4. Use 2D Finite Differences to solve the following BVP

$u_{xx} + u_{yy} = 0, \quad 0 < x < 1, 0 < y < 1$

subject to $u(x, y) = 0$ on the top, left, and ride sides of the square domain with $u(x, y) = \sin(\pi x)$ for $y = 0$ (i.e. the bottom of the square). Use 5 grid points (3 interior points) in each of the x and y directions. Code up your FD method into Matlab and plot the solution. Does your FD solution improve with a finer grid? Is it possible to use too many points? Discuss. Think about how you could compute the 'true' analytic solution

For Finite Differences Method

$$\frac{1}{h^2}(u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) + \frac{1}{h^2}(u_{i+1,j} - 2u_{i,j} + u_{i+1,j}) = 0$$

$$\frac{1}{h^2}(u_{i,j+1} - 4u_{i,j} + u_{i,j-1} + u_{i+1,j} + u_{i+1,j}) = 0$$

$$\frac{u_{i,j+1}}{h^2} - \frac{4u_{i,j}}{h^2} + \frac{u_{i,j-1}}{h^2} + \frac{u_{i+1,j}}{h^2} + \frac{u_{i+1,j}}{h^2} = 0$$

$$\frac{u_{i,j+1}}{h^2} + \frac{u_{i,j-1}}{h^2} + \frac{u_{i+1,j}}{h^2} + \frac{u_{i+1,j}}{h^2} - \frac{4u_{i,j}}{h^2} = 0$$

$$\frac{u_{i,j+1}}{h^2} + \frac{u_{i,j-1}}{h^2} + \frac{u_{i+1,j}}{h^2} + \frac{u_{i+1,j}}{h^2} - \frac{4}{h^2}u_{i,j} = 0$$

$$u_{i,j+1} + u_{i,j-1} + u_{i+1,j} + u_{i+1,j} = 4u_{i,j}$$

$$\frac{1}{4}(u_{i,j+1} + u_{i,j-1} + u_{i+1,j} + u_{i+1,j}) = u_{i,j}$$

$u_{i,j} = \frac{1}{4}(u_{i,j+1} + u_{i,j-1} + u_{i+1,j} + u_{i+1,j})$ for each i,j in the interval points

So for 5 grid points (3 interior points) there will be 9 unknows

The points are known if they are on the side.

Therefore we need to a 9 by 9 matrix to solve all of those 9 point.

Which is:

V*a=b

Where V is 9 by 9, a is vector contain 9 unknown interior point, and b is the known point from BC.

If we write up the equation for $i = 2$ and $j = 2$

$$u_{2,2} = \frac{1}{4}(u_{2,3} + u_{2,1} + u_{3,2} + u_{3,2})$$

$$4u_{2,2} = u_{2,3} + u_{2,1} + u_{3,2} + u_{3,2}$$

$$4u_{2,2} - u_{2,3} - u_{2,1} - u_{3,2} - u_{3,2} = 0$$

Since $u_{2,1}$is on the boundary

$$4u_{2,2} - u_{2,3} - u_{3,2} - u_{3,2} = u_{2,1}$$

If we arrange the unknown like:

$$a = \begin{bmatrix} u_{2,2} \\ u_{2,3} \\ u_{2,4} \\ u_{3,2} \\ u_{3,3} \\ u_{3,4} \\ u_{4,2} \\ u_{4,3} \\ u_{4,4} \end{bmatrix}$$

So the first row of V is

$$\begin{bmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The first element of b is $u_{2,1}$

And then if we write up for the second row

$$u_{2,3} = \frac{1}{4}(u_{2,4} + u_{2,2} + u_{3,3} + u_{3,3})$$

$$4u_{2,3} - u_{2,4} - u_{2,2} - u_{3,3} - u_{3,3} = 0$$

So the second row of V is

$$\begin{bmatrix} -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

we could see the number 4 and -1 appears lots of times.

4 is the coeffecient on the point we need to know and -1 is the coeffecient on the point left, right, up and down of it the point on the BC will move to the correspounding element in b.

If we form the above in a matrix:

$$k = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

So we can keep load the matrix k into a all zero matrix and cut off the BC point then flat the matrix left to a row.

Which will become the row for matrix V.

The method I generate the matrix V, b and calculate result is:

grid point on x=nx

grid point on x=ny

set $k = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

step 1: from the point $(2,2)$ to $(\mathrm{nx}-1, \mathrm{ny}-1)$:

step 2: build up an all 0 matrix M_z with dimension nx by ny (in this case is 5 by 5).

change the point and all the point around it (in total 9 points) to matrix k.

step 3: cut the matrix from $M_z(2,2)$ to $M_z(\mathrm{nx}-1, \mathrm{ny}-1)$

step 4 flat the cut matrix to 1 row and restore is to the matrix V.

step 5: for correspomding element in b, we could just add up all the point around it from the matrix of IC.

(therefore only BC will be added since the other point is 0)

step 6: calculate a by V\b then reform a to $(\mathrm{nx}-1)$ by $(\mathrm{ny}-1)$. Then put reformed a back into IC matrix.

By compute true solution

PDE $\quad u_{\mathrm{xx}} = -u_{\mathrm{yy}} \qquad 0 < x < 1, \quad 0 < y < 1$

BCs $\quad \begin{cases} u(1,y) = 0 & 0 < y < 1 \\ u(x,1) = 0 & 0 < x < 1 \end{cases}$

IC $\quad \begin{cases} u(0,y) = 0 \\ u(x,0) = \sin(\pi x) & 0 \le x \le 1 \end{cases}$

$u = X(x)Y(y)$

$X''(x)Y(y) = -X(x)Y''(y)$

$\frac{X''(x)}{X(x)} = -\frac{Y''(y)}{Y(y)} = k$

$\frac{X''(x)}{X(x)} = k$

$\frac{Y''(y)}{Y(y)} = -k$

Case 1 $k > 0$, $k = \lambda^2$,

$X''(x) = \lambda^2 X(x)$

$X(x) = C_1 e^{\lambda x} + C_2 e^{-\lambda x}$

IC $\begin{cases} u(0,y)=0 \\ u(x,0)=\sin(\pi x) \quad 0\le x\le 1 \end{cases}$

$X(0)=C_1e^{\lambda 0}+C_2e^{-\lambda 0}=C_1+C_2=0$

$C_1=-C_2$

And by BCs

BCs $\begin{cases} u(1,y)=0 \quad 0<y<1 \\ u(x,1)=0 \quad 0<x<1 \end{cases}$

$X(1)=C_1e^{\lambda 1}+C_2e^{-\lambda 1}=-C_2e^{\lambda 1}+C_2e^{-\lambda 1}=C_2(e^{-\lambda}-e^{\lambda})=0$

So either $C_2=0$ or $(e^{-\lambda}-e^{\lambda})=0$

If $C_2=0, C_1=0$

Which $X(x)=0$ and then $u(x,y)=0$

So $C_2=0$ is not true, $(e^{-\lambda}-e^{\lambda})=0$

$e^{-\lambda}-e^{\lambda}=0,$

$e^{-\lambda}=e^{\lambda}$

The only λ that satisfies this equation is $\lambda=0$

Which means $X(x)=C_1e^{0x}+C_2e^{-0x}$

Which also means $X(x)=0$ and then $u(x,t)=0$

So $k>0$ is not true.

Case 2 $k=0,\ k=\lambda^2=0$, so $\lambda=0$

$X(x)=C_1e^{\lambda x}+\mathrm{x}C_2e^{\lambda x}=C_1+C_2x$

Plug in ICs

IC $\begin{cases} u(0,y)=0 \\ u(x,0)=\sin(\pi x) \quad 0\le x\le 1 \end{cases}$

$X(0)=C_1+C_20=C_1=0$

and BCs:

BCs $\begin{cases} u(1,y)=0 \quad 0<y<1 \\ u(x,1)=0 \quad 0<x<1 \end{cases}$

$X(1) = C_1 + C_2 = C_2 = 0$

So $C_2 = 0$

Therefor both C_1 and C_2 equal to 0

$X(x) = 0$ for any situation and $u(x,t) = 0$

So $k = 0$ is not true.

Case 3 $k < 0$, $k = -\lambda^2, r_{1,2} = \pm\sqrt{-k}\,i = \pm\lambda i$

$a = 0, b = \lambda$

$X(x) = C_1\cos(\lambda x) + C_2\sin(\lambda x)$

Plug in ICs

IC $$\begin{cases} u(0,y) = 0 \\ u(x,0) = \sin(\pi x) \quad 0 \le x \le 1 \end{cases}$$

$X(0) = C_1\cos(0) + C_2\sin(0) = C_1 = 0$

and BCs:

BCs $$\begin{cases} u(1,y) = 0 \quad 0 < y < 1 \\ u(x,1) = 0 \quad 0 < x < 1 \end{cases}$$

$X(1) = C_2\sin(\lambda) = 0$

$\lambda = n\pi$

$X(x) = C_n\sin(n\pi x)$

Then for $Y(t), -\frac{Y''(y)}{Y(y)} = k$

for $k < 0$

$Y(y) = a_n e^{\lambda y} + b_n e^{-\lambda y}$

BCs $$\begin{cases} u(1,y) = 0 \quad 0 < y < 1 \\ u(x,1) = 0 \quad 0 < x < 1 \end{cases}$$

$Y(1) = a_n e^{\lambda} + b_n e^{-\lambda} = 0$

So

$$U(x,y) = X(x)Y(y) = \sum_{1}^{\infty} C_n \sin(n\pi x)\left[a_n e^{\lambda y} + b_n e^{-\lambda y}\right]$$

Let $C_n a_n = A_n C_n b_n = B_n$

$$u(x,y) = \sum_{n=1}^{\infty} \sin(n\pi x)\left[A_n e^{n\pi y} + B_n e^{-n\pi y}\right]$$

Then plug IC in:

IC $$\begin{cases} u(0,y) = 0 \\ u(x,0) = \sin(\pi x) \quad 0 \le x \le 1 \end{cases}$$

$$u(x,0) = \sum_{n=1}^{\infty} \sin(n\pi x)[A_n + B_n] = \sin(\pi x)$$

n only could be 1

$$A_n + B_n = 1$$
$$A_n = 1 - B_n$$

then from BCs

BCs $$\begin{cases} u(1,y) = 0 \quad 0 < y < 1 \\ u(x,1) = 0 \quad 0 < x < 1 \end{cases}$$

$$u(x,1) = \sum_{n=1}^{\infty} \sin(n\pi x)\left[A_n e^{n\pi} + B_n e^{-n\pi}\right] = 0$$

since $\sin(n\pi x)$ could not be 0

$$A_n e^{n\pi} + B_n e^{-n\pi} = 0$$
$$(1 - B_n)e^{n\pi} + B_n e^{-n\pi} = 0$$
$$e^{n\pi} - B_n e^{n\pi} + B_n e^{-n\pi} = 0$$
$$e^{n\pi} - B_n(e^{n\pi} + e^{-n\pi}) = 0$$
$$e^{n\pi} = B_n(e^{n\pi} + e^{-n\pi})$$
$$\frac{e^{n\pi}}{e^{n\pi} + e^{-n\pi}} = B_n$$
$$B_n = 1 - \frac{1}{e^{2n\pi} + 1}$$

So $A_n = \frac{1}{e^{2n\pi} + 1}$

since n only could be 1

$$u(x,y) = \sin(\pi x)\left[\frac{1}{e^{2n\pi}+1}e^{\pi y} + \left(1 - \frac{1}{e^{2n\pi}+1}\right)e^{-\pi y}\right] \qquad (10)$$

```
nx=5;%set grid point of x is 5
ny=5;%set grid point of y is 5

%general initial condition
x=linspace(0,1,nx);
y=linspace(0,1,ny);
u=zeros(nx,ny);
u(ny,:)=sin(pi*x);%set u(x,y)=sin(pi*x)
```

```
u = 5×5
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0         0
         0    0.7071    1.0000    0.7071    0.0000
```

```
[uk]=FD(nx,ny,u)%calculate result
```

```
uk = 5×5
         0    0.7071    1.0000    0.7071    0.0000
         0    0.3318    0.4693    0.3318         0
         0    0.1509    0.2134    0.1509         0
         0    0.0584    0.0825    0.0584         0
         0         0         0         0         0
```

```
mesh(uk)
title("solution from Finite Difference Method with grid 5")
xlabel('x')
ylabel('y')
zlabel('z')

%test for more grid point
nx50=50;
ny50=50;

x50=linspace(0,1,nx50);
y50=linspace(0,1,ny50);
u50=zeros(nx50,ny50);
u50(ny50,:)=sin(pi*x50);

[uk50]=FD(nx50,ny50,u50)

mesh(uk50)
title("solution from Finite Difference Method with grid 50")
xlabel('x')
ylabel('y')
zlabel('z')
```

From just looking at the graph we could see the graph with more grid is more smooth than it with less grid.

Now for testing the accuracy. We need to compute, the difference between the approximate value with the true value with different numbers of grid.

```
%true value for 5 grid point
[xx,yy]=meshgrid(x,y);
%equation (10)
tu=sin(pi*xx).*(1/(exp(2*pi)+1)*exp(pi.*yy)+(1-1/(exp(2*pi)+1))*exp(-pi.*yy))
%plot the result
mesh(tu)
%calculate the average difference between true value and approximate value
Diff_grid_5 = sum(abs(tu-uk),'all')/((nx-1)*(ny-1))

%true value for 50 grid point
[xx50,yy50]=meshgrid(x50,y50);
%equation (10)
tu50=sin(pi*xx50).*(1/(exp(2*pi)+1)*exp(pi.*yy50)+(1-1/(exp(2*pi)+1))*exp(-pi.*yy50));
%plot the result
mesh(tu50)
%calculate the average difference between true value and approximate value
Diff_grid_50 = sum(abs(tu50-uk50),'all')/((nx-1)*(ny-1))
```

So as result the more grid we have the less accuracy we will have.

```
function [uk] = FD(nx, ny, u)
%function of processing finite difference
%set k
    k = [0 -1 0;
        -1 4 -1;
        0 -1 0];
    V = [];
    b = [];
%step 1 loop from (2,2) to (nx-1),(ny-1)
    for j = 2:ny - 1
        for l = 2:nx - 1
            %step 2 set all 0 matrix
            ut = zeros(ny, nx);
            %step 2 change the points around to k
            ut(j - 1:j + 1, l - 1:l + 1) = k;
            %step 3 cut the matrix from (2,2) to (nx-1),(ny-1)
            uc = ut(2:nx - 1, 2:ny - 1);
            %step 4 flat the cut matrix to 1 row and store it to V
            V = [V;reshape(uc', 1, (nx - 2)*(ny - 2))];
            %step 5 calculate b by adding the around point from IC matrix
            %together
            b = [b;u(j + 1, l) + u(j, l - 1) + u(j - 1, l) + u(j, l + 1)];
        end
    end
    uin = V\b;%step 6 calculate the result.
    u(2:nx - 1, 2:ny - 1) = reshape(uin, (nx - 2), (ny - 2))';
    %put the solution back into IC matrix.
    uk = flipud(u);
end
```