Numerical Analysis

HW3

Name: Leung Ko Tsun(梁高浚)

SID: T10902303

Diagram, text

Description automatically generatedDiagram

Description automatically generatedText, letter

Description automatically generatedA piece of paper with writing on it

Description automatically generated with medium confidenceText

Description automatically generated

I have tried several functions using the 3rd order finite difference formula, and calculated the average errors over selecting a different h.

1. f(x) = sin(x), f’’’(x) = -cos(x)

Chart, line chart

Description automatically generated

We can see that the truncation errors accumulate when h is larger. However, as I only adopted a relatively large fixed interval(0.01) to draw the above graph due to computational limit, the effect of rounding error cannot be shown using my computer. Normally, the total error should occur as an open-upward curve, and an optimal h can be found.

2. f(x) = exp(x), f’’’(x) = exp(x)

Chart, line chart

Description automatically generated

Similar behaviour can be observed.

A picture containing text, whiteboard

Description automatically generatedA whiteboard with writing on it

Description automatically generated with medium confidence

Phase portrait:

Diagram, engineering drawing

Description automatically generated

So, we can observe that the contour lines around those three fixed points ((-1,0), (0,0), (1,0)), are greater than 0. So that the vetor field on the contour is always directed inward, meaning that the point starting within the contour will stay in that contour forever. Thus it is globally stable.

A piece of paper with writing on it

Description automatically generated with medium confidenceA picture containing text, whiteboard

Description automatically generated

Appendix:matlab code:

%q2code

clear

close all;

fig = figure();

hs = 0:0.01:0.5;

errors = zeros(1,length(hs));

i = 1;

for h = hs

    x = 0:h:10;

    y = exp(x);

    %y = sin(x);

    %plot(x(4:end-4), y(4:end-4), 'Linewidth', 2)

    yderiv = exp(x);

    %yderiv = -cos(x);

    yderiv = yderiv(4:end-4);

    %plot(x(4:end-4), yderiv,'b-', 'Linewidth', 2)

    hold on

    %%% Use Forward differencing to compute value %%

    yderivest = (-1 \* y(2:end - 6) -1 \* y(3:end - 5) + 10 \* y(4:end-4) - 14\* y(5:end-3) + 7\*y(6:end-2) -1\*y(7:end-1)) ./ (4\*h^3);

    errors(i) = (sum(abs(yderiv - yderivest)) / length(yderiv));

    i = i + 1;

    %plot(x(4:end-4), yderivest, 'r-', 'Linewidth', 2)

end

plot(hs,errors,'r-', 'Linewidth', 2)

set(fig, 'color', 'white')

xlabel('h')

%xlabel('x')

ylabel('error')

%ylabel('y')

%legend('truth', 'estimated')

grid on

%q3code

clc;

clear;

close all;

roots = [-sqrt(3/5), 0, sqrt(3/5)];

weights = [5/9, 8/9, 5/9];

a = 8;

b = 30;

%f = @func1;

%f = @func2;

f = @func3;

result = 11\*(weights(1)\*f(11\*roots(1)+19)+weights(2)\*f(11\*roots(2)+19)+weights(3)\*f(11\*roots(3)+19));

x = 8:1:30;

y = arrayfun(f, x);

trape\_result = trapz(y);

func1\_truth = 11061.34;

func3\_truth = (201520\*log(3) + 205891132088088) / log(3);

gauss\_error = abs(func3\_truth - result)/func3\_truth;

trape\_error = abs(func3\_truth - trape\_result)/func3\_truth;

function fx = func1(x)

    fx = 2000\*log(140000/(140000 - 2100\*x))-9.8\*x;

end

function fx = func2(x)

    fx = 1/(x-8);

end

function fx = func3(x)

    fx = x^3 + 3^x + 2;

end

%q4code

clc;

clear;

close all;

x = linspace(-2,2);

y = linspace(-2,2);

[X, Y] = meshgrid(x,y);

Z = X.^4/4 - X.^2/4 + + Y.^2/2;

contour(X,Y,Z, 'ShowText', 'on')