.

1 Introduction

Each group will be assigned a specific set of initial and boundary conditions and solve the two-dimensional heat equation both analytically and numerically. Let $u = u(x, y, t)$ satisfy the following:

$$\frac{\partial u}{\partial t} = k\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \qquad 0 < x < L, \quad 0 < y < W, t > 0, \tag{1}$$

$$u(x, y, 0) = f(x, y) \qquad 0 < x < L, \quad 0 < y < W. \tag{2}$$

Group	L	W	k	$x = 0$	$x = L$	$y = 0$	$y = W$	$f(x, y)$
A	1	1	$\frac{1}{10}$	$u = 0$	$u = 0$	$u = 0$	$u = 0$	$xy(1-x)(1-y)$
B	2	1	$\frac{1}{5}$	$u = 0$	$u = 0$	$u = 0$	$u_y = 0$	$xy(2-x)\left(1-\frac{y}{2}\right)$
C	1	2	$\frac{1}{10}$	$u_x = 0$	$u = 0$	$u = 0$	$u = 0$	$y(1-x^2)(2-y)$
D	1	2	$\frac{1}{5}$	$u_x = 0$	$u_x = 0$	$u = 0$	$u = 0$	$x^2y^2\left(1-\frac{2}{3}x\right)(2-y)$
E	1	1	$\frac{1}{10}$	$u_x = 0$	$u = 0$	$u_y = 0$	$u = 0$	$(1-x^2)(1-y^2)$
F	1	1	$\frac{1}{5}$	$u_x = 0$	$u = 0$	$u = 0$	$u_y = 0$	$y(1-x^2)\left(1-\frac{y}{2}\right)$
G	1	1	$\frac{1}{5}$	$u_x = 0$	$u_x = 0$	$u = 0$	$u_y = 0$	$x^2y\left(1-\frac{2}{3}x\right)\left(1-\frac{y}{2}\right)$
H	2	2	$\frac{1}{15}$	$u = 0$	$u_x = 0$	$u = 0$	$u = 0$	$xy(4-x)(2-y)$
I	2	1	$\frac{1}{15}$	$u = 0$	$u = 0$	$u_y = 0$	$u = 0$	$x(2-x)(1-y^2)$

Table 1: These are the parameters and conditions needed for your assigned group: length, L, width, W, heat constant, k, boundary conditions, and initial condition.

2 1D Heat Equation

For the first part of the project you will numerically solve the one-dimensional heat equation. Below is a copy of the MATLAB code you will be given to carry out this part of the project.

```
% Solution of the Heat Equation Using a Forward Difference Scheme

% Initialize Data
%     Length of Rod, Time Interval
%     Number of Points in Space, Number of Time Steps

clear
L=1;
```

```
T=0.1;
k=1;
N=50;
M=500;
dx=L/N;
dt=T/M;
alpha=k*dt/dx^2;

t0 = cputime;  % Combine with t1 to time the routine

% Position

for i=1:N+1
    x(i)=(i-1)*dx;
end

% Initial Condition

for i=1:N+1
    u0(i)=x(i)*(1-x(i));
end

% Partial Difference Equation (Numerical Scheme)

for j=1:M
   for i=2:N
      u1(i)=u0(i)+alpha*(u0(i+1)-2*u0(i)+u0(i-1));
   end
   u1(1)=0;
   u1(N+1)=0;
   u0=u1;
   % Plot solution
   hold on
   if mod(j,10)==0
       plot(x, u1);
   end
   hold off
end

t1=cputime;
telapsed = t1-t0
```

3 2D Heat Equation

For the second part of the project you will solve the two-dimensional heat equation by constructing the solution to the initial-boundary value problem you are assigned from Table 1. You will find the product solutions $\phi_{n,m}(x, y, t)$ and the Fourier coefficients, $c_{n,m}$. Be careful as problems D and G also have coefficients $c_{0,m}$.

Below is a copy of MATLAB code for plotting the exact solution. Besides generating a 3D plot of the solution evolving in time, frames are captured and placed in a movie file. The movie can be played using one of the commands at the end of the file.

```
% Solution of the 2D Heat Equation Using the series solution.
% u_t = k (u_xx + u_yy)

% Initialize Data
%       L, W   = Length and Width of Playe,
%       T      = for Time Interval [0, T]
%       Nx, Ny = Number of Points in Space Grid,
%       M      = Number of Time Steps
%       dx, dy = Delta x and Delta y.
% Set your own values of L, W, T, k
clear
L=1;
W=2;
T=1;
k=1/10;
Nx=20;
Ny=20;
dx=L/Nx;
dy=W/Ny;
M=20;
dt = T/M;

% Spatial grid
[x,y]=meshgrid(0:dx:L,0:dy:W);

% Initialize u % Change to your initial condition
u=zeros(Nx+1,Ny+1);
for n=1:10
    for m=1:10
        u=u+sin(n*pi*x/L).*sin(m*pi*y/W)/n^2/m^2;
    end
end
H=max(max(u));

% Plot initial condition
surf(x,y,u,'FaceColor','red','EdgeColor','none')
camlight left;
lighting phong
xlabel('x')
ylabel('y')
title('Solution at t = 0')
axis([0,L,0,W,0,H])
frame=1;
```

```
Mov(frame)=getframe(gcf);
pause(0.5)

% Time evolution
for j=1:M
    u=zeros(Nx+1,Ny+1);
    t=j*dt;
    for n=1:10
        for m=1:10
            lambda=(n*pi/L)^2+(m*pi/W)^2;    % Change lambda
            u=u+sin(n*pi*x/L).*sin(m*pi*y/W)/n^2/m^2*exp(-k*lambda*t);
        end
    end

   % Plot 3D solution every 10 time steps
   %if mod(m,10)==0
       frame=frame+1;
       surf(x,y,u,'FaceColor','red','EdgeColor','none')
       camlight left;
       lighting phong
       xlabel('x')
       ylabel('y')
       title(['Solution at t = ' num2str(t)])
       axis([0,L,0,W,0,H])
       Mov(frame)=getframe(gcf);
       pause (0.5)
   %end
end

% Extra - show or create a movie
%    movie(gcf, Mov)         % plays movie
%    movie(gcf, Mov,1,2)     % plays at 2 fps
%    movie(gcf, Mov,10,5)    % repeats 10 times at 5 fps
% Create movie
%    movie2avi(Mov, 'heat2d.avi', 'compression', 'None');
```

One can also make use of the Symbolic Toolbox in MATLAB. The following defines the product solutions and computes the Fourier coefficients.

```
clear
% Problem Test
syms n x y
n = sym('n','integer');
m = sym('m','integer');

L = 1;
W = 1;
Kx = n*pi/L;
Ky = m*pi/W;
```

```
f = sin(pi*x/L)*sin(3*pi*y/W);
phi = sin(Kx*x)*sin(Ky*y);

% Fourier Coefficients
c=simplify(4/L/W*int(int(f*phi, x, [0 L]), y, [0 W]) );
```

Now one can plot the initial condition

```
% Initial Condition
N=5;
M=5;
H=.07;
ff=symsum(symsum(c*phi,n,1,N),m,1,M);
h=fsurf(ff,[0,L,0,W],'FaceColor','red','EdgeColor','none');
    camlight left;
    lighting phong
    xlabel('x')
    ylabel('y')
    title(['Solution at t = ' num2str(0)])
    axis([0,L,0,W,-1,1])
%frame=1;
%Mov(frame)=getframe(gcf);
```

The solution can then be evolved in time.

```
k=1/200;     % Heat constant

% Time Evolution
T =   1;    % Final time
Nt = 20;    % Number of steps

dt=T/Nt;
lambda = Kx^2+Ky^2;

for j=1:20
    t=j*dt;
%   frame=frame+1;
    ff=symsum(symsum(c*phi*exp(-k*lambda*t),n,1,N),m,1,M);
    %fsurf(ff,[0,L,0,W],'FaceColor','red','EdgeColor','none')
    h.Function=ff; % Alternative to using fsurf by updating the function plotted

    title(['Solution at t = ' num2str(t)])

%    Mov(frame)=getframe(gcf);
%    pause(.05)
end
```