Lecture 6: Diffusion in a Cooling Fin

Introduction. In this lecture we will consider heat diffusion in a thin plate. The model of the temperature will have the form

newu = A oldu + b for the time dependent case and

u = A u + b for the steady state case.

In general, the matrix A can be extremely large, but it will also have a special structure with many more zeros than nonzero components.

Previously we considered the model of heat diffusion in a thin wire so that there was diffusion in only one direction. Now, we will extend the model to heat diffusion in two directions. One could attribute this to heat diffusion in the radial direction for the thin wire problem, or to heat diffusion for a thin cooling fin that might be used to cool a computer chip or an amplifier or a motor.

Model. The model is derived via the Fourier heat law. It can be formulated as either a continuous model or as a discrete model. A model for heat diffusion in a thin 2D plate where there is diffusion in both the x and y directions, but any diffusion in the z direction is minimal and ignored. The objective is to determine the temperature in the interior of the fin given the initial temperature and the temperature on the boundary. Problems similar to this come from the design of cooling fins or from the manufacturing of large metal objects, which must be cooled so as to not damage the interior of the object.

In order to generate a 2D time dependent model for heat transfer diffusion the Fourier heat law must be applied in both the x and y directions. The continuous and discrete 2D models are very similar to the 1D versions for the wire. In the continuous 2D model the temperature u will dependent on three variables, u(x,y,t).

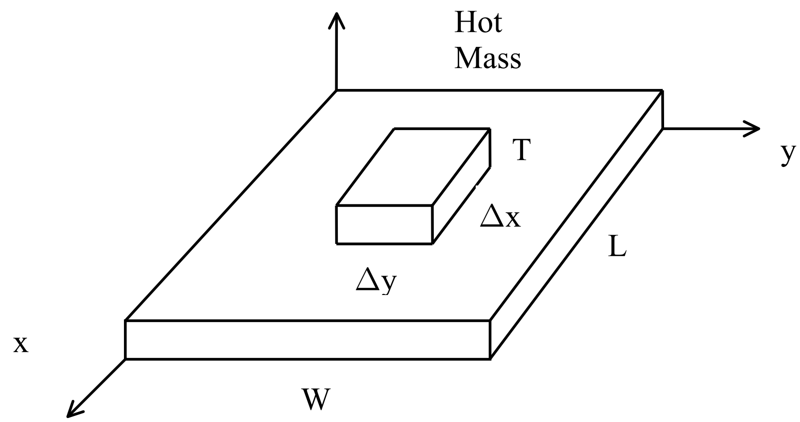


Figure: Diffusion in Two Directions

In order to model the temperature, we will first assume the temperature is given along the 2D boundary and that the thickness T is small. Consequently, there will be diffusion in just the x and y directions. Consider a small mass within the above plate whose volume is ($\Delta x\ \Delta y\ T$). This volume will have heat sources or sinks via the two ($\Delta x\ T$) surfaces, two ($\Delta y\ T$) surfaces, and two ($\Delta x\ \Delta y$) surfaces as well as any internal heat equal to f (heat/ (vol. time)). The top and bottom surfaces will be cooled by a Newton like law of cooling to the surrounding region whose temperature is u_{sur}. The Fourier heat law applied to each of the two directions will give the heat flowing through the four vertical surfaces:

$$\begin{aligned}\text{change in heat of } (\Delta x\ \Delta y\ T) = \rho c(u(x,y,t+\Delta t) - u(x,y,t)) &\approx f(x,y)\,(dx\ dy\ T)\,\Delta t \\ &+ (2\ \Delta x\ \Delta y)\,\Delta t\, C(u_{sur} - u(x,y,t)) \\ &+ (\Delta x\ T)\,\Delta t\,(Ku_y(x,y+\Delta y,t) - Ku_y(x,y,t)) \\ &+ (\Delta y\ T)\,\Delta t\,(Ku_x(x+\Delta x,y,t) - Ku_x(x,y,t)).\end{aligned}$$

This approximation gets more accurate as Δx , Δy and Δt go to zero. So, divide by ($\Delta x\ \Delta y\ T$) Δt and let Δx , Δy and Δt go to zero. This gives a differential equation with partial derivatives, and (1.1) is an example of a *partial differential equation*.

2D Diffusion Model for Cooling Fin.

$$\rho c u_t(x,y,t) = f(x,y,t) + (2C/T)(u_{sur} - u(x,y,t)) + (Ku_x(x,y,t))_x + (Ku_y(x,y,t))_y \quad (1.1)$$

for (x,y) in (0,L)x(0,W),
f(x,y) is the internal heat source,
ρ is the density,
c is the specific heat,
K is the thermal conductivity,
T is the small thickness of the plate and
C is the ability to transfer heat to the surrounding region.
u(x,y,0) = given initial condition and (1.2)
u(x,y,t) = given on the boundary of the fin. (1.3)

Explicit Finite Difference 2D Model of Heat Transfer.

Define $\alpha = (K/\rho c)(\Delta t/h^2)$ and let u_{ij}^{k+1} be the approximation of $u(ih,jh,(k+1)\Delta t)$.

$$u_{ij}^{k+1} = \Delta t/\rho c\,(f + 2C/T\,u_{sur}) + \alpha(u_{i-1,j}^k + u_{i+1,j}^k) + \alpha(u_{i,j-1}^k + u_{i,j+1}^k) + (1 - (\Delta t/\rho c)(2C/T) - 4\alpha)\,u_{i,j}^k \quad (2.1)$$

i,j = 1,...,n-1,
k = 0,..., maxk-1,
u_{ij}^0 = given for i,j = 1,..., n-1 and (2.2)
u_{ij}^k = given for k = 1,...,maxk and i,j = 0 or n. (2.3)

Stability Condition for (2).

$$1 - (\Delta t/\rho c)(2C/T) - 4\alpha > 0 \text{ and } \alpha > 0$$

Method. In order to execute the discrete model, there must be three nested loops. The outer loop must be the time loop (uses k), and the two inner loops are for the space grid (uses i and j). The order the i (x direction) and the j (y direction) is not important so long as they are both within the k (time loop).

Implementation. The following Matlab code is for heat diffusion on a thin plate, which has initial temperature equal to 70., and with temperature at boundary x = 0. equal to 370. for the first 120 time steps and then set equal to 70 after 120 time steps. The other temperatures on the boundary are always equal to 70. In the code we have used C = 0 so that no heat is lost via the large top and bottom surfaces of the fin; the heat flows from the hot mass through the fin to the other three smaller edges of the fin. The code in heat2d.m generates a 3D array with the temperatures for space and time. The code mov2dheat.m generates a sequence of 3D plots of the temperature versus space. One can see the heat moving from the hot side into the interior and then out the cooler boundaries. You may find it interesting to vary the parameters and also change the 3D plot to a contour plot by replacing mesh by contour.

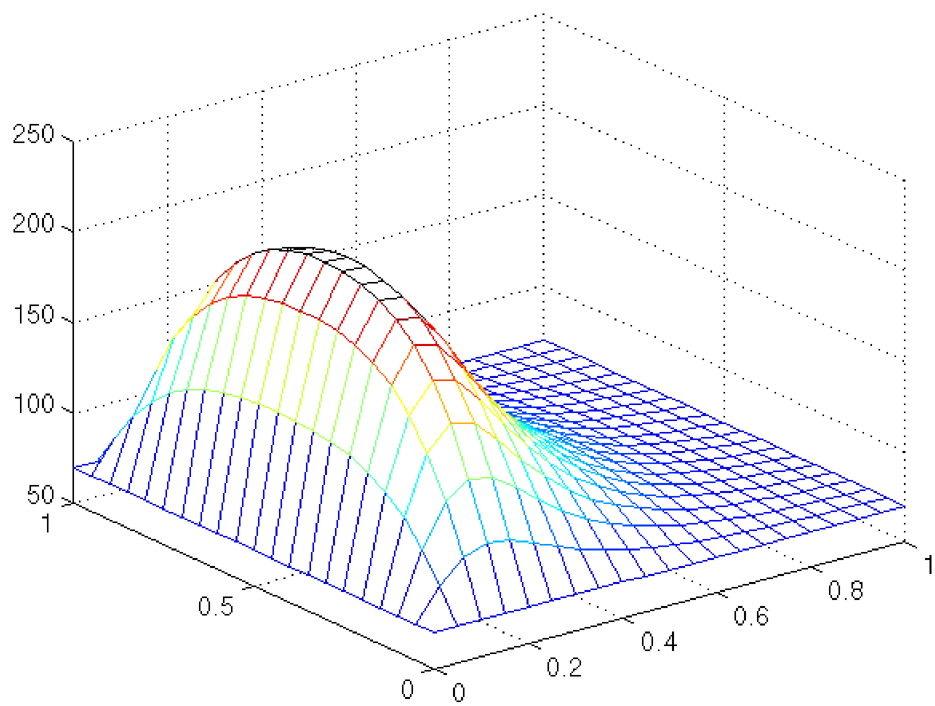
Matlab Codes for 2D Heat Flow (heat2d.m and mov2dheat.m).

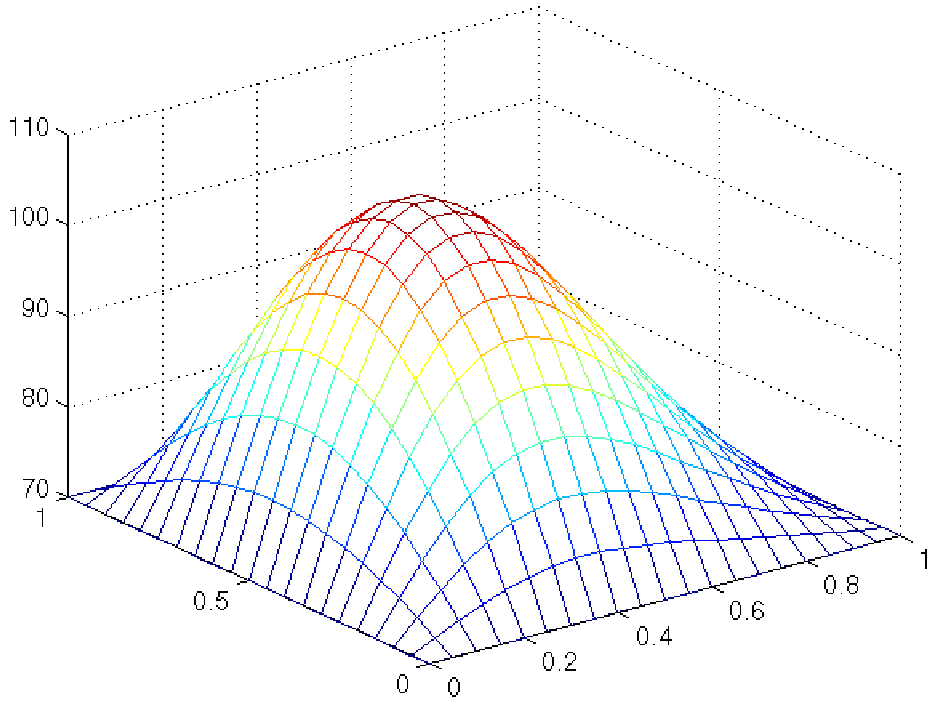
```
% Heat 2D Diffusion in File heat2d.m
clear;
L = 1.0;
W = L;
Tend = 80.;
maxk = 300;
dt = Tend/maxk;
n = 20.;
u(1:n+1,1:n+1,1:maxk+1) = 70.;
dx = L/n;
dy = W/n;
h = dx;
b = dt/(h*h);
cond = .002;
spheat = 1.0;
rho = 1.;
a = cond/(spheat*rho);
alpha = a*b;
for i = 1:n+1
   x(i) =(i-1)*h;
   y(i) =(i-1)*h;
end
for k=1:maxk+1
   time(k) = (k-1)*dt;
   for j=1:n+1
      u(1,j,k) =300.*(k<120)+ 70.;
   end
end
```

```
for k=1:maxk
    for j = 2:n
        for i = 2:n
            u(i,j,k+1) =0.*dt/(spheat*rho)+
                        (1-4*alpha)*u(i,j,k) + alpha*
                        (u(i-1,j,k)+u(i+1,j,k)+
                         u(i,j-1,k)+u(i,j+1,k));
        end
    end
end
mesh(x,y,u(:,:,maxk)')
```

% Generates Sequence of 3D plots in File mov2dheat.m

```
heat2d;
lim =[0 1 0 1 0 400];
for k=1:5:200
   mesh(x,y,u(:,:,k)')
   title ('heat versus space at different times' )
   axis(lim);
   k = waitforbuttonpress;
end
```





Assessment. The heat conduction in a thin plate has a number of approximations. Different mesh sizes in either the time or space variable will give different numerical results. However, if the stability conditions holds and the mesh size decreases, then the numerical computations will differ by smaller amounts. Also, the stability condition on the step sizes can still be a serious constraint. An alternative is to use an implicit time discretization, but this generates a sequence of linear problems, one for each time step!

Homework.

1. Experiment with the parameters in the 2D Matlab code. Observe the stability condition.
2. Observe the steady state solutions for both f = 0. (current code value) and f not equal to 0.
3. Modify the 2D Matlab code so that C is not zero. Experiment with different values of C to see how fast the fin cools.