**Pde implementation using MATLAB**

clear all, close all, clc

**part 1, detailed project specifications**

--in this project I will be implementing a program which can solve Pdes,

--I will be making use of a heat equation Pde to demonstrate how to

--implement a program in MATLAB capable of solving this equation successfully

--for inputs I will be using, a, to denote diffusion constant,L to denote

--length of the entire domain,N, for total number of discretization points

--for the output components, phat,this denotes the ultimate solution of the given solved Pde

**part 2, description of the problem**

--so in this case I will be using the method of Fourier series to solve the given Pde

\*\***derivation of the Pde**

--we let our heat equation be defined as follows:

-- Pt=alpha\*\*2 \* Pxx

--using Fourier transform to convert our heat equation into a spatial frequency

--let Px=i \* K \* Phat----(1)

--let Pxx=-K\*\*2 \* Phat

**substituting these into our heat equation (1) we have**

--Phatt=-alpha\*\*2 \* K\*\*2 \* Phat

--so basically, what we are doing is that we converting the given Pde equation

--into a system of decoupled Odes which we going to solve using the ode45 function

**part 3, coding the program**

function duhatdt=rhsHeat(t,phat,alpha,a)

duhatdt=-a^2\*(alpha. ^2)'.\*phat;

Implementing the Pde function using MATLAB

a=1;

L=100;

N=1000;

dx=L/N;

x=-L/2:dx:L/2-dx;

**defining the discreate wavenumbers**

alpha=(2\*pi/L) \* [-N/2: N/2-1];

alpha= fftshift(alpha) ; %this is reordering the Fourier transform wavenumbers

**defining the initial conditions**

p0=0\*x;

p0((L/2-L/10)/dx:(L/2 + L/10)/dx)=1;

**solving the system of pde using ode45**

\*\***defining a time t parameter**

t=0:0.1:20;

[t,phat]=ode45(@(t,phat)rhsHeat(t,phat,alpha,a),t,fft(p0));

**writing a for loop to iterate through the spatial domain**

for k=1: length(t)

u (k,:)=ifft(phat(k,:));

end

**plotting the Pde solution in time**

figure,h=waterfall(x,t(1:10:end),(u(1:10:end,:)));

set(h,'LineWidth',5,'FaceAlpha',0.5);

colormap(flipud(jet)/1.5)

set(gca,'FontSize',20)

xlabel('Space'); ylabel('Time');zlabel('Temperature')

set(gcf,'Position',[1400 400 1550 1100])

**conclusion**

--in this project we have used the ode45 function to generate solution to

--our heat equation pde, used the fftshift to fourier transform our pde,once

--we obtained the Fourier transform of our pde,we then inverse transformed

--the pde so as to obtain the solution outcome phat)

**references**

--1. https//peer.asee.org

--2. https//tandofonline.com

--3. http//www.math.chalmers.set