figure 2:

```
n = [2 4 8 16 32 128];
actual = 0.550676;
absErr = zeros(1,length(n));

for i = 1:length(n)
    % func is e^-x
   val = Simpson('func',0,0.8,n(i));
   absErr(i) = abs(val - actual);
   disp(val)
end

figure
loglog(n, absErr)
title('Absolute Error in trapezoid rule')
xlabel('Number of segments')
ylabel('Absolute Error')
```

When n is doubled here, at least from n = 2 to n = 4, the error is decreased about 70 times.

Figure 2 is the plot:

6 Problem 6

Romberg script:

```
function r = romberg(fun,a,b,n)
h = (b - a) ./ (2.^(0:n-1));
r(1,1) = (b - a) * (feval(fun,a) + feval(fun,b)) / 2;
for j = 2:n
    subtotal = 0;
    for i = 1:2^(j-2)
        subtotal = subtotal + feval(fun,a + (2 * i - 1) * h(j));
    end
    r(j,1) = r(j-1,1) / 2 + h(j) * subtotal;
    for k = 2:j
        r(j,k) = (4^(k-1) * r(j,k-1) - r(j-1,k-1)) / (4^(k-1) - 1);
    end
end
```

Function files:

```
function v=func(x)
    %v = exp(-1.*x);
    %v = cos(2.*x).*exp(-1.*x);
    v = sin(x);
    %v = x.^0.5;
end

function v=f(x)
    %v = exp(-1.*x);
    %v = cos(2.*x).*exp(-1.*x);
    %v = sin(x);
    v = x.^0.5;
end
```

Script file

```
rows = 5;
vals = romberg('func',0,pi,rows);
act = 2;

format short e;
disp(vals)
for i = 1:rows
    for j = 1:rows
        if vals(i,j) == 0
            continue
        else
            vals(i,j) = vals(i,j) - act;
        end
    end
end
```

```
disp('First Function Error')
disp(vals)

rows = 5;
vals = romberg('f',0,1,rows);
act = 2/3;
disp(vals)
for i = 1:rows
    for j = 1:rows
        if vals(i,j) == 0
            continue
        else
            vals(i,j) = vals(i,j) - act;
        end
    end
end

disp('Second Function Error')
disp(vals)
```

Tables in figure 3:

q = quad('f',0,1,1e-9)

q =

6.6667e-01

w = quad('func',0,pi,1e-9)

w =

2.0000e+00

q = quadl('f',0,1,1e-9)

q =

6.6667e-01

w = quadl('func',0,pi,1e-9)

w =

2.0000e+00

These quadrature approaches are spot on.

7 Problem 7

7.1 a. Calculate J at n = 1,3,9

I use the trapezoid formula:

$$T(f;h) = h[\frac{1}{2}f(x_0) + \Sigma f(x_i) + \frac{1}{2}f(x_n)]$$

```
>> problem6Script
   1.9237e-16            0            0            0            0
   1.5708e+00   2.0944e+00            0            0            0
   1.8961e+00   2.0046e+00   1.9986e+00            0            0
   1.9742e+00   2.0003e+00   2.0000e+00   2.0000e+00            0
   1.9936e+00   2.0000e+00   2.0000e+00   2.0000e+00   2.0000e+00

First Function Error
  -2.0000e+00            0            0            0            0
  -4.2920e-01   9.4395e-02            0            0            0
  -1.0388e-01   4.5598e-03  -1.4293e-03            0            0
  -2.5768e-02   2.6917e-04  -1.6869e-05   5.5500e-06            0
  -6.4297e-03   1.6591e-05  -2.4755e-07   1.6288e-08  -5.4127e-09

   5.0000e-01            0            0            0            0
   6.0355e-01   6.3807e-01            0            0            0
   6.4328e-01   6.5653e-01   6.5776e-01            0            0
   6.5813e-01   6.6308e-01   6.6352e-01   6.6361e-01            0
   6.6358e-01   6.6540e-01   6.6555e-01   6.6559e-01   6.6559e-01

Second Function Error
  -1.6667e-01            0            0            0            0
  -6.3113e-02  -2.8595e-02            0            0            0
  -2.3384e-02  -1.0140e-02  -8.9101e-03            0            0
  -8.5364e-03  -3.5874e-03  -3.1505e-03  -3.0591e-03            0
  -3.0855e-03  -1.2685e-03  -1.1139e-03  -1.0816e-03  -1.0738e-03
```

Figure 3:

```
Columns 1 through 8

 0.420735492403948                  0                  0                  0                  0                  0                  0                  0
 0.689793284806177  0.779479215606920                  0                  0                  0                  0                  0                  0
 0.819513521665390  0.862753600618460  0.868305226285896                  0                  0                  0                  0                  0
 0.883190863582701  0.904416644221805  0.907194180462028  0.907811465448951                  0                  0                  0                  0
 0.914735029934386  0.925249752051614  0.926638625906935  0.926947267898124  0.927022310260669                  0                  0                  0
 0.930433560962768  0.935666404638895  0.936360848144714  0.936515169132615  0.936552690313927  0.936562006423461                  0                  0
 0.938264443060063  0.940874737092495  0.941[illegible]21959256068  0.941299119749899  0.941317880340555  0.941322538395322  0.941323700918392                  0
 0.942175288543152  0.943478903704182  0.943652514811628  0.943691095058541  0.943700475353869  0.943702804381253  0.943703385642788  0.943703530896082
 0.944129562411347  0.944780987034079  0.944867792589405  0.944887082712862  0.944891772860526  0.944892937374218  0.944893228004985  0.944893300631633

Column 9

                 0
                 0
                 0
                 0
                 0
                 0
                 0
                 0
 0.944893318786355
```

Figure 4:

```
n = [1 3 9];

for i = 1:length(n)
    space = 1 / n(i);
    x = 0:space:1;
    y = zeros(1,length(x));
    y(1) = 1;
    y(2:end) = func(x(2:end));
    val = trapz(space,y);
    disp(val)
end
```

Gives us:

n	1	3	9
J	0.9207355	0.9432914	0.9457732

I use my romberg fomula:

```
function r = romberg(fun,a,b,n)
h = (b - a) ./ (2.^(0:n-1));
r(1,1) = (b - a) * (feval(fun,a) + feval(fun,b)) / 2;
for j = 2:n
    subtotal = 0;
    for i = 1:2^(j-2)
        subtotal = subtotal + feval(fun,a + (2 * i - 1) * h(j));
    end
    r(j,1) = r(j-1,1) / 2 + h(j) * subtotal;
    for k = 2:j
        r(j,k) = (4^(k-1) * r(j,k-1) - r(j-1,k-1)) / (4^(k-1) - 1);
    end
end
```

Gives us figure 4:

8 Problem 8

8.1 a.

We want to assure that this quadrature gives us the exact value for any polynomial of degree 7 or less.

Let's check:

f(x) = 1, The actual value from [-1,1] is 2, x is 0, x^2 is 2/3, x^3 is 0, x^4 is 2/5, x^5 is 0, x^6 is 2/7, and x^7 is 0.

Using the values in matlab we get the following printout:

```
x = [-((3 - 4 * 0.3 ^ 5) / 7)^0.5 -((3 + 4 * 0.3 ^ 5) / 7)^0.5 ((3 - 4 * 0.3 ^ 5) / 7)^0.5 ((3 + 4 * 0.
```

-0.653592271330420 -0.655713352006805 0.653592271330420 0.655713352006805

```
a = [0.5 + (10/3)^0.5 / 12 0.5 - (10/3)^0.5 / 12 0.5 + (10/3)^0.5 / 12 0.5 - (10/3)^0.5 / 12]
```

0.652145154862546 0.347854845137454 0.652145154862546 0.347854845137454

a(1)+a(2)+a(3)+a(4)

ans =

2

a(1)*x(1) + a(2)*x(2) + a(3)*x(3) + a(4)*x(4)

ans =

-5.551115123125783e-17

a(1)*x(1)^2 + a(2)*x(2)^2 + a(3)*x(3)^2 + a(4)*x(4)^2

ans =

0.856297799482706

a(1)*x(1)^3 + a(2)*x(2)^3 + a(3)*x(3)^3 + a(4)*x(4)^3

ans =

-1.387778780781446e-17

a(1)*x(1)^4 + a(2)*x(2)^4 + a(3)*x(3)^4 + a(4)*x(4)^4

ans =

0.366626459899463

a(1)*x(1)^5 + a(2)*x(2)^5 + a(3)*x(3)^5 + a(4)*x(4)^5

ans =

0

a(1)*x(1)ˆ6 + a(2)*x(2)ˆ6 + a(3)*x(3)ˆ6 + a(4)*x(4)ˆ6

ans =

0.156973714736308

a(1)*x(1)ˆ7 + a(2)*x(2)ˆ7 + a(3)*x(3)ˆ7 + a(4)*x(4)ˆ7

ans =

0

8.2 b.

If we want our polynomial to be correct for all polynomials for degree less than or equal to two we can set up a system of equations to find our uknowns:

$$
\begin{gathered}
f(x) = 1 : a_1 + a_2 + a_3 = 2 \\
f(x) = x : -0.5 * a_1 + 0 * a_2 + 0.5 * a_3 = 0 \\
f(x) = x^2 : -0.5 * a_1 + 0 * a_2 + 0.5 * a_3 = \frac{2}{3}
\end{gathered}
$$

Now solve!

$$
\begin{gathered}
a_1 + a_2 + a_3 = 2 \\
-0.5a_1 + 0.5a_3 = 0 \\
-0.5a_1 + 0.5a_3 = \frac{2}{3}
\end{gathered}
$$

We can add the second and third:

$$a_3 = \frac{2}{3}$$

Plug back into number 2:

$$
\begin{gathered}
-\frac{1}{2}a_1 = -\frac{1}{2}\frac{2}{3} \\
a_1 = \frac{2}{3}
\end{gathered}
$$

Plugging into the first equation we have:

$$
\begin{gathered}
\frac{2}{3} + a_2 + \frac{2}{3} = 2 \\
a_2 + \frac{4}{3} = 2 \\
a_2 = \frac{6}{3} - \frac{4}{3} \\
a - 2 = \frac{2}{3}
\end{gathered}
$$

So all three values $a_1, a_2, a_3 = \frac{2}{3}$.