Numerical Analysis Assignment #5

Xinglu Wang   Student Number: 3140102282
College of Information Science & Electronic Engineering

problem 1

a). For this problem, if we use formula derived on the book, which means using $f = p1x + p2$ to fit. We get the coefficients (with 95% confidence bounds):

$$p1 = 0.7525(0.2317, 1.273) \text{ and } p2 = -3.199(-8.307, 1.909)$$

Here, $E = \frac{-p2}{p1} = 4.25 \neq 5.3$ then the true value will be unused. So we must regard that E is known.

$$E(a_0, a_1) = E_2(a_0, ka_0) = \sum_{i=1}^{m} [y_i - (ka_0 x_i + a_0)]^2$$

where k is constant $-\frac{1}{E}$, then

$$\frac{dE}{da_0} = 0$$
$$\Rightarrow a_0 = -4.59$$
$$\Rightarrow a_1 = -\frac{1}{5.3} a_0 = 0.8996$$

And the error should be mean error: $\frac{1}{3} \sum_{i=1}^{m} [y_i - (ka_0 x_i + a_0)]^2 = 0.136$

Meanwhile, we can use linear fittype in matlab to fit a_1.

```
function res=LeastSquare(X,Y)
F=fittype(@(k,x) k*(x-5.3));
res = fit( X, Y, F, 'StartPoint',1 );
figure('Color',[1 1 1]);
plot( res, X, Y );
%legend off
```

b). Similarly, $a1 = 0.9052$ and $E(a1) = \frac{1}{7} \sum_{i=1}^{m} [y_i - (ka_0 x_i + a_0)]^2 = 0.128$. So $a_1 = 0.9052$ is more accurate with more data points.

problem 2

On the one hand, we can use the formula

$$\sum_{k=0}^{n} a_k \int_a^b x^{j+k}\,dx = \int_a^b x^j f(x)\,dx, \quad \text{for each } j = 0, 1, \ldots, n, \qquad (8.6)$$

But the complexity of calculating an linear system will be great. So we choose The set of Legendre polynomials $P_n(x)$, which is orthogonal on $[-1, 1]$ with respect to the weight function $w(x) \equiv 1$.

$$P_0(x) = 1$$
$$P_1(x) = x$$
$$P_2 x = x^2 - \frac{1}{3}$$

a). My answer is

$$f(x) = \frac{10}{3} - 2x$$

I find that the linear least sqaure polynominal approximation is just discard the item with high power, because we can see the accurate form with respect to Legendre polynomials is $f(x) = \frac{10}{3} - 2x + (x^2 - \frac{1}{3})$, and to approximate is to discard $(x^2 - \frac{1}{3})$.

b). Similarly, my answer is

$$f(x) = 0.6x$$

problem 3

According to Gram-Schmidt Process,

$$B_1 = \frac{\int_a^b xw(x)[\phi_0(x)]^2\,dx}{\int_a^b w(x)[\phi_0(x)]^2\,dx},$$

$$B_k = \frac{\int_a^b xw(x)[\phi_{k-1}(x)]^2\,dx}{\int_a^b w(x)[\phi_{k-1}(x)]^2\,dx}$$

$$C_k = \frac{\int_a^b xw(x)\phi_{k-1}(x)\phi_{k-2}(x)\,dx}{\int_a^b w(x)[\phi_{k-2}(x)]^2\,dx}.$$

These are the first few Laguerre polynomials:

$$\phi_0(x) = 1$$
$$\phi_1(x) = x - 1$$
$$\phi_2(x) = x^2 - 4x + 2$$
$$\phi_3(x) = x^3 - 9x^2 + 18x - 6$$

problem 4

Using the Three-point Startpoint formula for x_1 , Three-point Midpoint formula for x_2, x_3 and Endpoint for x_4. We get

a).

x	f'(x)
1.1	17.76
1.2	22.19
1.3	27.1
1.4	32.51

b).

x	f'(x)
8.1	3.09
8.3	3.12
8.5	3.14
8.7	3.16

problem 5

We need to complete the table below,

$O(h^2)$	$O(h^4)$	$O(h^6)$
$N_0(h)$		
$N_0(\frac{h}{3})$	$N_1(h)$	
$N_0(\frac{h}{3^2})$	$N_1(\frac{h}{3})$	$N_2(h)$

Table 5.1 Iteration of a Richardson's Extrapolation Method

First, to calculate N_1, we know

$$\begin{aligned} M &= N_0(h) + Kh^2 + O(h^4) \\ M &= N(\tfrac{h}{3}) + K\tfrac{h^2}{9} + O(h^4) \end{aligned} \tag{1}$$

We get

$$\begin{aligned} M &= \tfrac{1}{8}\left[9N(\tfrac{h}{3}) - N(h)\right] + O(h^4) \\ &\doteq N_1(h) + O(h^4) \end{aligned}$$

Substitute h with $\frac{h}{3}$, we get $N_1(\frac{h}{3}) = \frac{1}{8}\left[9N(\frac{h}{9}) - N(\frac{h}{3})\right]$
Then iterate again, we get $N_2(h) = \frac{1}{640}N_0(h) - \frac{9}{64}N_0(\frac{h}{3}) + \frac{729}{640}N_0(\frac{h}{9})$. So finally,

$$M = \frac{1}{640}N_0(h) - \frac{9}{64}N_0(\frac{h}{3}) + \frac{729}{640}N_0(\frac{h}{9}) + O(h^6)$$

Problem 6

My answer is:

	Trapezoidal	Simpson
$\int_{-0.25}^{0.25} cos^2x$	0.469395640472593	0.489798546824198
$\int_0^{0.5} xln(x+1)$	0.086643397569993	0.052854638560979
$\int_0 .75^1 .3sin^2x - 2xsinx + 1$	-0.037024252723997	-0.020271589910295
$\int_e^{e+1} \frac{1}{x\ln x}dx$	0.286334172478335	0.272670452444963

The code to implement methods is shown below:

```
function res=Trape(f,Xi,Xe)
F=matlabFunction(f);
res=(Xe-Xi)/2*(F(Xi)+F(Xe));
```

```
function res=Simps(f,xi,xe)
xm=xi+xe;xm=xm/2;
F=matlabFunction(f);
res=(xe-xm)/3*(F(xi)+4*F(xm)+F(xe));
```

Problem 7

Using $h_n = \frac{1}{2^n}(b-a)$, Romberg method can be inductively defined by

$$\begin{aligned} &R(0,0) = h_1(f(a) + f(b)) \\ &R(n,0) = \tfrac{1}{2}R(n-1,0) + h_n \sum_{k=1}^{2^{n-1}} f(a + (2k-1)h_n) \\ &R(n,m) = R(n,m-1) + \tfrac{1}{4^m-1}(R(n,m-1) - R(n-1,m-1)) \end{aligned}$$

where $n \geq m$ and $m \geq 1$. Just consider $n = m = 3$, my answer is

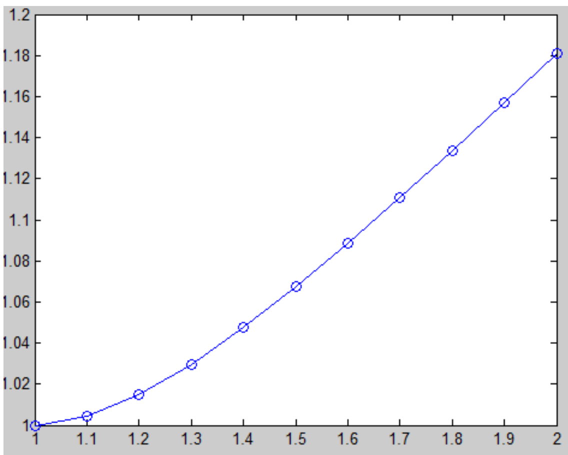
$\int_{-1}^{1} cos^2x$	1.452814
$\int_{-0.75}^{0.75} xln(x+1)$	0.327959
$\int_1^4 sin^2x - 2xsinx + 1$	1.387063
$\int_e^{2e} \frac{1}{x\ln x}dx$	0.526816

```
function res=Romb(f,xi,xe)
R(1,1)=Trape(f,xi,xe);
h=(xe-xi)/2;
R(2,1)=Trape(f,xi,(xi+xe)/2)+Trape(f,(xi+xe)/2,xe);
h=h/2; R(3,1)=0;
for xx=xi:h:(xe-h)
R(3,1)=R(3,1)+Trape(f,xx,xx+h);
end
R(2,2)=R(2,1)+1/3*(R(2,1)-R(1,1));
R(3,2)=R(3,1)+1/3*(R(3,1)-R(2,1));
R(3,3)=R(3,2)+1/15*(R(3,2)-R(2,2));
res=R(3,3);
```

Problem 8

a). We can compare Euler's method with analytic solution, since this ODE have analytic solution $\frac{t}{\log(t)+1}$:

T	Euler's method	$\frac{t}{\log(t)+1}$
1	1	1
1.1	1.004281728	1
1.2	1.014952314	1.008264463
1.3	1.029813689	1.021689472
1.4	1.047533919	1.038514734
1.5	1.067262354	1.057668192
1.6	1.088432687	1.078461094
1.7	1.110655052	1.100432165
1.8	1.133653556	1.123262052
1.9	1.157228433	1.146723597
2	1.181232218	1.17065157



Apparently, Euler's method is not very accurate. On the one hand h is a kind of large, and on the other hand the LTE is $O(h)$ itself.

```
function [T,Y]=myode(f,T,yi)
h=T(2)-T(1);
Y=zeros(size(T));
Y(1)=yi;
for iter=2:size(T,2)
Y(iter)=Y(iter-1)+h*f(T(iter-1),Y(iter-1));
end
```

b). Similarly, my answer is

T	Y
1	0
1.2	0.2
1.4	0.438888889
1.6	0.721242756
1.8	1.052038032
2	1.437251148
2.2	1.884260805
2.4	2.402269589
2.6	3.002837165
2.8	3.700600705
3	4.514277428