《机械工程中的数值分析技术》

作业

学　　生：易弘睿

学　　号：20186103

专业班级：机械一班

作业编号：2021071308

重庆大学-辛辛那提大学联合学院

二〇二一年七月

Catalog

Chap19 Numerical Integration Formulas

1.1 Question 19.2

19.2 Evaluate the following integral:

$$\int_0^4 (1 - e^{-x})\,dx$$

(a) analytically, **(b)** single application of the trapezoidal rule, **(c)** composite trapezoidal rule with $n = 2$ and 4, **(d)** single application of Simpson's 1/3 rule, **(e)** composite Simpson's

1/3 rule with $n = 4$, **(f)** Simpson's 3/8 rule, and **(g)** composite Simpson's rule, with $n = 5$. For each of the numerical estimates **(b)** through **(g)**, determine the true percent relative error based on **(a)**.

1.1.1 Question a

The Matlab code is below:

```
clc;clear all;
f = @(x) 1-exp(-x);
int = integral(f,0,4);
fprintf('The analytically result is: %f\n', int)
```

The output is below:

```
The analytically result is: 3.018316
```

1.1.2 Question b

The Matlab code is below:

```
clc;clear all;
f = @(x) 1-exp(-x);
```

```
int = integral(f,0,4);
Single_trap_result = 0.5*(f(0)+f(4))*4;
Error = abs(Single_trap_result-int)/int;
fprintf('The result using single application of the
trapezoidal rule is: %f\n', Single_trap_result)
fprintf('Relative error is : %f\n', Error)
```

The output is below:

```
The result using single application of the trapezoidal rule
is: 1.963369
Relative error is : 0.349515
```

1.1.3 Question c

The Matlab code is below:

```
clc;clear all;
f = @(x) 1-exp(-x);
int = integral(f,0,4);
Trap_with_n2 = mytrap([0,4],f,2);
Error2 = abs(Trap_with_n2-int)/int;
fprintf('The result of composite trapezoidal rule with n
= 2 is: %f\n', Trap_with_n2)
fprintf('Relative error is : %f\n', Error2)
Trap_with_n4 = mytrap([0,4],f,4);
Error4 = abs(Trap_with_n4-int)/int;
fprintf('The result of composite trapezoidal rule with n
= 4 is: %f\n', Trap_with_n4)
fprintf('Relative error is : %f\n', Error4)
function integral = mytrap(x,f,m)
    integral = 0;
    x = min(x):(max(x)-min(x))/m:max(x);
    for k = 1:length(x)-1
        integral =
integral+0.5*(f(x(k))+f(x(k+1)))*(x(k+1)-x(k));
    end
end
```

The output is below:

```
The result of composite trapezoidal rule with n = 2 is:
2.711014
Relative error is : 0.101812
The result of composite trapezoidal rule with n = 4 is:
2.937840
Relative error is : 0.026662
```

1.1.4 Question d

The Matlab code is below:

```
clc;clear all;
f = @(x) 1-exp(-x);
int = integral(f,0,4);
Single_simpson_result = (4-0)/6*(f(0)+4*f((4+0)/2)+f(4));
Error = abs(Single_simpson_result-int)/int;
fprintf('The result of single application of Simpson¡¯s
1/3 rule is: %f\n', Single_simpson_result)
fprintf('Relative error is : %f\n', Error)
```

The output is below:

```
The result of single application of Simpson’s 1/3 rule is:
2.960229
Relative error is : 0.019245
```

1.1.5 Question e

The Matlab code is below:

```
clc;clear all;
f = @(x) 1-exp(-x);
int = integral(f,0,4);
Single_simpson_with_n4 = simpson([0,4],f,4);
Error = abs(Single_simpson_with_n4-int)/int;
fprintf('The result of composite Simpson¡¯s 1/3 rule with
n = 4 is: %f\n', Single_simpson_with_n4)
fprintf('Relative error is : %f\n', Error)

function integral = simpson(x,f,m)
```

```
    integral = 0;
    x = min(x):(max(x)-min(x))/m:max(x);
        for k = 1:length(x)-1
           h = (x(k+1)-x(k))/2;
           integral =
integral+h/3*(f(x(k))+4*f((x(k+1)+x(k))/2)+f(x(k+1)));
        end
end
```

The output is below:

```
The result of composite Simpson's 1/3 rule with n = 4 is:
3.017985 Relative error is : 0.000110
```

1.1.6 Question f

The Matlab code is below:

```
clc;clear all;
f = @(x) 1-exp(-x);
int = integral(f,0,4);
Simpson_38rule = (4-0)/8*(f(0)+3*f(4/3)+3*f(8/3)+f(4));
Error = abs(Simpson_38rule-int)/int;
fprintf('The result of Simpson¡¯s 3/8 rule is: %f\n',
Simpson_38rule)
fprintf('Relative error is : %f\n', Error)
```

The output is below:

```
The result of Simpson's 3/8 rule is: 2.991221
Relative error is : 0.008977
```

1.1.7 Question g

The Matlab code is below:

```
clc;clear all;
f = @(x) 1-exp(-x);
int = integral(f,0,4);
```

```
X = linspace(0,4,6);
simpson_with_n5 = 0;
for i = 1:length(X)-1
 if i <= 2
 simpson_with_n5 = simpson_with_n5+(X(i+1)-
X(i))/6*(f(X(i))+4*f((X(i)+X(i+1))/2)+f(X(i+1)));
 else
 in = (X(i+1)-X(i))/3; simpson_with_n5 =
simpson_with_n5+(X(i+1)-
X(i))/8*(f(X(i))+3*f(X(i)+in)+3*f(X(i)+2*in)+f(X(i+1)));
 end
end
Error = abs(simpson_with_n5-int)/int;
fprintf('The result of composite Simpson¡¯s rule with n =
5 is: %f\n', simpson_with_n5)
fprintf('Relative error is : %f\n', Error)
```

The output is below:

```
The result of composite Simpson’s rule with n = 5 is:
3.018193
Relative error is : 0.000041
```

1.2 Question 19.4

19.4 Evaluate the following integral:

$$\int_{-2}^{4} (1 - x - 4x^3 + 2x^5)\, dx$$

(a) analytically, **(b)** single application of the trapezoidal rule, **(c)** composite trapezoidal rule with $n = 2$ and 4, **(d)** single application of Simpson’s 1/3 rule, **(e)** Simpson’s 3/8 rule, and **(f)** Boole’s rule. For each of the numerical estimates **(b)** through **(f)**, determine the true percent relative error based on **(a)**.

1.2.1 Question a

The Matlab code is below:

```
clc;clear all;
f = @(x)1-x-4*x.^3+2*x.^5;
int = integral(f,-2,4);
fprintf('The analytically result is: %f\n', int)
```

The output is below:

```
The analytically result is: 1104.000000
```

1.2.2 Question b

The Matlab code is below:

```
clc;clear all;
f = @(x)1-x-4*x.^3+2*x.^5;
int = integral(f,-2,4);
Single_trap_result = (f(-2)+f(4))*3;
Error = abs(Single_trap_result-int)/int;
fprintf('The result using single application of the
trapezoidal rule is: %f\n', Single_trap_result)
fprintf('Relative error is : %f\n', Error)
```

The output is below:

```
The result using single application of the trapezoidal rule
is: 5280.000000
Relative error is : 3.782609
```

1.2.3 Question c

The Matlab code is below:

```
clc;clear all;
f = @(x)1-x-4*x.^3+2*x.^5;
int = integral(f,-2,4);
Trap_with_n2 = mytrap([0,4],f,2);
Error2 = abs(Trap_with_n2-int)/int;
fprintf('The result of composite trapezoidal rule with n
= 2 is: %f\n', Trap_with_n2)
```

```
fprintf('Relative error is : %f\n', Error2)
Trap_with_n4 = mytrap([0,4],f,4);
Error4 = abs(Trap_with_n4-int)/int;
fprintf('The result of composite trapezoidal rule with n
= 4 is: %f\n', Trap_with_n4)
fprintf('Relative error is : %f\n', Error4)
function integral = mytrap(x,f,m)
    integral = 0;
    x = min(x):(max(x)-min(x))/m:max(x);
        for k = 1:length(x)-1
            integral =
integral+0.5*(f(x(k))+f(x(k+1)))*(x(k+1)-x(k));
        end
end
```

The output is below:

```
The result of composite trapezoidal rule with n = 2 is:
1852.000000
Relative error is : 0.677536
The result of composite trapezoidal rule with n = 4 is:
1300.000000
Relative error is : 0.177536
```

1.2.4 Question d

The Matlab code is below:

```
clc;clear all;
f = @(x)1-x-4*x.^3+2*x.^5;
int = integral(f,-2,4);
Single_simpson_result = (4+2)/6*(f(-2)+4*f((4-
2)/2)+f(4));
Error = abs(Single_simpson_result-int)/int;
fprintf('The result of single application of Simpson¡¯s
1/3 rule is: %f\n', Single_simpson_result)
fprintf('Relative error is : %f\n', Error)
```

The output is below:

```
The result of single application of Simpson’s 1/3 rule is:
1752.000000
Relative error is : 0.586957
```

1.2.5 Question e

The Matlab code is below:

```
clc;clear all;
f = @(x)1-x-4*x.^3+2*x.^5;
int = integral(f,-2,4);
Simpson_38rule = (4+2)/8*(f(-2)+3*f(0)+3*f(2)+f(4));
Error = abs(Simpson_38rule-int)/int;
fprintf('The result of Simpson¡¯s 3/8 rule is: %f\n',
Simpson_38rule)
fprintf('Relative error is : %f\n', Error)
```

The output is below:

```
The result of Simpson’s 3/8 rule is: 1392.000000
Relative error is : 0.260870
```

1.2.6 Question f

The Matlab code is below:

```
clc;clear all;
f = @(x)1-x-4*x.^3+2*x.^5;
int = integral(f,-2,4);
Inteval = 1.5;
Bool = 6/90*(7*f(-2)+32*f(-2+Inteval)+12*f(-
2+2*Inteval)+32*f(-2+3*Inteval)+7*f(4));
Error = abs(Bool-int)/int;
fprintf('The result of Boole?¢ã?s rule is: %f\n', Bool)
fprintf('Relative error is : %f\n', Error)
```

The output is below:

```
The result of Booleâ€™s rule is: 1104.000000
Relative error is : 0.000000
```

1.3 Question 19.10

19.10 The force on a sailboat mast can be represented by the following function:

$$f(z) = 200\left(\frac{z}{5+z}\right)e^{-2z/H}$$

where z = the elevation above the deck and H = the height of the mast. The total force F exerted on the mast can be determined by integrating this function over the height of the mast:

$$F = \int_0^H f(z)\,dz$$

The line of action can also be determined by integration:

$$d = \frac{\int_0^H z f(z)\,dz}{\int_0^H f(z)\,dz}$$

(a) Use the composite trapezoidal rule to compute F and d for the case where $H = 30$ ($n = 6$).

(b) Repeat **(a)**, but use the composite Simpson's 1/3 rule.

1.3.1 Question a

The Matlab code is below:

```
clc;clear all;
f = @(x)200*x/(5+x)*exp(-2*x/30);
F = @(x)x*200*x/(5+x)*exp(-2*x/30);
Fa = mytrap([0,30],f,6);
da = mytrap([0,30],F,6)/mytrap([0,30],f,6);
fprintf('F = %f\n', Fa)
fprintf('d = %f\n', da)
```

The output is below:

```
F = 1402.728197
d = 13.719864
```

1.3.2 Question b

The Matlab code is below:

```
clc;clear all;
f = @(x)200*x/(5+x)*exp(-2*x/30);
F = @(x)x*200*x/(5+x)*exp(-2*x/30);
Fb = simpson([0,30],f,6);
db = simpson([0,30],F,6)/mytrap([0,30],f,6);
fprintf('F = %f\n', Fb)
fprintf('d = %f\n', db)

function integral = simpson(x,f,m)
    integral = 0;
    x = min(x):(max(x)-min(x))/m:max(x);
        for k = 1:length(x)-1
            h = (x(k+1)-x(k))/2;
            integral =
integral+h/3*(f(x(k))+4*f((x(k+1)+x(k))/2)+f(x(k+1)));
        end
end
```

The output is below:

```
F = 1478.611181
d = 13.784660
```

Chap20 Numerical Integration of Functions

2.1 Question 20.1

20.1 Use Romberg integration to evaluate

$$I = \int_1^2 \left(x + \frac{1}{x} \right)^2 dx$$

to an accuracy of $\varepsilon_s = 0.5\%$. Your results should be presented in the format of Fig. 20.1. Use the analytical solution of the integral to determine the percent relative error of the result obtained with Romberg integration. Check that ε_t is less than ε_s.

The Matlab code is below:

```
clc;clear all;
f = @(x)(x+1./x).^2;
es = 0.5;
[Result,Error] = romberg(f,1,2,es);
fprintf('The calculated result is: %f\n',Result)
fprintf('Relative error is : %f\n', Error)
if Error < es
 disp('The relative error is less than the required
accuracy.')
else
 disp('Sorry, the error is still large.')
end

function [q,ea]=romberg(func,a,b,es,maxit,varargin)
if nargin<3
 error('at least 3 input arguments required')
end
if nargin<4 || isempty(es)
 es=0.000001;
end
if nargin<5 || isempty(maxit)
 maxit=50;
end
actual = integral(func,a,b);
n = 1;
```

```
I(1,1) = trap(func,a,b,n);
iter = 0;
while iter < maxit
 iter = iter+1;
 n = 2^iter;
 I(iter+1,1) = trap(func,a,b,n);
 for k = 2:iter+1
 j = 2+iter-k;
 I(j,k) = (4^(k-1)*I(j+1,k-1)-I(j,k-1))/(4^(k-1)-1);
 end
 ea = abs((I(1,iter+1)-actual)/actual)*100;
 if ea <= es
     break;
 end
end
q = I(1,iter+1);
end
```

The output is below:

```
The calculated result is: 4.837963
Relative error is : 0.095785
The relative error is less than the required accuracy.
```

2.2 Question 20.8

20.8 The amount of mass transported via a pipe over a period of time can be computed as

$$M = \int_{t_1}^{t_2} Q(t)c(t)\,dt$$

where M = mass (mg), t_1 = the initial time (min), t_2 = the final time (min), $Q(t)$ = flow rate (m^3/min), and $c(t)$ = concentration (mg/m^3). The following functional representations define the temporal variations in flow and concentration:

$$Q(t) = 9 + 5\cos^2(0.4t)$$
$$c(t) = 5e^{-0.5t} + 2e^{0.15t}$$

Determine the mass transported between $t_1 = 2$ and $t_2 = 8$ min with **(a)** Romberg integration to a tolerance of 0.1% and **(b)** the MATLAB `quad` function.

2.2.1 Question b

The Matlab code is below:

```
clc;clear all;
f = @(x)(9+5*cos(0.4*x).^2).*(5*exp(-
0.5*x)+2*exp(0.15*x));
es = 0.1;
[Result,Error] = romberg(f,2,8,es);
fprintf('The calculated result is: %f\n',Result)
fprintf('Relative error is : %f\n', Error)
```

The output is below:

```
The calculated result is: 335.959198
Relative error is : 0.001443
```

2.2.2 Question b

The Matlab code is below:

```
clc;clear all;
f = @(x)(9+5*cos(0.4*x).^2).*(5*exp(-
0.5*x)+2*exp(0.15*x));
Result= quad(f,2,8);
fprintf('The result calculated by matlab
is: %f\n',Result)
```

The output is below:

```
The result calculated by matlab is: 335.962530
```