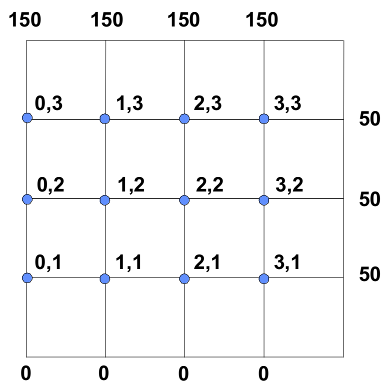
CHAPTER 29

29.1 Use Liebmann's method to solve for the temperature of the square heated plate in Fig. 29.4, but with the upper boundary condition increased to 175°C and the left boundary insulated. Use a relaxation factor of 1.2 and iterate to $\varepsilon_s = 1\%$.

The new representation of the plate is



Because the left edge is insulated, the finite-difference equations for the nodes on that edge are written as

(0, 3): $4T_{0,3} - 2T_{1,3} - T_{0,2} = 150$

(0, 2): $4T_{0,2} - T_{0,3} - 2T_{1,2} - T_{0,1} = 0$

(0, 1): $4T_{0,1} - 2T_{1,1} - T_{0,2} = 0$

All the other nodes are represented by Eq. (29.11). The first two iterations of Liebmann's method are

150	150	150	150	
45	58.5	62.55	84.615	50
0	0	0	19.5	50
0	0	0	15	50
0	0	0	0	

150	150	150	150	
75.15	81.09	91.935	86.2509	50
13.5	21.6	32.445	51.978	50
0	0	4.5	19.2	50
0	0	0	0	

After 10 iterations, the maximum approximate error is 0.754% with the result

150	150	150	150	
109.8519	108.8637	104.6117	91.54351	50
71.75664	70.97995	68.0271	61.55182	50
35.49834	35.32963	34.98164	36.63082	50
0	0	0	0	

Note that the ultimate result is

150	150	150	150	
109.9655	108.9359	104.6497	91.55766	50
71.9903	71.12847	68.10531	61.58093	50
35.73874	35.48233	35.0621	36.66076	50
0	0	0	0	

29.8 With the exception of the boundary conditions, the plate in Fig. P29.8 has the exact same characteristics as the plate used in Examples 29.1 through 29.3. Simulate both the temperatures and fluxes for the plate.

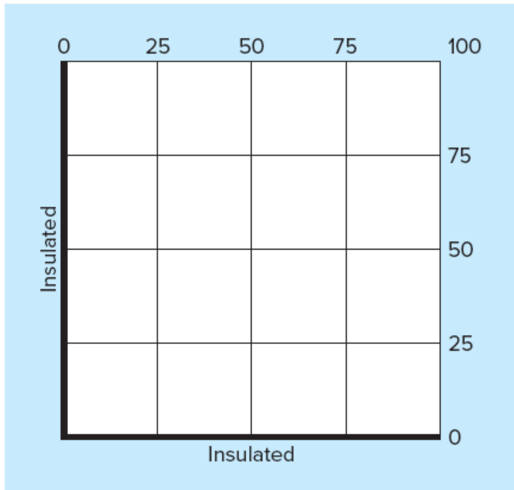
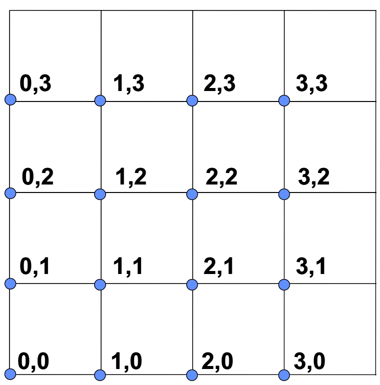


FIGURE P29.8

The nodes to be simulated are



Simple Laplacians are used for all interior nodes. Balances for the edges must take insulation into account. For example, node 1,0 is modeled as

$$4T_{1,0} - T_{0,0} - T_{2,0} - 2T_{1,1} = 0$$

The corner node, 0,0 would be modeled as

$$4T_{0,0} - 2T_{1,0} - 2T_{0,1} = 0$$

The resulting set of equations can be solved for

0	25	50	75	100
23.89706	**32.16912**	**45.58824**	**60.29412**	75
31.25	**34.19118**	**39.88971**	**45.58824**	50
32.72059	**33.45588**	**34.19118**	**32.16912**	25
32.72059	**32.72059**	**31.25**	**23.89706**	0

The fluxes can be computed as

J_x

-1.225	-1.225	-1.225	-1.225	-1.225
-0.40533	-0.53143	-0.68906	-0.72059	-0.72059
-0.14412	-0.21167	-0.27923	-0.2477	-0.21618
-0.03603	-0.03603	0.031526	0.225184	0.351287
0	0.036029	0.216176	0.765625	1.170956

J_y

1.170956	0.351287	-0.21618	-0.72059	-1.225
0.765625	0.225184	-0.2477	-0.72059	-1.225
0.216176	0.031526	-0.27923	-0.68906	-1.225
0.036029	-0.03603	-0.21167	-0.53143	-1.225
0	-0.03603	-0.14412	-0.40533	-1.225

J_n

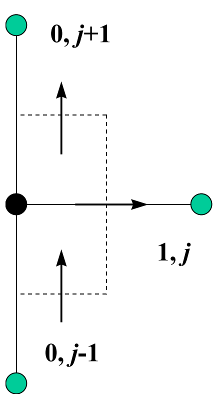
1.694628	1.274373	1.243928	1.421222	1.732412
0.866299	0.577174	0.732232	1.019066	1.421222
0.259812	0.214008	0.394888	0.732232	1.243928
0.050953	0.050953	0.214008	0.577174	1.274373
0	0.050953	0.259812	0.866299	1.694628

θ (degrees)

136.2922	163.999	-169.992	-149.534	-135
117.8973	157.0362	-160.228	-135	-120.466
123.6901	171.5289	-135	-109.772	-100.008
135	-135	-81.5289	-67.0362	-73.999
0	-45	-33.6901	-27.8973	-46.2922

29.11 Apply the control-volume approach to develop the equation for node (0, j) in Fig. 29.7.

The control volume is drawn as in



A flux balance around the node can be written as (note $\Delta x = \Delta y = h$)

$$-kh\Delta z \frac{T_{1,j} - T_{0,j}}{h} + k(h/2)\Delta z \frac{T_{0,j} - T_{0,j-1}}{h} - k(h/2)\Delta z \frac{T_{0,j+1} - T_{0,j}}{h} = 0$$

Collecting and canceling terms gives

$$4T_{0,j} - T_{0,j-1} - T_{0,j+1} - 2T_{1,j} = 0$$

29.20 Determine the temperature distribution and fluxes for the plate depicted in Fig. P29.20. The plate is 60 × 60 × 1 cm, is made out of aluminum [k' = 0.49 cal/(s cm °C)], with an input of 10 cal/s into the middle node.

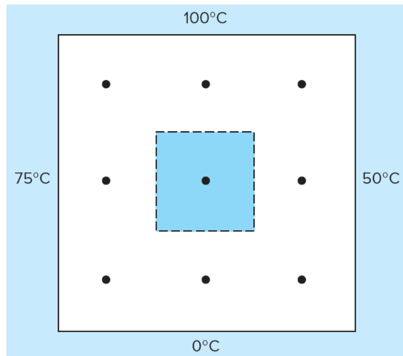


FIGURE P29.20

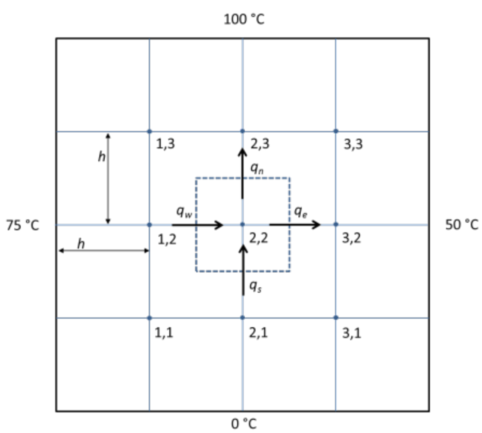


Figure P29.20

All of the internal node equations are the same as for the example from Section 29.2.1, except for node (2,2), where the input heat needs to be taken in to account.

$$q_{in} = q_w + q_s - q_n - q_e$$

$$Q_{in} = -k'\frac{T_{22} - T_{12}}{h} h\Delta z + -k'\frac{T_{22} - T_{21}}{h} h\Delta z - k'\frac{T_{22} - T_{32}}{h} h\Delta z - k'\frac{T_{22} - T_{32}}{h} h\Delta z$$

and with $\Delta z = 1$, dividing by h, and putting all of the temperatures on the left hand side,

$$T_{12} + T_{21} - 4T_{22} + T_{33} + T_{32} = \frac{Q_{in}}{k'}$$

Proceeding with this equation and Liebmann's method (λ = 1.5) in Excel, we generate the following coefficients and iterations

T_{11}	T_{21}	T_{31}	T_{12}	T_{22}	T_{32}	T_{13}	T_{23}	T_{33}	{T}	{b}
4	-1	0	-1	0	0	0	0	0	T_{11}	75
-1	4	-1	0	-1	0	0	0	0	T_{21}	0
0	-1	4	0	0	-1	0	0	0	T_{31}	50
-1	0	0	4	-1	0	-1	0	0	T_{12}	75
0	-1	0	-1	4	-1	0	-1	0	T_{22}	20.4082

0	0	-1	0	-1	4	0	0	-1	**T_{32}**	50
0	0	0	-1	0	0	4	-1	0	**T_{13}**	175
0	0	0	0	-1	0	-1	4	-1	**T_{23}**	100
0	0	0	0	0	-1	0	-1	4	**T_{33}**	150

where the above table leads to iterative equations (first 3 are shown):

$$T_{11}^{1} = \frac{75 + T_{21}^{0} + T_{12}^{0}}{4}\lambda + (1-\lambda)T_{11}^{0}$$

$$T_{21}^{1} = \frac{0 + T_{11}^{1} + T_{31}^{0} + T_{22}^{0}}{4}\lambda + (1-\lambda)T_{21}^{0}$$

$$T_{31}^{1} = \frac{50 + T_{21}^{1} + T_{32}^{0}}{4}\lambda + (1-\lambda)T_{31}^{0}$$

i	***T_{11}***	***T_{21}***	***T_{31}***	***T_{12}***	***T_{22}***	***T_{32}***	***T_{13}***	***T_{23}***	***T_{33}***	***max ε_a (%)***
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
1	28.13	10.55	22.71	38.67	26.11	37.06	80.13	77.34	99.15	100.00
2	32.52	25.23	30.75	60.82	69.76	75.10	77.37	91.19	69.03	62.57
3	44.13	41.63	47.15	69.44	76.78	53.56	87.17	79.28	71.55	40.21
4	47.71	43.55	31.59	72.78	62.70	54.16	79.06	77.85	69.98	49.23
5	47.89	31.55	35.09	62.85	61.21	54.03	78.86	77.34	70.52	38.05
6	39.58	35.18	34.66	64.07	63.53	55.00	79.22	78.81	71.17	21.01
7	45.55	36.31	35.66	66.71	64.70	55.57	80.58	79.26	71.23	13.12
8	43.98	35.97	35.25	65.74	64.01	54.90	79.71	78.48	70.65	3.58
9	44.28	35.84	35.15	65.75	63.76	54.89	79.85	78.61	70.99	0.67

Temperatures, arranged in order of Fig. P29.20

79.85	78.61	70.99
65.75	63.76	54.89
44.28	35.84	35.15

Fluxes and directions

Total Flux

0.42184	0.45704	0.609506
0.45704	0.540594	0.443035
0.724111	0.45305	0.467456

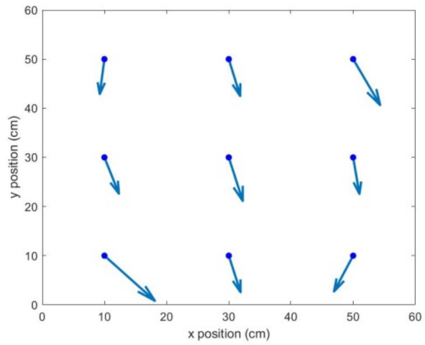
q_x

-0.04424	0.10863	0.257093
0.137692	0.133117	0.05987
0.47971	0.111795	-0.18188

q_y

-0.41951	-0.44394	-0.55263
-0.43581	-0.52395	-0.43897
-0.54242	-0.43904	-0.43062

And a plot of the directions (using `quiver` in MATLAB)



CHAPTER 30

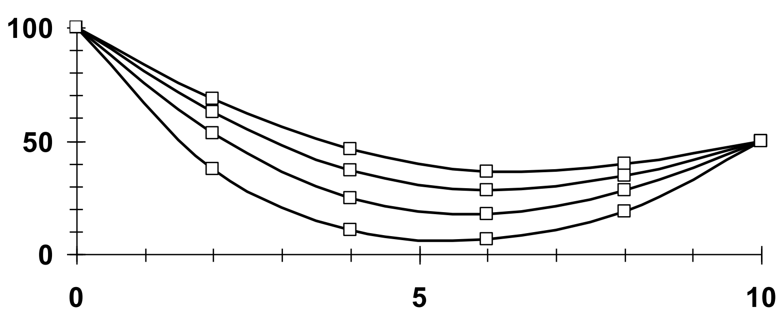
30.1 Repeat Example 30.1, but use the midpoint method to generate your solution.

The key to approaching this problem is to recast the PDE as a system of ODEs. Thus, by substituting the finite-difference approximation for the spatial derivative, we arrive at the following general equation for each node

$$\frac{dT_i}{dt} = k\frac{T_{i-1} - 2T_i + T_{i+1}}{\Delta x^2}$$

By writing this equation for each node, the solution reduces to solving 4 simultaneous ODEs with Heun's method. The results for the first two steps along with some later selected values are tabulated below. In addition, a plot similar to Fig. 30.4, is also shown

t	*x* = 0	*x* = 2	*x* = 4	*x* = 6	*x* = 8	*x* = 10
0	100	0	0	0	0	50
0.1	100	2.043923	0.021788	0.010894	1.021962	50
0.2	100	4.005178	0.084022	0.042672	2.002593	50
•						
•						
•						
3	100	37.54054	10.27449	6.442319	18.95732	50
6	100	53.24294	24.66052	17.4603	27.92251	50
9	100	62.39032	36.64937	27.84901	34.34692	50
12	100	68.71331	46.03498	36.54213	39.5355	50

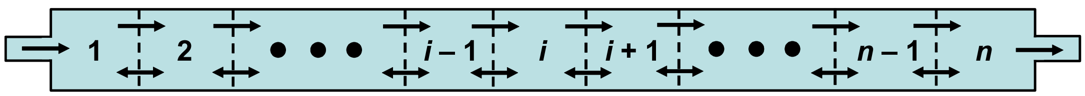


30.7 The advection-diffusion equation is used to compute the distribution of concentration along the length of a rectangular chemical reactor (see Sec. 32.1),

$$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial x^2} - U\frac{\partial c}{\partial x} - kc$$

where c = concentration (mg/m^3), t = time (min), D = a diffusion coefficient (m^2/min), x = distance along the tank's longitudinal axis (m) where $x = 0$ at the tank's inlet, U = velocity in the x direction (m/min), and k = a reaction rate (min^{-1}) whereby the chemical decays to another form. Develop an explicit scheme to solve this equation numerically. Test it for $k = 0.2$, $D = 75$, and $U = 2.5$ for a tank of length 20 m. Use $\Delta x = 2$ m and a step size of $\Delta t = 0.005$. Assume that the inflow concentration is 100 and that the initial concentration in the tank is zero. Perform the simulation from $t = 0$ to 100 and plot the final resulting concentrations versus x.

Although this problem can be modeled with the finite-difference approach (see Sec. 32.1), the control-volume method provides a more straightforward way to handle the boundary conditions. Thus, the tank is idealized as a series of control volumes:



The boundary fluxes and the reaction term can be used to develop the discrete form of the advection-diffusion equation for the interior volumes as

$$\Delta x\frac{dc_i^l}{dt} = -D\frac{c_i^l - c_{i-1}^l}{\Delta x} + D\frac{c_{i+1}^l - c_i^l}{\Delta x} + U\frac{c_i^l + c_{i-1}^l}{2} - U\frac{c_{i+1}^l + c_i^l}{2} - k\Delta x c_i^l$$

or dividing both sides by Δx,

$$\frac{dc_i^l}{dt} = D\frac{c_{i+1}^l - 2c_i^l + c_{i-1}^l}{\Delta x^2} - U\frac{c_{i+1}^l - c_{i-1}^l}{2\Delta x} - kc_i^l$$

which is precisely the form that would have resulted by substituting centered finite difference approximations into the advection-diffusion equation.

For the first boundary node, no diffusion is allowed up the entrance pipe and advection is handled with a backward difference,

$$\Delta x\frac{dc_1^l}{dt} = D\frac{c_2^l - c_1^l}{\Delta x} + Uc_0^l - U\frac{c_2^l + c_1^l}{2} - k\Delta x c_1^l$$

or dividing both sides by Δx,

$$\frac{dc_1^l}{dt} = D\frac{c_2^l - c_1^l}{\Delta x^2} + U\frac{2c_0^l - c_2^l - c_1^l}{2\Delta x} - kc_1^l$$

For the last boundary node, no diffusion is allowed through the exit pipe and advection out of the tank is again handled with a backward difference,

$$\Delta x\frac{dc_n^l}{dt} = D\frac{c_{n-1}^l - c_n^l}{\Delta x} + U\frac{c_n^l + c_{n-1}^l}{2} - Uc_n^l - k\Delta x c_n^l$$

or dividing both sides by Δx,

$$\frac{dc_n^l}{dt} = D\frac{c_{n-1}^l - c_n^l}{\Delta x^2} + U\frac{c_{n-1}^l - c_n^l}{2\Delta x} - kc_n^l$$

By writing these equations for each equally-spaced volume, the PDE is transformed into a system of ODEs. Explicit methods like Euler's method or other higher-order RK methods can then be used to solve the system.

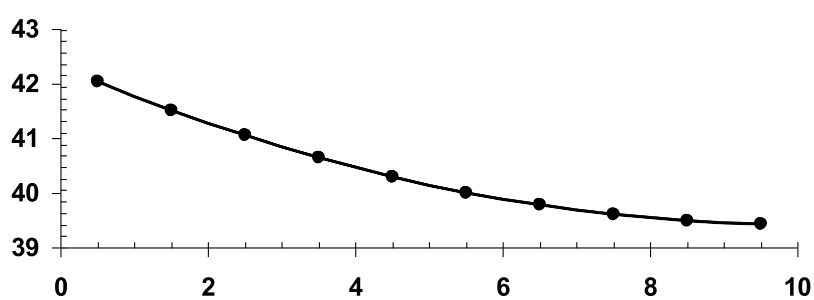
We can set up the Euler's method as follows using the system of equations:

$$c_i^l = c_i^l + c_i^l \times \Delta t$$

The results with an initial condition that the reactor has zero concentration with an inflow concentration of 100 (using Euler with a step size of 0.005) for t = 100 are

x	0.5	2.5	4.5	6.5	8.5	10.5	12.5	14.5	16.5	18.5
c	49.8987	46.9940	44.4076	42.1328	40.1660	38.5068	37.1581	36.1264	35.4222	35.0603

A plot of the results is shown below:



30.13 The nondimensional form for the transient heat conduction in an insulated rod (Eq. 30.1) can be written as

$$\frac{\partial^2 u}{\partial \bar{x}^2} = \frac{\partial u}{\partial \bar{t}}$$

where nondimensional space, time, and temperature are defined as

$$\bar{x} = \frac{x}{L} \qquad \bar{t} = \frac{T}{(\rho C L^2 / k)} \qquad u = \frac{T - T_o}{T_L - T_o}$$

where L = the rod length, k = thermal conductivity of the rod material, ρ = density, C = specific heat, T_o = temperature at x = 0, and T_L = temperature at $x = L$. This yields the following boundary and initial conditions:

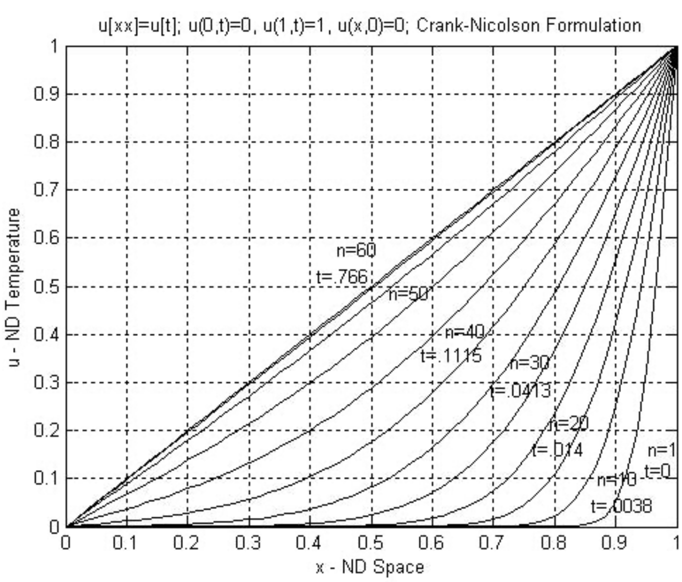
Boundary conditions	$u(0, \bar{t}) = 0$	$u(1, \bar{t}) = 0$
Initial conditions	$u(\bar{x}, 0) = 0$	$0 \le \bar{x} \le 1$

Solve this nondimensional equation for the temperature distribution using finite-difference methods and a second-order accurate Crank-Nicolson formulation to integrate in time. Write a computer program to obtain the solution. Increase the value of $\Delta\bar{t}$ by 10% for each time step to more quickly obtain the steady-state solution, and select values of $\Delta\bar{x}$ and $\Delta\bar{t}$ for good accuracy. Plot the nondimensional temperature versus nondimensional length for various values of nondimensional times.

MATLAB solution:

```
%PDE Parabolic Problem - Heat conduction in a rod
%   u[xx]=u[t]
%   BC u(0,t)=0   u(1,t)=1
%   IC u(x,0)=0   x<1
%   i=spatial index, from 1 to imax
%   imax = no. of x points
%   n=time index  from 1 to nmax
%   nmax = no. of time steps,
%   Crank-Nicolson Formulation
imax=61;
nmax=60;     % last time step = nmax+1
%  Constants
dx=1/(imax-1);
dx2=dx*dx;
dt=dx2; % Setting dt to dx2 for good stability and results
%  Independent space variable
x=0:dx:1;
%  Sizing matrices
u=zeros(imax,nmax+1); t=zeros(1,nmax+1);
a=zeros(1,imax);  b=zeros(1,imax);
c=zeros(1,imax);  d=zeros(1,imax);
ba=zeros(1,imax); ga=zeros(1,imax);
up=zeros(1,imax);
%  Boundary Conditions
u(1,1)=0;
u(imax,1)=1;
%  Time step loop
%  n=1 represents 0 time, n+1 = next time step
t(1)=0;
for n=1:nmax
  t(n+1)=t(n)+dt;
  %    Boundary conditions & Constants
  u(1,n+1)=0;
  u(imax,n+1)=1;
  dx2dt=dx2/dt;
  % coefficients
  b(2)=-2-2*dx2dt;
  c(2)=1;
  d(2)=(2-2*dx2dt)*u(2,n)-u(3,n);
  for i=3:imax-2
    a(i)=1;
```

```
    b(i)=-2-2*dx2dt;
    c(i)=1;
    d(i)=-u(i-1,n)+(2-2*dx2dt)*u(i,n)-u(i+1,n);
  end
  a(imax-1)=1;
  b(imax-1)=-2-2*dx2dt;
  d(imax-1)=-u(imax-2,n)+(2-2*dx2dt)*u(imax-1,n)-2;
  % Solution by Thomas Algorithm
  ba(2)=b(2);
  ga(2)=d(2)/b(2);
  for i=3:imax-1
    ba(i)=b(i)-a(i)*c(i-1)/ba(i-1);
    ga(i)=(d(i)-a(i)*ga(i-1))/ba(i);
  end
        % Back substitution step
  u(imax-1,n+1)=ga(imax-1);
  for i=imax-2:-1:2
    u(i,n+1)=ga(i)-c(i)*u(i+1,n+1)/ba(i);
  end
  dt=1.1*dt;
end
% end of time step loop
% Plot
% Storing plot value of u as up, at every 5 time steps, np=5
% j=time index
% i=space index
np=5;
for j=np:np:nmax
  for i=1:imax
    up(i)=u(i,j);
  end
  plot(x,up)
  hold on
end
grid
title('u[xx]=u[t]; u(0,t)=0, u(1,t)=1, u(x,0)=0; Crank-Nicolson Formulation')
xlabel('x - ND Space')
ylabel('u - ND Temperature')
hold off
gtext('n=60');gtext('n=50');gtext('n=40');gtext('n=30');
gtext('n=20');gtext('n=10');gtext('n=1');gtext('t=.766');
gtext('t=.1115');gtext('t=.0413');gtext('t=.014');
gtext('t=.0038');gtext('t=0')
```



30.14 The problem of transient radial heat flow in a circular rod in nondimensional form is described by

$$\frac{\partial^2 u}{\partial \bar{r}^2} + \frac{1}{\bar{r}}\frac{\partial u}{\partial \bar{r}} = \frac{\partial u}{\bar{t}}$$

Boundary conditions	$u(1, \bar{t}) = 1$	$\frac{\partial u}{\partial \bar{t}}(0, \bar{t}) = 0$
Initial conditions	$u(\bar{x}, 0) = 0$	$0 \le \bar{x} \le 1$

Solve the nondimensional transient radial heat-conduction equation in a circular rod for the temperature distribution at various times as the rod temperature approaches steady state. Use second-order accurate finite-difference analogues for the derivatives, with a Crank-Nicolson formulation. Write a computer program for the solution. Select values of $\Delta \bar{r}$ and $\Delta \bar{t}$ for good accuracy. Plot the temperature u versus radius $\bar{r}$ for various times $\bar{t}$.

$$\frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} = \frac{\partial u}{\partial t}$$

Substituting of second order correct Crank-Nicolson analogues

$$\frac{\partial^2 u}{\partial r^2} = \frac{1}{2}\left[\frac{u_{i+1,n+1} - 2u_{i,n+1} + u_{i-1,n+1}}{\Delta r^2} + \frac{u_{i+1,n} - 2u_{i,n} + u_{i-1,n}}{\Delta r^2}\right]$$

$$\frac{\partial u}{\partial r} = \frac{1}{2}\left[\frac{u_{i+1,n+1} - u_{i-1,n+1}}{2\Delta r} + \frac{u_{i+1,n} - u_{i-1,n}}{2\Delta r}\right]$$

$$r = (i-1)\Delta r$$

$$\frac{\partial u}{\partial t} = \frac{u_{i,n+1} - u_{i,n}}{\Delta t}$$

into the governing equation give the following finite difference equations:

$$\left[1 - \frac{1}{2(i-1)}\right]u_{i-1,n+1} + \left[-2 - 2\frac{\Delta r^2}{\Delta t}\right]u_{i,n+1} + \left[1 + \frac{1}{2(i-1)}\right]u_{i+1,n+1} = \left[-1 + \frac{1}{2(i-1)}\right]u_{i-1,n}$$
$$+ \left[2 - 2\frac{\Delta r^2}{\Delta t}\right]u_{i,n} + \left[-1 - \frac{1}{2(i-1)}\right]u_{i+1,n}$$

For the end points:

$x = 1$ (i = R), substitute the value of $u_R = 1$ into the above FD equation
$x = 0$ ($i = 1$), set the FD analog to the first derivative = 0

$$\left[\frac{\partial u}{\partial r}\right]_{i=1} = \frac{1}{2}\left[\frac{u_{2,n+1} - u_{0,n+1}}{2\Delta r} + \frac{u_{2,n} - u_{0,n}}{2\Delta r}\right] = 0$$

Also substitute in $i = 1$ into the finite difference equation and algebraically eliminate $u_{0,n+1} + u_{0,n}$ from the two equations and get the FD equation at $i = 1$:

$$\left[-2-2\frac{\Delta r^2}{\Delta t}\right]u_{1,n+1}+\left[2\right]u_{2,n+1}=-\left[2-2\frac{\Delta r^2}{\Delta t}\right]u_{1,n}+\left[-2\right]u_{2,n}$$

Here is a MATLAB implementation:

```
%PDE Parabolic Problem - Heat conduction in the radial direction in a circular rod
%  u[rr]+(1/r)u[r]=u[t]        0<r<1
%  BC   u(1,t)=1   u[r](0,t)=0
%  IC   u(r,0)=0          0<r<1
%  i=spatial index, from 1 to imax
%  imax = no. of r points (imax=21 for 20 dr spaces)
%  n=time index  from 1 to nmax
%  nmax = no. of time steps,
%  Crank-Nicolson Formulation
imax=41;
nmax=60;                                    % last time step = nmax+1
%  Constants
dr=1/(imax-1);
dr2=dr*dr;
dt=dr2; % Setting dt to dr2 for good stability and results
%  Independent space variable
r=0:dr:1;
%  Sizing matrices
u=zeros(imax,nmax+1); t=zeros(1,nmax+1);
a=zeros(1,imax);  b=zeros(1,imax);
c=zeros(1,imax);  d=zeros(1,imax);
ba=zeros(1,imax); ga=zeros(1,imax);
up=zeros(1,imax);
%  Boundary Conditions
u(imax,1)=1;
%  Time step loop
%  n=1 represents 0 time, new time = n+1
t(1)=0;
for n=1:nmax
  t(n+1)=t(n)+dt;
  % Boundary conditions & Constants
  u(imax,n+1)=1;
  dr2dt=dr2/dt;
  % coefficients
  b(1)=-2-2*dr2dt;
  c(1)=2;
  d(1)=(2-2*dr2dt)*u(1,n)-2*u(2,n);
  for i=2:imax-2
    a(i)=1-1/(2*(i-1));
    b(i)=-2-2*dr2dt;
    c(i)=1+1/(2*(i-1));
    d(i)=(-1+1/(2*(i-1)))*u(i-1,n)+(2-2*dr2dt)*u(i,n)+(-1-1/(2*(i-1)))*u(i+1,n);
  end
  a(imax-1)=1-1/(2*(imax-2));
  b(imax-1)=-2-2*dr2dt;
  d(imax-1)=(-1+1/(2*(imax-2)))*u(imax-2,n)+(2-2*dr2dt)*u(imax-1,n)-2*(1+1/(2*(imax-2)));
  % Solution by Thomas Algorithm
  ba(1)=b(1);
  ga(1)=d(1)/b(1);
  for i=2:imax-1
    ba(i)=b(i)-a(i)*c(i-1)/ba(i-1);
    ga(i)=(d(i)-a(i)*ga(i-1))/ba(i);
  end
  % Back substitution step
  u(imax-1,n+1)=ga(imax-1);
  for i=imax-2:-1:1
    u(i,n+1)=ga(i)-c(i)*u(i+1,n+1)/ba(i);
  end
  dt=1.1*dt;
end
```

```
% end of time step loop
% Plot
% Storing plot value of u as up, at every 5 time steps
% j=time index
% i=space index
istart=4;
for j=istart:istart:nmax+1
  for i=1:imax
    up(i)=u(i,j);
  end
  plot(r,up)
  hold on
end
grid
title('u[rr]+(1/r)u[r]=u[t]; u(1,t)=1; u[r](0,t)=0; u(r,0)=0')
xlabel('r - ND Space')
ylabel('u - ND Temperature')
hold off
```