**HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY**

**OFFICE FOR INTERNATIONAL STUDY PROGRAMS**

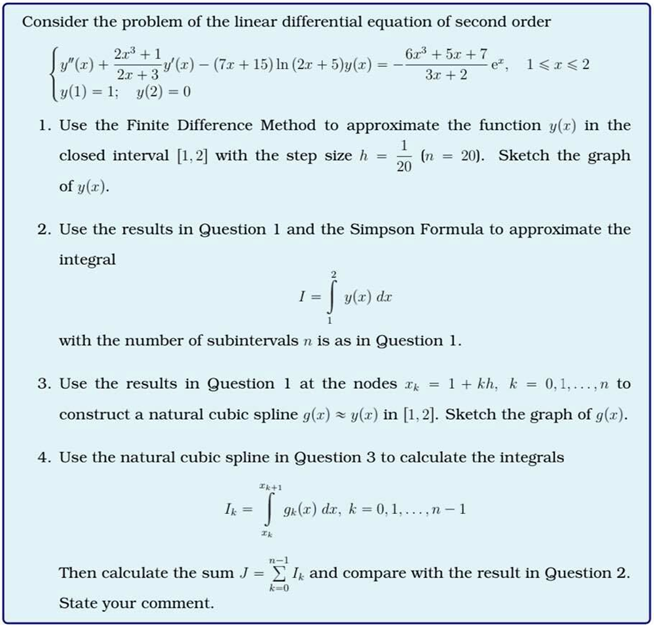


**REPORT ASSIGNMENT**

**NUMERICAL METHODS**

**Lecturer: Mr. Lê Thái Thanh**

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Student ID** |
| 1 | Nguyễn Duy Thành | 1952456 |
| 2 | Võ Duy Thành | 2050019 |
| 3 | Lê Ngọc Thành | 2052257 |
| 4 | Nguyễn Đô Trưởng | 1752586 |
| **Assignment – Group 8** | | |



|  |  |
| --- | --- |
| **Name** | **Workload** |
| Nguyễn Duy Thành | Report + Formatting |
| Võ Duy Thành | Report |
| Lê Ngọc Thành | Matlab Code |
| Nguyễn Đô Trưởng | Formula + Report |

+ In order to solve functions and sketch graphs, we use Matlab (version may vary depends on each individual personal computer). The following code entries will show the detailed functions of each assignments that are executed in Matlab (complete program will be put in the end of report). The solving method involved many formulas and methods, which includes:

1. Finite difference method
2. Simpson formula
3. Natural cubic spline construction

* **Question 1**: Finite Difference Method

%test 1

x = 1:1/20:2;

%Functions settings and work environment

%Rearrange functions and equations

X = x;

B = (6\*x.^3 + 5\*x + 7)./(3\*x + 2).\*exp(x);

U = (2\*x.^3 + 1)./(2\*x + 3);

L = -(7\*x + 15).\*log(2\*x + 5);

h = 1/20; %dentax

A = zeros(length(x));

A(1,1) = 1;

A(end,end) = 1;

B(1) = 1;

B(end) = 0;

for i = 2:length(x)-1

A(i,i) = -2/h^2 + U(i)/h +L(i);

A(i,i-1) = 1/h^2 - U(i)/h;

A(i,i+1) = 1/h^2;

end

Y = inv(A)\*B';

plot(x,Y,'.')

disp('TEST 1')

Y

Graphical user interface, chart

Description automatically generated with medium confidenceOutput:

Test 1:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| N | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| X | 1 | 1.05 | 1.1 | 1.15 | 1.2 | 1.25 | 1.3 |
| Y | 1 | 0.6128 | 0.3319 | 0.1271 | -0.0236 | -0.1362 | -0.2222 |
|  |  |  |  |  |  |  |  |
| N | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| X | 1.35 | 1.4 | 1.45 | 1.5 | 1.55 | 1.6 | 1.65 |
| Y | -0.2897 | -0.3444 | -0.3900 | -0.4290 | -0.4625 | -0.4905 | -0.5119 |
|  |  |  |  |  |  |  |  |
| N | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| X | 1.7 | 1.75 | 1.8 | 1.85 | 1.9 | 1.95 | 2 |
| Y | -0.5245 | -0.5244 | -0.5057 | -0.4596 | -0.3735 | -0.2291 | 0 |

* **Question 2**: Simpson Formula

%Apply Simpson formula for problem 2%

disp('Test 2')

I =

(h/3)\*( ( Y(1) + Y(end))

+ 4\*( Y(2) + Y(4) + Y(6) + Y(8) + Y(12) + Y(14) + Y(16) + Y(18) + Y(20) )

+ 2\*(Y(3) + Y(5) + Y(7) + Y(9) + Y(11) + Y(13) + Y(15) + Y(17) + Y(19)))

Test 2:

I = - 0.1943

* **Question 3**: Natural Cubic Spline Construction

%Construct a natural cubic spline

disp('Test 3')

f = Spline\_bac\_3\_tu\_nhien(x,Y)

%%

%Graph sketch for y(x) in problem 1

function f = Spline\_bac\_3\_tu\_nhien(x,y)

plot(x,y,'.')

hold on

% x = input('nhap x:')

% y = input('nhap y:');

h = diff(x); % h (j) = x(j+1) -x(j)

a = y; % a(j) = y(j);

% d = diff(c)/(3\*h); % d(j) = (c(j+1)-c(j))/(3h(j))

% b = diff(a)/h - h(c+2\*c)/3

%Natural Cubic Spline graph sketch for g(x) based on calculated y(x)nodes

n = length(x);

A = zeros(n);

A(1,1) = 1;

A(n,n) = 1;

j = 1;

for i = 2 : n-1

A(i,i) = 2\*(h(j) + h(j+1));

A(i,i-1) = h(j);

A(i,i+1) = h(j+1);

j = j+1;

end

% B

v = diff(a);

B = zeros(n,1);

for i = 2 : length(B) - 1

B(i) = 3/h(i) \* v(i) - 3/h(i-1) \* v(i-1);

end

c = inv(A) \* B;

m = diff(c);

for i =1 : length(c)-1

d(i) = m(i) / (3\*h(i));

end

for j = 1 : length(a)-1

b(j) =

(a(j+1)-a(j))/h(j)-h(j) \*( c(j+1) + 2\*c(j))/3;

end

X = x;

syms x

a = double(a);

b = double(b);

c = double(c);

d = double(d);

for i = 1 : length(d)

f(i)= a(i) + b(i)\*(x - X(i)) + c(i) \*(x - X(i))^2 + d(i)\*(x - X(i) )^3;

ezplot(f(i), [X(i) X(i+1)])

end

plot(X,y,'\*')

axis([min(X) max(X) min(y) max(y)])

f = f.';

Test 3:

Chart

Description automatically generated

y(x) g(x) =



* **Question 4**: Comparison

%Calculate the intergral and sum J (noted by S in this project)

disp('Test 4')

syms x

S = 0;

for i = 1: length(f)

S = S + int(f(i), x, X(i), X(i+1));

end

%Comparision format between S and I in problem 2

S = double(S)

if S > I

disp('S>I')

elseif S == I

disp('S==I')

else

disp('S<I')

end

Test 4:

S = - 0.2200

S<I at the difference of: -0.1943 - (-0.2200) = 0.0257

* **Source code**

clc

clear all

%test 1

x = 1:1/20:2;

%Functions settings and work environment

%Rearrange functions and equations

X =x;

B = (6\*x.^3 +5\*x+7)./(3\*x+2).\*exp(x);

U = (2\*x.^3+1)./(2\*x+3);

L = -(7\*x+15).\*log(2\*x+5);

h=1/20; %dentax

A =zeros(length(x));

A(1,1) =1;

A(end,end) = 1;

B(1) = 1;

B(end) = 0;

for i =2:length(x)-1

A(i,i) = -2/h^2 +U(i)/h +L(i);

A(i,i-1) = 1/h^2 -U(i)/h;

A(i,i+1) = 1/h^2;

end

Y = inv(A)\*B';

plot(x,Y,'.')

disp('TEST 1')

Y

%Apply Simpson formula for problem 2%

disp('Test 2')

I=(h/3)\*((Y(1)+Y(end))+4\*(Y(2)+Y(4)+Y(6)+Y(8)+Y(12)+Y(14)+Y(16)+Y(18)+Y(20))+2\*(Y(3)+Y(5)+Y(7)+Y(9)+Y(11)+Y(13)+Y(15)+Y(17)+Y(19)))

%Construct a natural cubic spline

disp('Test 3')

f = Spline\_bac\_3\_tu\_nhien(x,Y)

%Calculate the intergral and sum J (noted by S in this project)

disp('Test 4')

syms x

S =0;

for i = 1: length(f)

S = S + int(f(i),x,X(i),X(i+1));

end

%Comparision format between S and I in problem 2

S = double(S)

if S > I

disp('S>I')

elseif S == I

disp('S==I')

else

disp('S<I')

end

%%

%Graph sketch for y(x) in problem 1

function f =Spline\_bac\_3\_tu\_nhien(x,y)

plot(x,y,'.')

hold on

% x = input('nhap x:')

% y = input('nhap y:');

h = diff(x) ; % h (j) = x(j+1) -x(j)

a = y; % a(j) = y(j);

% d = diff(c)/(3\*h); % d(j) = (c(j+1)-c(j))/(3h(j))

% b = diff(a)/h - h(c+2\*c)/3

%Natural Cubic Spline graph sketch for g(x) based on calculated y(x)nodes

n = length(x);

A = zeros(n);

A(1,1) = 1;

A(n,n) = 1;

j = 1;

for i =2 : n-1

A(i,i) = 2\*(h(j) +h(j+1));

A(i,i-1) = h(j);

A(i,i+1) = h(j+1);

j =j+1;

end

% B

v =diff(a);

B = zeros(n,1);

for i =2 : length(B)-1

B(i) = 3/h(i)\*v(i) -3/h(i-1)\*v(i-1);

end

c = inv(A)\*B;

m = diff(c);

for i =1 : length(c)-1

d(i)= m(i)/(3\*h(i));

end

for j = 1 : length(a)-1

b(j) = (a(j+1) -a(j))/h(j) -h(j) \*(c(j+1)+2\*c(j))/3;

end

X = x;

syms x

a = double(a);

b = double(b);

c = double(c);

d = double(d);

for i =1 : length(d)

f(i)= a(i) + b(i)\*(x-X(i)) + c(i) \*(x-X(i))^2+d(i)\*(x-X(i))^3;

ezplot(f(i), [X(i) X(i+1)])

end

plot(X,y,'\*')

axis([min(X) max(X) min(y) max(y)])

f =f.';

end