Parabolic PDE’s in Matlab

Jake Blanchard
University of Wisconsin - Madison
Introduction

- Parabolic partial differential equations are encountered in many scientific applications.
- Think of these as a time-dependent problem in one spatial dimension.
- Matlab’s `pdepe` command can solve these.
Model Problem

\[ \rho c_p \frac{\partial T}{\partial x} = k \frac{\partial^2 T}{\partial x^2} \]

\[ T(x,0) = 0 \]

\[ -k \frac{\partial T}{\partial x} \bigg|_{x=0} = q'' \]

\[ T(L,t) = 0 \]
pdepe Solves the Following

\[ c\left(x, t, u, \frac{\partial u}{\partial x}\right) \frac{\partial u}{\partial t} = x^{-m} \frac{\partial}{\partial x} \left( x^m f\left(x, t, u, \frac{\partial u}{\partial x}\right) \right) + s\left(x, t, u, \frac{\partial u}{\partial x}\right) \]

\[ u(x, t_0) = u_0(x) \]

\[ p(x, t, u) + q(x, t)f\left(x, t, u, \frac{\partial u}{\partial x}\right) = 0 \]

**Boundary Conditions** – one at each boundary

**m=0** for Cartesian, 1 for cylindrical, 2 for spherical

**Initial Conditions**
pdepe Solves the Following

\[ c \left( x, t, u, \frac{\partial u}{\partial x} \right) \frac{\partial u}{\partial t} = x^{-m} \frac{\partial}{\partial x} \left( x^m f \left( x, t, u, \frac{\partial u}{\partial x} \right) \right) + s \left( x, t, u, \frac{\partial u}{\partial x} \right) \]

\[ \rho c_p \frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2} \]

\[ c = \rho c_p \]

\[ f = k \frac{\partial T}{\partial x} \]

\[ s = 0 \]
Differential Equations

function [c,f,s] = pdex1pde(x,t,u,DuDx)
global rho cp k
c = rho*cp;
f = k*DuDx;
s = 0;
Initial Conditions

function u0 = pdex1ic(x)
u0 = 0;
pdepe Solves the Following

\[-k \frac{\partial T}{\partial x}\bigg|_{x=0} = q''\]

or

\[q'' + k \frac{\partial T}{\partial x}\bigg|_{x=0} = 0\]

remember

\[f = k \frac{\partial T}{\partial x}\]

\[T(L,t) = 0\]
Boundary Conditions

function [pl,ql,pr,qr] = pdex1bc(xl,ul,xr,ur,t)

global q
pl = q;
ql = 1;
pr = ur;
qr = 0;

At left edge, q+k*dT/dx=0
At right edge, ur=0
Calling the solver

tend=10
m = 0;
x = linspace(0,L,200);
t = linspace(0,tend,50);
sol =
pdepe(m,@pdex1pde,@pdex1ic,@pdex1bc,x,t);

200 spatial mesh points
50 time steps from t=0 to tend
Postprocessing

Temperature = sol(:,:,1);
figure, plot(x, Temperature(end,:))

figure, plot(t, Temperature(:,1))
function parabolic
global rho cp k
global q
L=0.1 %m
k=200 %W/m-K
rho=10000 %kg/m^3
cp=500 %J/kg-K
q=1e6 %W/m^2
tend=10 %seconds

m = 0;
x = linspace(0,L,200);
t = linspace(0,tend,50);

sol = pdepe(m,@pdex1pde,@pdex1ic,@pdex1bc,x,t);
Temperature = sol(:,:,1);
figure, plot(x,Temperature(end,:))

function [c,f,s] = pdex1pde(x,t,u,DuDx)
global rho cp k

c = rho*cp;
f = k*DuDx;
s = 0;

function u0 = pdex1ic(x)
u0 = 0;

function [pl,ql,pr,qr] = pdex1bc(xl,ul,xr,ur,t)
global q
pl = q;
ql = 1;
pr = ur;
qr = 0;
A Second Problem

- Suppose we want convection at $x=L$
- That is

\[-k \frac{dT}{dx} = h(T - T_{bulk})\]

or

\[hT - hT_{bulk} + k \frac{dT}{dx} = 0\]

\[p(x, t, u) + q(x, t)f\left(x, t, u, \frac{\partial u}{\partial x}\right) = 0\]

\[x = L\]

\[p = T = h(ur - T_{bulk})\]

\[q = 1\]
function u0 = pdex1ic(x)
global q hcoef Tbulk
u0 = Tbulk;

function [pl,ql,pr,qr] = pdex1bc(xl,ul,xr,ur,t)
global q hcoef Tbulk
pl = q;
ql = 1;
pr = hcoef*(ur-Tbulk);
qr = 1;
Download Scripts

http://blanchard.ep.wisc.edu/