Find the appropriate stoichiometric coeffs for the following chemical reactions by conserving the number of atoms of each element between the reactants and products.

(a) $a_1 FeS + a_2 O_2 \rightarrow a_3 FeO + a_4 SO_2$

Fe balance $\rightarrow$ $a_3 - a_1 = 0$

O balance $\rightarrow$ $a_3 + 2a_4 - 2a_2 = 0$

S balance $\rightarrow$ $a_4 - a_1 = 0$

and $a_1 = \alpha$ ← chosen to give integer coeffs

In matrix form, these become

$$\begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -2 & 1 & 2 \\ -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \alpha \end{bmatrix}$$

Program this into matlab (see reaction_eqns1.m) to get

with $\alpha = 1$ $a_1 = 1$ $a_2 = 1.5$ $a_3 = 1$ $a_4 = 1$

$\therefore$ let $\alpha = 2$ then

$a_1 = 2$ $a_2 = 3$ $a_3 = 2$ $a_4 = 2$

ok

$\therefore$ $\boxed{2FeS + 3O_2 \rightarrow 2FeO + 2SO_2}$

(b) $a_1 (Cr(N_2H_4CO)_6)_4 (Cr(CN)_6)_3 + a_2 KMnO_4$
$+ a_3 H_2SO_4 \longrightarrow a_4 K_2Cr_2O_7 + a_5 MnSO_4$
$+ a_6 CO_2 + a_7 KNO_3 + a_8 K_2SO_4 + a_9 H_2O$

K balance $\rightarrow 2a_4 + a_7 + 2a_8 - a_2 = 0$

Cr balance $\rightarrow 2a_4 - 7a_1 = 0$

O balance $\rightarrow 7a_4 + 4a_5 + 2a_6 + 3a_7 + 4a_8 + a_9 - 24a_1 - 4a_2 - 4a_3 = 0$

Mn balance $\rightarrow a_5 - a_2 = 0$

S balance $\rightarrow a_5 + a_8 - a_3 = 0$

C balance $\rightarrow a_6 - 42a_1 = 0$

N balance $\rightarrow a_7 - 66a_1 = 0$

H balance $\rightarrow 2a_9 - 96a_1 - 2a_3 = 0$

and $a_1 = \alpha \leftarrow$ chosen to give integer results

In matrix form, these become

$$\begin{bmatrix} 0 & -1 & 0 & 2 & 0 & 0 & 1 & 2 & 0 \\ -7 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ -24 & -4 & -4 & 7 & 4 & 2 & 3 & 4 & 1 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 1 & 0 \\ -42 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -66 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -96 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 2 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \alpha \end{bmatrix}$$

program above matrix in Matlab to give with $\alpha = 1$

$a_1 = 1$ $a_2 = 117.6$ $a_3 = 139.9$ $a_4 = 3.5$ $a_5 = 117.6$

$a_6 = 42$ $a_7 = 66$ $a_8 = 22.3$ $a_9 = 187.9$

To make these integers, I need to multiply by 10

$\therefore$ set $\alpha = 10$

then $a_1 = 10$ $a_2 = 1176$ $a_3 = 1399$ $a_4 = 35$

$a_5 = 1176$ $a_6 = 420$ $a_7 = 660$ $a_8 = 223$

$a_9 = 1879$

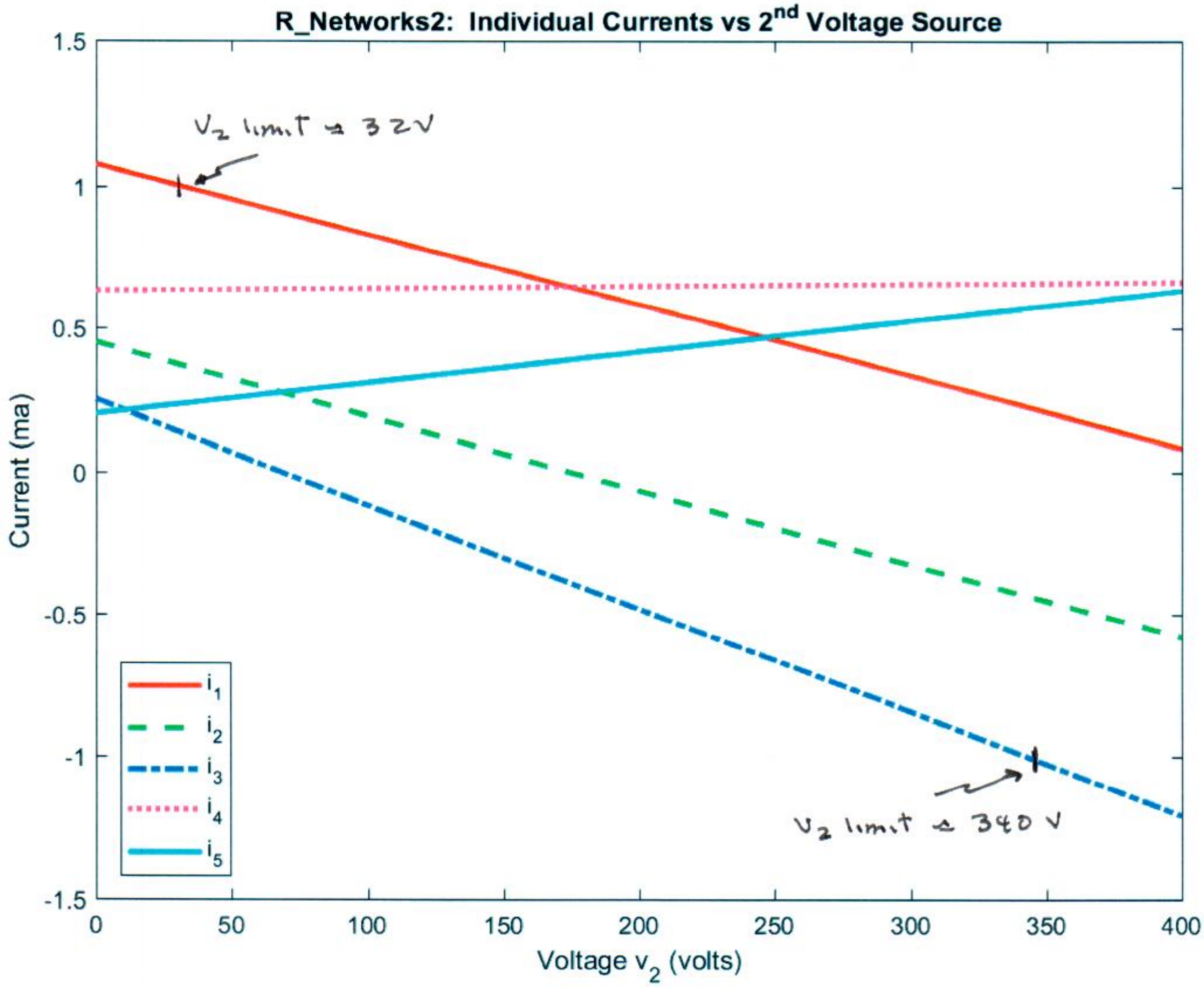
thus, the balanced reaction eqn is:

$$10\,(Cr(N_2H_4CO)_6)_4(Cr(CN)_6)_3 + 1176\,KMnO_4 + 1399\,H_2SO_4 \longrightarrow 35\,K_2Cr_2O_7 + 1176\,MnSO_4 + 420\,CO_2 + 660\,KNO_3 + 223\,K_2SO_4 + 1879\,H_2O$$

```
%
%   REACTION_EQNS1.M   Balances for Chemical Reaction Equations
%
%   This file computes the stoichiometric coefficients for a couple of reaction
%   equations -- see the hand documentation for the development of the appropriate
%   matrix equations...
%
%   File prepared by J. R. White, UMass-Lowell  (last update: Nov. 2017)
%

      clear all,   close all
      format long e
%
%   find coeffs for Case a with a1 = 2
      A = [-1 0 1 0; 0 -2 1 2;-1 0 0 1;1 0 0 0];
      b = [0 0 0 2]';
      disp('Stoichiometric coeffs for Case a: ');  a = A\b
%
%   find coeffs for Case b with a1 = 10
      A = [  0 -1  0 2 0 0 1 2 0; -7  0  0 2 0 0 0 0 0;
           -24 -4 -4 7 4 2 3 4 1;  0 -1  0 0 1 0 0 0 0;
             0  0 -1 0 1 0 0 1 0;-42  0  0 0 0 1 0 0 0;
           -66  0  0 0 0 0 1 0 0;-96  0 -2 0 0 0 0 0 2;
             1  0  0 0 0 0 0 0 0];
      b = [0 0 0 0 0 0 0 0 10]';
      disp('Stoichiometric coeffs for Case b: ');  a = A\b
%
%  end of problem
```

```
>> reaction_eqns1
Stoichiometric coeffs for Case a:
a =
     2
     3
     2
     2
Stoichiometric coeffs for Case b:
a =
     1.000000000000001e+01
     1.176000000000001e+03
     1.399000000000000e+03
     3.500000000000001e+01
     1.176000000000000e+03
     4.200000000000002e+02
     6.600000000000003e+02
     2.230000000000001e+02
     1.879000000000001e+03
```



$\therefore$ for $|i| < 1$ ma $\quad 32 < V_2 < 340$ volts

```
%
%   RESISTIVE_NETWORKS2.M    Find the currents and maximum allowed voltage range
%                                for a particular resistive network
%
%   A resistive network is given along with the applied voltage for source #1.
%   This file finds the allowed range of values (using graphical techniques) for a
%   second voltage source such that the maximum current in the circuit is 1 ma.
%   This requires solving a system of equations for each potential value of v2 and
%   plotting the resulting currents versus v2.  A point where any of the currents
%   exceeds 1 ma represents a value of v2 that is not allowed.  In this way, we
%   should be able to easily visualize an appropriate range of values for v2.
%
%   Concerning units, all the resistor values are given in k-ohm = 1000 ohm.  Thus,
%   since i = v/R, where v is in volts, the result will automatically be in
%   milliamps (ma) if we use resistance in k-ohm and voltage in volts.
%
%   Reference:  This problem was derived from Prob. 24 in Chapter 4 of Palm's text,
%   "Introduction to Matlab 6 for Engineers," McGraw Hill (2001).
%
%   File prepared by J. R. White, UMass-Lowell  (last update: Nov. 2017)
%

      clear all,   close all
%
%   define given problem parameters
      R = [5 100 200 150 250];   % resistances (k-ohm)
      v1 = 100;                  % voltage source #1 (volts)
%
%   set potential range for second voltage source
      Nv2 = 101;  v2 = linspace(0,400,Nv2)';
%
%   start loop over number of voltage values
      cur = zeros(Nv2,5);                      % initialize 2-D matrix for results
      A = [R(1)   0     0    R(4)    0  ;      % coeff matrix (independent of v2)
            0    R(2)   0   -R(4)   R(5);
            0     0    R(3)   0    -R(5);
            1    -1     0    -1      0  ;
            0     1    -1     0     -1 ];
      for j = 1:Nv2
        b = [v1   0   -v2(j)  0      0]';      % RHS vector for specific v2
        I = A\b;                               % solve for current vector in ma
        cur(j,:) = I';                         % store currents in jth row of 2-D matrix
      end
%
%   let's plot the results to visualize the allowed range for v2
      plot(v2,cur(:,1),'r-',v2,cur(:,2),'g--',v2,cur(:,3),'b-. ',...
           v2,cur(:,4),'m:',v2,cur(:,5),'c-','LineWidth',2),grid
      xlabel('Voltage v_2 (volts)'),ylabel('Current (ma)')
      title('R\_Networks2:  Individual Currents vs 2^{nd} Voltage Source')
      legend('i_1','i_2','i_3','i_4','i_5','Location','SouthWest')
%
%  end of problem
```

Consider heat conduction through the wall of a circular pipe. The inside wall temperature is 200°c and the outside wall temperature is 80°c. The inner radius of the pipe is a = 0.05 m and the outer radius is b = 0.1 m.

The differential eqn which describes the temperature distribution in the pipe is

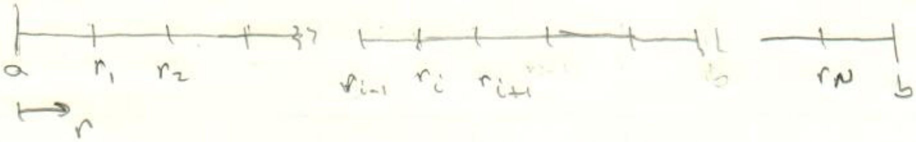
$$\frac{d^2T}{dr^2} + \frac{1}{r}\frac{dT}{dr} = 0$$

with boundary conditions

$$T(a) = 200 \quad \text{and} \quad T(b) = 80$$

Although this boundary value problem (BVP) can be solved analytically, here you will develop a numerical solution (see the Example Application in "Introduction to Finite Difference Methods" in the notes for Lesson #4).

One always starts by descretization of the independent variable. Recognizing that the temperatures at the inner and outer surfaces are known, we have the following 1-D descrete representation of the spatial grid for this problem



where r_i represents the radius of the i^{th} descrete unknown temperature T_i. The vector of spatial points can be generated within matlab as follows

N = # of unknown Temperatures in domain of interest

$$h = \Delta r = \frac{b-a}{N+1} = \text{step size}$$

r = a+h : h : b−h = vector of locations for unknown temperatures

(a) Now, using 2nd order central finite difference (FD) approximations for the derivatives

$$\left.\frac{d^2T}{dr^2}\right|_i = \frac{T_{i-1} - 2T_i + T_{i+1}}{h^2}$$

and

$$\left.\frac{dT}{dr}\right|_i = \frac{T_{i+1} - T_{i-1}}{2h}$$

formally develop the discrete FD equations needed for this problem, being careful to treat the boundary nodes as special cases (i.e. when $i=1$ and $i=N$). The final result should be as follows:

central node

$$\left.\frac{d^2T}{dr^2}\right|_i + \frac{1}{r}\left.\frac{dT}{dr}\right|_i = 0$$

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

$$T_{i-1} - 2T_i + T_{i+1} + \frac{h}{2r_i}\left(T_{i+1} - T_{i-1}\right) = 0$$

$$\boxed{\left(1 - \frac{h}{2r_i}\right)T_{i-1} - 2T_i + \left(1 + \frac{h}{2r_i}\right)T_{i+1} = 0} \qquad i = 2:N-1$$

node 1

simply let $i=1$

$$\boxed{-2T_1 + \left(1 + \frac{h}{2r_1}\right)T_2 = -\left(1 - \frac{h}{2r_1}\right)T(a)}$$

node N

simply let $i=N$

$$\boxed{\left(1 - \frac{h}{2r_N}\right)T_{N-1} - 2T_N = -\left(1 + \frac{h}{2r_N}\right)T(b)}$$

(b) Now, implement the discrete eqns developed in Part a along with the discrete geometry information and actually solve the resultant matrix equations, $\underline{\underline{A}}\,\underline{T} = \underline{B}$, for the unknown discrete temperature profile, $\underline{T}$, within the pipe. Plot the temperature profile (including the two known end-point temperatures) for this problem. Does your solution make sense? Are the boundary conditions (BCs) satisfied by your solution?

$$\frac{d}{dr}\left(\frac{dT}{dr}\right) + \frac{1}{r}\frac{dT}{dr} = 0$$

$$\frac{d}{dr}(T') = -\frac{1}{r}T'$$

$$\frac{dT'}{T'} = -\frac{1}{r}\,dr$$

$$\ln T' = -\ln r + \ln c_1$$

$$T' = c_1/r$$

$$dT = \frac{c_1}{r}\,dr \qquad T(r) = c_1 \ln r + c_2$$

(c) The analytical soln to this problem is given by

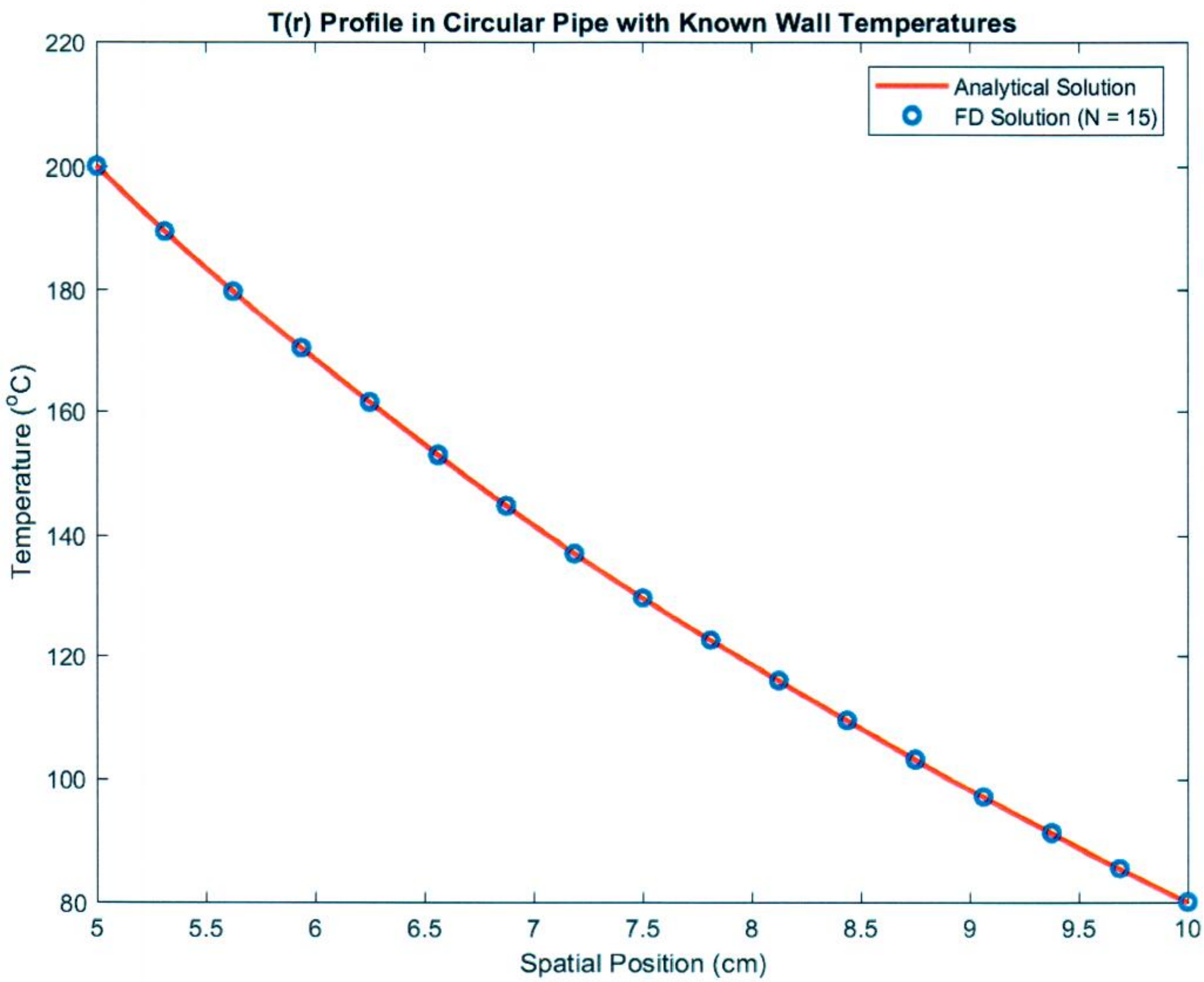
$$T(r) = c_1 \ln r + c_2$$

where c_1 and c_2 are constants that are chosen to satisfy the given BCs., or

see htprob_fd1.m

$$T(a) = c_1 \ln a + c_2$$
$$T(b) = c_1 \ln b + c_2$$

Evaluate and plot this exact solution for the conditions given above. Does the FD solution from Part b compare favorably? Explain any differences.



T(r) Profile in Circular Pipe with Known Wall Temperatures
Analytical Solution
FD Solution (N = 15)
Temperature (°C)
Spatial Position (cm)
220
200
180
160
140
120
100
80
5
5.5
6
6.5
7
7.5
8
8.5
9
9.5
10

```
%
%    HTPROB_FD1.M   Introduction to FD Methods for Solving BVPs
%                   Heat Conduction through a thick Circular Pipe
%
%    This file illustrates the finite difference (FD) method for solving BVPs.
%    We solve a simple heat transfer problem that involves heat conduction in a thick
%    circular pipe with known temperatures at the inner and outer radius of the pipe.
%    For comparison, we also evaluate the analytical solution -- showing that the
%    two methods give identical solutions.
%
%    File prepared by J. R. White, UMass-Lowell  (last update:  Nov. 2017)
%

      clear all,  close all,  nfig = 0;
%
%    identify basic problem data
      a = 0.05;  b = 0.1;               % inner and outer radii of pipe (m)
      Ta = 200;  Tb = 80;               % boundary temperatures (C)
%
%%   Analytical Solution
%
      Nr = 51;  re = linspace(a,b,Nr)';                        % independent variable
      Ae = [log(a) 1;log(b) 1];  Be = [Ta Tb]';  c = Ae\Be;   % coeffs in exact soln
      Te = c(1)*log(re) + c(2);                                % exact soln
%
%%   FD Solution
%
%    setup and solve matrix eqn for discrete temperatures
      N = input('Enter number of discrete unknowns (N) for problem: ');
      h = (b-a)/(N+1);   rn = a+h:h:b-h;   rn = rn';
      A = zeros(N,N);     B = zeros(N,1);
%    node 1
      A(1,1) = -2;  A(1,2) = 1 + h/(2*rn(1));  B(1) = -(1 - h/(2*rn(1)))*Ta ;
%    interior nodes
      for i = 2:N-1
        A(i,i-1) = 1 - h/(2*rn(i));  A(i,i) = -2;  A(i,i+1) = 1 + h/(2*rn(i));
      end
%    last node
      A(N,N-1) = 1 - h/(2*rn(N));  A(N,N) = -2;   B(N) = -(1 + h/(2*rn(N)))*Tb;
%    solve
      Tn = A\B;
%    add on the known endpoint temperatures to the solution vector
      rnp = [a; rn; b];   Tnp = [Ta; Tn; Tb];
%
%%   plot both solutions
      nfig = nfig+1;   figure(nfig)
      plot(re*100,Te,'r-',rnp*100,Tnp,'bo','LineWidth',2),grid
      title('T(r) Profile in Circular Pipe with Known Wall Temperatures')
      xlabel('Spatial Position (cm)')
      ylabel('Temperature (^oC)')
      legend('Analytical Solution',['FD Solution (N = ',num2str(N),')'])
%
%   end of problem
```