

CHAPTER 24

24.1 (a) The solution can be assumed to be $T = e^{\lambda x}$. This, along with the second derivative $T'' = \lambda^2 e^{\lambda x}$, can be substituted into the differential equation to give

$$\lambda^2 e^{\lambda x} - 0.15 e^{\lambda x} = 0$$

which can be used to solve for

$$\lambda^2 - 0.15 = 0$$

$$\lambda = \pm\sqrt{0.15}$$

Therefore, the general solution is

$$T = Ae^{\sqrt{0.15} x} + Be^{-\sqrt{0.15} x}$$

The constants can be evaluated by substituting each of the boundary conditions to generate two equations with two unknowns,

$$240 = A + B$$

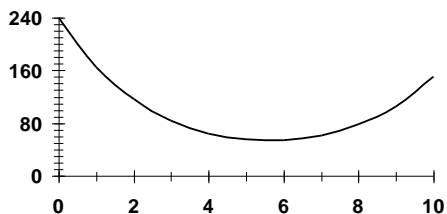
$$150 = 48.08563A + 0.020796B$$

which can be solved for $A = 3.016944$ and $B = 236.9831$. The final solution is, therefore,

$$T = 3.016944e^{\sqrt{0.15} x} + 236.9831e^{-\sqrt{0.15} x}$$

which can be used to generate the values below:

x	T
0	240
1	165.329
2	115.7689
3	83.79237
4	64.54254
5	55.09572
6	54.01709
7	61.1428
8	77.55515
9	105.7469
10	150



(b) Re-express the second-order equation as a pair of ODEs:

$$\frac{dT}{dx} = z \quad \frac{dz}{dx} = 0.15T$$

These can be stored in a function

```
function dy=prob2401sys(x,y)
dy=[y(2);0.15*y(1)];
```

The solution was then generated with the following script. Note that we have generated a plot of all the shots as well as the analytical solution.

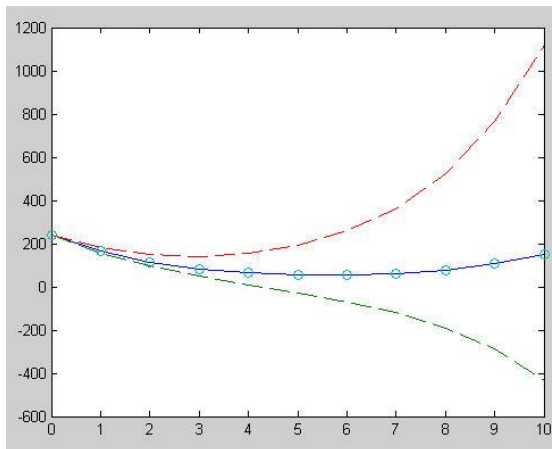
```
xi=0;xf=10;
xe=[xi:xf];Te=3.016944*exp(sqrt(0.15)*xe)+236.9831*exp(-
sqrt(0.15)*xe);
za1=-100;za2=-75;Ta=240;Tb=150;
[x1,T1]=ode45(@prob2401sys,xe,[Ta za1]);
Tb1=T1(length(T1));
[x2,T2]=ode45(@prob2401sys,xe,[Ta za2]);
Tb2=T2(length(T2));
za=za1+(za2-za1)/(Tb2-Tb1)*(Tb-Tb1);
[x,T]=ode45(@prob2401sys,xe,[Ta za]);
plot(x,T(:,1),x1,T1(:,1),'--',x2,T2(:,1),'--',xe,Te,'o')
disp('results:')
fprintf('1st shot:  za1 = %8.4g Tb1 = %8.4g\n',za1,Tb1)
fprintf('2nd shot:  za2 = %8.4g Tb2 = %8.4g\n',za2,Tb2)
fprintf('Final shot: za = %8.4g T = %8.4g\n',za,T(length(T)))
fprintf('\n      x      T      dT/dx\n')
disp([x T])
```

The results are

```
results:
1st shot:  za1 =      -100 Tb1 =    -432.4
2nd shot:  za2 =      -75 Tb2 =     1119
Final shot: za =    -90.61 T =     150

      x      T      dT/dx
      0    240.0000   -90.6147
1.0000    165.3278   -60.5889
```

2.0000	115.7683	-39.7664
3.0000	83.7921	-24.9838
4.0000	64.5424	-13.9958
5.0000	55.0957	-5.1334
6.0000	54.0171	2.9492
7.0000	61.1429	11.4799
8.0000	77.5553	21.7541
9.0000	105.7471	35.3325
10.0000	150.0000	54.2772



(c) A centered finite difference can be substituted for the second derivative to give,

$$\frac{T_{i-1} - 2T_i + T_{i+1}}{h^2} - 0.15T_i = 0$$

or for $h = 1$,

$$-T_{i-1} + 2.15T_i - T_{i+1} = 0$$

The first interior node would be

$$2.15T_1 - T_2 = 240$$

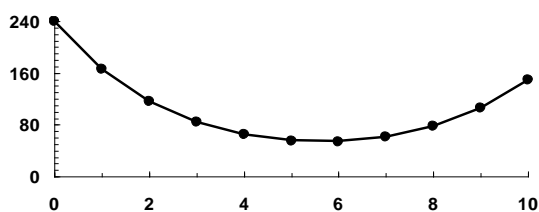
and the last interior node would be

$$-T_8 + 2.15T_9 = 150$$

The tridiagonal system can be solved with the Thomas algorithm or Gauss-Seidel for (the analytical solution is also included)

x	T	Analytical
0	240	240
1	165.7573	165.3290
2	116.3782	115.7689
3	84.4558	83.7924
4	65.2018	64.5425
5	55.7281	55.0957
6	54.6136	54.0171
7	61.6911	61.1428
8	78.0223	77.5552
9	106.0569	105.7469
10	150	150

The following plot of the results (with the analytical shown as filled circles) indicates close agreement.



24.2 (a) The solution can be assumed to be $T = e^{\lambda x}$. This, along with the second derivative $T'' = \lambda^2 e^{\lambda x}$, can be substituted into the differential equation to give

$$\lambda^2 e^{\lambda x} - 0.15 e^{\lambda x} = 0$$

which can be used to solve for

$$\lambda^2 - 0.15 = 0$$

$$\lambda = \pm\sqrt{0.15}$$

Therefore, the general solution is

$$T = Ae^{\sqrt{0.15}x} + Be^{-\sqrt{0.15}x}$$

For the second condition, we can differentiate the

$$\frac{dT}{dx} = A\sqrt{0.15}e^{\sqrt{0.15}x} - B\sqrt{0.15}e^{-\sqrt{0.15}x}$$

and evaluate the result at $x = 10$,

$$\frac{dT}{dx} = A(0.387298)e^{(0.387298)10} - B(0.387298)e^{-(0.387298)10} = 18.62348A - 0.00805B = 0$$

Therefore, the boundary conditions are expressed as two equations with two unknowns,

$$240 = A + B$$

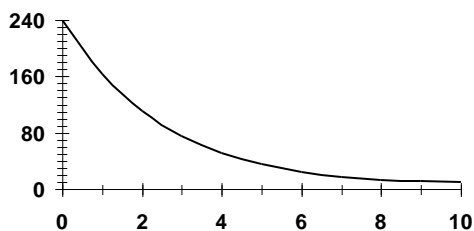
$$0 = 18.62348 - 0.00805B$$

which can be solved for $A = 0.10375$ and $B = 239.896$. The final solution is, therefore,

$$T = 0.10375e^{\sqrt{0.15}x} + 239.89625e^{-\sqrt{0.15}x}$$

which can be used to generate the values below:

x	T
0	240.000
1	163.016
2	110.791
3	75.393
4	51.447
5	35.315
6	24.546
7	17.506
8	13.124
9	10.736
10	9.978



(b) Re-express the second-order equation as a pair of ODEs:

$$\frac{dT}{dx} = z \quad \frac{dz}{dx} = 0.15T$$

These can be stored in a function

```
function dy=prob2401sys(x,y)
dy=[y(2);0.15*y(1)];
```

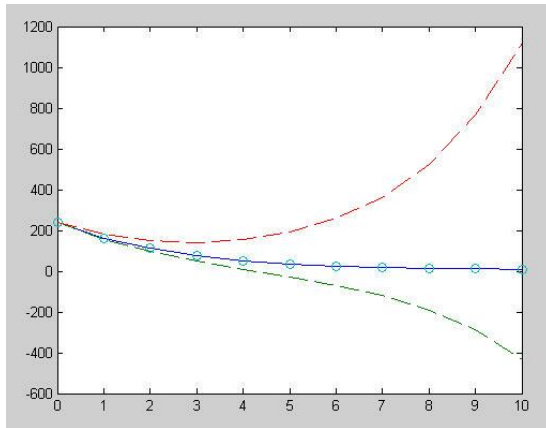
The solution was then generated with the following script. Note that we have generated a plot of all the shots as well as the analytical solution.

```
xi=0;xf=10;
xe=[xi:xf];Te=0.10375*exp(sqrt(0.15)*xe)+239.89625*exp(-
sqrt(0.15)*xe);
za1=-100;za2=-75;Ta=240;zb=0;
[x1,T1]=ode45(@prob2401sys,xe,[Ta za1]);
zb1=T1(length(T1),2);
[x2,T2]=ode45(@prob2401sys,xe,[Ta za2]);
zb2=T2(length(T2),2);
za=za1+(za2-za1)/(zb2-zb1)*(zb-zb1);
[x,T]=ode45(@prob2401sys,xe,[Ta za]);
plot(x,T(:,1),x1,T1(:,1),'--',x2,T2(:,1),'--',xe,Te,'o')
disp('results:')
fprintf('1st shot:  za1 = %8.4g zb1 = %8.4g\n',za1,zb1)
fprintf('2nd shot:  za2 = %8.4g zb2 = %8.4g\n',za2,zb2)
fprintf('Final shot: za = %8.4g\n',za)
fprintf('\n      x      T      dT/dx\n')
disp([x T])
```

The results are

```
1st shot:  za1 =      -100 zb1 =     -171.5
2nd shot:  za2 =       -75 zb2 =      429.9
Final shot: za =     -92.87
```

x	T	dT/dx
0	240.0000	-92.8712
1.0000	163.0144	-63.0168
2.0000	110.7902	-42.7345
3.0000	75.3931	-28.9428
4.0000	51.4469	-19.5470
5.0000	35.3146	-13.1200
6.0000	24.5461	-8.6858
7.0000	17.5056	-5.5707
8.0000	13.1240	-3.3018
9.0000	10.7357	-1.5344
10.0000	9.9780	-0.0000



(c) A centered finite difference can be substituted for the second derivative to give,

$$\frac{T_{i-1} - 2T_i + T_{i+1}}{h^2} - 0.15T_i = 0$$

or for $h = 1$,

$$-T_{i-1} + 2.15T_i - T_{i+1} = 0$$

The first interior node would be

$$2.15T_1 - T_2 = 240$$

For the last node, we double the weight on the interior node to give

$$-2T_9 + 2.15T_{10} = 0$$

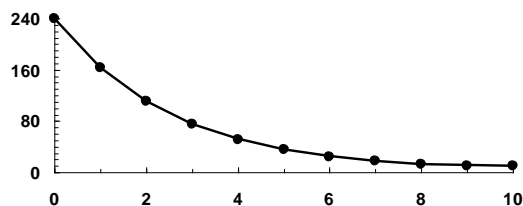
The tridiagonal system can be solved with the Thomas algorithm or Gauss-Seidel for (the analytical solution is also included)

x	T	Analytical
0	240	240
1	163.4080	163.0156
2	111.3270	110.7908
3	75.9448	75.3934
4	51.9541	51.4470
5	35.7561	35.3146
6	24.9212	24.5460
7	17.8241	17.5056

PROPRIETARY MATERIAL. © The McGraw-Hill Companies, Inc. All rights reserved. No part of this Manual may be displayed, reproduced or distributed in any form or by any means, without the prior written permission of the publisher, or used beyond the limited distribution to teachers and educators permitted by McGraw-Hill for their individual course preparation. If you are a student using this Manual, you are using it without permission.

8	13.4003	13.1239
9	10.9862	10.7356
10	10.2197	9.9778

The following plot of the results (with the analytical shown as filled circles) indicates close agreement.



24.3 The second-order ODE can be expressed as the following pair of first-order ODEs,

$$\begin{aligned}\frac{dy}{dx} &= z \\ \frac{dz}{dx} &= \frac{2z + y - x}{7}\end{aligned}$$

These can be solved for two guesses for the initial condition of z . For our cases we used -1 and -0.5 . We solved the ODEs with the Heun method without iteration using a step size of 0.125 . The results are

$z(0)$	-1	-0.5
$y(20)$	$-11,837.64486$	$22,712.34615$

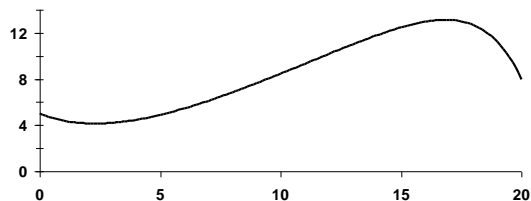
Clearly, the solution is quite sensitive to the initial conditions. These values can then be used to derive the correct initial condition,

$$z(0) = -1 + \frac{-0.5 + 1}{22712.34615 - (-11837.64486)}(8 - (-11837.64486)) = -0.82857239$$

The resulting fit is displayed below:

x	y
0	5
2	4.151601
4	4.461229
6	5.456047
8	6.852243
10	8.471474
12	10.17813

14	11.80277
16	12.97942
18	12.69896
20	8



24.4 Centered finite differences can be substituted for the second and first derivatives to give,

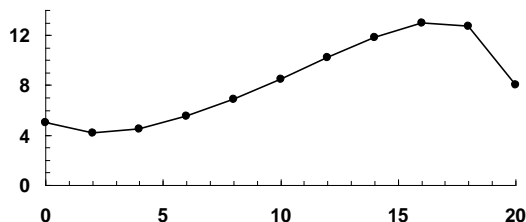
$$7 \frac{y_{i+1} - 2y_i + y_{i-1}}{\Delta x^2} - 2 \frac{y_{i+1} - y_{i-1}}{2\Delta x} - y_i + x_i = 0$$

or substituting $\Delta x = 2$ and collecting terms yields

$$-2.25y_{i-1} + 4.5y_i - 1.25y_{i+1} = x_i$$

This equation can be written for each node and solved with methods such as the Tridiagonal solver, the Gauss-Seidel method or LU Decomposition. The following solution can be computed:

x	y
0	5
2	4.199592
4	4.518531
6	5.507445
8	6.893447
10	8.503007
12	10.20262
14	11.82402
16	13.00176
18	12.7231
20	8



24.5 The linearized term can be substituted into Eq. (P24.5) to give

$$0 = \frac{d^2T}{dx^2} + 0.05(200 - T) + 2.7 \times 10^{-9}(200)^4 - 2.7 \times 10^{-9}(300)^4 - 4(2.7 \times 10^{-9})(300)^3(T - 300)$$

Collecting terms and substituting the parameter values gives

$$0 = \frac{d^2T}{dx^2} + 79.93 - 0.3416T$$

A centered-difference can be substituted for the derivative to give

$$0 = \frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} + 79.93 - 0.3416T_i$$

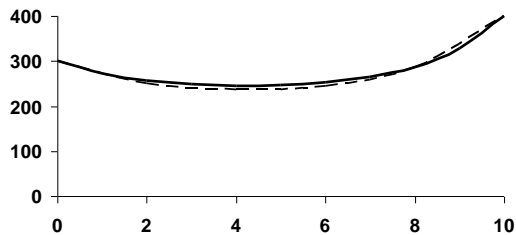
Collecting terms yields,

$$-T_{i+1} + 2.3416T_i - T_{i-1} = 79.93$$

This equation can be applied to each interior node and the resulting tridiagonal system solved for

x	T
0	300
1	271.7125
2	256.3119
3	248.5374
4	245.7334
5	246.9419
6	252.5757
7	264.5593
8	286.9864
9	327.5181
10	400

The comparison with the solution of the nonlinear version (dashed) is depicted in this plot:



24.6

```
function [x,T]=bvshoot(func,tspan,bc,tout,varargin)
% bvshoot: shooting method boundary value ODEs
% [x,T]=bvshoot(func,tspan,bc,tout,p1,p2,...):
%     uses the shooting method to solve a linear 2nd-order ODE
% input:
% func = name of the M-file that evaluates the ODEs
% tspan = [ti , tf]; initial and final times
% bc = boundary values of Dirichlet conditions
% tout = desired times for output
% p1,p2,... = additional parameters used by func
% output:
% x = vector of independent variable
% T = vector of solution for dependent variables

if nargin<3,error('at least 3 input arguments required'),end
if nargin<4||isempty(tout),tout=tspan;end
%first shot
za1=1;Ta=bc(:,1);Tb=bc(:,2);
[x1,T1]=ode45(func,tspan,[Ta za1]);
Tb1=T1(length(T1));
%second shot
za2=za1*1.1;
[x2,T2]=ode45(func,tspan,[Ta za2]);
Tb2=T2(length(T2));
%final shot
za=za1+(za2-za1)/(Tb2-Tb1)*(Tb-Tb1);
[x,T]=ode45(func,tout,[Ta za]);
plot(x,T(:,1))
disp('results:')
fprintf('Final shot: za = %8.4g    T = %8.4g\n',za,T(length(T)))
fprintf('\n      x          T          dT/dx\n')
disp([x T])
```

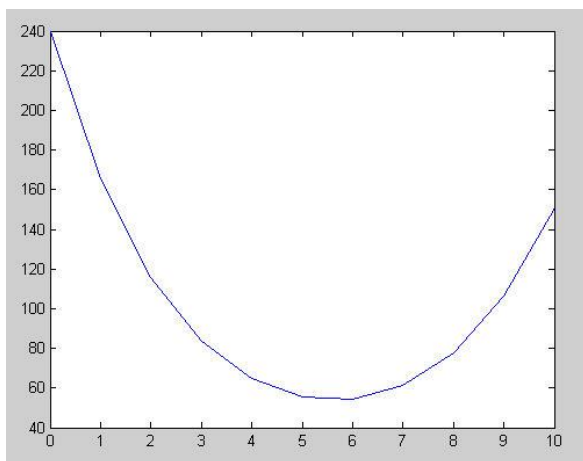
M-file to hold ODEs to test for Prob. 24.1:

```
function dy=prob2401sys(y)
dy=[y(2);0.15*y(1)];
```

Test run:

```
>> [x,T]=bvshoot(@prob2401sys,[0 10],[240 150],[0:1:10]);
results:
Final shot: za =      -90.6      T =      150.8
```

x	T	dT/dx
0	240.0000	-90.6016
1.0000	165.3413	-60.5748
2.0000	115.7973	-39.7492
3.0000	83.8409	-24.9608
4.0000	64.6186	-13.9634
5.0000	55.2108	-5.0869
6.0000	54.1886	3.0169
7.0000	61.3968	11.5791
8.0000	77.9302	21.8999
9.0000	106.3000	35.5470
10.0000	150.8148	54.5931



24.7 A general formulation that describes Example 24.5 as well as Probs. 24.1 and 24.3 is

$$a \frac{d^2 y}{dx^2} + b \frac{dy}{dx} + cy + f(x) = 0$$

Finite difference approximations can be substituted for the derivatives:

$$a \frac{y_{i+1} - 2y_i + y_{i-1}}{\Delta x^2} + b \frac{y_{i+1} - y_{i-1}}{2\Delta x} + cy_i + f(x_i) = 0$$

Collecting terms

$$-(a - 0.5b\Delta x)y_{i-1} + (2a - c\Delta x^2)y_i - (a + 0.5b\Delta x)y_{i+1} = f(x_i)\Delta x^2$$

PROPRIETARY MATERIAL. © The McGraw-Hill Companies, Inc. All rights reserved. No part of this Manual may be displayed, reproduced or distributed in any form or by any means, without the prior written permission of the publisher, or used beyond the limited distribution to teachers and educators permitted by McGraw-Hill for their individual course preparation. If you are a student using this Manual, you are using it without permission.

Dividing by Δx^2 ,

$$-(a/\Delta x^2 - 0.5b/\Delta x)y_{i-1} + (2a/\Delta x^2 - c)y_i - (a/\Delta x^2 + 0.5b/\Delta x)y_{i+1} = f(x_i)$$

The following M-file is set up to solve this general system:

```
function [x,y] = bvFD(xspan,n,bc,a,b,c,fx)
% bvFD: finite-difference method for boundary value ODEs
% [x,y] = bvFD(xspan,n,bc,a,b,c,fx):
%     uses the finite-difference method to solve a linear 2nd-
%     order ODE
% input:
%     func = name of the M-file that evaluates the ODEs
%     xspan = [xi xf]; initial and final values of independent
%     variable
%     n = number of segments
%     bc = boundary values of Dirichlet conditions
%     a,b,c,fx = parameters
% output:
%     x = vector of independent variable
%     y = vector of solution for dependent variable
m=n-1;
dx=(max(xspan)-min(xspan))/n;
x=[min(xspan):dx:max(xspan)];
%a=1;b=0;c=-0.05;Tinf=200;Ta=300;Tb=400;
e=(-(a/dx^2-b/(2*dx)))*ones(m,1);
g=-(a/dx^2+b/(2*dx))*ones(m,1);
f=(2*a/dx^2-c)*ones(m,1);
r=-fx*ones(m,1);
r(1)=r(1)-e(1)*bc(1);
r(m)=r(m)-g(m)*bc(2);
y = Tridiag(e,f,g,r);
y=[bc(1) y bc(2)];
```

For Example 24.5, $a = 1$, $b = 0$, $c = -h' = -0.05$ and $f(x) = h'T_a = -0.05(200)$. These parameters can be passed to the M-file to yield the solution:

```
>> [x,T]=bvFD([0 10],5,[300 400],1,0,-0.05,-0.05*200)

x =
    0         2         4         6         8        10
T =
 300.0000  283.2660  283.1853  299.7416  336.2462  400.0000
```

24.8 First, the 2nd-order ODE can be re-expressed as the following system of 1st-order ODE's

$$\frac{dT}{dx} = z \qquad \frac{dz}{dx} = -25$$

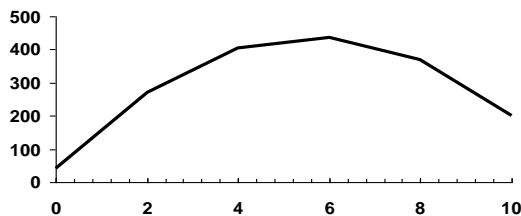
(a) Shooting method: These can be solved for two guesses for the initial condition of z . For our cases we used -1 and -0.5 . We solved the ODEs with the 4th-order RK method using a step size of 0.125. The results are

$z(0)$	-1	-0.5
$T(10)$	-1220	-1215

These values can then be used to derive the correct initial condition,

$$z(0) = -1 + \frac{-0.5 + 1}{-1215 - (-1220)}(200 - (-1220)) = 141$$

The resulting fit is displayed below:



(b) Finite difference: Centered finite differences can be substituted for the second and first derivatives to give,

$$\frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} + 25 = 0$$

or substituting $\Delta x = 2$ and collecting terms yields

$$-T_{i+1} + 2T_i - T_{i-1} = 100$$

This equation can be written for each node with the result

$$\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} 140 \\ 100 \\ 100 \\ 300 \end{bmatrix}$$

These equations can be solved with MATLAB:

```
>> A=[2 -1 0 0;-1 2 -1 0;0 -1 2 -1;0 0 -1 2];
>> b=[140 100 100 300]';
>> T=A\b
T =
    272.0000
    404.0000
    436.0000
    368.0000
```

24.9 First, the 2nd-order ODE can be re-expressed as the following system of 1st-order ODE's

$$\begin{aligned}\frac{dT}{dx} &= z \\ \frac{dz}{dx} &= -(0.12x^3 - 2.4x^2 + 12x)\end{aligned}$$

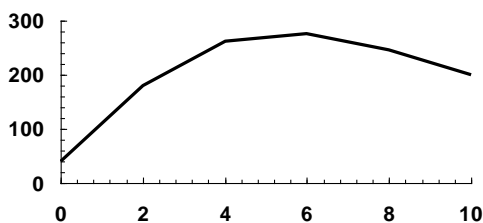
(a) Shooting method: These can be solved for two guesses for the initial condition of z . For our cases we used -1 and -0.5 . We solved the ODEs with the 4th-order RK method using a step size of 0.125. The results are

$z(0)$	-1	-0.5
$T(10)$	-570	-565

These values can then be used to derive the correct initial condition,

$$z(0) = -1 + \frac{-0.5 + 1}{-565 - (-570)}(200 - (-570)) = 76$$

The resulting fit is displayed below:



(b) Finite difference: Centered finite differences can be substituted for the second and first derivatives to give,

$$\frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} + 0.12x_i^3 - 2.4x_i^2 + 12x_i = 0$$

or substituting $\Delta x = 2$ and collecting terms yields

$$-T_{i+1} + 2T_i - T_{i-1} = \Delta x^2 (0.12x_i^3 - 2.4x_i^2 + 12x_i)$$

This equation can be written for each node with the result

$$\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} 101.44 \\ 69.12 \\ 46.08 \\ 215.36 \end{bmatrix}$$

These equations can be solved with MATLAB:

```
>> A=[2 -1 0 0
-1 2 -1 0
0 -1 2 -1
0 0 -1 2];
>> b=[101.44 69.12 46.08 215.36]';
>> x=A\b
```

```
x =
184.1280
266.8160
280.3840
247.8720
```

24.10 This problem can be solved with MATLAB:

```
% ODE Boundary Value Problem
% Tapered conical cooling fin
% u[xx]+(2/x)(u[x]-pu)=0
% BC. u(x=0)=0 u(x=1)=1
% i=spatial index, from 1 to R
% numbering for points is i=1 to i=R for (R-1) dx spaces
% u(i=1)=0 and u(i=R)=1

R=21;
%Constants
dx=1/(R-1);
dx2=dx*dx;
%Parameters
p(1)=10; p(2)=20; p(3)=50; p(4)=100;
%sizing matrices
u=zeros(1,R); x=zeros(1,R);
a=zeros(1,R); b=zeros(1,R); c=zeros(1,R); d=zeros(1,R);
ba=zeros(1,R); ga=zeros(1,R);
%Independent Variable
x=0:dx:1;
%Boundary Conditions
```

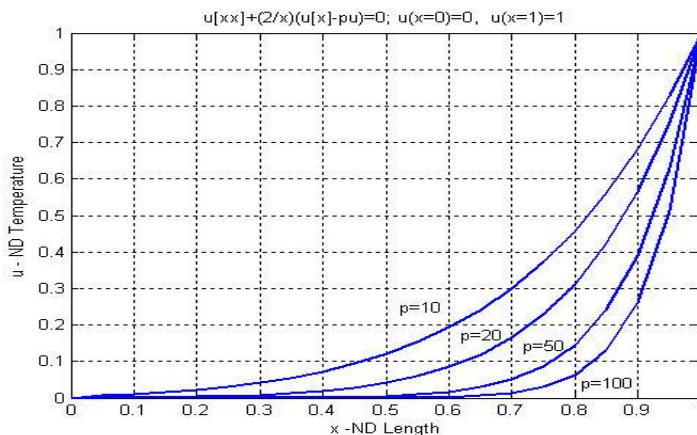


```

u(1)=0; u(R)=1;

for k=1:4;
    %Coefficients
    b(2)=-2-2*p(k)*dx2/dx;
    c(2)=2;
    for i=3:R-2,
        a(i)=1-dx/(dx*(i-1));
        b(i)=-2-2*p(k)*dx2/(dx*(i-1));
        c(i)=1+1/(i-1);
    end
    a(R-1)=1-dx/(dx*(R-2));
    b(R-1)=-2-2*p(k)*dx2/(dx*(R-2));
    d(R-1)=-(1+1/(R-2));
    %Solution by Thomas Algorithm
    ba(2)=b(2);
    ga(2)=d(2)/b(2);
    for i=3:R-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(R-1)=ga(R-1);
    for i=R-2:-1:2,
        u(i)=ga(i)-c(i)*u(i+1)/ba(i);
    end
    %Plot
    plot(x,u)
    title('u[xx]+(2/x)(u[x]-pu)=0; u(x=0)=0, u(x=1)=1')
    xlabel('x -ND Length')
    ylabel('u - ND Temperature')
    hold on
end
grid
hold off
gtext('p=10');gtext('p=20');gtext('p=50');gtext('p=100');

```



24.11 Several methods could be used to obtain a solution for this problem (e.g., finite-difference, shooting method, finite-element). The finite-difference approach is straightforward:

PROPRIETARY MATERIAL. © The McGraw-Hill Companies, Inc. All rights reserved. No part of this Manual may be displayed, reproduced or distributed in any form or by any means, without the prior written permission of the publisher, or used beyond the limited distribution to teachers and educators permitted by McGraw-Hill for their individual course preparation. If you are a student using this Manual, you are using it without permission.

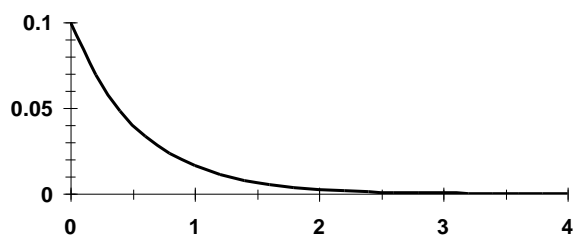
$$D \frac{A_{i-1} - 2A_i + A_{i+1}}{\Delta x^2} - kA_i = 0$$

Substituting parameter values and collecting terms gives

$$-1.5 \times 10^{-6} A_{i-1} + (3 \times 10^{-6} + 5 \times 10^{-6} \Delta x^2) A_i - 1.5 \times 10^{-6} A_{i+1} = 0$$

Using a $\Delta x = 0.2$ cm this equation can be written for all the interior nodes. The resulting linear system can be solved with an approach like the Gauss-Seidel method. The following table and graph summarize the results.

x	A	x	A	x	A	x	A
0	0.1						
0.2	0.069544	1.2	0.011267	2.2	0.001779	3.2	0.000257
0.4	0.048359	1.4	0.007814	2.4	0.001224	3.4	0.000166
0.6	0.033621	1.6	0.005415	2.6	0.000840	3.6	9.93E-05
0.8	0.023368	1.8	0.003748	2.8	0.000574	3.8	4.65E-05
1	0.016235	2	0.002591	3	0.000389	4	0



24.12 Substitute centered difference for the derivatives,

$$D \frac{c_{i+1} - 2c_i + c_{i-1}}{\Delta x^2} - U \frac{c_{i+1} - c_{i-1}}{2\Delta x} - kc_i = 0$$

Collecting terms gives

$$-\left(\frac{D}{\Delta x^2} + \frac{U}{2\Delta x}\right)c_{i-1} + \left(\frac{2D}{\Delta x^2} + k\right)c_i - \left(\frac{D}{\Delta x^2} - \frac{U}{2\Delta x}\right)c_{i+1} = 0$$

Substituting the parameter values yields

$$-55c_{i-1} + 102c_i - 45c_{i+1} = 0$$

For the inlet node ($i = 1$), we must use a finite difference approximation for the first derivative. We use a second-order version (recall Table 21.3) so that the accuracy is comparable to the centered differences we employ for the interior nodes,

$$U_{c_{\text{in}}} = U_{c_1} - D \frac{-c_3 + 4c_2 - 3c_1}{2\Delta x}$$

which can be solved for

$$\left(\frac{3D}{2\Delta x^2} + \frac{U}{\Delta x} \right) c_1 - \left(\frac{2D}{\Delta x^2} \right) c_2 + \left(\frac{D}{2\Delta x^2} \right) c_3 = \frac{U}{\Delta x} c_{\text{in}}$$

Substituting the parameters gives

$$85c_1 - 100c_2 + 25c_3 = 1000$$

For the outlet node ($i = 10$), the zero derivative condition implies that $c_{11} = c_9$,

$$-\left(\frac{2D}{\Delta x^2} \right) c_9 + \left(\frac{2D}{\Delta x^2} + k \right) c_{10} = 0$$

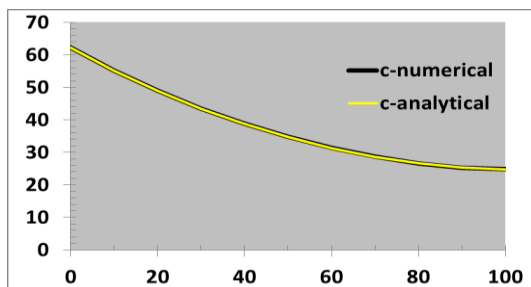
Substituting the parameters gives

$$-100c_9 + 102c_{10} = 0$$

The tridiagonal system can be solved. Here are the results together with the analytical solution. As can be seen, the results are quite close.

x (m)	c -numerical	c -analytical
0	62.3030	62.1767
10	55.2012	55.0818
20	48.9746	48.8634
30	43.5410	43.4390
40	38.8351	38.7437
50	34.8094	34.7300
60	31.4362	31.3708
70	28.7106	28.6615
80	26.6553	26.6258
90	25.3280	25.3223
100	24.8313	24.8552

This is also evident when the results are plotted and the two solutions are visually indistinguishable.



24.13 Centered differences can be substituted for the derivatives to give

$$\begin{aligned}
 D \frac{c_{a,i+1} - 2c_{a,i} + c_{a,i-1}}{\Delta x^2} - U \frac{c_{a,i+1} - c_{a,i-1}}{2\Delta x} - k_1 c_{a,i} &= 0 \\
 D \frac{c_{b,i+1} - 2c_{b,i} + c_{b,i-1}}{\Delta x^2} - U \frac{c_{b,i+1} - c_{b,i-1}}{2\Delta x} + k_1 c_{a,i} - k_2 c_{b,i} &= 0 \\
 D \frac{c_{c,i+1} - 2c_{c,i} + c_{c,i-1}}{\Delta x^2} - U \frac{c_{c,i+1} - c_{c,i-1}}{2\Delta x} + k_2 c_{b,i} &= 0
 \end{aligned}$$

Collecting terms gives

$$\begin{aligned}
 -50c_{a,i-1} + 83c_{a,i} - 30c_{a,i+1} &= 0 \\
 -50c_{b,i-1} + 81c_{b,i} - 30c_{b,i+1} &= 3c_{a,i} \\
 -50c_{c,i-1} + 80c_{c,i} - 30c_{c,i+1} &= c_{b,i}
 \end{aligned}$$

For the inlet node ($i = 1$), we must use a finite difference approximation for the first derivative. We use a second-order version (recall Table 21.3) so that the accuracy is comparable to the centered differences we employ for the interior nodes. For example, for reactant A,

$$Uc_{a,\text{in}} = Uc_{a,1} - D \frac{-c_{a,3} + 4c_{a,2} - 3c_{a,1}}{2\Delta x}$$

which can be solved for

$$\left(\frac{3D}{2\Delta x^2} + \frac{U}{\Delta x} \right) c_{a,1} - \left(\frac{2D}{\Delta x^2} \right) c_{a,2} + \left(\frac{D}{2\Delta x^2} \right) c_{a,3} = \frac{U}{\Delta x} c_{a,\text{in}}$$

Because the condition does not include reaction rates, similar equations can be written for the other nodes. Substituting the parameters gives

PROPRIETARY MATERIAL. © The McGraw-Hill Companies, Inc. All rights reserved. No part of this Manual may be displayed, reproduced or distributed in any form or by any means, without the prior written permission of the publisher, or used beyond the limited distribution to teachers and educators permitted by McGraw-Hill for their individual course preparation. If you are a student using this Manual, you are using it without permission.

$$80c_{a,1} - 80c_{a,2} + 20c_{a,3} = 200$$

$$80c_{b,1} - 80c_{b,2} + 20c_{b,3} = 0$$

$$80c_{c,1} - 80c_{c,2} + 20c_{c,3} = 0$$

For the outlet node ($i = 10$), the zero derivative condition implies that $c_{i1} = c_9$,

$$-\left(\frac{2D}{\Delta x^2}\right)c_9 + \left(\frac{2D}{\Delta x^2} + k_1\right)c_{10} = 0$$

Again, because the condition does not include reaction rates, similar equations can be written for the other nodes. Substituting the parameters gives

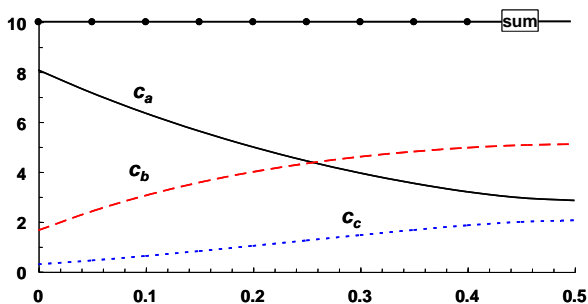
$$-80c_{a,9} + 83c_{a,10} = 0$$

$$-80c_{b,9} + 81c_{b,10} = 3c_{a,10}$$

$$-80c_{c,9} + 80c_{c,10} = c_{b,10}$$

Notice that because the reactions are in series, we can solve the systems for each reactant separately in sequence. The result is

x	c_a	c_b	c_c	sum
0	8.0646	1.6479	0.2875	10
0.05	7.1492	2.4078	0.4430	10
0.1	6.3385	3.0397	0.6218	10
0.15	5.6212	3.5603	0.8184	10
0.2	4.9878	3.9846	1.0276	10
0.25	4.4309	4.3258	1.2433	10
0.3	3.9459	4.5955	1.4587	10
0.35	3.5320	4.8035	1.6644	10
0.4	3.1955	4.9572	1.8473	10
0.45	2.9542	5.0591	1.9867	10
0.5	2.8474	5.1021	2.0505	10



24.14 Centered differences can be substituted for the derivatives to give

$$D \frac{c_{a,i+1} - 2c_{a,i} + c_{a,i-1}}{\Delta x^2} = 0 \quad 0 \leq x < L$$

$$D_f \frac{c_{a,i+1} - 2c_{a,i} + c_{a,i-1}}{\Delta x^2} - kc_{a,i} = 0 \quad L \leq x < L + L_f$$

Collecting terms

$$-\frac{D}{\Delta x^2} c_{a,i-1} + \frac{2D}{\Delta x^2} c_{a,i} - \frac{D}{\Delta x^2} c_{a,i+1} = 0 \quad 0 \leq x < L$$

$$-\frac{D_f}{\Delta x^2} c_{a,i-1} + \left(\frac{2D_f}{\Delta x^2} + k \right) c_{a,i} - \frac{D_f}{\Delta x^2} c_{a,i+1} = 0 \quad L \leq x < L + L_f$$

The boundary conditions can be developed. For the first node ($i = 1$),

$$\frac{2D}{\Delta x^2} c_{a,1} - \frac{D}{\Delta x^2} c_{a,2} = \frac{D}{\Delta x^2} c_{a,0}$$

For the last,

$$-\frac{2D_f}{\Delta x^2} c_{a,n-1} + \left(\frac{2D_f}{\Delta x^2} + k \right) c_{a,n} = 0$$

A special equation is also required at the interface between the diffusion layer and the biofilm ($x = L$). A flux balance can be written around this node as

$$D \frac{c_{a,i-1} - c_{a,i}}{\Delta x} + D_f \frac{c_{a,i+1} - c_{a,i}}{\Delta x} - k \frac{\Delta x}{2} c_{a,i} = 0$$

Collecting terms gives

$$-\frac{D}{\Delta x} c_{a,i-1} + \left(\frac{D + D_f}{\Delta x} + k \frac{\Delta x}{2} \right) c_{a,i} - \frac{D_f}{\Delta x} c_{a,i+1} = 0$$

Substituting the parameters gives

$$\text{first node:} \quad 160,000c_{a,1} - 80,000c_{a,2} = 8,000,000$$

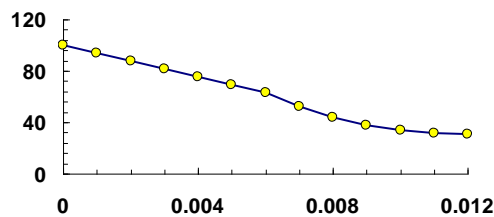
$$\text{interior nodes (diffusion layer):} \quad -80,000c_{a,i-1} + 160,000c_{a,i} - 80,000c_{a,i+1} = 0$$

PROPRIETARY MATERIAL. © The McGraw-Hill Companies, Inc. All rights reserved. No part of this Manual may be displayed, reproduced or distributed in any form or by any means, without the prior written permission of the publisher, or used beyond the limited distribution to teachers and educators permitted by McGraw-Hill for their individual course preparation. If you are a student using this Manual, you are using it without permission.

$$\begin{aligned}
\text{boundary node } (i = 6): & \quad -80,000c_{a,5} + 121,000c_{a,6} - 40,000c_{a,7} = 0 \\
\text{interior nodes (biofilm):} & \quad -40,000c_{a,i-1} + 82,000c_{a,i} - 40,000c_{a,i+1} = 0 \\
\text{last node:} & \quad -80,000c_{a,n-1} + 82,000c_{a,n} = 0
\end{aligned}$$

The solution is

x	c_a
0	100.0000
0.001	93.8274
0.002	87.6549
0.003	81.4823
0.004	75.3097
0.005	69.1372
0.006	62.9646
0.007	52.1936
0.008	44.0322
0.009	38.0725
0.01	34.0164
0.011	31.6611
0.012	30.8889



24.15 Main Program:

```

% Hanging static cable - w=w(x)
% Parabolic solution w=w(x)
% CUS Units (kg,m,s)
% w = wo(1+sin(pi/2*x/l))
% Independent Variable x
% initial conditions [y(1)=cable y-coordinate, y(2)=cable
slope];
clear,clc,format compact
global wToP
es=0.5e-7; w0=450;
xspan=[0,60]; ic=[0 0];
wToP=0.0025;
[x,y]=ode45(@slp,xspan,ic);
yf(1)=y(length(x));
wTo(1)=wToP;
ea(1)=1;

```

```

wToP=0.002;
[x,y]=ode45(@slp,xspan,ic);
yf(2)=y(length(x));
wTo(2)=wToP;
ea(2)=abs((yf(2)-yf(1))/yf(2));
for k=3:10
    wTo(k)=wTo(k-1)+(wTo(k-1)-wTo(k-2))/(yf(k-1)-yf(k-2))*(15-
yf(k-1));
    wToP=wTo(k);
    [x,y]=ode45(@slp,xspan,ic);
    yf(k)=y(length(x));
    ea(k)=abs((yf(k)-yf(k-1))/yf(k)));
    if (ea(k)<=es)
        plot(x,y(:,1)); grid;
        xlabel('x-coordinate - m'); ylabel('y-coordinate - m');
        title('Cable - w=wo(1+sin(pi/2*x/l))');
        break
    end
end
wToP
To=w0/wToP

```

Function 'slp':

```

function dxy=slp(x,y)
global wToP
dxy=[y(2);(wToP)*(1+sin(pi/2*x/l))];

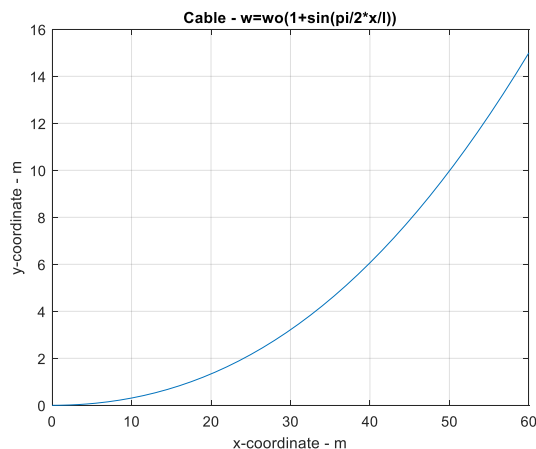
```

Results:

```

wToP =
    0.0057
To =
    7.8984e+04

```



24.16 This is a boundary-value problem because the values of only one of the variables are known at two separate points; i.e., at $x = 0, y = 0$ and at $x = L, y = 0$. We can either use finite differences or the shooting method to obtain results.

PROPRIETARY MATERIAL. © The McGraw-Hill Companies, Inc. All rights reserved. No part of this Manual may be displayed, reproduced or distributed in any form or by any means, without the prior written permission of the publisher, or used beyond the limited distribution to teachers and educators permitted by McGraw-Hill for their individual course preparation. If you are a student using this Manual, you are using it without permission.

For example, for the finite-difference approach, we can substitute centered differences for the second derivative gives

$$EI \frac{y_{i-1} - 2y_i + y_{i+1}}{(\Delta x)^2} = \frac{wLx_i}{2} - \frac{wx_i^2}{2}$$

which can be expressed as

$$y_{i-1} - 2y_i + y_{i+1} = \frac{(\Delta x)^2}{EI} \left(\frac{wLx_i}{2} - \frac{wx_i^2}{2} \right)$$

Substituting the parameter values with $\Delta x = 0.6$ m yields

$$y_{i-1} - 2y_i + y_{i+1} = \frac{0.6^2}{2 \times 10^{11} (3 \times 10^{-4})} \left(\frac{15,000(3)}{2} x_i - \frac{15,000}{2} x_i^2 \right)$$

or

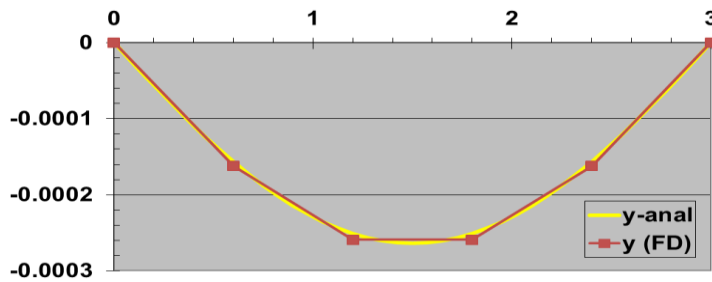
$$y_{i-1} - 2y_i + y_{i+1} = 1.35 \times 10^{-4} x_i - 4.5 \times 10^{-5} x_i^2$$

Writing this equation for the four internal nodes gives

$$\begin{bmatrix} -2 & 1 & & \\ 1 & -2 & 1 & \\ & 1 & -2 & 1 \\ & & 1 & -2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 6.48 \times 10^{-5} \\ 9.72 \times 10^{-5} \\ 9.72 \times 10^{-5} \\ 6.48 \times 10^{-5} \end{bmatrix}$$

These equations can be solved for the displacements. The results together with the analytical solution are tabulated and graphed as

x	y (FD)	y (analytical)
0	0	0
0.6	-0.000162	-0.0001566
1.2	-0.0002592	-0.0002511
1.8	-0.0002592	-0.0002511
2.4	-0.000162	-0.0001566
3	0	0



Thus, the results are pretty close. If a finer grid is used the results would be improved.

(b) Shooting method

```
clear,clc,clf,format compact
E = 2e11;I =3e-4;w= 15000;L = 3;
xspan=[0:L/5:L];bc=[0 0];
[x,y]=bvshoot(@prob2416sys,xspan,bc,E,I,w,L);

function [x,y]=bvshoot(func,tspan,bc,varargin)
% bvshoot: shooting method boundary value ODEs
% [x,y]=bvshoot(func,tspan,bc,tout,p1,p2,...):
%     uses the shooting method to solve a linear 2nd-order ODE
% input:
% func = name of the M-file that evaluates the ODEs
% tspan = [ti , tf]; initial and final times
% bc = boundary values of Dirichlet conditions
% tout = desired times for output
% p1,p2,... = additional parameters used by func
% output:
% x = vector of independent variable
% y = vector of solution for dependent variables

%Shooting method
%first shot
za1=1;ya=bc(:,1);yb=bc(:,2);
[x1,y1]=ode45(func,tspan,[ya za1],[],varargin{:});
yb1=y1(length(y1));
%second shot
za2=za1*1.1;
[x2,y2]=ode45(func,tspan,[ya za2],[],varargin{:});
yb2=y2(length(y2));
%final shot
za=za1+(za2-za1)/(yb2-yb1)*(yb-yb1);
[x,y]=ode45(func,tspan,[ya za],[],varargin{:});
% analytical solution
E = 2e11;I =3e-4;w= 15000;L = 3;
tanal=[0:L/100:L];
yanal=w*L*tanal.^3./(12*E*I)-w*tanal.^4/(24*E*I)-
w*L^3*tanal/(24*E*I);
% plot results
subplot(2,1,1)
plot(x,y(:,1),tanal,yanal)
```

```

grid;legend('shooting','analytical')
ylabel('y (m)')
subplot(2,1,2)
plot(x,y(:,2))
xlabel('x (m)'),ylabel('dy/dx (m/m)')
% display results
disp('results:')
fprintf('Final shot: za = %8.4g    y = %8.4g\n',za,y(length(y)))
fprintf('\n          x          y          dy/dx\n')
disp([x y])

function dy=prob2416sys(x,y,E,I,w,L)
dy=[y(2);(w*L*x/2-w*x^2/2)/(E*I)];

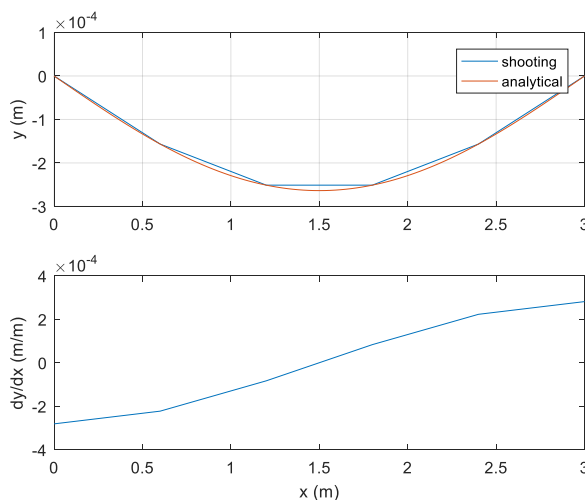
```

Results:

results:

Final shot: za = -0.0002812 y = 3.965e-15

x	y	dy/dx
0	0	-0.00028125
0.6	-0.0001566	-0.00022275
1.2	-0.0002511	-8.325e-05
1.8	-0.0002511	8.325e-05
2.4	-0.0001566	0.00022275
3	3.9652e-15	0.00028125



24.17 We can substitute centered differences for the fourth derivative to give

$$EI \frac{y_{i+2} - 4y_{i+1} + 6y_i - 4y_{i-1} + y_{i-2}}{\Delta x^4} = -w$$

Substituting the parameters and collecting terms gives

PROPRIETARY MATERIAL. © The McGraw-Hill Companies, Inc. All rights reserved. No part of this Manual may be displayed, reproduced or distributed in any form or by any means, without the prior written permission of the publisher, or used beyond the limited distribution to teachers and educators permitted by McGraw-Hill for their individual course preparation. If you are a student using this Manual, you are using it without permission.

$$y_{i-2} - 4y_{i-1} + 6y_i - 4y_{i+1} + y_{i+2} = -3.24 \times 10^{-5}$$

Writing the finite-difference equation for the four internal nodes gives the following simultaneous equations

$$y_{-1} - 4y_0 + 6y_1 - 4y_2 + y_3 = -3.24 \times 10^{-5}$$

$$y_0 - 4y_1 + 6y_2 - 4y_3 + y_4 = -3.24 \times 10^{-5}$$

$$y_1 - 4y_2 + 6y_3 - 4y_4 + y_5 = -3.24 \times 10^{-5}$$

$$y_2 - 4y_3 + 6y_4 - 4y_5 + y_6 = -3.24 \times 10^{-5}$$

The boundary conditions can be used to reduce the number of unknowns. The zero end conditions mean that $y_0 = y_5 = 0$. The zero second derivative at node 1 can be represented in finite-difference form as

$$\frac{y_{-1} - 2y_0 + y_1}{(\Delta x)^2} = 0$$

Because $y_0 = 0$, this condition implies that $y_{-1} = -y_1$. Similarly, the zero second derivative at node 5 yields $y_6 = -y_4$. Substituting these results into the system of equations gives,

$$\begin{aligned} 5y_1 - 4y_2 + y_3 &= -3.24 \times 10^{-5} \\ -4y_1 + 6y_2 - 4y_3 + y_4 &= -3.24 \times 10^{-5} \\ y_1 - 4y_2 + 6y_3 - 4y_4 &= -3.24 \times 10^{-5} \\ y_2 - 4y_3 + 5y_4 &= -3.24 \times 10^{-5} \end{aligned}$$

The equations can be solved for the displacements.

x	y (FD)	y (anal)
0	0	0
0.6	-0.000162	-0.000157
1.2	-0.000259	-0.000251
1.8	-0.000259	-0.000251
2.4	-0.000162	-0.000157
3	0	0

Here is a MATLAB script to generate the solution:

```
clear,clc,clf
E=200e9; I=30000/100^4; w=15e3; L=3;
n=5; dx=L/n;
b=ones(n-1,1)*-w*dx^4/E/I;
```

PROPRIETARY MATERIAL. © The McGraw-Hill Companies, Inc. All rights reserved. No part of this Manual may be displayed, reproduced or distributed in any form or by any means, without the prior written permission of the publisher, or used beyond the limited distribution to teachers and educators permitted by McGraw-Hill for their individual course preparation. If you are a student using this Manual, you are using it without permission.

```

xw=dx:dx:L-dx;
a(1,1)=5;a(1,2)=-4;a(1,3)=1;
a(2,1)=-4;a(2,2)=6;a(2,3)=-4;a(2,4)=1;
for i =3:n-3
    a(i,i-2)=1;a(i,i-1)=-4;a(i,i)=6;a(i,i+1)=-4;a(i,i+2)=1;
end
a(n-2,n-4)=1;a(n-2,n-3)=-4;a(n-2,n-2)=6;a(n-2,n-1)=-4;
a(n-1,n-3)=1;a(n-1,n-2)=-4;a(n-1,n-1)=5;
a
y=a\b
x=0:dx:L;
y=[0 y' 0];
plot(x,y,'o-')

```

When this is run, the results are:

```

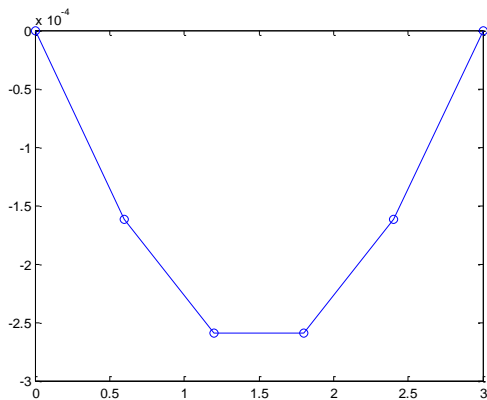
a =
     5     -4      1      0
    -4      6     -4      1
     1     -4      6     -4
     0      1     -4      5

```

```

y =
    1.0e-03 *
    -0.1620
    -0.2592
    -0.2592
    -0.1620

```



24.18 (a) The second-order equation can be re-expressed as a pair of first-order equations:

$$\frac{dh}{dx} = z$$

$$\frac{dz}{dx} = -\frac{N}{Kh} = -\frac{0.0001}{1(7.5)} = -0.0000133333$$

These can be stored in a function

PROPRIETARY MATERIAL. © The McGraw-Hill Companies, Inc. All rights reserved. No part of this Manual may be displayed, reproduced or distributed in any form or by any means, without the prior written permission of the publisher, or used beyond the limited distribution to teachers and educators permitted by McGraw-Hill for their individual course preparation. If you are a student using this Manual, you are using it without permission.

```
function dy=prob2418sys(x,y)
hb=7.5;N=0.0001;K=1;
dy=[y(2);-N/K/hb];
```

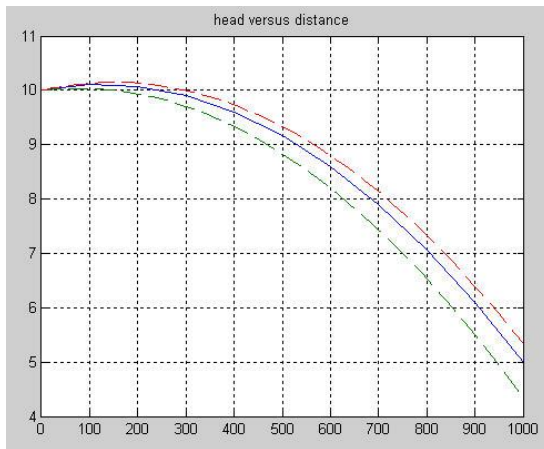
The solution was then generated with the following script. Note that we have generated a plot of all the shots:

```
xi=0;xf=1000;
za1=0.001;za2=0.002;ha=10;hb=5;
[x1,h1]=ode45(@prob2418sys,[xi xf],[ha za1]);
hb1=h1(length(h1));
[x2,h2]=ode45(@prob2418sys,[xi xf],[ha za2]);
hb2=h2(length(h2));
za=za1+(za2-za1)/(hb2-hb1)*(hb-hb1);
[x,h]=ode45(@prob2418sys,[xi:(xf-xi)/10:xf],[ha za]);
plot(x,h(:,1),x1,h1(:,1),'--',x2,h2(:,1),'--')
grid;title('head versus distance')
disp('results:')
fprintf('1st shot:  za1 = %8.4g  hb1 = %8.4g\n',za1,hb1)
fprintf('2nd shot:  za2 = %8.4g  hb2 = %8.4g\n',za2,hb2)
fprintf('Final shot: za = %8.4g    h = %8.4g\n',za,h(length(h)))
z=[x';h(:,1)'];
fprintf('\n          x          h\n');
fprintf('%6d    %10.5f\n',z);
```

The results are

```
results:
1st shot:  za1 =    0.001  hb1 =    4.333
2nd shot:  za2 =    0.002  hb2 =    5.333
Final shot: za = 0.001667    h =    5
```

x	h
0	10.00000
100	10.10000
200	10.06667
300	9.90000
400	9.60000
500	9.16667
600	8.60000
700	7.90000
800	7.06667
900	6.10000
1000	5.00000



(b) We can substitute a centered difference for the second derivative to give

$$K\bar{h} \frac{h_{i-1} - 2h_i + h_{i+1}}{(\Delta x)^2} + N = 0$$

Collecting terms,

$$-h_{i-1} + 2h_i - h_{i+1} = \frac{N(\Delta x)^2}{K\bar{h}}$$

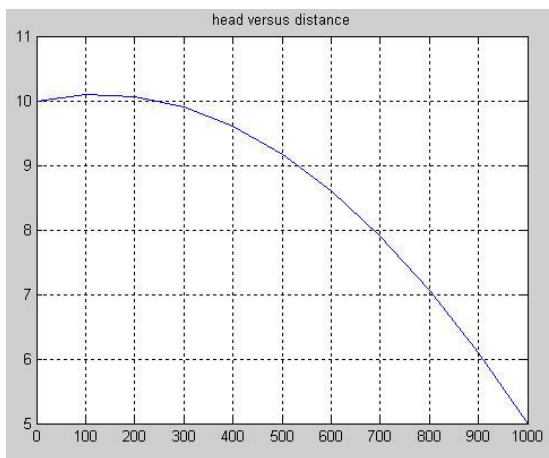
Substituting the parameter values gives a tridiagonal system of linear algebraic equations. These can be solved with MATLAB as in the following script,

```
h0=10; hn=5; hb=(h0+hn)/2; N=0.0001; K=1; dx=100;
L=1000; n=L/dx-1;
rhs=N*dx^2/K/hb;
a=-1; b=2; c=-1;
A = b*diag(ones(n,1)) + c*diag(ones(n-1,1),1) + a*diag(ones(n-1,1),-1);
b = rhs*ones(n,1);
b(1)=b(1)+h0;
b(n)=b(n)+hn;
h=A\b;
h=[h0 h' hn];
x=[0:dx:L];
z=[x;h];
disp('Results:')
fprintf('\n      x      h\n');
fprintf('%6d    %10.5f\n',z);
plot(x,h);grid;title('head versus distance')
```

The output is

Results:

x	h
0	10.00000
100	10.10000
200	10.06667
300	9.90000
400	9.60000
500	9.16667
600	8.60000
700	7.90000
800	7.06667
900	6.10000
1000	5.00000



24.19 (a) The product rule can be used to evaluate the derivative

$$Kh \frac{d^2 h}{dx^2} + K \left(\frac{dh}{dx} \right)^2 + N = 0$$

We can then define a new variable, $z = dh/dx$ and express the second-order ODE as a pair of first-order ODEs,

$$\begin{aligned} \frac{dh}{dx} &= z \\ \frac{dz}{dx} &= -\frac{1}{h} z^2 - \frac{N}{Kh} \end{aligned}$$

These equations can be solved iteratively using an approach similar to Example 24.4. First, an M-file can be developed to compute the right-hand sides of these equations,

```
function dy=dydxGW(x,y)
```

PROPRIETARY MATERIAL. © The McGraw-Hill Companies, Inc. All rights reserved. No part of this Manual may be displayed, reproduced or distributed in any form or by any means, without the prior written permission of the publisher, or used beyond the limited distribution to teachers and educators permitted by McGraw-Hill for their individual course preparation. If you are a student using this Manual, you are using it without permission.


```
dy=[y(2); -1/y(1)*y(2)^2-0.0001/y(1)/1];
```

Next, we can build a function to hold the residual that we will try to drive to zero as

```
function r=resGW(za)
[x,y]=ode45(@dydxGW,[0 1000],[10 za]);
r=y(length(x),1)-5;
```

We can then find the root with the `fzero` function (note that we obtain the initial guess from the linear solution obtained in Prob. 24.18),

```
za=fzero(@resGW, 0.001667)
```

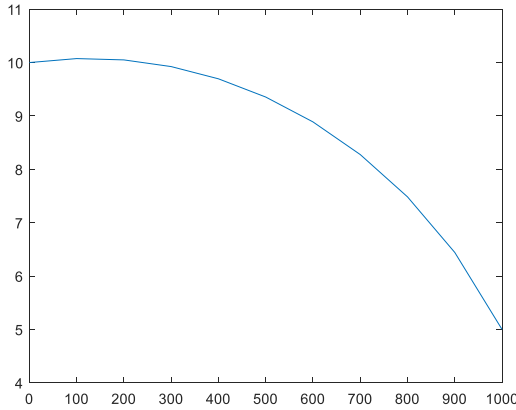
```
za =
    0.00125
```

Thus, we see that if we set the initial trajectory $z(0) = 0.00125$, the residual function will be driven to zero and the head boundary condition, $h(1000) = 5$ should be satisfied. This can be implemented by developing a script to generate the entire solution and plotting head versus distance,

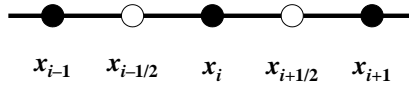
```
clear,clc,format long,format compact
za=fzero(@resGW, 0.001667)
[x,h]=ode45(@dydxGW,[0:100:1000],[10 za]);
z=[x';h(:,1)'];
fprintf('\n      x      h\n');
fprintf('%6d    %10.5f\n',z);
plot(x,h(:,1))
```

```
za =
    0.001250002356082
```

x	h
0	10.00000
100	10.07472
200	10.04988
300	9.92472
400	9.69536
500	9.35414
600	8.88820
700	8.27648
800	7.48332
900	6.44206
1000	5.00000



(b) A nodal scheme for this problem is shown below:



We can substitute a centered difference for the outer first derivative to give

$$\frac{\left(Kh \frac{dh}{dx}\right)_{i+1/2} - \left(Kh \frac{dh}{dx}\right)_{i-1/2}}{\Delta x} + N = 0 \quad (1)$$

We can then apply centered differences to evaluate the remaining derivatives

$$\begin{aligned} \left(Kh \frac{dh}{dx}\right)_{i+1/2} &= Kh_{i+1/2} \frac{h_{i+1} - h_i}{\Delta x} \\ \left(Kh \frac{dh}{dx}\right)_{i-1/2} &= Kh_{i-1/2} \frac{h_i - h_{i-1}}{\Delta x} \end{aligned}$$

which can be substituted into (1) to give

$$K \frac{\frac{h_i + h_{i+1}}{2} \frac{h_{i+1} - h_i}{\Delta x} - \frac{h_i + h_{i-1}}{2} \frac{h_i - h_{i-1}}{\Delta x}}{\Delta x} + N = 0$$

Collecting terms yields

$$(h_i + h_{i+1})(h_{i+1} - h_i) - (h_i + h_{i-1})(h_i - h_{i-1}) = -\frac{2\Delta x^2}{K} N$$

The terms on the left-hand side can be multiplied out to give

$$h_i h_{i+1} - h_{i+1} h_i - h_i h_i + h_{i+1} h_{i+1} - h_i h_i + h_i h_{i-1} - h_{i-1} h_i + h_{i-1} h_{i-1} = -\frac{2\Delta x^2}{K} N$$

Cancelling and collecting terms gives

$$h_{i-1}^2 - 2h_i^2 + h_{i+1}^2 = -\frac{2\Delta x^2}{K} N$$

An iterative solution similar to Gauss-Seidel can then be developed as

$$h_i = \sqrt{\frac{h_{i-1}^2 + h_{i+1}^2}{2} + \frac{\Delta x^2}{K} N}$$

The results are

x	h
0	10
100	10.07472
200	10.04988
300	9.924717
400	9.69536
500	9.354143
600	8.888194
700	8.276473
800	7.483315
900	6.442049
1000	5

Note that these values are quite similar to those obtained with the shooting method in part (a).

24.20 A centered differences can be substituted for the derivative to give

$$\frac{V_{i+1} - 2V_i + V_{i-1}}{\Delta x^2} = -\frac{\rho_v}{\varepsilon}$$

Collecting variables and substituting parameters gives

$$-V_{i+1} + 2V_i - V_{i-1} = 60$$

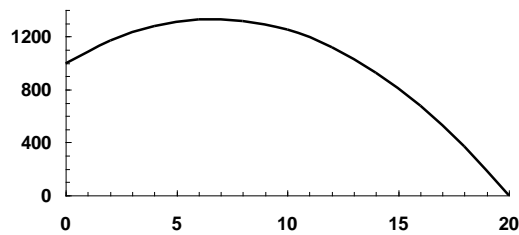
The equations for the first and last node reflect the boundary conditions

$$2V_1 - V_2 = 1060$$

$$-V_8 + 2V_9 = 60$$

The equations can be solved for

x	V
0	1000
2	1170
4	1280
6	1330
8	1320
10	1250
12	1120
14	930
16	680
18	370
20	0



24.21 The second-order equation can be re-expressed as a pair of first-order equations:

$$\frac{dx}{dt} = v$$

$$\frac{dv}{dt} = g - \frac{c}{m}v$$

A function can be developed to hold these equations,

```
function dy=prob2421sys(x,y)
g=9.81;c=12.5;m=70;
dy=[y(2);g-c/m*y(2)];
```

The solution was then generated with the following script. Note that we have generated a plot of all the shots as well as the analytical solution.

PROPRIETARY MATERIAL. © The McGraw-Hill Companies, Inc. All rights reserved. No part of this Manual may be displayed, reproduced or distributed in any form or by any means, without the prior written permission of the publisher, or used beyond the limited distribution to teachers and educators permitted by McGraw-Hill for their individual course preparation. If you are a student using this Manual, you are using it without permission.

```

clear,clc,format short,format compact
ti=0;tf=12;
za1=0;za2=50;xa=0;xb=500;
[t1,x1]=ode45(@prob2421sys,[ti:(tf-ti)/10:tf],[xa za1]);
xb1=x1(length(x1));
[t2,x2]=ode45(@prob2421sys,[ti:(tf-ti)/10:tf],[xa za2]);
xb2=x2(length(x2));
za=za1+(za2-za1)/(xb2-xb1)*(xb-xb1);
[t,x]=ode45(@prob2421sys,[ti:(tf-ti)/10:tf],[xa za]);
plot(t,x(:,1),t1,x1(:,1),'--',t2,x2(:,1),'--')
disp('results:')
fprintf('1st shot:  za1 = %8.4g  xb1 = %8.4g\n',za1,xb1)
fprintf('2nd shot:  za2 = %8.4g  xb2 = %8.4g\n',za2,xb2)
fprintf('Final shot: za = %8.4g    x = %8.4g\n',za,x(length(x)))
fprintf('\n      t      x      dx/dt\n')
disp([t x])

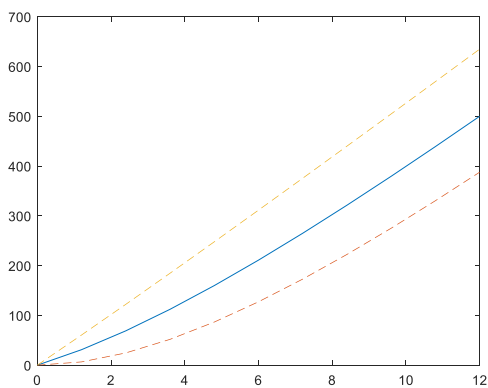
```

```

results:
1st shot:  za1 =          0  xb1 =      387.7
2nd shot:  za2 =          50  xb2 =      634.8
Final shot: za =      22.72    x =          500

```

t	x	dx/dt
0	0	22.7224
1.2000	31.1280	28.9359
2.4000	68.9674	33.9508
3.6000	112.2237	37.9985
4.8000	159.8520	41.2654
6.0000	211.0091	43.9023
7.2000	265.0143	46.0305
8.4000	321.3183	47.7482
9.6000	379.4776	49.1346
10.8000	439.1345	50.2536
12.0000	500.0000	51.1567



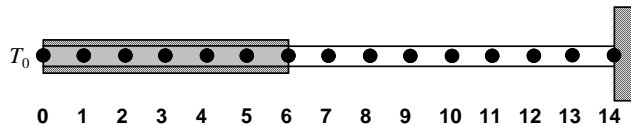
24.22 Heat balances for the two sections can be written as

rod: $\frac{d^2 T}{dx^2} = 0$

PROPRIETARY MATERIAL. © The McGraw-Hill Companies, Inc. All rights reserved. No part of this Manual may be displayed, reproduced or distributed in any form or by any means, without the prior written permission of the publisher, or used beyond the limited distribution to teachers and educators permitted by McGraw-Hill for their individual course preparation. If you are a student using this Manual, you are using it without permission.

tube: $\frac{d^2T}{dx^2} + \frac{2h}{rk_2}(T_\infty - T) = 0$

The nodes can be set up as shown:



Substituting finite differences for the interior nodes gives

Rod nodes $i = 1$ through 5: $\frac{T_{i-1} - 2T_i + T_{i+1}}{\Delta x^2} = 0$

Tube nodes $i = 7$ through 13: $\frac{T_{i-1} - 2T_i + T_{i+1}}{\Delta x^2} + \frac{2h}{rk_2}(T_\infty - T_i) = 0$

At the left end (node 0), $T = T_0$, so the heat balance for node 1 is

$$2T_1 - T_2 = T_0$$

For nodes 2 through 5:

$$-T_1 + 2T_2 - T_3 = 0$$

$$-T_2 + 2T_3 - T_4 = 0$$

$$-T_3 + 2T_4 - T_5 = 0$$

$$-T_4 + 2T_5 - T_6 = 0$$

For nodes 7 through 13

$$-T_{i+1} + \left(2 + \frac{2h\Delta x^2}{rk_2}\right)T_i - T_{i-1} = \frac{2h\Delta x^2}{rk_2}T_\infty$$

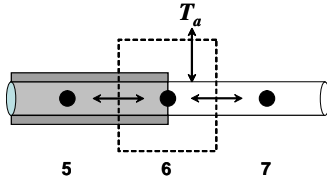
or substituting the parameters,

$$-T_{i+1} + \left(2 + \frac{2 \times 3000 \text{ J/(s m}^2\text{K)}(0.1 \text{ m})^2}{0.03 \text{ m} \times 0.615 \text{ J/(s m K)}}\right)T_i - T_{i-1} = \frac{2 \times 3000 \text{ J/(s m}^2\text{K)}(0.1 \text{ m})^2}{0.03 \text{ m} \times 0.615 \text{ J/(s m K)}}T_\infty$$

which gives

$$-T_{i-1} + 3254T_i - T_{i+1} = 975,610$$

To write a heat balance at the node between the rod and the tube, a heat balance can be written for the element enclosed by the dashed line depicted below:



The steady-state heat balance for this element is

$$0 = -k_1 \frac{T_6 - T_5}{\Delta x} A_c - k_2 \frac{T_6 - T_7}{\Delta x} A_c + h(T_\infty - T_6) A_s$$

where A_c = the cross-sectional area (m^2) = πr^2 , and A_s = the surface area (m^2) of the tube within the element boundary = $\pi \Delta x$. Substituting these areas gives

$$0 = -k_1 \frac{T_6 - T_5}{\Delta x} \pi r^2 - k_2 \frac{T_6 - T_7}{\Delta x} \pi r^2 + h(T_\infty - T_6) \pi r \Delta x$$

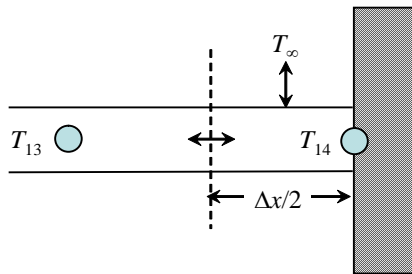
or collecting terms

$$-k_1 T_5 + \left(k_1 + k_2 + \frac{h}{r} \Delta x^2 \right) T_6 - k_2 T_7 = \frac{h}{r} \Delta x^2 T_\infty$$

Substituting the parameters gives

$$-80.2T_5 + 1080.8T_6 - 0.615T_7 = 300,000$$

At the right end ($i = 14$), a heat balance is written as



$$0 = -k_2 \frac{T_{14} - T_{13}}{\Delta x} A_c + h(T_\infty - T_{14})A_s$$

Substituting the areas and collecting terms gives

$$-T_{13} + \left(1 + \frac{h\Delta x^2}{k_2 r}\right) T_{14} = \frac{h\Delta x^2}{k_2 r} T_\infty$$

Substituting parameters yields

$$-T_{13} + 1627T_{14} = 487805$$

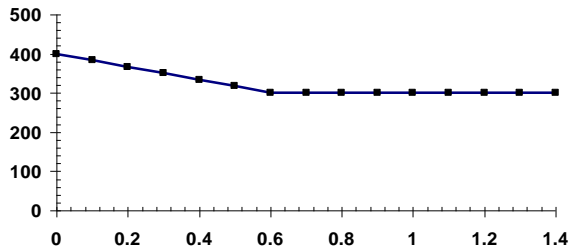
The foregoing equations can be assembled in matrix form as

$$\begin{bmatrix} 2 & -1 & & & & & & & & & & & & \\ -1 & 2 & -1 & & & & & & & & & & & \\ & -1 & 2 & -1 & & & & & & & & & & \\ & & -1 & 2 & -1 & & & & & & & & & \\ & & & -1 & 2 & -1 & & & & & & & & \\ & & & & -1 & 2 & -1 & & & & & & & \\ & & & & & -80.2 & 1080.8 & -0.615 & & & & & & \\ & & & & & & -1 & 3254 & -1 & & & & & \\ & & & & & & & -1 & 3254 & -1 & & & & \\ & & & & & & & & -1 & 3254 & -1 & & & \\ & & & & & & & & & -1 & 3254 & -1 & & \\ & & & & & & & & & & -1 & 3254 & -1 & \\ & & & & & & & & & & & -1 & 3254 & -1 & \\ & & & & & & & & & & & & -1 & 3254 & -1 & \\ & & & & & & & & & & & & & -1 & 1627 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ T_7 \\ T_8 \\ T_9 \\ T_{10} \\ T_{11} \\ T_{12} \\ T_{13} \\ T_{14} \end{bmatrix} = \begin{bmatrix} 400 \\ 0 \\ 0 \\ 0 \\ 0 \\ 300000 \\ 975610 \\ 975610 \\ 975610 \\ 975610 \\ 975610 \\ 975610 \\ 975610 \\ 487805 \end{bmatrix}$$

These can be solved for

$$\begin{array}{lllll} T_1 = 383.553 & T_2 = 367.106 & T_3 = 350.659 & T_4 = 334.212 & T_5 = 317.765 \\ T_6 = 301.318 & T_7 = 300.0004 & T_8 = 300 & T_9 = 300 & T_{10} = 300 \\ T_{11} = 300 & T_{12} = 300 & T_{13} = 300 & T_{14} = 300 & \end{array}$$

A plot of the results can be developed as



24.23 For this case, with the exception of the node at the boundary between the rod and the tube, all interior nodes are represented by

$$-T_{i-1} + 2T_i - T_{i+1} = 0$$

The first interior node is

$$2T_1 - T_2 = 400$$

and the last interior node is

$$-T_{12} + 2T_{13} = 200$$

A heat balance can be written for the node at the boundary between the rod and the tube as

$$0 = -k_1 \frac{T_6 - T_5}{\Delta x} - k_2 \frac{T_6 - T_7}{\Delta x}$$

or multiplying by Δx and substituting the parameters

$$-80.2T_5 + 80.815T_6 - 0.615T_7 = 0$$

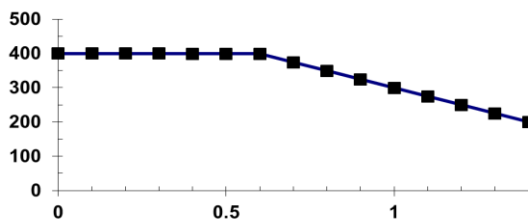
The foregoing equations can be assembled in matrix form as

$$\begin{bmatrix}
 2 & -1 & & & & & & & & & & & \\
 -1 & 2 & -1 & & & & & & & & & & \\
 & -1 & 2 & -1 & & & & & & & & & \\
 & & -1 & 2 & -1 & & & & & & & & \\
 & & & -1 & 2 & -1 & & & & & & & \\
 & & & & -1 & 2 & -1 & & & & & & \\
 & & & & & -80.2 & 80.815 & -0.615 & & & & & \\
 & & & & & & -1 & 2 & -1 & & & & \\
 & & & & & & & -1 & 2 & -1 & & & \\
 & & & & & & & & -1 & 2 & -1 & & \\
 & & & & & & & & & -1 & 2 & -1 & \\
 & & & & & & & & & & -1 & 2 & -1 \\
 & & & & & & & & & & & -1 & 2 \\
 & & & & & & & & & & & & -1 & 2
 \end{bmatrix}
 \begin{bmatrix}
 T_1 \\
 T_2 \\
 T_3 \\
 T_4 \\
 T_5 \\
 T_6 \\
 T_7 \\
 T_8 \\
 T_9 \\
 T_{10} \\
 T_{11} \\
 T_{12} \\
 T_{13}
 \end{bmatrix}
 =
 \begin{bmatrix}
 400 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 200
 \end{bmatrix}$$

These can be solved for

$$\begin{array}{lll}
 T_1 = 399.809388 & T_2 = 399.6187761 & T_3 = 399.4281641 \\
 T_4 = 399.2375521 & T_5 = 399.0469401 & T_6 = 398.8563282 \\
 T_7 = 373.9992871 & T_8 = 349.1422461 & T_9 = 324.2852051 \\
 T_{10} = 299.4281641 & T_{11} = 274.5711231 & T_{12} = 249.714082 \\
 T_{13} = 224.857041 & &
 \end{array}$$

A plot of the results can be developed as



24.24 Re-express as a system of odes

$$\frac{dy}{dx} = z \qquad \frac{dz}{dx} = -y$$

```

clear, clc
solinit = bvpinit(linspace(0,1,10),[1 1]);
sol = bvp4c(@odes2424,@bc2424,solinit);
x = linspace(0,1);

```

PROPRIETARY MATERIAL. © The McGraw-Hill Companies, Inc. All rights reserved. No part of this Manual may be displayed, reproduced or distributed in any form or by any means, without the prior written permission of the publisher, or used beyond the limited distribution to teachers and educators permitted by McGraw-Hill for their individual course preparation. If you are a student using this Manual, you are using it without permission.

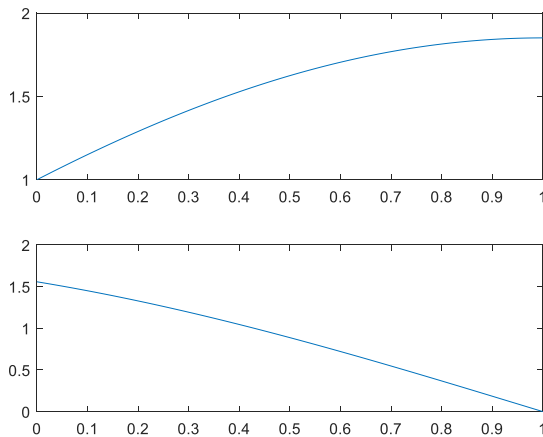
```

y = deval(sol,x);
subplot(2,1,1)
plot(x,y(1,:))
subplot(2,1,2)
plot(x,y(2,:))

function res = bc2424(ya,yb)
res = [ya(1)-1; yb(2)];

function dydx = odes2424(x,y)
dydx = [y(2); -y(1)];

```



24.25 The differential equation can be expressed as a pair of first-order odes

$$\frac{dy}{dx} = z$$

$$\frac{dz}{dx} = \frac{w_0}{6EI} \left(0.6Lx - \frac{x^3}{L} \right)$$

```

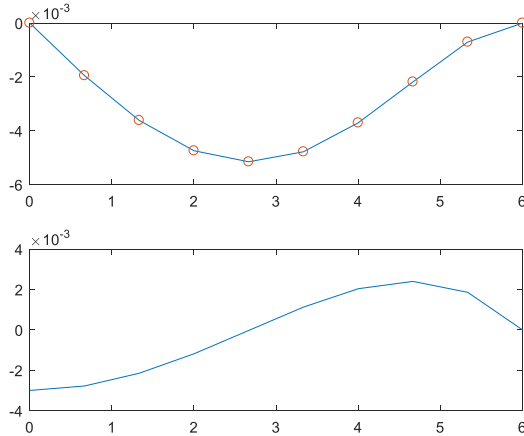
clear, clc
L = 600/100; E=50000*1e3*1e4; I=30000/1e8; w0 = 2.5*1e3*100;
solinit = bvpinit(linspace(0,L,10),[1 1]);
sol = bvp4c(@odes2425,@bc2425,solinit);
x = linspace(0,L,10);
y = deval(sol,x);
yp=w0/(120*E*I*L)*(-x.^5+2*L^2*x.^3-L^4*x);
subplot(2,1,1)
plot(x,y(1,:),x,yp,'o')
subplot(2,1,2)
plot(x,y(2,:))

function res = bc2425(ya,yb)
res = [ya(1); yb(1)];

function dydx = odes2425(x,y)

```

```
L = 600/100;E=50000*1e3*1e4;I=30000/1e8;w0 = 2.5*1e3*100;
dydx = [y(2);w0/(6*E*I)*(0.6*L*x-x^3/L)];
```



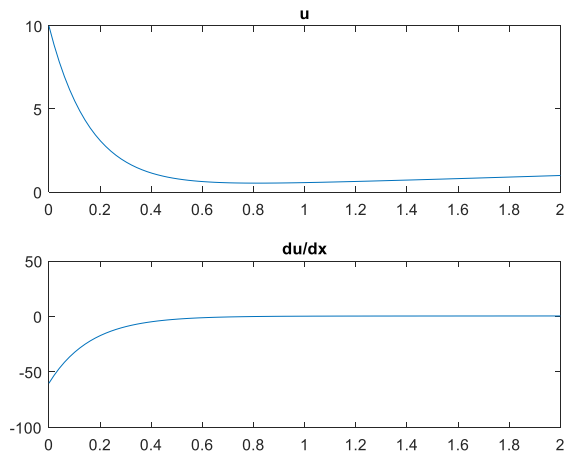
24.26 The differential equation can first be expressed as a pair of first-order odes

$$\begin{aligned}\frac{du}{dx} &= w \\ \frac{dw}{dx} &= -6w + u + 2\end{aligned}$$

```
clear, clc, clf
solinit = bvpinit(linspace(0,2,10),[1 1]);
sol = bvp4c(@odes2426,@bc2426,solinit);
x = linspace(0,2);
y = deval(sol,x);
subplot(2,1,1)
plot(x,y(1,:))
title('u')
subplot(2,1,2)
plot(x,y(2,:))
title('du/dx')

function res = bc2426(ya,yb)
res = [ya(1)-10; yb(1)-1];

function dydx = odes2426(x,y)
dydx = [y(2); -6*y(2)+y(1)+2];
```



24.27 The differential equation can first be expressed as a pair of first-order odes

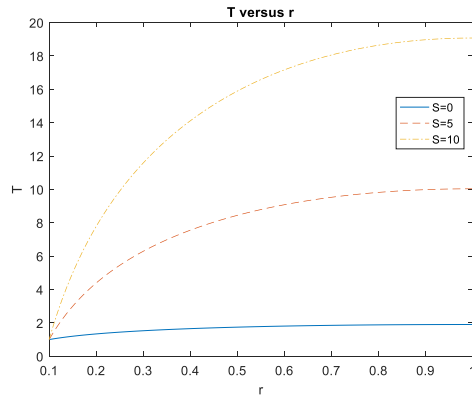
$$\frac{dT}{dr} = W$$

$$\frac{dW}{dr} = -\frac{1}{r}W - S$$

```
clear, clc, clf
global S
solinit = bvpinit(linspace(0.1,1,10),[1 1]);
x = linspace(0.1,1);
S=1;
sol = bvp4c(@odes2427,@bc2427,solinit);
y1 = deval(sol,x);
S=10;
sol = bvp4c(@odes2427,@bc2427,solinit);
y2 = deval(sol,x);
S=20;
sol = bvp4c(@odes2427,@bc2427,solinit);
y3 = deval(sol,x);
plot(x,y1(1,:),x,y2(1,:), '--',x,y3(1,:), '-.')
legend('S=0','S=5','S=10','location','best')
title('T versus r'),xlabel('r'),ylabel('T')

function res = bc2427(ya,yb)
res = [ya(1)-1; yb(2)];

function dydx = odes2427(x,y,S)
global S
dydx = [y(2); -y(2)/x-S];
```



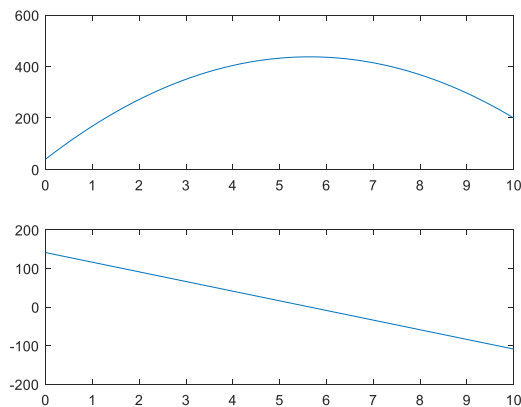
24.28 The differential equation can first be expressed as a pair of first-order odes

$$\begin{aligned}\frac{dT}{dx} &= W \\ \frac{dW}{dx} &= -25\end{aligned}$$

```
clear, clc, clf
solinit = bvpinit(linspace(0,10,10),[1 1]);
x = linspace(0,10);
sol = bvp4c(@odes2428,@bc2428,solinit);
y = deval(sol,x);
subplot(2,1,1)
plot(x,y(1,:))
subplot(2,1,2)
plot(x,y(2,:))
```

```
function res = bc2428(ya,yb)
res = [ya(1)-40; yb(1)-200];
```

```
function dydx = odes2428(x,y)
dydx = [y(2); -25];
```



24.29 The differential equation can first be expressed as a pair of first-order odes

$$\frac{dT}{dx} = W$$

$$\frac{dW}{dx} = -0.12x^3 + 2.4x^2 - 12x$$

```
clear, clc, clf
solinit = bvpinit(linspace(0,10,10),[1 1]);
x = linspace(0,10);
sol = bvp4c(@odes2429,@bc2429,solinit);
y = deval(sol,x);
subplot(2,1,1)
plot(x,y(1,:))
subplot(2,1,2)
plot(x,y(2,:))

function res = bc2429(ya,yb)
res = [ya(1)-40; yb(1)-200];

function dydx = odes2429(x,y)
dydx = [y(2); -0.12*x^3+2.4*x^2-12*x];
```

