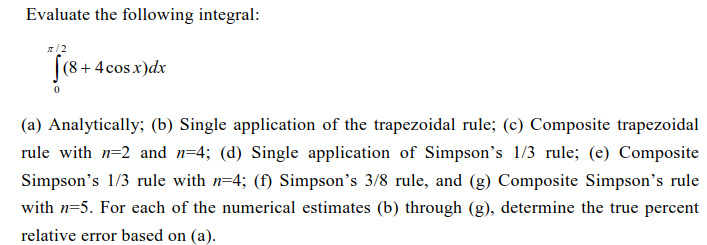
Numerical Methods Homework-8

B10602110 四電子三乙 呂和軒

1.



ANS :

1. Code\_function :

function [area] = trap\_I(f,x0,x1,n)

xx = linspace(x0,x1,n+1);

area = 0;

for i = 1:n

area = area + (xx(i+1)-xx(i))\*0.5\*(f(xx(i+1))+f(xx(i)));

end

end

function [area] = Simpson\_1\_3\_I(f,x0,x1,n)

xx = linspace(x0,x1,3\*n-n+1);

area = 0;

for i = 1:2:3\*(n-1)-(n-1-1)

area = area + (xx(i+1)-xx(i))/3 ...

\*(f(xx(i))+4\*f(xx(i+1))+f(xx(i+2)));

end

end

function [area] = Simpson\_3\_8\_I(f,x0,x1,n)

xx = linspace(x0,x1,4\*n-n+1);

area = 0;

for i = 1:3:4\*(n-1)-(n-1-1)

area = area + (xx(i+1)-xx(i))\*3/8 ...

\*(f(xx(i))+3\*f(xx(i+1))+3\*f(xx(i+2))+f(xx(i+3)));

end

end

1. Code\_main:

close all

clear all

format long

f = @(x) (8+4.\*cos(x))

x0 = 0; x1 = 0.5\*pi;

%% (a)

syms x

f\_s = 8+4\*cos(x);

f\_s\_I = int(f\_s)

f\_I = matlabFunction(f\_s\_I)

area\_a = f\_I(x1) - f\_I(x0)

fprintf("(a)real: %6.15f \n",area\_a)

%% (b)

area\_b = trap\_I(f,x0,x1,1);

error = abs(area\_a - area\_b)/area\_a;

fprintf("(b) n=1: %6.15f , error = %6.15f\n",area\_b,error)

%% (c)

n = 2;

area\_c\_2 = trap\_I(f,x0,x1,n);

error = abs(area\_a - area\_c\_2)/area\_a;

fprintf("(c) n=2: %6.15f , error = %6.15f \n",area\_c\_2,error)

n = 4;

area\_c\_2 = trap\_I(f,x0,x1,n);

error = abs(area\_a - area\_c\_2)/area\_a;

fprintf(" n=4: %6.15f , error = %6.15f\n",area\_c\_2,error)

%% (d)

n = 1;

area\_d = Simpson\_1\_3\_I(f,x0,x1,n);

error = abs(area\_a - area\_d)/area\_a;

fprintf("(d) n=1: %6.15f , error = %6.15f\n",area\_d,error)

%% (e)

n = 4;

area\_e = Simpson\_1\_3\_I(f,x0,x1,n);

error = abs(area\_a - area\_e)/area\_a;

fprintf("(e) n=4: %6.15f , error = %6.15f\n",area\_e,error)

%% (f)

n = 1;

area\_f = Simpson\_3\_8\_I(f,x0,x1,n);

error = abs(area\_a - area\_f)/area\_a;

fprintf("(f) n=1: %6.15f , error = %6.15f\n",area\_f,error)

%% (g)

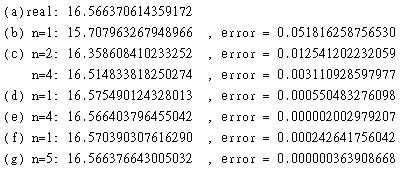
n = 5;

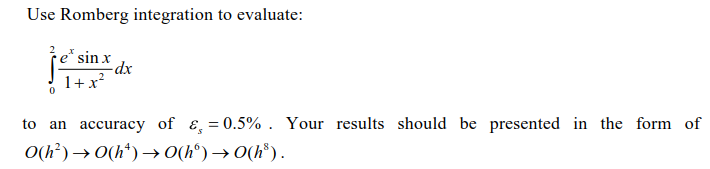
area\_g = Simpson\_3\_8\_I(f,x0,x1,n);

error = abs(area\_a - area\_g)/area\_a;

fprintf("(g) n=5: %6.15f , error = %6.15f\n",area\_g,error)

1. Result:



2. 

ANS :

1. Code\_function :

function [d,iter] = Romberg\_I(f,x0,x1,maxit,es)

%d = zeros(n,n);

n\_h = 1;

i = 1;

d(1,1) = trap\_I(f,x0,x1,1);

iter = 1;

while iter < maxit

n\_h = 2^iter;

d(iter+1,1) = trap\_I(f,x0,x1,n\_h);

for i = 2:iter+1

d(iter+1,i) = (4^(i-1)\*d(iter+1,i-1)-d(iter,i-1))/(4^(i-1)-1);

end

ea = abs(d(iter+1,iter+1)-d(iter,iter))/d(iter,iter);

iter = iter + 1;

if(ea <= es)

break;

end

end

end

1. Code\_main:

close all

clear all

format long

f = @(x) (exp(x).\*sin(x))./(1+x.^2);

x0 = 0;x1 = 2;

n = 12;

es = 0.005;

[d2,iter] = Romberg\_I(f,x0,x1,n,es)

str = [];

str\_r = [];

for i = 1:iter

str = [str;'O\_'+string(2\*i)+'\_'];

str\_r = [str\_r,'n ='+string(2^(i-1))];

end

T = array2table(round(d2,12));

for i = 1:iter

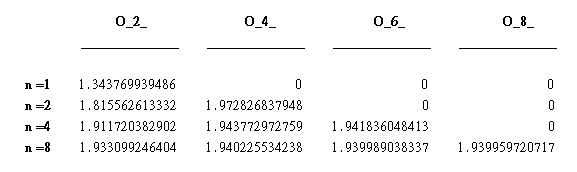
T.Properties.VariableNames{i} = char((str(i)));

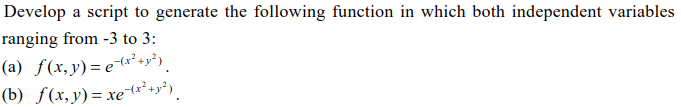
T.Properties.RowNames{i} = char(str\_r(i));

end

T

1. Result:



3. 

ANS :

1. Code\_main :

close all

clear all

format long

f\_a = @(x,y) exp(-(x.^2+y.^2))

f\_b = @(x,y) x.\*exp(-(x.^2+y.^2))

[xx,yy] = meshgrid(-3:0.01:3,-3:0.01:3);

figure(1)

mesh(xx,yy,f\_a(xx,yy))

figure(2)

mesh(xx,yy,f\_b(xx,yy))

4.



ANS:

1. Code\_ function:

function [y,tt] = Heun\_D(fy,t0,t1,y0,h)

tt = t0:h:t1;

n = length(tt);

y = y0\*ones(n,1);

for i = 1:n-1

k1 = fy(tt(i),y(i));

k2 = fy(tt(i+1),y(i)+h\*k1);

y(i+1) = y(i) + h\*0.5\*(k1+k2);

end

end

5.



ANS :

1. Code\_function :

function [y,tt] = midpoint\_D(fy,t0,t1,y0,h)

tt = t0:h:t1;

n = length(tt);

y = y0\*ones(n,1);

for i = 1:n-1

k1 = fy(tt(i),y(i));

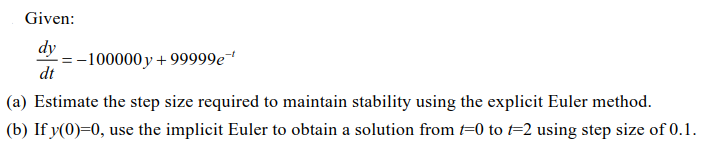
k2 = fy(tt(i)+0.5\*h,y(i)+0.5\*h\*k1);

y(i+1) = y(i) + h\*(k2);

end

end

6.



ANS :

1. Code\_function :

function [y,tt] = Eulr\_D(fy,t0,t1,y0,h)

tt = t0:h:t1;

n = length(tt);

y = y0\*ones(n,1);

for i = 1:n-1

k1 = fy(tt(i),y(i));

y(i+1) = y(i) + h\*(k1);

end

end

1. Code\_main:

close all

clear all

format long

fy = @(t,y) -1e5.\*y+99999.\*exp(-t)

tspan = [0,2]

y0 = 0;

h = 3e-5;

[y,tt] = Eulr\_D(fy,tspan(1),tspan(2),y0,h);

figure(1)

plot(tt,y);

xlabel("t")

ylabel("y")

title("h = "+string(h))

h = 2e-5;

[y,tt] = Eulr\_D(fy,tspan(1),tspan(2),y0,h);

figure(2)

plot(tt,y);

xlabel("t")

ylabel("y")

title("h = "+string(h))

h = 1e-5;

[y,tt] = Eulr\_D(fy,tspan(1),tspan(2),y0,h);

figure(3)

plot(tt,y);

xlabel("t")

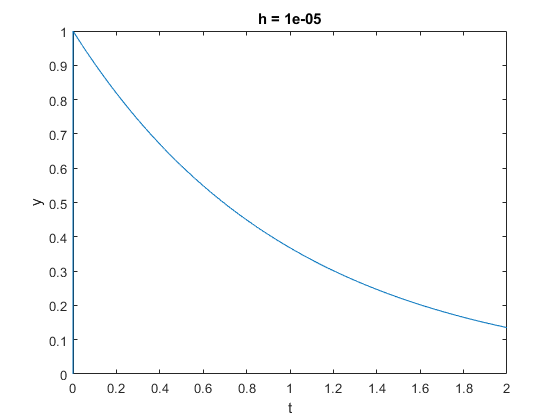
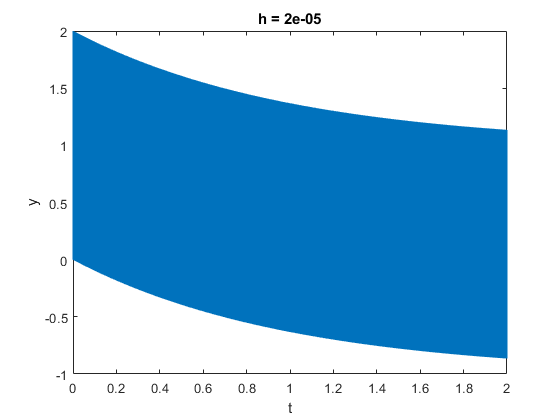
ylabel("y")

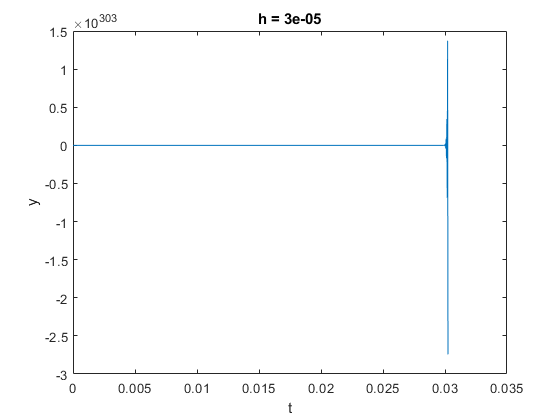
title("h = "+string(h))

1. Result:

令

，為了保持穩定度，|1-100000h| < 1 ，因此得

 ，h = 1e-5時保持穩定收斂，h=2e-5時振盪，h=3e-5發散。





ANS :

1. Code\_function :

function [y,tt] = Eulr\_D\_back(fy,t0,t1,y0,h)

tt = t0:h:t1;

n = length(tt);

y = y0\*ones(n,1);

iter = 50;

for i = 1:n-1

f\_z = @(y\_1) y\_1-h.\*fy(tt(i+1),y\_1)-y(i);

y(i+1) = newton\_back(f\_z,y(i),iter);

end

end

1. Code\_main:

close all

clear all

format long

fy = @(t,y) -1e5.\*y+99999.\*exp(-t)

tspan = [0,2]

y0 = 0;

h = 0.1;

[y,tt] = Eulr\_D\_back(fy,tspan(1),tspan(2),y0,h);

figure(1)

plot(tt,y);

xlabel("t")

ylabel("y")

1. Result:

