Applied Numerical Methods with MATLAB for Engineers and Scientists (4th Edition)

Problem

Repeat Prob. 24.28, but for the following heat source: $f(x) = 0.12x^3 - 2.4x^2 + 12x$.

Step-by-step solution

Step 1 of 12

The insulated heated rod with uniform heat source is modelled as,

$$\frac{d^2T}{dx^2} = -\left(0.12x^3 - 2.4x^2 + 12x\right)$$

The boundary conditions are:

$$T(x=0) = 40^0\mathrm{C}$$
$$T(x=10) = 200^0\mathrm{C}$$

Comment

Step 2 of 12

Convert the above boundary value problem into an initial value problem by defining the rate of change of temperature, that is, $\frac{dT}{dx}$.

Let $\frac{dT}{dx} = z$, $\frac{d^2T}{dx^2} = \frac{dz}{dx}$

Therefore,

$$\frac{dT}{dx} = z,$$
$$\frac{dz}{dx} = -\left(0.12x^3 - 2.4x^2 + 12x\right)$$

Let $T(0) = 40\ \mathrm{K}$, $T(10) = 200\ \mathrm{K}$

Comment

Step 3 of 12

(a)

Use the shooting method to develop a solution for the above initial-value problem.

Write the following MATLAB code 'ShootingMethod2.m'.

```
function ShootingMethod2
% Inputs are L,h,Ta,T0,T10,z0
% Outputs are y
T0 = 40;
T10 = 200;
npts = 100;
xspan = linspace(0,10,101);
xlim([50 500])
disp(['-------------------------------------'])
for i = 1:2
z0(i) = input('\nEnter a guess for dT/dx(0): ');
disp(['------------------------------------'])
y0 = [T0 z0(i)]';
[x,y] = ode45(@HeatTransfer,xspan,y0);
T = y(:,1);
Tlow(i) = y(length(y));
plot(x,T)
xlabel('x')
ylabel('T')
hold on
fprintf('The final value of T is %5.2f\n', y(length(y)))
end
slope = (z0(2)-z0(1))/(Tlow(2)-Tlow(1));
z1 = z0(1)+slope*(T10-Tlow(1));
[x,y] = ode45(@HeatTransfer,[0 10],[300,z1]);
T = y(:,1);
disp(['----------------------------------'])
fprintf('The final value of T is %5.2f\n',y(length(y)))
disp(['----------------------------------'])
plot(x,T,'r')
function dy = HeatTransfer(x,y)
T = y(1,1);
z = y(2,1);
% Calculate state derivative vectory
dTdx = z;
dzdx = -(0.12*x^3-2.4*x^2+12*x);
% Save state derivative vector
dy(1,1) = dTdx;
dy(2,1) = dzdx;
```

Comment

Step 4 of 12

Choose the below two guesses for the initial condition of z.

Solve the ODEs with the 4th-order RK method using a step size of 0.125.

The results are as follows:

```
>> ShootingMethod2
--------------------------------------

Enter a guess for dT/dx(0): -1
-------------------------------------
The final value of T is -570.00

Enter a guess for dT/dx(0): -0.5
-------------------------------------
The final value of T is -565.00
-----------------------------------
The final value of T is 460.00
-----------------------------------
```

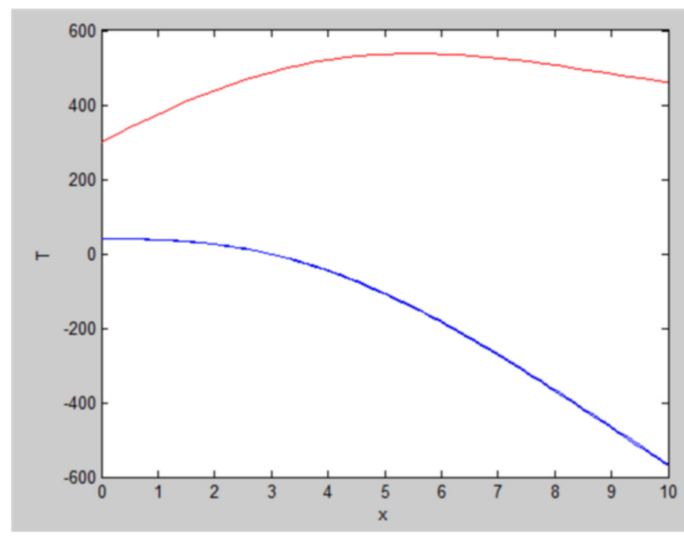
Comment

Step 5 of 12

Use these values to derive the correct initial condition,

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

The resulting fit is displayed below:



Comment

Step 6 of 12

(b)

Finite Difference Method:

The second derivative at node *i* represented by,

$$\frac{d^2T}{dx^2} = \frac{T_{i-1} - 2T_i + T_{i+1}}{\Delta x} \quad \text{(centred finite difference)}$$

Substitute $\frac{d^2T}{dx^2} = \frac{T_{i-1} - 2T_i + T_{i+1}}{(\Delta x)^2}$ into $\frac{d^2T}{dx^2} = -\left(0.12x^3 - 2.4x^2 + 12x\right)$, we find,

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

Substitute $\Delta x = 2$ and collect the terms.

$$T_{i-1} - 2T_i + 4\left(0.12x_i^3 - 2.4x_i^2 + 12x_i\right) + T_{i+1} = 0 \quad ......(1)$$

or

$$-T_{i+1} + 2T_i - T_{i-1} = 4\left(0.12x_i^3 - 2.4x_i^2 + 12x_i\right)$$

Comment

Step 7 of 12

This equation can be written for each node with the result

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

This equation can be written for each node with the result

```
>> 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
```

Comment

Step 8 of 12

The first and last equations (from (1)) along with the boundary conditions are given by,

$$T_a - 2T_1 + 4\left(0.12x_1^3 - 2.4x_1^2 + 12x_1\right) + T_2 = 0$$
$$T_{n-1} - 2T_n + 4\left(0.12x_1^3 - 2.4x_1^2 + 12x_1\right) + T_b = 0$$

Here T_a and T_b are the boundary conditions.

Differentiate $f(T) = T_a - 2T_1 + 4\left(0.12x_1^3 - 2.4x_1^2 + 12x_1\right) + T_2$ with respect to T_1.

Thus, $f'(T) = -2$

Comment

Step 9 of 12

Use the following MATLAB program, 'fda2.m' to approximate solutions for the linear BVP $\frac{d^2T}{dx^2} = -\left(0.12x^3 - 2.4x^2 + 12x\right), T(0) = 40\ \mathrm{K}, T(10) = 200\ \mathrm{K}$:

```
function W=fda2(inter,bv,n)
% Inputs:interval(inter), boundary values(bv), number of steps(n) are inputs
% Command used: w=fda2( [0 10],[300 400],4)
a=inter(1); b=inter(2);ya=bv(1); yb=bv(2);
h=(b-a)/(n+1); % This gives the step size h
W=zeros(n,1) ; % This initializes solution array w
for i=1:20
W=W-jac(W,inter,bv,n)\f(W,inter,bv,n);
end
plot([a a+(1:n)*h b],[ya W' yb]); % This gives plot w with boundary data
xlabel('x'),ylabel('T')
title('Approximate solutions of BVP by the Finite Difference Method')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function y=f(W,inter,bv,n)
y=zeros(n,1) ; h=(inter(2)-inter(1))/(n+1);
y(1)=bv(1)-2*W(1)+100 +W(2);
y(n)=W(n-1)-2*W(n)+100+bv(2) ;
for i=2:n-1
y(i)=W(i-1)-2*W(i)+100+W(i+1);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function a=jac(W,inter,bv,n)
a=zeros(n,n); h=(inter(2)-inter(1))/(n+1);
for i=1:n
a(i,i)=-2;
end
for i=1:n-1
a(i,i+1)=1;
a (i+1,i)=1;
end
```

Comment

Step 10 of 12

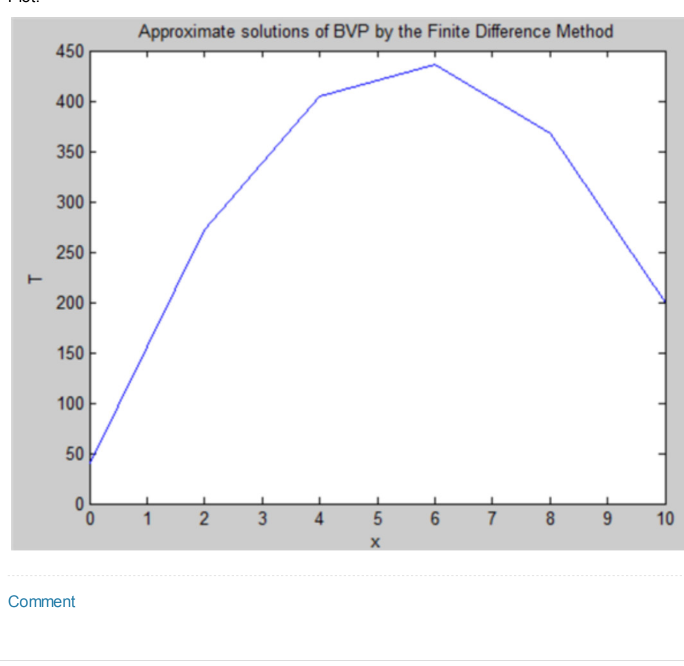
Output:

```
>> w=fda2( [0 10],[40 200],4)

w =

   272.0000
   404.0000
   436.0000
   368.0000
```

Plot:



Comment

Step 11 of 12

(c)

Bvp4c:

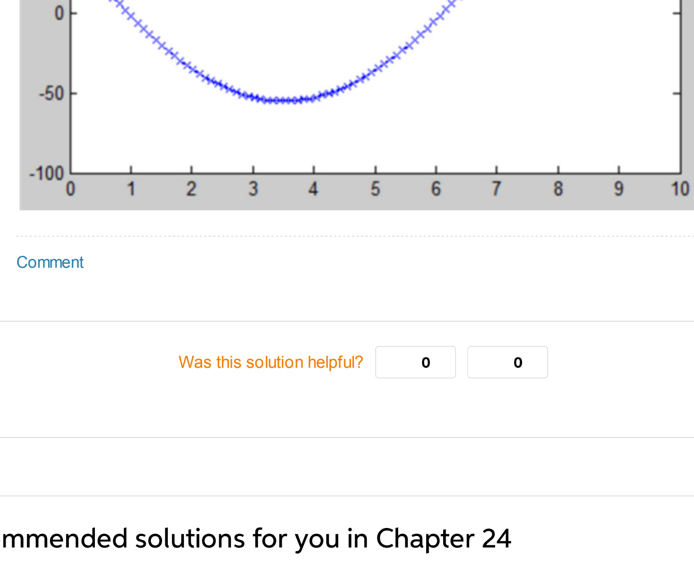
Write the following SolveBVP.m file and then run:

```
% Function to solve y''+ 25 = 0.
function SolveBVP()
solinit = bvpinit(linspace(0,10,100),[1,-1]); %initial guess 1,-1
sol = bvp4c(@deriv,@bcs,solinit);
x=linspace(0,10,100);
y=deval(sol,x);
plot(x,y(1,:),'b-x');
function dYdx = deriv(x,Y)
dYdx(1) = Y(2);
dYdx(2) = 0.12*x^3-2.4*x^2+12*x;
% boundary conditions y(0)=40, y(10)=200.
function res = bcs(ya,yb)
res = [ ya(1)-40; yb(1)-200];
```

Comment

Step 12 of 12

Run the function to obtain the below graph:



Comment