DC011407 Choi Wan Ioi

The code and figures included in this report can be found on

https://github.com/Michael1119/finalForMATH3005

1. Problem 1

Consider the two-point boundary value problem .

The finite difference scheme is .

Where , , and .

Since , the finite difference scheme can be written as

,

* 1. Local Truncation Error and Consistency

The local truncation errors of the finite difference scheme can be obtained by replacing by .

By Taylor’s theorem, and

.

Therefore the local truncation errors of the finite difference scheme at , are

By Taylor’s theorem, and

.

The local truncation errors of the finite difference scheme at is

Since and , the finite difference scheme is consistent.

* 1. System matrix

The linear system of the finite difference scheme is

Therefore the system matrix is .

Since , the system matrix is not symmetric.

The system matrix is neither diagonally dominant nor weakly diagonally dominant because

, and for any .

* 1. Discussion

Notice that the first two rows of are identical, so is singular.

Therefore the linear system above either have no solution when or infinitely many solutions when .

This is because when , implies .

Also when , from the question.

cannot be equal to and simultaneously if , so the linear system has no solutions if .

1. Problem 2
   1. Derivation of the numerical scheme

Consider the two-point boundary value problem

where .

The central finite difference scheme with the ghost point method is

,

Where , , and .

Since , the first equation becomes when .

The third equation can be rewritten as .

Substitute into the second equation to obtain .

Therefore the scheme becomes

,

The linear system for the scheme is

where .

* 1. Theoretical order of accuracy

The local truncation errors of the scheme can be obtained by replacing by .

By Taylor’s theorem, and

.

Therefore the local truncation errors of the scheme at , are

By Taylor’s theorem, .

and .

Therefore .

The local truncation error of the scheme at is

Therefore and .

Let where and . Page 24 implies .

It can be shown that for all where is a constant independent of .

Therefore . Similarly, .

* 1. Numerical results

The exact solution is . Let be the computed solution with step size .

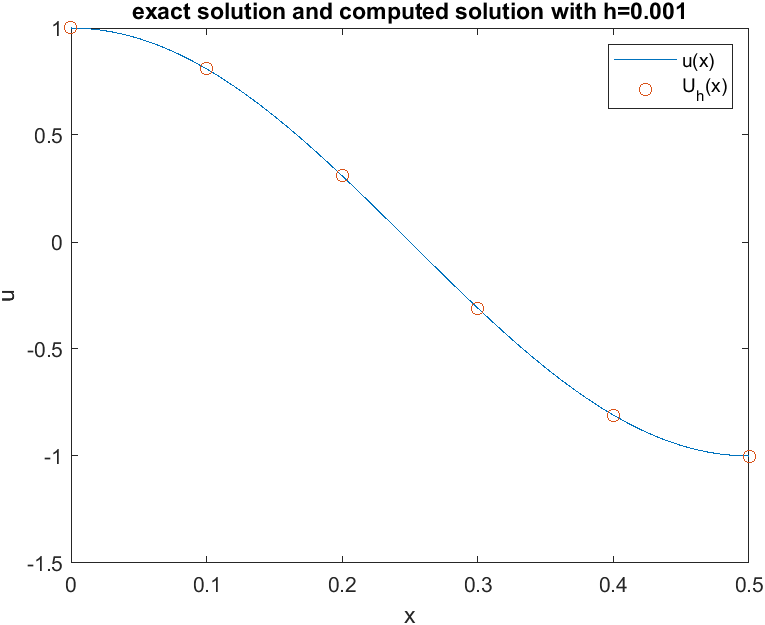
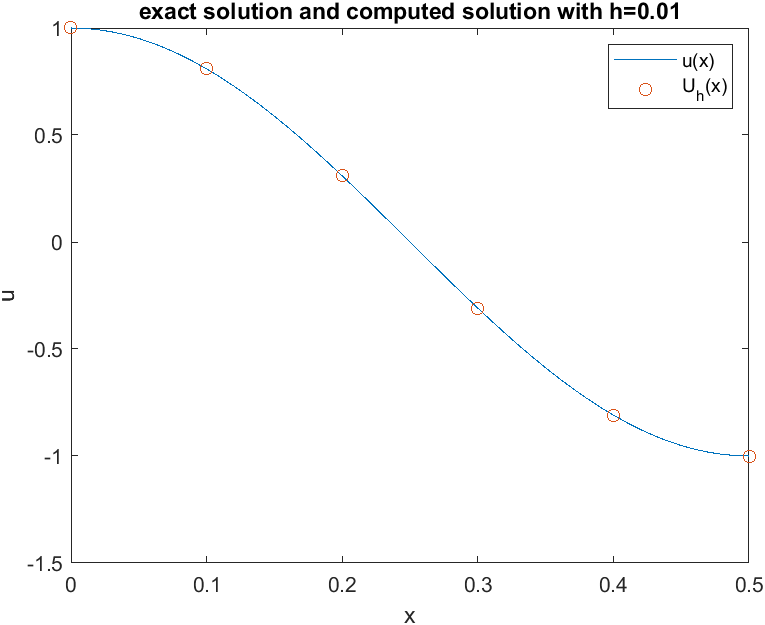
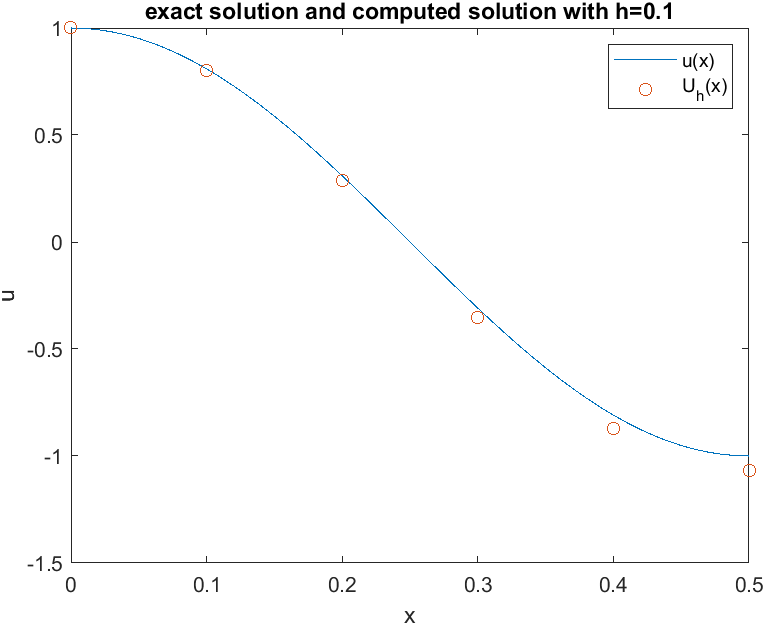
Denote the error between the exact solution and the computed solution with step size as .

Then , and .

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 0.1 | 0.1 | 0.80261 | 6.4091e-03 |
| 0.2 | 0.28583 | 2.3188e-02 |
| 0.3 | -0.35295 | 4.3929e-02 |
| 0.4 | -0.86972 | 6.0708e-02 |
| 0.5 | -1.0671 | 6.7117e-02 |
| 0.01 | 0.1 | 0.80895 | 6.2843e-05 |
| 0.2 | 0.30879 | 2.2737e-04 |
| 0.3 | -0.30945 | 4.3073e-04 |
| 0.4 | -0.80961 | 5.9526e-04 |
| 0.5 | -1.0007 | 6.5810e-04 |
| 0.001 | 0.1 | 0.80902 | 6.2831e-07 |
| 0.2 | 0.30901 | 2.2732e-06 |
| 0.3 | -0.30902 | 4.3065e-06 |
| 0.4 | -0.80902 | 5.9514e-06 |
| 0.5 | -1 | 6.5797e-06 |

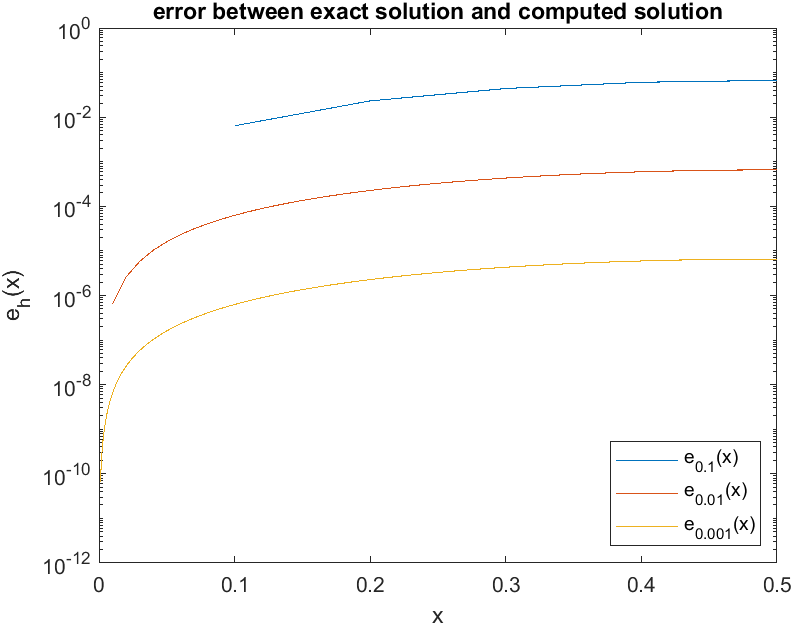
The exact solution and computed solution are shown in the following figure.

It can be seen that the computed solution is an accurate estimate of the exact solution.



The error between the exact solution and computed solution are shown in the following semi-log plot.

It can be seen that the error between the exact solution and computed solution decreases as step size decreases from 0.1 to 0.001.



From the figure above, it can be seen that the error between the exact solution and computed solution is the largest at the right endpoint, so .

Section 2.2 implies , so the expected value of is .

The actual value of is , which agrees with theoretical results.

|  |  |
| --- | --- |
|  | 6.5810e-04 |
|  | 6.5797e-06 |
|  | 10201 |

Alternatively, one can use log-log plot to verify the order of accuracy of the numerical scheme.

Suppose , then , so the function appears as a straight line with slope and intercept in the log-log plot.

To estimate , one can find the slope of the least square regression line .

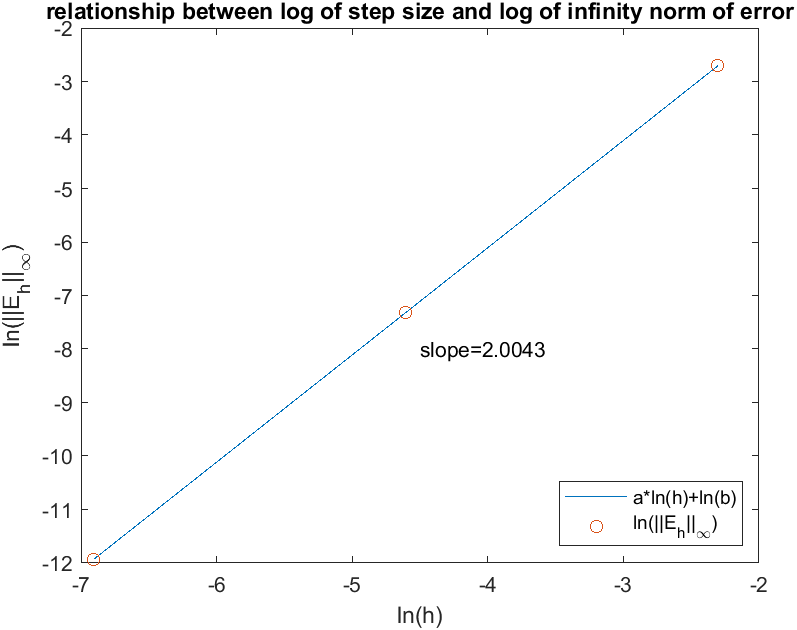
The slope and intercept of the least square regression line is found by solving the least square problem:

Where , and

It can be shown that satisfies the linear system .

By solving the linear system, one can obtain which agrees with theoretical results.

The errors and the regression line are shown in the following log-log plot.



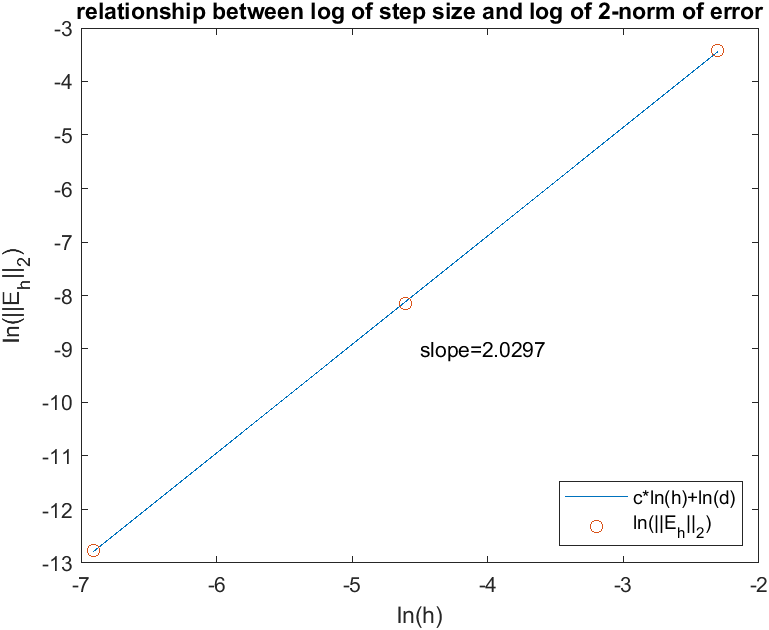
Suppose , then can be estimated by solving the least square problem:

Where , and

It can be shown that satisfies the linear system .

By solving the linear system, one can obtain which agrees with theoretical results.

The errors and the regression line are shown in the following log-log plot.



1. Appendix
   1. Central finite difference scheme with the ghost point method for problem 2

function [x,U] = ghost\_at\_b(a,b,ua,uxb,f,h)

n = (b-a)/h; h1=h\*h;

A = zeros(n,n);

x = zeros(n,1);

F = zeros(n,1);

for i=1:n-1

A(i,i) = -2/h1;

A(i+1,i) = 1/h1;

A(i,i+1)= 1/h1;

end

A(n,n) = -2/h1;

A(n,n-1) = 2/h1;

for i=1:n

x(i) = a+i\*h;

F(i) = feval(f,x(i));

end

F(1) = F(1) - ua/h1;

F(n) = F(n) - 2\*uxb/h;

U = A\F;

return

* 1. Main function for problem 2

clc

clear

f=@(x) -4\*pi^2\*cos(2\*pi\*x);

u=@(x) cos(2\*pi\*x);

[x1,U1]=ghost\_at\_b(0,0.5,1,0,f,0.1);

[x2,U2]=ghost\_at\_b(0,0.5,1,0,f,0.01);

[x3,U3]=ghost\_at\_b(0,0.5,1,0,f,0.001);

error1=abs(U1-u(x1));

error2=abs(U2-u(x2));

error3=abs(U3-u(x3));

format shortG

U1(mod(x1,0.1)<eps)

U2(mod(x2,0.1)<eps)

U3(mod(x3,0.1)<eps)

format shortE

error1(mod(x1,0.1)<eps)

error2(mod(x2,0.1)<eps)

error3(mod(x3,0.1)<eps)

norm(error1,"inf")

norm(error2,"inf")

norm(error3,"inf")

norm(error1,"inf")/norm(error3,"inf")

figure

plot([0;x2],[1;u(x2)],'-',[0;x1],[1;U1],'o')

title('exact solution and computed solution with h=0.1')

xlabel('x')

ylabel('u')

legend('u(x)','U\_h(x)')

figure

plot([0;x2],[1;u(x2)],'-',[0;x1],[1;U2(mod(x2,0.1)<eps)],'o')

title('exact solution and computed solution with h=0.01')

xlabel('x')

ylabel('u')

legend('u(x)','U\_h(x)')

figure

plot([0;x2],[1;u(x2)],'-',[0;x1],[1;U3(mod(x3,0.1)<eps)],'o')

title('exact solution and computed solution with h=0.001')

xlabel('x')

ylabel('u')

legend('u(x)','U\_h(x)')

figure

semilogy(x1,error1,'-',x2,error2,'-',x3,error3,'-')

title('error between exact solution and computed solution')

xlabel('x')

ylabel('e\_h(x)')

legend({'e\_{0.1}(x)','e\_{0.01}(x)','e\_{0.01}(x)'},'Location','southeast')

x=transpose(log([0.1,0.01,0.001]));

y1=transpose(log([error1(end),error2(end),error3(end)]));

y2=transpose(log([sqrt(0.1)\*norm(error1,2),sqrt(0.01)\*norm(error2,2),sqrt(0.001)\*norm(error3,2)]));

A=[x,ones(length(x),1)];

par1=(A'\*A)\(A'\*y1);

par2=(A'\*A)\(A'\*y2);

a=par1(1);

b=par1(2);

c=par2(1);

d=par2(2);

figure

plot(log([0.1,0.01,0.001]),a\*log([0.1,0.01,0.001])+b,'-',x,y1,'o')

title('relationship between log of step size and log of infinity norm of error')

xlabel('ln(h)')

ylabel('ln(||E\_h||\_\infty)')

legend({'a\*ln(h)+ln(b)','ln(||E\_h||\_\infty)'},'Location','southeast')

text(-4.5,-8,strcat('slope=',string(a)))

figure

plot(log([0.1,0.01,0.001]),c\*log([0.1,0.01,0.001])+d,'-',x,y2,'o')

title('relationship between log of step size and log of 2-norm of error')

xlabel('ln(h)')

ylabel('ln(||E\_h||\_2)')

legend({'c\*ln(h)+ln(d)','ln(||E\_h||\_2)'},'Location','southeast')

text(-4.5,-9,strcat('slope=',string(c)))