One-Dimensional Transient Conduction

We have discussed how to approximate the steady one-dimensional conduction equations by integrating the equations over a control volume and taking energy balance at the control volume. In carrying out the integration, it was assumed that the temperatures in the control volumes do not change in time. In a transient conduction, temperature of the control volume is a function of time as well as the space. Additional consideration is needed to handle this dependency of temperature on time. Recall that one-dimensional, transient conduction equation is given by

$$\rho C \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}(k \frac{\partial T}{\partial x}) + S \qquad (1)$$

It is important to point out here that no assumptions are made regarding the specific heat, C. In general, specific heat is a function of temperature. The source term is assumed to be in a linearized form as discussed previously for the steady conduction.

Finite Volume Equation

Temperature depends on time (t) as well as on space (x) in a transient conduction. We may thus treat time as an additional coordinate. In the spatial coordinates, energy transfer occurs in both directions. That is heat flux can be negative or positive at a spatial point depending on the temperature gradient at the given point. This is called a "two-way" coordinate implying that information can travel in both spatial directions. On the other hand, time coordinate is a "one-way" coordinate. That is temperature at a point is influenced only by the temperature of the previous time and not by the future time. Two-way behavior is sometimes called "elliptic" and one-way behavior, "parabolic" [1].

The temperature at a time $(t+\Delta t)$ is determined by temperature at a time (t). This is schematically illustrated in Fig. 1. Now we integrate Eq.(1) over the given control volume during a time step (Δt) to obtain a finite volume representation of Eq.(1). Integrating the first term;

$$\int_{t^0}^{t^1}\int_0^1\int_0^1\int_w^e (\rho C\frac{\partial T}{\partial t})\,dxdydzdt = (1x1)\int_{t^0}^{t^1}\int_w^e (\rho C\frac{\partial T}{\partial t})\,dxdt$$

$$= (1x1)\Delta x\int_{t^0}^{t^1}(\rho C\frac{\partial T}{\partial t})\,dt = (1x1)\Delta x\,\rho\int_{t^0}^{t^1}(C\frac{\partial T}{\partial t})\,dt$$

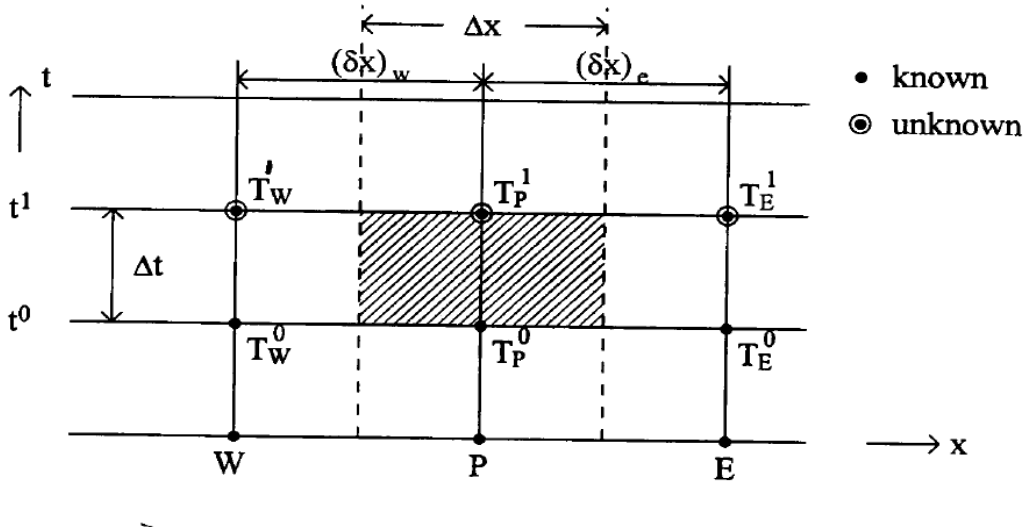


Figure 1 Control volume in x-t coordinates

In general, C=C(T) and $\frac{\partial T}{\partial t}$ changes continuously during the integration time interval, $t^0 \le t \le t^1$. Let $C\overline{\frac{\partial T}{\partial t}}$ be an average value during integration time, then the above integration becomes

$$1st\ Term = \Delta x(1x1)\rho(\overline{C\frac{\partial T}{\partial t}})\int_{t^0}^{t^1} dt = \rho\overline{C}(\Delta x)(1x1)\Delta t(\overline{\frac{\partial T}{\partial t}}) \qquad (2)$$

Integrating the second term,

$$\int_{t^0}^{t^1}\int_0^1\int_0^1\int_w^e \frac{\partial}{\partial x}(k\frac{\partial T}{\partial x})\,dxdydzdt = (1x1)\int_{t^0}^{t^1}[(k\frac{\partial T}{\partial x})_e - (k\frac{\partial T}{\partial x})_w]dt$$

Again, the diffusion flux will change continuously during an integration interval. Let $\overline{k\frac{\partial T}{\partial x}}$ denotes the average diffusion flux during this time step, then

$$2\,\text{nd term} = (1\,x\,1)\left[\overline{(k\frac{\partial T}{\partial x})_e} - \overline{(k\frac{\partial T}{\partial x})_w}\right]\int_{t^0}^{t^1} dt = (1\,x\,1)\Delta t\left[\overline{(k\frac{\partial T}{\partial x})_e} - \overline{(k\frac{\partial T}{\partial x})_w}\right] \qquad (3)$$

Integrating the source term, we have

$$\int_{t^0}^{t^1}\int_0^1\int_0^1\int_w^e (S_C + S_P\,T)\,dxdydzdt = (1\,x\,1)\Delta x\int_{t^0}^{t^1}(S_C + S_P\,T_P)\,dt = (1\,x\,1)\Delta x(S_C + S_P\,\overline{T_P})\Delta t \qquad (4)$$

where $\overline{T_P}$ is the average temperature at which source terms are evaluated.

We have yet to decide at which time various properties should be evaluated. We may choose $t=t^0$, $t=t^1$ or in between these two times. A generalization can be made by introducing a weight factor f, such that

$$\overline{\phi} = f\,\phi^1 + (1 - f)\,\phi^0 \qquad (5)$$

where $\overline{\phi}$ is the averaged property and $0 \le f \le 1.0$. Thus

$$\overline{\frac{\partial T}{\partial t}} = f(\frac{\partial T}{\partial t})^1 + (1 - f)(\frac{\partial T}{\partial t})^0 \qquad (6)$$

Evaluating the time derivatives by the 2nd order accurate scheme evaluated at $t= (t^0+ t^1)/2$ in time, we get

$$\overline{\frac{\partial T}{\partial t}} = \frac{T^1 - T^0}{\Delta t} \qquad (7)$$

The diffusion fluxes in Eq.(3) are approximated by

$$\overline{\left(k\frac{\partial T}{\delta x}\right)_e} = f\left(k\frac{\partial T}{\partial x}\right)_e^1 + (1-f)\left(k\frac{\partial T}{\partial x}\right)_e^0$$

$$= f\left(k_e^1\frac{T_E^1 - T_P^1}{(\delta x)_e}\right) + (1-f)\left[k_e^0\frac{T_E^0 - T_P^0}{(\delta x)_e}\right] \tag{8}$$

and

$$\overline{\left(k\frac{\delta T}{\delta x}\right)_w} = f\left[k_w^1\frac{T_P^1 - T_W^1}{(\delta x)_w}\right] + (1-f)\left[k_w^0\frac{T_P^0 - T_W^0}{(\delta x)_w}\right] \tag{9}$$

Finally, $\overline{T_P}$ in Eq.(4) is

$$\overline{T_P} = f\,T_P^1 + (1-f)\,T_P^0 \tag{10}$$

Substituting Eqns.(8),(9) and (10) into Eqns.(2),(3) and (4), respectively and collecting terms, we obtain (left as an exercise), after dropping the superscript "1" for convenience;

$$a_P T_p = a_E f\,T_E + a_E^0(1-f)\,T_E^0 + a_W f\,T_W + a_W^0(1-f)\,T_W^0$$

$$+[a_P^0 - (1-f)\,a_E^0 - (1-f)\,a_W^0 + S_P(1-f)\Delta x(1x1)]\,T_P^0 + S_C\,\Delta x(1x1) \tag{11}$$

where

$$a_E = \frac{k_e(1x1)}{(\delta x)_e};\; a_E^0 = \frac{k_e^0(1x1)}{(\delta x)_e} \tag{12a}$$

$$a_W = \frac{k_w(1x1)}{(\delta x)_w};\; a_W^0 = \frac{k_w^0(1x1)}{(\delta x)_w} \tag{12b}$$

$$a_P^0 = \frac{\rho\overline{C}\Delta x(1x1)}{\Delta t} \tag{12c}$$

$$a_P = f\,a_E + f\,a_W + a_P^0 - f\,S_P\,\Delta x(1x1) \tag{12d}$$

and

$$\overline{C} = fC + (1 - f)C^0 \qquad (12e)$$

Explicit and Implicit Schemes

Eq.(11) shows that temperature T_P depends on the previous time-level temperatures T_P^0, T_E^0 and T_W^0 as well as T_E and T_W when f is not equal to zero.

Explicit scheme

If f=0, Eq.(11) reduces to

$$a_P^0 T_P = a_E^0 T_E^0 + a_W^0 T_W^0 + [a_P^0 - a_E^0 - a_W^0 + S_P \Delta x(1x1)] T_P^0 + S_C \Delta x(1x1) \qquad (13)$$

The right hand side of Eq.(13) contains all the known values and T_P can be found explicitly without solving simultaneous equations. This method is called the "explicit scheme". Explicit scheme does not need to solve simultaneous equations and temperatures at a new time-level can be calculated point by point. This solution scheme is simple to program and requires less computer memory space but has a serious constraint on the size of time steps. This shortcoming can be easily understood by examining the coefficients appearing in Eq.(13). All coefficients must be positive to ensure proper physical process as discussed in a previous chapter. Otherwise, temperature can become negative. Coefficients a_E^0, and a_W^0 and a_P^0 are obviously positive. However, coefficient in front of T_P^0 can go negative. To ensure this coefficient stays on positive, we must require that

$$a_P^0 - a_E^0 - a_W^0 + S_P \Delta x(1x1) \geq 0$$

Since $S_P \leq 0$ always, we must require

$$a_P^0 - a_E^0 - a_W^0 \geq 0 \quad \text{or} \quad \frac{\rho C \Delta x(1x1)}{\Delta t} - \frac{k_e^0}{(\delta x)_e} - \frac{k_w^0}{(\delta x)_w} \geq 0$$

For a special case in which $k_e = k_w = k$, and $(\delta x)_e = (\delta x)_w = \Delta x$, the last expression becomes

$$\frac{\rho C \Delta x}{\Delta t} - \frac{2k}{\Delta x} \geq 0$$

or,

$$\Delta t \leq \frac{\rho C (\Delta x)^2}{2k} = \frac{1}{2}\frac{(\Delta x)^2}{\alpha} \qquad (14a)$$

where α is the thermal diffusivity of the materials. In terms of the local Fourier number, Eq.(14a) is

$$F_0 = \frac{\alpha \Delta t}{(\Delta x)^2} \leq \frac{1}{2} \qquad (14b)$$

Thus, a time step greater than the value given by Eq.(14a) should not be used in the explicit scheme. Eq.(14a) is called the CFL (Courant-Friedrich-Levy) stability criterion. The stability requirement can severely limit the size of time steps when Δx becomes very small. The CFL condition becomes even more stringent when there is a large source term $S_P \leq 0$.

Crank-Nicolson scheme

By setting f=1/2, Eq.(11) becomes an implicit scheme, implying that the temperature T_P is influenced by T_W and T_E as well as the old-time-level temperatures. Simultaneous equations must be solved to find the temperatures at a new time-level. The Crank-Nicolson scheme assumes that property changes occur linearly during an integration time step as depicted in Figure 3.

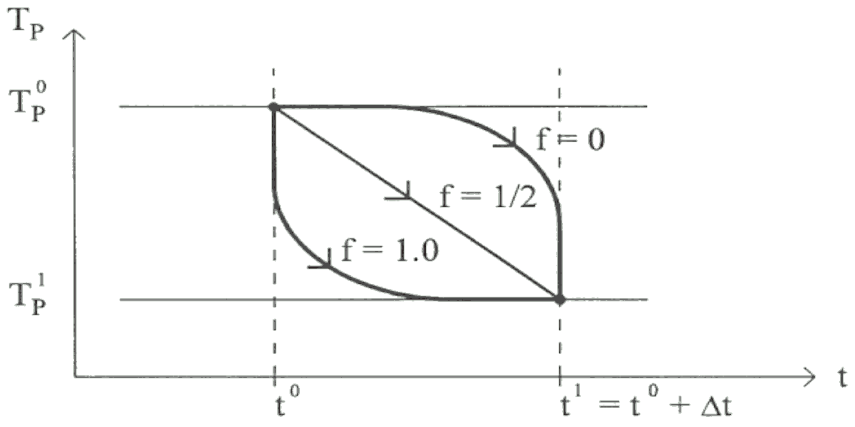


Figure 2 Assumed temperature variation during a time step

The Crank-Nicolson scheme is known to be stable for all sizes of time steps. However, if the time steps are too large, solutions could be locally oscillatory and eventually become nonphysical. This behavior is predictable by examining the coefficient for T_P in Eq.(11) with f=1/2. Following the procedure as used for the explicit scheme, it can be shown that

$$\Delta t \le \frac{(\Delta x)^2}{\alpha} \quad \text{or} \quad F_0 \le 1.0 \tag{15}$$

The stability requirement for the Crank-Nicolson scheme is less stringent than the explicit scheme, but time steps still cannot be too large.

5.2.3 Fully-Implicit scheme

By choosing f=1.0, Eq.(11) simplifies to

$$a_P T_P = a_E T_E + a_W T_W + b \tag{16}$$

where

$$a_E = \frac{k_e (1x1)}{(\delta x)_e} \tag{17a}$$

$$a_W = \frac{k_w (1x1)}{(\delta x)_w} \tag{17b}$$

$$a_P^0 = \frac{\rho C \Delta x (1x1)}{\Delta t} \tag{17c}$$

$$b = S_C \Delta x (1x1) + a_P^0 T_P^0 \tag{17d}$$

and

$$a_P = a_E + a_W + a_P^0 - S_P \Delta x(1x1) \qquad (17e)$$

The fully-implicit scheme offers simplicity in formulation. We observe only minor differences between the steady-state conduction and transient conduction formulations. The only difference is the term a_P^0. If we multiply T_P to a_P^0, we have

$$a_P^0 T_P = \frac{\rho C \Delta x(1x1)}{\Delta t} T_P = \frac{m_{CV} C T_P}{\Delta t}$$

which is the rate of energy change in the given control volume. For a steady state, the rate of change of energy in the control volume should be zero, that is

$$a_P^0 T_P \to 0 \quad \text{as} \quad \Delta t \to \infty$$

Therefore, by setting the time step very large, steady state formulation is recovered from transient formulation.

One-Dimensional Transient Conduction Program

One dimensional steady state conduction program (std1da.m) is modified to obtain a transient one dimensional conduction program. These modifications are rather straightforward. The modified program is called the tran1d.m. Computational steps used in the tran1d.m are as follow:

(1) Define the calculation domain.
(2) Prescribe the initial temperatures T^0 at $t=t^0$.
(3) Assign the guessed temperatures T^* at the next time level $t=t^1$. Usually $T^*=T^0$.
(4) Evaluate the coefficients and source terms.
(5) Prescribe the boundary conditions, T_1 and T_{N+2}.
(6) Solve the simultaneous equations by TDMA for the new temperature T at $t=t^1$.
(7) Chech the convergence. Is

$$\frac{\text{abs}(T - T^*)}{T} \le \varepsilon \quad ?$$

(a) Yes. Solution at $t=t^1$ is obtained. Go to step (8).

(b) No. More iteration is needed within this time step. Let $T^* = T$ and go to step (4).

(8) Solution at $t^1 = t^0 + \Delta t$ is obtained. Write the results, if desired.
(9) Reinitialize the dependent variable by setting $t^0 = t^1$ and $T^0 = T^1$. Go to step (3).
(10) Repeat steps (1) through (9) until $t^1 = t_{stop}$.

Fig.3 shows the flow chart for the transient 1-D conduction program. There are two iteration loops; an inner loop for the convergence within a given time step and an outer one for the time marching step. It is clear that a transient solution procedure is nothing more than the repetitions of steady state solution over many time step periods. If the repetition continues for a long time, the solutions would cease to change anymore if the physical problem allows a steady-state. Otherwise, the solutions will continue to change as time advances.

The structure of the tran1d.m program is best illustrated by an example application as follows.

Description of the problem

As an illustration of how to apply the tran1d.m, let us consider a transient heat transfer through a semi-infinite copper slab initially at 20 ^{0}C. Initially temperature of the slab is uniformly at 20 ^{0}C. The temperature at x=0 is suddenly raised to 120 ^{0}C. (Fig.5). The transient temperature distribution in the slab is calculated and compared with the exact solution. This example problem is selected to test three integration methods for the time steps, f=0, 1/2 and 1.0. It is assumed that thermal conductivity and specific heat are constants.

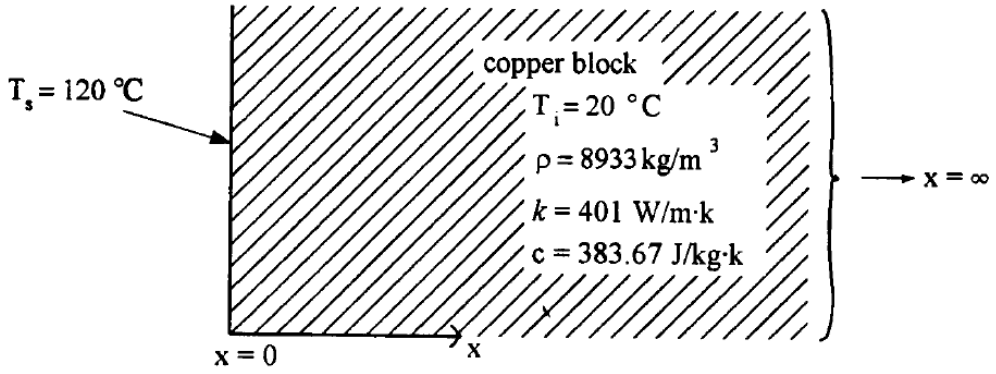


Figure 4 Transient conduction through an infinite medium

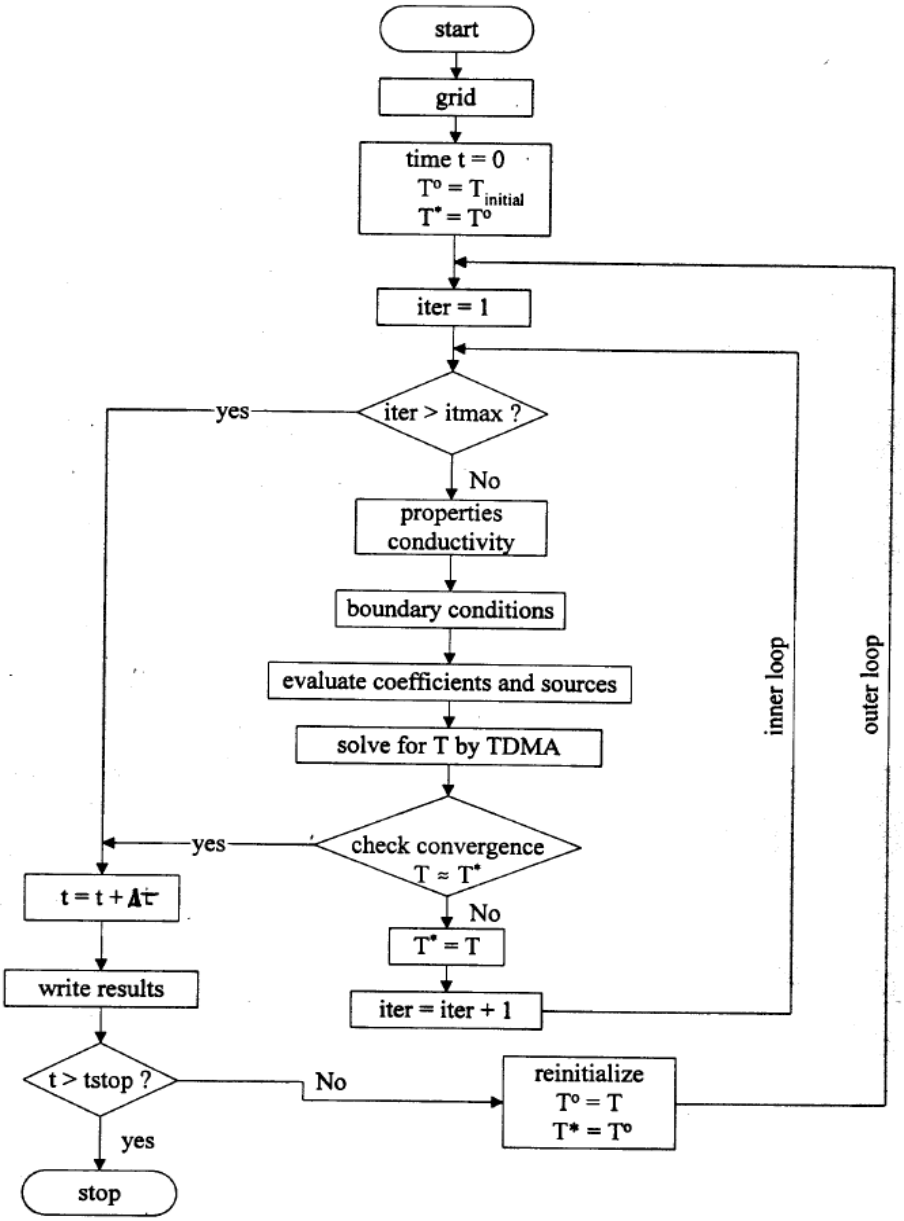


Figure 3 Flow Chart for transient 1-D conduction Program

The exact solution is given by [2],

$$T(x,t) = T_s + (T_i - T_s)\,\mathrm{erf}\left(\frac{x}{2\sqrt{\alpha t}}\right) \qquad (18)$$

where $\alpha = \dfrac{k}{\rho C}$ is the thermal diffusivity.

Listing of tran1d.m

```
%tran1d.m. transient,1-dimensional conduction with varying cross section area
%,nonuniform condutivity and arbitry sources. Finite volume formulation
%using matlab program. (By Dr. S. Han, sep 9, 2008)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%specify the number of control volumes
n=10; % nmber of control volumes
maxiter=20; % maximum number of iteration in each time step. set large for
steady state
np1=n+1;
np2=n+2;
np3=n+3;
%f=1; %weighting function =1 (fully-implicit);=1/2 (Crank-Nicolson); =0
(explicit)
%assign time step and maximum time
tstop=120; %time to stop calculation
dt=48; % time step. set dt=1.0e10 for steady state
mwrite=1; %if dt>tstop mwrite must be set to 0.
%define calculation domain
tl=1; %total length of medium
delx=tl/n; % control volume size
dx=ones(np2);
dx=delx*dx;
%replace fictitious boundary volume size to small value
dx(1)=1.0e-10;
dx(np2)=1.0e-10;
%assign x-coordinate
x(1)=0;
for m=1:np2
    x(m+1)=x(m)+dx(m);
end
%define cross-sectional area
%ac=ones(1,np3);%cross-sectional area
for i=1:np3
    ac(i)=1; %constant cross section, dia=0.0025 m
end
%time derivitive test
plotte=[];
f=0;
for j=1:3% weighting factor choice
%prescribe intitial temperatures for all control volumes
for i=1:np2
    te0(i)=20;
    te(i)=te0(i);
    tep(i)=te(i);
```

```
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%time loop begins here
t=0; % starting time
iwrite=1; % print out counter, iwrite<mwrite means skip print out
while t<tstop % calculation continues until t>tstop
%iteration for convergence
iter=0; % set iteration counter in each time step
iflag=1; % iflag=1 means convergence is not reached
%iteration loop for the convergence in each time step
while iflag==1      % end is at the end of program *********
%prescribe thermal conductivity, density and specific heat
    for i=1:np2
        tk(i)=401; %conductivity
        ro(i)=8933; %density
        cp(i)=383.67; %specific heat
    end
%prescribe boundary temperature
    te(1)=120;%at the left boundary given temperature
    te(np2)=te0(np2); %far away boundary
%evaluate the diffusion conductance and source terms
for i=2:np1
% diffusion conductance
    ke=tk(i)*tk(i+1)*(dx(i)+dx(i+1))/...
        (dx(i)*tk(i+1)+dx(i+1)*tk(i)); %east interface conductivity
    de=2.0*ke*ac(i+1)/(dx(i)+dx(i+1)); % east side diffusion conductance
%
    kw=tk(i)*tk(i-1)*(dx(i)+dx(i-1))/...
        (dx(i-1)*tk(i)+dx(i)*tk(i-1)); %west interface conductivity
    dw=2.0*kw*ac(i)/(dx(i-1)+dx(i));%west side diffusion conductance
%
%time derivitive selection
    ae=f*de;
    aw=f*dw;
%linearlized source term evaluation
     sp=0;
     sc=0;
    vol=0.5*(ac(i)+ac(i+1))*dx(i); % volume of cv
    a0=ro(i)*cp(i)*vol/dt; % this term is zero for steady state
    ap=ae+aw+a0-f*sp*vol;
    b=sc*vol+de*(1-f)*te0(i+1)+dw*(1-f)*te0(i-1)+...
        (a0-(1-f)*de-(1-f)*dw+sp*(1-f)*vol)*te0(i);
%setting coefficients for tdma matrix
    ta(i)=ap;
    tb(i)=ae;
    tc(i)=aw;
    td(i)=b;
%incorporate boundary conditions
    if i==2
        td(i)=td(i)+aw*te(1); %at x=0
    elseif i==np1
     td(i)=td(i)+ae*te(np2); %at x=L
    end
end
%solve the simultaneous equations by using tdma%%%%%%%%%%%%%%%%%%%%%
nq=n;
nqp1=nq+1;
nqm1=nq-1;
%forward substitution
beta(2)=tb(2)/ta(2);
```

```
alpa(2)=td(2)/ta(2);
for i=3:nqp1
    beta(i)=tb(i)/(ta(i)-tc(i)*beta(i-1));
    alpa(i)=(td(i)+tc(i)*alpa(i-1))/...
         (ta(i)-tc(i)*beta(i-1));
end
%backward substitution
dum(nqp1)=alpa(nqp1);
for j=1:nqm1
    i=nqp1-j;
    dum(i)=beta(i)*dum(i+1)+alpa(i);
end
%end of tdma%%%%%%%%
%update the temperature
for i=2:np1
    te(i)=dum(i);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%check the convergence
errote=ones(1,np2); %initialze error
for i=1:np2
    errote(1,i)=abs(te(1,i)-tep(1,i))/te(1,i);
end
error=1.0e-6; %prescribed error tolerance
    if (max(errote)>error) %solution not converged
    iter=iter+1; %increase the iteration counter
    tep=te; % update the guessed value
    iflag=1; % keep the flag red
    else
    iflag=0; % solution converged, flag is green
    end
    if iter>maxiter % need to increase maxiter
        break
    end
end % this end goes with the while iflag==1 at the top********
%solution converged
%advance to the next time level and reinitialize the temperature
t=t+dt; %increase time
for i=1:np2
    te0(i)=te(i); %reinitialize temperature
    tep(i)=te0(i);
end
%write the results at this time?
if iwrite>mwrite
%print the results at selected time interval
fprintf('iteration number is %i \n',iter)
iwrite=0;
end
iwrite=iwrite+1;
end %this end goes with while t<tstop%%%%%%%%%%%%%%%%time loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%plot at the end of calculation
plotte=[plotte te'];
f=f+0.5;
end %this goes with weighting factor f
%plot the result x.vs.te
%locate the midpoint of control volumes
for i=1:np2
    xc(i)=0.5*(x(i)+x(i+1));
end
%exact solution
```

```
alpha=tk(1)/(ro(1)*cp(1)); %thermal diffusivity
for i=1:np2
    t1=erf(xc(i)/(2*sqrt(alpha*t)));
teexact(i)=te(1)+(te(np2)-te(1))*t1;
end
%
plot(xc,teexact,'-',xc,plotte(:,1),'-o',xc,plotte(:,2),'--
x',xc,plotte(:,3),':*')
legend('exact','f=0','f=1/2','f=1')
title('effect of time derivitive scheme'), xlabel('distance from the wall,
m'),
ylabel('temperature ,C')
grid on
```

Numerical Results

Ten uniform control volumes are used with a time step 48 sec, that is slightly greater than CFL condition (42.73 sec). Fig.6 show explicit scheme gives unrealistic solution while Crank-Nicolson and fully implicit scheme give reasonable agreement with exact solution.

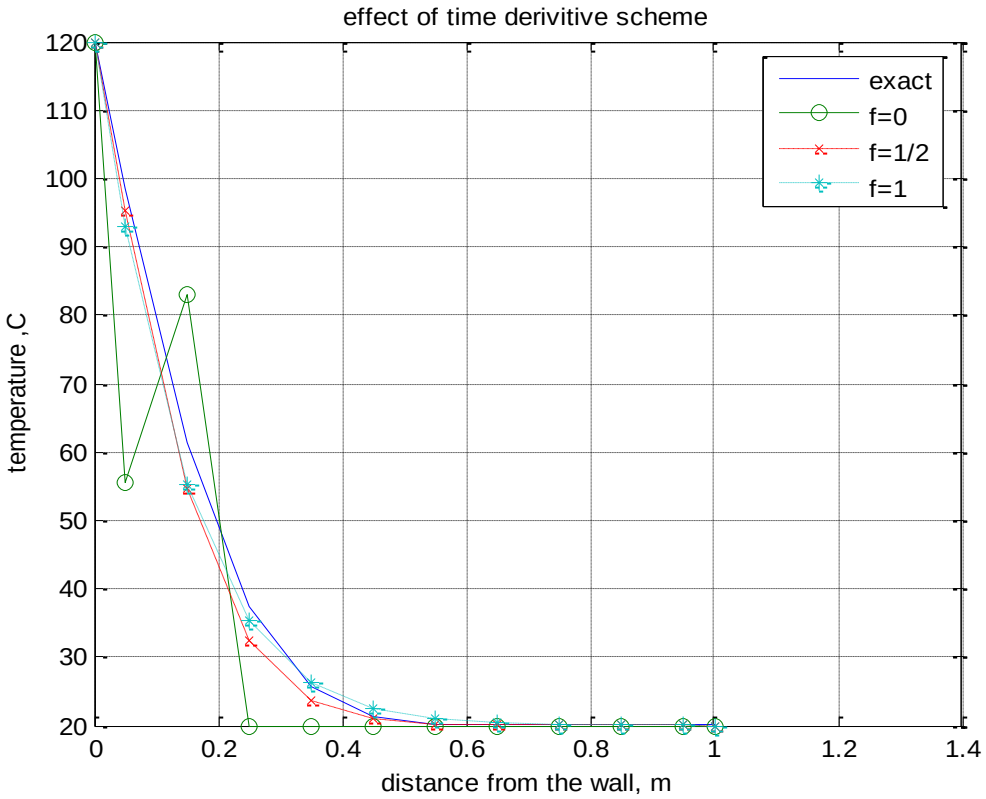


Figure 5 Effect of time derivative term on the temperature distribution at t=120 sec

Relaxation

In all iterative numerical schemes, it is often necessary to speed-up or slow-down the changes in the values of computed dependent variables and other related quantities. This is known as the relaxation. In the explicit scheme, unknown values are calculated by point by point and thus information travels slower in the computational domain. For this reason, it is common to use an over-relaxation to speed-up the convergence in the explicit scheme. In the implicit schemes, on the other hand, under-relaxation is preferred to slow down the changes since information travels much faster than desired. In the context of the present numerical formulation, under-relaxation is thus preferred [1].

The finite volume expression for the unknown temperature T_P is

$$a_P T_P = a_E T_E + a_W T_W + b \tag{19}$$

By solving this equation, an updated temperature T_P is obtained. The difference between the newly updated temperature and the immediate previously known temperature T_P^* is then

$$\Delta T_p = T_P - T_P^* = \frac{a_E T_E + a_W T_W + b}{a_P} - T_P^* \tag{20}$$

In writing the updated temperature, however, we modify the change ΔT_P by multiplying α such that

$$T_P = T_P^* + \alpha \Delta T_P$$

Substituting Eq.(20) into the last equation, we obtain

$$T_P = T_P^* + \alpha\left(\frac{a_E T_E + a_W T_W + b}{a_P} - T_P^*\right)$$

After rearranging, we have

$$\frac{a_P}{\alpha} T_P = a_E T_E + a_W T_W + b + \frac{a_P}{\alpha}(1-\alpha) T_P^* \tag{21}$$

If $\alpha = 1.0$, Eq.(21) reduces to Eq.(19). If α is greater than 1.0, changes in T_P will accelerate resulting in an over-relaxation. If $0. < \alpha < 1.0$, changes in the predicted temperature at the control volume would decelerate giving an under-relaxation. No matter what values of α are used for the relaxation, however, the converged final solution will be the same. This can be seen by setting $T_P = T_P^*$ in Eq.(21). The resulting equation is Eq.(19). The optimum value of relaxation coefficient depends on the problems at hand and should be determined by a trial and error. For the majority of pure conduction problems, effects of relaxation coefficient are not significant and can be set to 1.0. Relaxation becomes very critical when the momentum equations are solved iteratively for velocities in convection heat transfer problems, which will be discussed in later chapters. The relaxation coefficients less than 0.5 are frequently used in the convection problems.

Temperature at the boundary subject to nonlinear boundary condition, such as applied heat flux at the boundary, is calculated as a part of solution. If the heat flux applied to the boundary is very large, calculated boundary temperature can become too large and the solution becomes nonphysical. To alleviate this problem, boundary temperature can be relaxed by using

$$T = \alpha T + (1 - \alpha) T^*$$

where $\alpha \leq 1.0$ is the relaxation factor.

For a quasi-one dimensional transient conduction, Eqn.(1) is replaced by

$$\rho C \frac{\partial T}{\partial t} = \frac{1}{A} \frac{\partial}{\partial x} \left(A \kappa \frac{\partial T}{\partial x} \right) + S \qquad (22)$$

General Purpose 1-D Conduction Program

It was shown in previous sections that there are not many differences between a steady and a transient formulation of conduction problems. Transient formulation requires an additional term and a do-loop, which repeats the calculation of steady state solution at each time step. Depending on the choice of time derivative, one can use the explicit, and the implicit schemes in time marching. The fully-implicit scheme was shown to be the most reliable method regardless of time step sizes as long as the time step sizes are not extremely larger than CFL condition. The steady state solution, if exist, will be obtained as the time approaches to a large value.

We have written several programs to solve steady and transient 1-dimensional conduction problems. The structure of the programs was slightly modified to incorporate additional ideas as they were introduced in the course of presentation. We notice that one has to change only a small portion of the program to solve different problems. These differences arise because of the geometry, properties, boundary conditions, source terms and the steady or transient nature of the problems. The majority of the remaining parts of the program need not be changed for all problems. Thus we can construct a general-purpose program that can be used for a wide range of

conduction problems. By combining these programs, a transient program called cond1d.m is created. This program calls for number of MATLAB functions. There are two types of functions: one is user dependent and other is independent.

User dependent functions are:

grid1d: geometry of the problem
propty: property of materials
inital: initial conditions
boundy: boundary conditions
source1d: source terms
figure1: plot the result

User independent functions are:

solve: evaluate diffusion conductance and source and solve simultaneous equations
tdma: tri-diagonal matrix solver
convchek: check the convergence

As an example of transient conduction, a fin problem as discussed in previous chapter is solved as a transient conduction problem. All physical properties and boundary conditions remain the same. Transient temperature in the fin is to be calculated. Number of control volumes is 10. Time step size is 100 sec, that is larger than 62 sec required for a CFL stability criterion. Calculation is to be performed until the physical time equals $2x10^4$ sec. The maximum iteration in each time step is set to 10. Time step size should be small enough to have solution converges within 10 iterations in each time step. Results are printed at every 30 iterations. That is at every 3000 sec time intervals.

Listing of the program

```
%cond1d.m
%transient, 1-dimensional conduction with varying cross section area
%,nonuniform condutivity and sources. Finite volume formulation
%using matlab program. (By Dr. S. Han, Sep 25, 2007)
%modified to include all types of boundary conditions. (May 5, 2012)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%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
%funtions called in sequence:
%   function grid1d.m:assigns grid system
%   function inital.m:initialize the dependent variable, te and tep
```

```
%   function propty.m:=evaluate thermal properties
%   function boundy.m:set boundary condition for dependent variable
%   function source1d.m: prescribe source terms
%   function solve.m:evaulate coefficients and sources and solve
%        the resulting simultaneous equations by calling function tdma
%   function convcheck.m:check the convergence, te=tep?
%   function figure1.m: plots line graph x vs te
%*************************************************************************
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 calling function grid1d
[x,dx,ac]=grid1d(iperiodic,n);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%prescribe intitial temperatures calling function inital
[te,tep,te0]=inital(n);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%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
%iteration for convergence
iter=0;
iflag=1;
%iteration loop for the convergence
while iflag==1      % end is at the end of program ********************(1)
%prescribe thermal property calling function propty.
[tk,ro,cp]=propty(n); %thermal conductivity, density and specific heat
%incorpolate boundary conditions calling boundy_modified.
[te,bx0,qx0c,qx0p,qx0,bx1,qx1c,qx1p,qx1]=boundy(te,tk,dx,n,t,dt);
%incorporate source terms calling source1d.
[sp,sc]=source1d(te,n,x,dx,ac,t);
%solve equation calling function solve_modified.
[te,qx0,qx1]=solve(dx,ac,tk,te,tep,te0,ro,cp,dt,n,sp,sc,re,...
    bx0,qx0c,qx0p,qx0,bx1,qx1c,qx1p,qx1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%check the convergence of solution by calling convcheck.
[iflag,iter,tep]=convcheck(te,tep,n,iter,maxiter);
    if iter>maxiter
        disp('iter is greater than maxiter')
        break
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end % this end goes with the while iflag==1 at the top*****************
%
%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%%%%%%%%%%%%%%%%%%%%%%%%%%%%(1)
%plot the result x.vs.te by calling function figure1
figure1(n,x,plotte);


function [te,bx0,qx0c,qx0p,qx0,bx1,qx1c,qx1p,qx1]=boundy(te,tk,dx,n,t,dt);
% prescribe the boundary conditions
np1=n+1;
np2=n+2;
%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;
te(1)=473;%base temperature
%at x=xmax
bx1(1)=2;
    hf=10;% convection coefficient
    tf=298;%room temperature
    qx1p(1)=hf;
    qx1c(1)=-hf*tf;
    qx1(1)=qx1c(1)+qx1p(1)*te(np2);
```

Results

Fig. 6 shows the converged solution that is identical to the previous example.

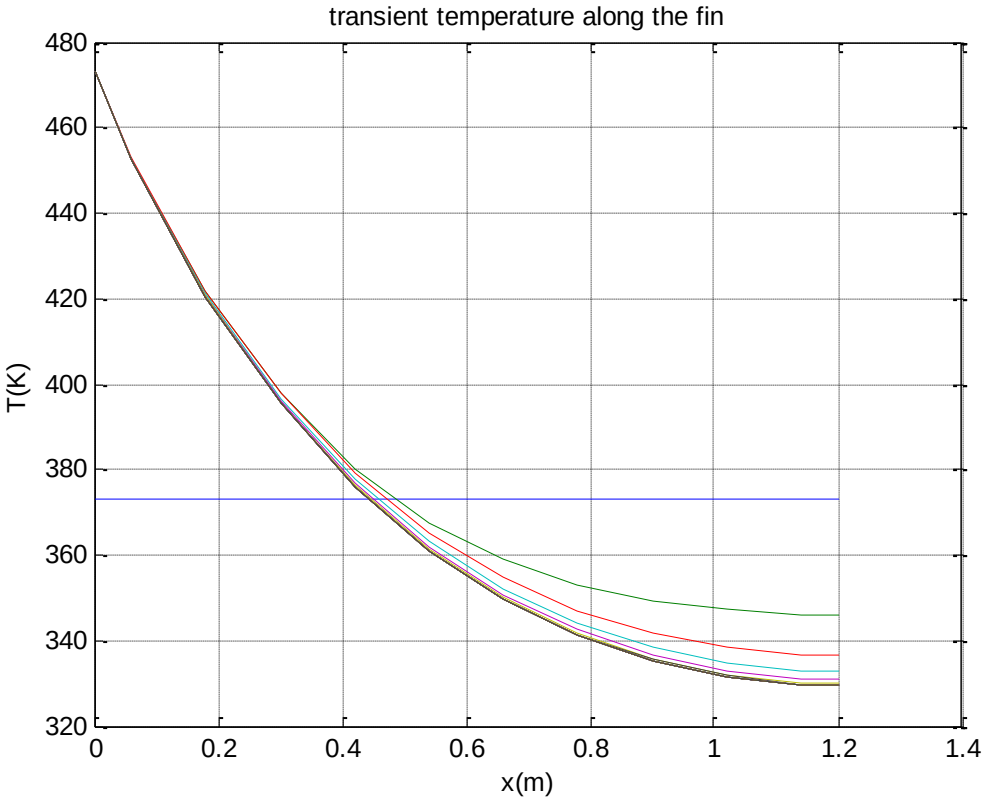


Figure 6 Transient temperature profiles for a pin fin

References

1. Patankar, S. V., Numerical Heat Transfer and Fluid Flow,
2. Incropera,

Project

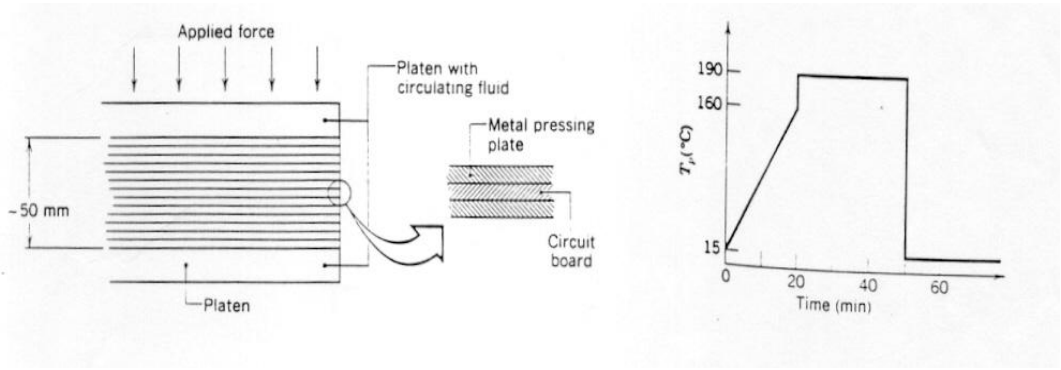
1. Copper-coated, epoxy-filled fiberglass circuit boards are treated by heating a stack of them under high pressure as shown in the figure. The purpose of the pressure-heating operation is to cure the epoxy which bonds the fiberglass sheets imparting stiffness to the boards. The stack is comprised of 10 boards and 11 pressing plates, which prevent epoxy from flowing between the boards and impart a smooth finish to the cured boards. Each of the boards and plates has a thickness of 2.36 mm. Thermodynamic properties are:

	board	plate
density kg/m^3	1000	8000
specific heat J/kg.K	1500	480
conductivity W/m.K	0.3	12

When the module is activated at t=0, the initial temperature of the stack is at T_i=15 ^{0}C. To reduce the excessive thermal stress, temperature of the platen is changed as shown in the figure.

(a) Using a time step of 60 sec, determine the temperature history at the mid-plane of the stack and determine whether curing will occur (170 ^{0}C for 5 min).

(b) Following the reduction of the platen temperatures to 15 ^{0}C (t=50 min), how long will it take for the mid-plane to reach 37 ^{0}C, a safe temperature at which the operator can begin unloading the press ?



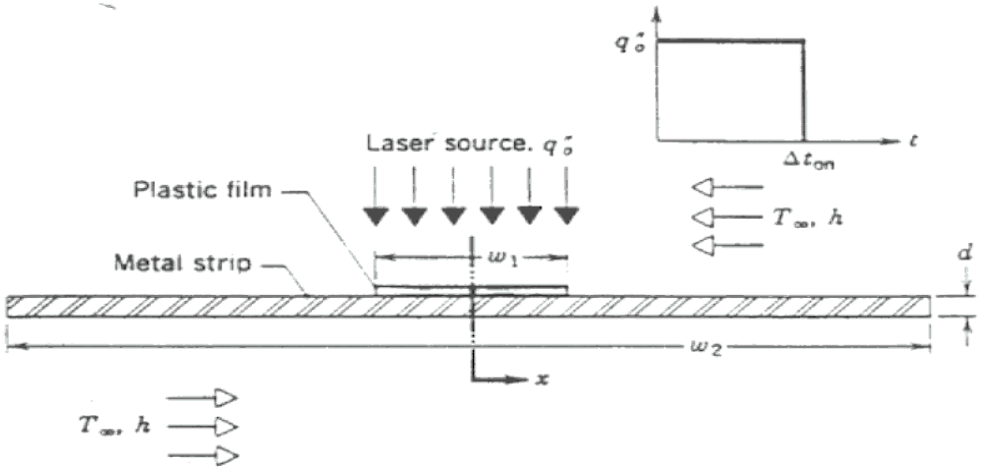
2. A bonding operation utilizes a laser to provide a constant heat flux, q", across the top of a thin adhesive-backed plastic film to be affixed to a metal strip as shown in the sketch. The metal strip has a thickness d=1.25 mm and its width (w_2) is large relative to that of the film (w_1=40 mm). The strip and film are very long in the direction normal to the page. The properties of the metal

strips are: density (7850 kg/m^3), specific heat (435 J/kg.K) and conductivity (60 W/m.K). The thermal resistance of the plastic film (including the adhesive) can be neglected.

The strip is initially at 25 ^{0}C and the laser provides a uniform heat flux of 85,000 W/m^2 over a time interval of 10 sec. The upper and lower surfaces of the strip (including the plastic film) experience convective cooling with air at 25 ^{0}C and a convective coefficient of 100 W/m^2.K. To cure the adhesive satisfactorily, temperature must be above 90 ^{0}C for a duration of 10 sec. However, temperature should be kept below 200 ^{0}C to prevent thermal degradation of the film.

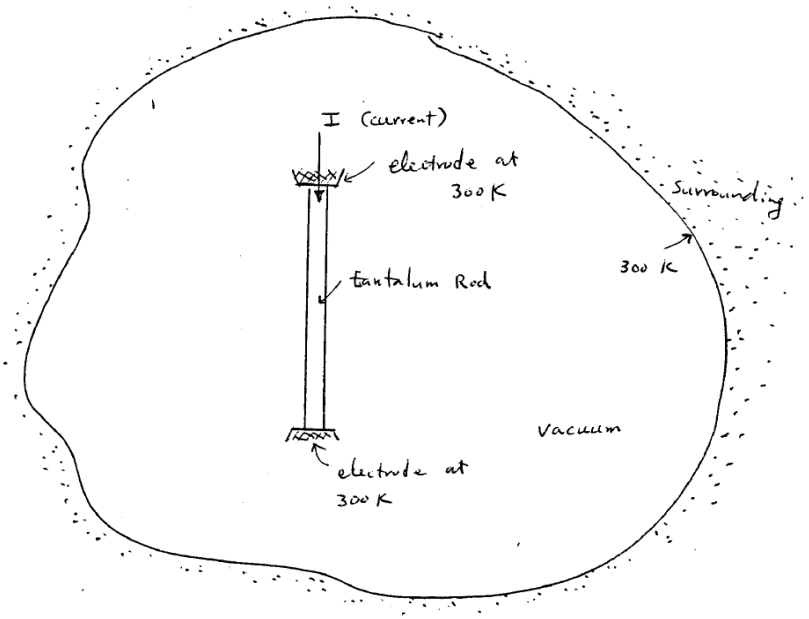
(a) Obtain temperature histories at the center (x=0) and film edge (x=w$_1$/2) for 30 sec. Use x=4 mm and t=1.0sec.

(b) Is the curing satisfactory?



3. A tantalum rod of 3mm-dia and 120 mm-long is supported by two electrodes within a large vacuum enclosure. Initially the rod is in thermal equilibrium with the electrodes and its surroundings, which are maintained at 300 K. Suddenly, electrical current of 80 Amperes is being passed through the rod. Assume the emissivity of the rod is 0.1 and the electrical resistivity is 9.5x10^{-7} ohm.m. The density of the tantalum is 1.66x10^4 kg/m^3, C=147 J/kg.K, and k=58.8 W/m.K. Assume all thermodynamic properties are constant. Use 12 uniform control volumes.

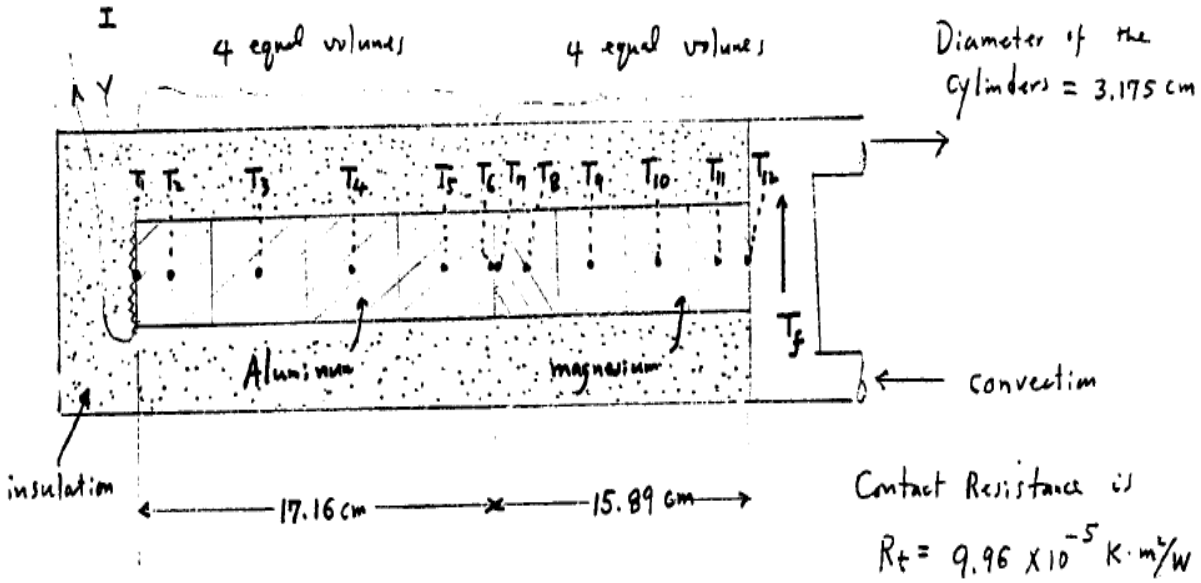
(a) Derive the conduction equation appropriate for this problem.
(b) Linearize the source term.
(c) Determine the steady state temperature and estimate the time to reach steady state.
(d) Take an energy balance for the steady state.



4. The figure shows a laboratory equipment to measure contact resistance between different materials. Aluminum and magnesium cylinders are used in the present problem. At the beginning of an experiment, the cylinders are at the room temperature of 20 ^{0}C. At time greater than zero, electric current flows through resistance attached to the aluminum cylinder. Energy dissipated by the electric current is 27.037 W. At the same time right end of the magnesium cylinder is exposed to a convective environment with T_f=319 K and the convective coefficient is very large due to very high flow speed. The contact resistance is 9.96×10^{-5} K.m^2/W.

(a) Determine the approximate time at which steady-state is reached.

(b) Plot temperature profiles (including contact surface temperatures) at several time steps.



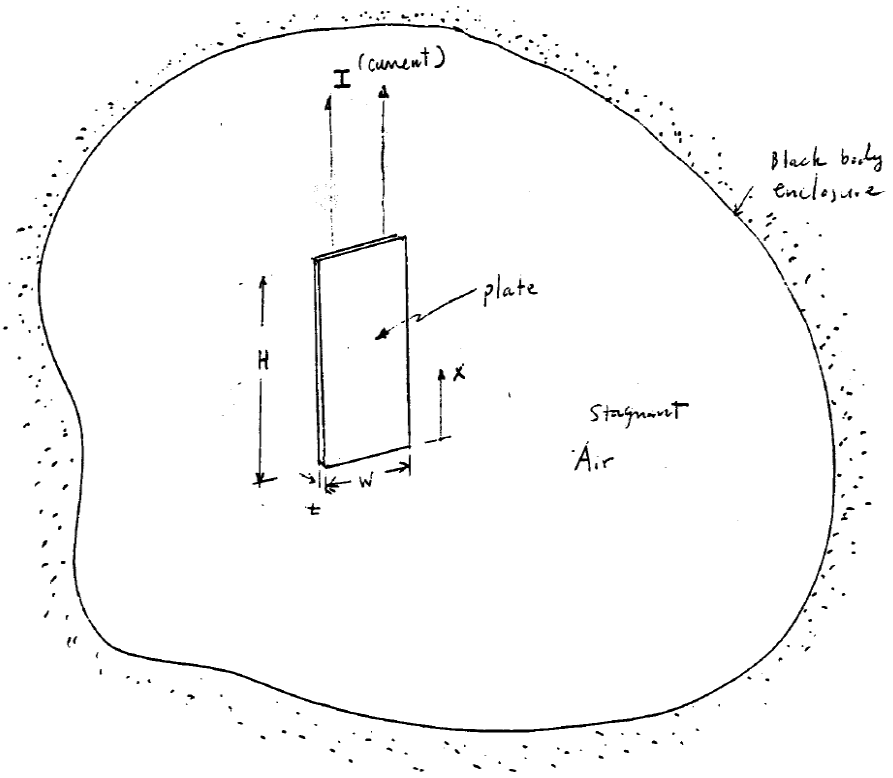
5. A small rectangular plate is suspended vertically inside a large enclosure. The medium between the plate and the bounding surface of the enclosure is air. The air is stagnant except near the plate caused by the buoyancy force. At the beginning, the plate is in thermal equilibrium with its surroundings, which are at a room temperature of 70 ^{0}F. At time greater than zero, energy is generated uniformly in the plate. The total energy generation is 320 W. The plate is made of aluminum. Total mass of the plate is 3.13 kg and the dimensions are H=25.875 inch, W=6.96 inch and thickness=0.86 inch. Emissivity of the plate is 0.86. Assume 1-dimension transient conduction along the vertical direction. Assume all edges are insulated.

(a) Derive the conduction equation appropriate for this problem. (Energy loss from the plate to its surroundings is caused by natural convection and radiation.)

(b) Calculate the temperature distributions along the plate and plot the results at several time steps until the steady-state is reached. Assume thermodynamic properties of the plate are temperature independent.

(c) Take the energy balance at the steady-state.

(d) Assume thermodynamic properties are temperature dependent, repeat the calculations and compare the results with the results of constant property case.



6. A Trombe wall is often used in passive solar energy system for residential building. A wall is fabricated using 20-cm thick solid concrete block. Exterior wall is coated with black paint to absorb solar energy and is exposed to the atmospheric air. Solar radiation flux absorbed at the surface is approximated by

$$q''_{solar} = t(375 - 46.875t)\,W/m^2$$

where t is in hr. Note that solar energy is available for 8 hours during day time. When the solar energy is not available, a shutter made of good insulating blanket is drawn to prevent energy loss from the outside wall. During the day time when the solar energy is available, the atmospheric air temperature is constant at 0 ^{0}C. So there is energy loss from the outer wall by natural convection and radiation. The inside room temperature is 15 ^{0}C and constant. Heat transfer from the inner wall to the room air is also by radiation and natural convection. Assume combined heat transfer coefficient on both walls is 10 W/m^2.K. Thermal properties of the concrete wall are density (1800 kg/m^3), specific heat (800 J/kg.K) and conductivity (0.7 W/m.K). Initial temperature of the wall is 15 ^{0}C.

We would like to analyze the effectiveness of this Trombe wall by numerical simulation. Modify the input file (C5E1INP.FOR) and answer the following questions.

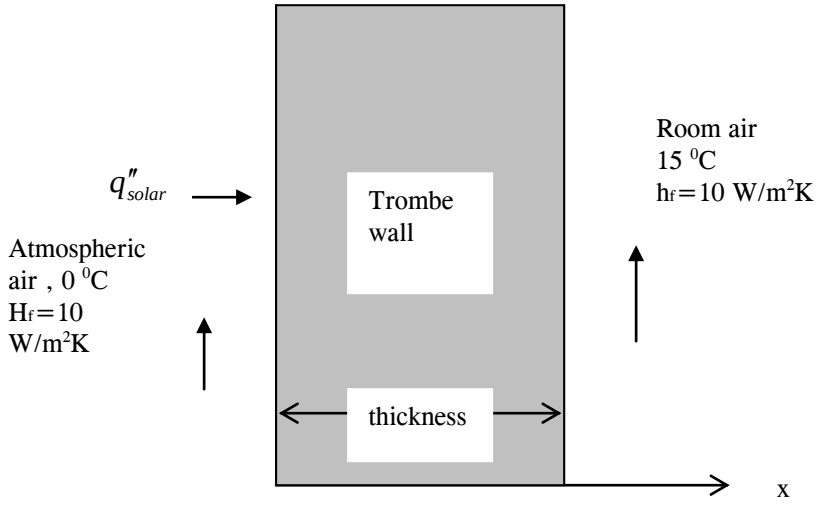
(1)Simulate the transient conduction for the first 24 hr duration and plot temperature history at the inner wall and at the outer wall. Use 10 control volumes and 10 min time step.

(2)Plot heat flux at the outer wall and at the inner wall as a function of time for the first 24 hour period.

Run the program for a longer period (say 72 hours) to find the cyclic behavior of Trombe wall performance.

(3) Plot heat flux at the inner wall as a function of time for one cycle and calculate the total energy delivered to the room air per unit area of Trombe wall per a cycle.

(4) What would you suggest to improve the performance of the Trombe wall? Performance will depend on the thickness, and materials of the wall and how to minimize loss from the outer wall during day hours, to name a few. (Extra credits)

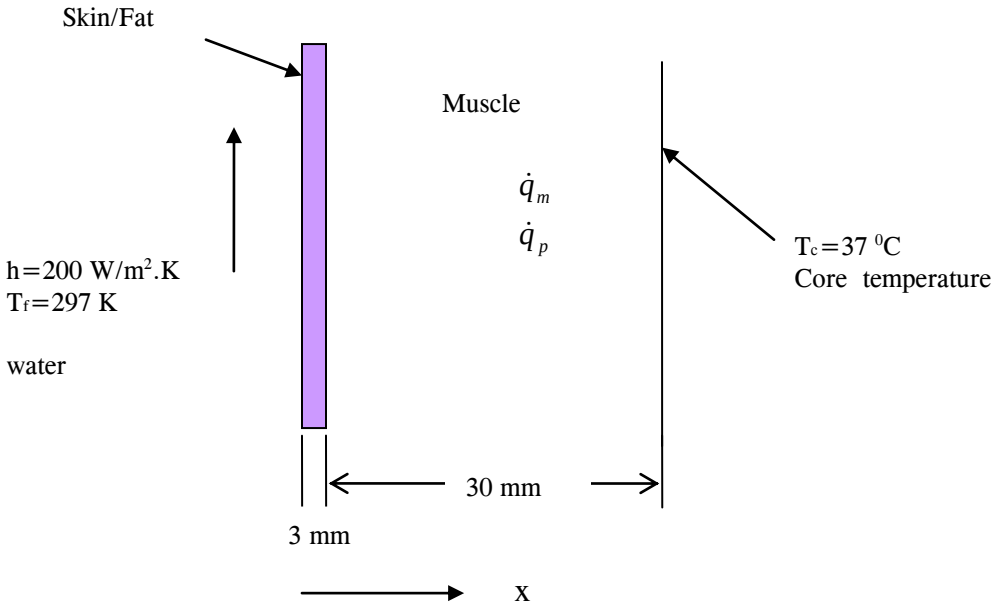


7. The topic of heat transfer in biological system is becoming increasingly important as new medical treatments are developed that involve extreme temperature environments. Heat transfer processes through living tissues are more complex than in conventional engineering materials because of metabolic heat generation and energy exchange between the flowing blood and the surrounding tissues. A simplified bio-heat transfer model introduced by Pennes is given by [Example 3.12, in our text]

$$\rho C \frac{\partial T}{\partial t} = \frac{1}{A_c} \frac{\partial}{\partial x}\left(A_c k \frac{\partial T}{\partial x} \right) + \dot{q}_m + \dot{q}_p$$

where $\dot{q}_m$ is metabolic heat source and $\dot{q}_p$ is perfusion heat source term in the muscle tissues. $\dot{q}_m = 700\ W/m^3$ *and* $\dot{q}_p = \omega\rho_b C_b(T_c - T)$. Perfusion rate, blood density and specific heat are $\omega = 0.0005/\text{sec}$, $\rho_b = 1000\ kg/m^3$, $C_b = 3600\ J/kg.K$, respectively.

The muscle and skin/fat layers subject to sudden convection are shown in the figure. The initial temperature of the muscle and skin/fat is at the core temperature inside the muscle (T_c). Thermal conductivity of the muscle is 0.5 W/m.K and of skin/fat is 0.3 W/m.K. Assume density and specific heat of both muscle and skin/fat are those of water, 1000 kg/m^3 and 4100 J/kg.K.



Calculate transient temperature variation in the system by modifying cond1d.m. Use 11 uniform control volumes, 10 sec time step for a duration of 10 min.

(a) Plot temperature profiles at several time steps

(b) Calculate the heat loss rate when t=10 min, per unit m^2. If average surface area of an adult is about 1.8 m^2, what is the total heat loss rate? The nominal heat generation rate is about 100.