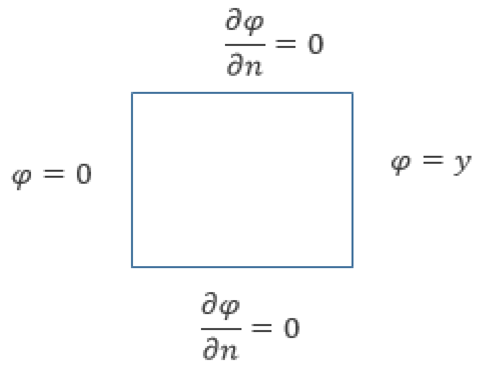
MATH5004 LAB 3

Finite Difference Method for 2D BVP

Laplace's equation

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} = 0$$

subject to BCs:



is solved using the 5-point finite difference stencil using both implicit matrix inversion techniques and explicit iterative solutions. The boundary conditions used include both Dirichlet and Neumann type conditions.

The following MATLAB program implements this procedure

```
% Solving the 2-D Laplace's equation by the Finite Difference Method
% Numerical scheme used is a second order central difference in space
% (5-point difference)
%Specifying parameters
nx=60;                      %Number of steps in space(x)
ny=60;                      %Number of steps in space(y)
niter=10000;                %Number of iterations
dx=2/(nx-1);                %Width of space step(x)
dy=2/(ny-1);                %Width of space step(y)
x=0:dx:2;         %Range of x(0,2) and specifying the grid points
y=0:dy:2;         %Range of y(0,2) and specifying grid points
%Initial Conditions
p=zeros(ny,nx);           %Preallocating p
pn=zeros(ny,nx);          %Preallocating pn
%Boundary conditions
p(:,1)=0;
p(:,nx)=y;
p(1,:)=p(2,:);            %Neumann conditions
p(ny,:)=p(ny-1,:);        %same as above
```

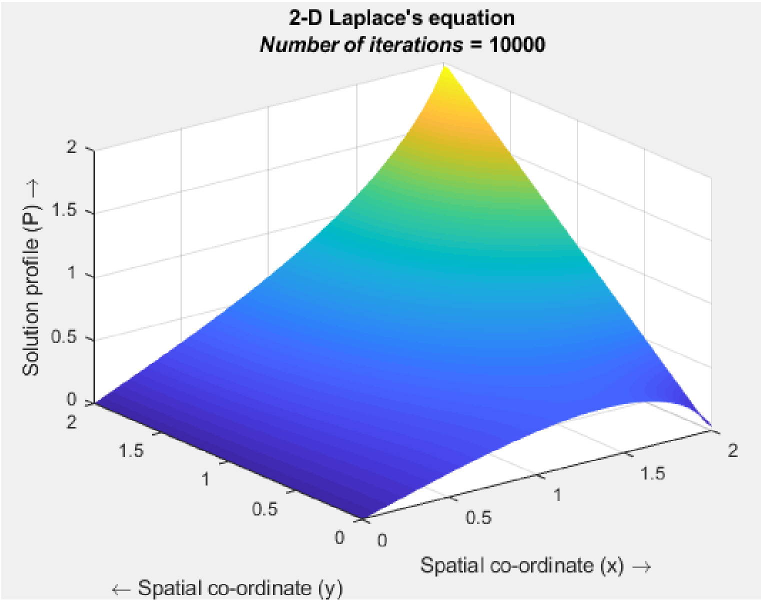
```
%Explicit iterative scheme with C.D in space (5-point difference)
j=2:nx-1;
i=2:ny-1;
for it=1:niter
  pn=p;
  p(i,j)=((dy^2*(pn(i+1,j)+pn(i-1,j)))+(dx^2*(pn(i,j+1)+...
          pn(i,j-1))))/(2*(dx^2+dy^2));
  %Boundary conditions (Neumann conditions)
  p(:,1)=0;
  p(:,nx)=y;
  p(1,:)=p(2,:);
  p(ny,:)=p(ny-1,:);
end
%Plotting the solution
surf(x,y,p,'EdgeColor','none');
shading interp
title({'2-D Laplace''s equation';['{\itNumber of iterations}=',...
        num2str(it)]})
xlabel('Spatial co-ordinate (x) \rightarrow')
ylabel('{\leftarrow} Spatial co-ordinate (y)')
zlabel('Solution profile (P) \rightarrow')
```

The above problem can be solved by an implicit second order central difference in space (5-point difference) which can be solved by the following program

```
%Specifying parameters
nx=100;                      %Number of steps in space(x)
ny=100;                      %Number of steps in space(y)
dx=2/(nx-1);                 %Width of space step(x)
dy=2/(ny-1);                 %Width of space step(y)
x=0:dx:2;                    %Range of x(0,2) and specifying the grid points
y=0:dy:2;                    %Range of y(0,2) and specifying the grid points
UW=0;                        %x=0 Dirichlet B.C
UE=y;                        %x=L Dirichlet B.C
US=0;                        %y=0 Dirichlet B.C
UN=0;                        %y=L Dirichlet B.C
UnW=0;                       %x=0 Neumann B.C (du/dn=UnW)
UnE=0;                       %x=L Neumann B.C (du/dn=UnE)
UnS=0;                       %y=0 Neumann B.C (du/dn=UnS)
UnN=0;                       %y=L Neumann B.C (du/dn=UnN)
```

```
%Pre-allocating u
u=zeros(nx,ny);

%B.C vector
bc=zeros(nx-2,ny-2);
bc(1,:)=UW/dx^2; bc(nx-2,:)=UE(2:ny-1)/dx^2;  %Dirichlet B.Cs
%bc(:,1)=US/dy^2; bc(:,ny-2)=UN/dy^2;  %Dirichlet B.Cs
%bc(1,:)=-UnW/dx; bc(nx-2,:)=UnE/dx;  %Neumann B.Cs
bc(:,1)=-UnS/dy; bc(:,ny-2)=UnN/dy;  %Neumann B.Cs
%B.Cs at the corners:
bc(1,1)=UW/dx^2-UnS/dy; bc(nx-2,1)=UE(2)/dx^2-UnS/dy;
bc(1,ny-2)=UW/dx^2+UnN/dy; bc(nx-2,ny-2)=UE(ny-1)/dx^2+UnN/dy;

%Calculating the coefficient matrix for the implicit scheme
Ex=sparse(2:nx-2,1:nx-3,1,nx-2,nx-2);
Ax=Ex+Ex'-2*speye(nx-2);         %Dirichlet B.Cs
%Ax(1,1)=-1; Ax(nx-2,nx-2)=-1;  %Neumann B.Cs
Ey=sparse(2:ny-2,1:ny-3,1,ny-2,ny-2);
Ay=Ey+Ey'-2*speye(ny-2);         %Dirichlet B.Cs
Ay(1,1)=-1; Ay(ny-2,ny-2)=-1;  %Neumann B.Cs
A=kron(Ay/dy^2,speye(nx-2))+kron(speye(ny-2),Ax/dx^2);
%%
S=zeros(nx-2,ny-2);              %Source term
S=reshape(S-bc,[],1);
S=A\S;
S=reshape(S,nx-2,ny-2);
u(2:nx-1,2:ny-1)=S;

%Boundary conditions
%Dirichlet:
u(1,:)=UW;
u(nx,:)=UE;
%u(:,1)=US;
%u(:,ny)=UN;
%Neumann:
%u(1,:)=u(2,:)-UnW*dx;
%u(nx,:)=u(nx-1,:)+UnE*dx;
u(:,1)=u(:,2)-UnS*dy;
u(:,ny)=u(:,ny-1)+UnN*dy;

%Plotting the solution
surf(x,y,u','EdgeColor','none');
shading interp
title('2-D Laplace''s equation')
xlabel('Spatial co-ordinate (x) \rightarrow')
ylabel('{\leftarrow} Spatial co-ordinate (y)')
zlabel('Solution profile (P) \rightarrow')
```

Result

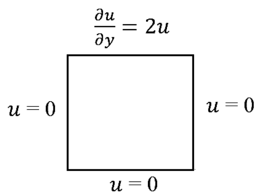


Assignment I

Question 1. (LAB-WK3)

Implement the above MATLAB program for the solution of 2-D BVP:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = x - 2y \quad \text{on a square with boundary condition as shown.}$$



Note: Assignments I & II (50%): Assignment Questions will be given weekly.

In this week, Questions 1 (LAB-WK3) is a part of Assignment I, please submit a document file with MATLAB code via Blackboard by the due date of Assignment I on Friday 11 September 2020