conduction problems. A user defined transient program called **cond1d\_xxx.m** is created by modifying user defined steady state program, **std1d\_fin.m**. In **cond1d\_xxx.m**, problem independent program, **cond1d\_invariant.m** is called. This invariant part is almost identical to **std1d\_invariant.m** except that few lines are added to handle transient nature of the problem.

In user defined program (**cond1d\_xxx.m**), one has to specify the followings:

* Time step, maximum time allowed
* Geometry, such as length and cross sectional area
* Material properties, such as conductivity, specific heat and density
* Initial conditions
* Boundary conditions
* Source terms
* Desired output, such as temperature profile, energy balance, etc.

User independent program **(cond1d\_invariant.m)** should not be changed at all.

As an example of how to construct a user defined program, cond1d\_example is created to solve a fin problem already discussed in previous chapter. In this example, we will treat it as a transient problem.

The length of the fin is 1.2 m and its diameter is 2.5 cm. The conductivity is 401 W/m.K, density, 8933 kg/m^3, and specific heat 383.67 J/kg.K. Initial temperature is 373 K. The boundary conditions are 473

K at x=0 and convective boundary at x=L, hconv=10 W/m^2.K, Tf=298 K. Number of control volumes is 10, time step is 100 sec (It is greater than 62 sec required for CFL criterion.) and maximum time where computation stops is 2x104 sec. The maximum iteration within a time step is set at 10. Transient temperatures are saved at every 30 iterations (3000 sec interval).

**Listing of cond1d.example.m**

%cond1d\_input\_example.m

%transient, 1-dimensional conduction with varying cross section area

%,nonuniform condutivity and sources. Finite volume formulation

%using matlab program. (By Dr. S. Han, Feb, 2013)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%list of symbols:

% ac=cross-sectional area

% dt=time step

% iflag=0 (solution converged) ,iflag=1 (solution is not converged)

% iter=iteration counter

% maxiter= maximum iteration allowed in a time step

% n=number of control volumes

% te=temperature at new time level

% tep=projected temperature at new time level

% te0=temperature at old time level

% tk=diffusion coefficent(thermal conductivity)

% tstop=time to stop computation

% x=independent variable (spatial coordinate), dx=delta x

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

clear all;close all;clc

%specify periodic boundary

iperiodic=0; %not a periodic boundary; 1=periodic boundary

%specify the number of control volumes

n=10; %number of conrol volumes

maxiter=100;% maximum number iteration in each time step

mwrite=10;%%print results at every 10 time steps

dt=100;%time step

tstop=23000;%time to stop the calculation

np1=n+1;np2=n+2;np3=n+3;

re=1.0;%relaxation coefficient for simultaneous equation

%define calculation domain

dx=ones(1,np2);

tl=1.2;%total length

delx=tl/n;

dx=delx\*dx;

if iperiodic==0; %non-periodic boundary

dx(1,1)=1.0e-10;dx(1,np2)=1.0e-10;

else

dx(1,1)=dx(1,np1);dx(1,np2)=dx(1,2);

end

%

x(1,1)=0;

for i=1:np2

x(1,i+1)=x(1,i)+dx(1,i);

end

%define cross sectional area

dia=0.025;

for i=1:np3

ac(1,i)=pi/4\*dia^2;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%

%prescribe intitial temperatures calling function inital

for i=1:np2

te0(i)=373;

te(i)=te0(1,i);

tep(i)=te(1,i); %predicted value at the next iteration

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%time loop begins here

t=0; %starting time

plotte=[te'];%save temperatures for plot

iwrite=1; %printout counter, iwrite<mwrite then skip printout

while t<tstop ;% time loop (outer do loop)......................(1)

%iteration for convergence

iter=0;

iflag=1;

%iteration loop for the convergence

while iflag==1 % Inner do loop within a time step..........(2)

%prescribe thermal property

for i=1:np2

tk(1,i)=401;%conductivity

ro(1,i)=8933;%density

cp(1,i)=385;%specific heat

end

%prescribe the boundary conditions

%intialize fluxes

qx0c=zeros(1);qx0p=zeros(1);qx0=zeros(1);

qx1c=zeros(1);qx1p=zeros(1);qx1=zeros(1);

%bx0(1)=1,2,3(known temperature, known flux, periodic) at x=0

%bx1(1)=1,2,3 (same) at x=xmax

%at x=0

bx0(1)=1;%known temperature

te(1)=473;%base temperature

%at x=xmax

bx1(1)=2;%known flux

hf=10;% convection coefficient

tf=298;%room temperature

qx1p(1)=hf;

qx1c(1)=-hf\*tf;

qx1(1)=qx1c(1)+qx1p(1)\*te(np2);

%incorporate source terms

hf=10;

tf=298;

dia=0.025;

per=pi\*dia;

for i=1:np2

sp(1,i)=-hf\*per/ac(i);

sc(1,i)=hf\*per/ac(i)\*tf;

end

%solve equations calling cond1d\_invariant.

cond1d\_invariant;

%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

end % this end goes with the while iflag==1, inner do loop.......(2)

%

%advance to next time

t=t+dt;%increase time to next level

for i=1:np2

te0(1,i)=te(1,i);%reinitialize variable

tep(1,i)=te0(1,i);

end

%write the results at this time?

if iwrite>mwrite

%print the results at selected time intervals

fprintf('iteration number is %i \n',iter)

disp('transient temperatures are')

fprintf('%9.3f\n',te)

plotte=[plotte te'];

%

iwrite=0;

end %this end goes with if iwrite>mwrite

iwrite=iwrite+1;

end %this end goes with time loop while t<tstop, outer do loop.....(1)

%plot the result x.vs.te at several time steps

for i=1:np2

xc(i)=0.5\*(x(i)+x(i+1));

end

plot(xc,plotte)

xlabel('x'), ylabel('K')

Fig. 6 shows the transient temperature distributions at several time steps. The converged solution is the steady state solution that is identical to the previous steady state solution.

****

Figure 6 Transient temperature profiles for a pin fin

**References**

1. Patankar, S. V., Numerical Heat Transfer and Fluid Flow, McGraw Hill, 1980.
2. Incropera, F. P., D. P. Dewitt, T. L. Bergman and A. S. Lavine, Fundamentals of Heat and Mass Transfer, 6th Ed., John Wiley, 2007.