Matlab Implementation of GWS
1D SS Heat Conduction with Variable Thermal Conductivity and Source

For our one-dimensional heat flow in a thick slab with x-dependent thermal conductivity and source, our governing differential equation and associated boundary conditions were

$$\mathcal{L}(T) = -\frac{d}{dx}\left(k(x)\frac{dT}{dx}\right) - s(x) = 0 \qquad \Omega \in (0, L) \tag{1a}$$

$$\ell(T) = k\frac{dT}{dn} - q_{in} = 0 \qquad \partial\Omega \in 0 \tag{1b}$$

$$T = T_b \qquad \partial\Omega \in L \tag{1c}$$

Applying the GWS recipe resulted in

$$GWS^h = S_e GWS_e^h = S_e\left(\int_{\Omega_e} \frac{d\{N_k\}}{dx} k(x) \frac{d\{N_k\}^T}{dx} d\overline{x}\{Q\}_e - \int_{\Omega_e} \{N_k\} s(x) d\overline{x} - k(x)\frac{dT^N}{dx}\{\delta_k\} = \{0_k\}\right) \tag{2a}$$

which was expressed compactly as

$$GWS^h = S_e GWS_e^h = S_e\left([\mathrm{DIFF}]_e \{Q\}_e = \{\mathrm{SRC}\}_e + \{\mathrm{BFLX}\}_e\right) \tag{2b}$$

It is at this point that we
1. Select the degree of the basis function.
2. Decide how to handle the x-dependent properties (element average values or element interpolated values).
3. Define a discretization.
4. Perform the required calculus and assemble.
5. Apply the Dirichlet data and solve.

For this problem, our physical data, all of which have appropriate units, will be

$$L = 1 \qquad k(x) = 0.1 + 0.001x$$
$$q = 5 \qquad s(x) = 25\left(1 - x^2\right)$$
$$T_b = 10$$

Making our decisions and solving:
1. We shall choose the linear basis function $\{N_1\}$.
2. Examining the variation in $k(x)$ and $s(x)$, we see that the conductivity is nearly constant across the domain, hence we will use element average values for $[\mathrm{DIFF}]_e$. The source term, however, varies considerably. We will therefore use element interpolated values for $\{\mathrm{SRC}\}_e$
3. We will initiate the analysis with a four element (five node) uniform discretization.
4. Our syntactical element level GWS_e^h for the terms in (2b) thus becomes

$$
\begin{aligned}
[\text{DIFF}]_e \{Q\}_e &= \int_{\Omega_e} \frac{d\{N_k\}}{dx} k(x) \frac{d\{N_k\}^T}{dx} d\bar{x} \{Q\}_e \\
&\approx \bar{k}_e \int_{\Omega_e} \frac{d\{N_k\}}{dx} \frac{d\{N_k\}^T}{dx} d\bar{x} \{Q\}_e \\
&= \quad (\ )(\bar{k})_e \ \{\ \}_e^T \ (-1)[\text{A211L}]\{Q\}_e
\end{aligned} \tag{3a}
$$

$$
\begin{aligned}
\{\text{SRC}\}_e &= \int_{\Omega_e} \{N_k\} s(x) d\bar{x} \\
&\approx \int_{\Omega_e} \{N_k\}\{N_k\}^T d\bar{x} \{S\}_e \\
&= \quad (\ )(\ )_e \ \{\ \}_e^T \ (1)[\text{A200L}]\{S\}_e
\end{aligned} \tag{3b}
$$

Note that $\{\text{BFLX}\}_e$ will assemble to a vector of zeros with the applied fluxes in the first and last entries, hence no special syntax is required.

5. We now invoke the powers of Matlab and FeMPSE.

```
% FEm.PSE template - Lab01
% steady conduction with variable conductivity and source

clear all;
format short;

global X1;        % array for node coordinates

% load the FE matrix library
load femlib;

% mesh parameters
XL = 0;
XR = 1;
% uniform discretization, M=4
nnodes=5;
X1 = linspace(XL,XR,nnodes);

% Boundary Conditions
q  = 5;           % BC: Heat flux at x=0
Tb = 10;          % BC: Prescribed temperature at x=L=1

% average element data - conductivity
K = 0.1+0.001*X1';
Kbar = (K(1:nnodes-1)+K(2:nnodes))/2;

% nodal element data - source
S = 25*(1-X1.^2);

%------------------------------------------------------------------------------

% Assemble all matrices on the LHS:
      % assemble Matrix for GWSe term [DIFF]
      DIFF = asjac1D([],[Kbar],[],-1,A211L,[]);

      % since [DIFF] is only term in LHS
      LHS = DIFF;

% Assemble all vectors on the RHS
      % assemble Vector for GWSe term SRC
      SRC = asres1D([],[],[],1,A200L,S);

      % form Vector for GWSe term BFLX
      BFLX = zeros(nnodes,1);  % an nnodes by 1 matrix of zeros
      % modify BFLX for flux BC at node 1
      BFLX(1,1) = q;

      % Build RHS
      RHS = SRC + BFLX;

% modify RHS for the Dirichlet boundary condition
RHSdiri = RHS;
RHSdiri(nnodes,1) = Tb;

% modify LHS for Dirichlet BC direct solve
LHSdiri = LHS; % copy DIFF
LHSdiri(nnodes,:) = zeros(1,nnodes); % zero out the last row
LHSdiri(nnodes,nnodes) = 1;   % i.e., 1*QM = Tb
```

```
% solve the linear system
Q =  LHSdiri \ RHSdiri;

Tleft = Q(1,1)

% compute Energy Norm using [DIFF] (without Dirichlet BC)
format long; % display Enorm in long format
Enorm = 0.5*Q'*DIFF*Q

% create nice graphics
figure(1)
plot(X1,Q,'ko-')
xlabel('x')
ylabel('T')
title('Temperature Distribution - Lab 01')
text(0.1,30,'Zachariah Chambers')
text(0.1,20,'March 18, 20002')
```