```
close all;
clear all;
% This codes solves the 1D Diffusion equation: u_t=D*u_xx
% for Dirichlet and Neumann boundary conditions
% u(0,t)=A or du(0,t)/dx = A, and u(Lx,0)=B or du(Lx,t)/dx = B
% and initial conditions u(x,0)=f(x)
% Set the diffusion coefficient
alpha_s = 1.0;

% Domain size
Lx = 1;

% Boundary conditions
% Boundary type (D for Dirichlet, N for Neumann):
Tleft_type = 'D';
Tright_type = 'D';
% Boundary value:
Tleft = 1;
Tright = 0;

% Initial condition:
f = @(x) cos(pi*x/2);

% Number of grid points in x direction
Nx = 51;

% Step size in x direction
dx = Lx / ( Nx - 1 );

% Create the mesh
x = linspace(0,Lx,Nx);

% Specify a time interval
T0 = 0; Tfinal = 0.5;

% Divide the time interval to Nt+1 sub intervals
Nt = 10001;
t = linspace(T0,Tfinal,Nt);

% The size of the time step
dt = (Tfinal - T0) / (Nt-1);

% Generate a matrix of solution u
u =zeros( Nx, Nt);

% Set the initial conditions
u( :, 1 ) = f(x);

% Set the boundary conditions
% left BC, x = 0:
if Tleft_type == 'D';
    u( 1, : ) = Tleft;
```

```
end
% right BC, x = Lx
if Tright_type == 'D';
    u( Nx, : ) = Tright;
end

% Set a forcing vector for inhomogeneous Neumann B.C.
F=zeros(Nx,1);
if Tleft_type == 'N'
 F(1) = -2*Tleft*alpha_s*dt/dx;
end

if Tright_type == 'N'
 F(Nx) = 2*Tright*alpha_s*dt/dx;
end

% Set up the coefficient matrix
D=zeros( Nx, Nx);
C = alpha_s * dt /dx^2;
for  i = 2:Nx-1
    D( i, i-1 ) = C;
    D( i, i ) = 1 - 2*C;
    D( i, i+1 ) =  C;
end

% Now, put in the boundary conditions
% x = 0
if Tleft_type == 'D'
    D( 1, 1) = 1;
elseif Tleft_type == 'N'
    D( 1, 1 ) = 1- 2*C;
    D( 1, 2 ) = 2*C;
end

% x = L

if Tright_type == 'D'
    D( Nx, Nx) = 1;
elseif  Tright_type == 'N'
    D( Nx, Nx-1 ) = 2*C;
    D( Nx, Nx ) = 1 -2*C;
end

% Time step the PDE
for n = 1:Nt-1
  u( :, n+1 ) =  D*u( :, n)+ F;
end
```

plotting the results

```
figure;

for n =1:round(Nt/50):Nt
```

```
    plot(x,u(:,n)); hold all;

    % formatting
    xlabel('x','FontSize',20);
    ylabel('u(x,t)','FontSize',20);
    set(gca,'FontSize',20);
    xlim( [ 0 Lx]);


end
print -dpdf 'Diffusion.pdf';
```

Published with MATLAB® R2014a