Transient Heat Conduction

From last time, the governing partial differential equation for transient, one-dimensional heat equation was

$$\mathcal{L}(Q) = \frac{\partial Q(x,t)}{\partial t} - \alpha \frac{\partial^2 Q(x,t)}{\partial x^2} - s = 0 \qquad \Omega \in (x_L, x_R) \cup t \tag{1}$$

Choosing to place the time dependence in the expansion coefficient led to a series approximation of the unknown solution

$$Q(x,t) \approx Q^N(x,t) = \sum_{\alpha=1}^{N} \Psi_\alpha(x) Q(t)_\alpha \tag{2}$$

which, upon turning the GWS crank, resulted in

$$GWS^h = \mathrm{S}_e \left(\int_{\Omega_e} \{N_k\}\{N_k\}^T d\overline{x} \frac{d\{Q\}_e}{dt} + \alpha \int_{\Omega_e} \frac{d\{N_k\}}{dx} \frac{d\{N_k\}^T}{dx} d\overline{x} \{Q\}_e - \int_{\Omega_e} \{N_k\}\{N_k\}^T d\overline{x} \{S\}_e = \{0\}_e \right) \tag{3}$$

Upon assembly, the following ordinary differential equation in time was realized

$$[MASS]\{Q\}' + [DIFFA]\{Q\} - \{SRCS\} = \{0\} \tag{4}$$

From the Theta Taylor Series we came up with a general-purpose time integration procedure for advancing our solution:

$$\{Q\}_{n+1} = \{Q\}_n + \Delta t \left(\Theta \{Q\}'_{n+1} + (1-\Theta)\{Q\}'_n \right) \tag{5}$$

We must now substitute (5) into (4) to temporally discretize our ODE statement. First, we shall evaluate (4) at the old and new time stations, t_n and t_{n+1}

Old: $$[\mathrm{MASS}]\{Q\}'_n + [\mathrm{DIFFA}]\{Q\}_n - \{\mathrm{SRC}\}_n = \{0\} \tag{6a}$$

New: $$[\mathrm{MASS}]\{Q\}'_{n+1} + [\mathrm{DIFFA}]\{Q\}_{n+1} - \{\mathrm{SRC}\}_{n+1} = \{0\} \tag{6b}$$

Solving for the primes

Old: $$\begin{aligned} \{Q\}'_n &= [\mathrm{MASS}]^{-1}\left(-[\mathrm{DIFFA}]\{Q\}_n + \{\mathrm{SRC}\}_n\right) \\ &= [\mathrm{MASS}]^{-1}\{-\mathrm{RES}\}_n \end{aligned} \tag{7a}$$

New: $$\begin{aligned} \{Q\}'_{n+1} &= [\mathrm{MASS}]^{-1}\left(-[\mathrm{DIFFA}]\{Q\}_{n+1} + \{\mathrm{SRC}\}_{n+1}\right) \\ &= [\mathrm{MASS}]^{-1}\{-\mathrm{RES}\}_{n+1} \end{aligned} \tag{7b}$$

where $\{\mathrm{RES}\}_{n,n+1}$ is the *spatial residual* of the assembled GWS^h (4). Substituting (7a,b) into (5) to replace the nodal derivatives with nodal solutions

$$\begin{aligned} \{Q\}_{n+1} &= \{Q\}_n + \Delta t \left(\Theta [\mathrm{MASS}]^{-1}\left(-[\mathrm{DIFFA}]\{Q\}_{n+1} + \{\mathrm{SRC}\}_{n+1}\right) + (1-\Theta)[\mathrm{MASS}]^{-1}\left(-[\mathrm{DIFFA}]\{Q\}_n + \{\mathrm{SRC}\}_n\right)\right) \\ &= \{Q\}_n + \Delta t \left(\Theta [\mathrm{MASS}]^{-1}\{-\mathrm{RES}\}_{n+1} + (1-\Theta)[\mathrm{MASS}]^{-1}\{-\mathrm{RES}\}_n \right) \\ &= \{Q\}_n - \Delta t [\mathrm{MASS}]^{-1}\left(\Theta\{\mathrm{RES}\}_{n+1} + (1-\Theta)\{\mathrm{RES}\}_n \right) \end{aligned} \tag{8}$$

Defining the temporal change in solution as

$$\{\Delta Q\} = \{Q\}_{n+1} - \{Q\}_n \tag{9}$$

and clearing the mass inverse, the terminal form of our GWS^h + ΘTS is

$$\boxed{\mathrm{GWS}^h + \Theta\mathrm{TS} = [\mathrm{MASS}]\{\Delta Q\} + \Delta t\left(\Theta\{\mathrm{RES}\}_{n+1} + (1-\Theta)\{\mathrm{RES}\}_n\right) = \{0\}} \quad (10)$$

If we set Θ=0, the algorithm is explicit and readily solved. However, for any other choice of theta, an iterative algorithm is required since we need to know $\{Q\}_{n+1}$ to evaluate $\{RES\}_{n+1}$. Recalling the Newton iteration algorithm :

$$\{Q\}_{n+1}^{p+1} = \{Q\}_{n+1}^{p} + \{\delta Q\}_{n+1}^{p+1}$$

$$[JAC]_{n,n+1}^{p}\{\delta Q\}_{n+1}^{p+1} = -\{FQ\}_{n,n+1}^{p}$$

$$\boxed{\{FQ\}_{n,n+1}^{p} = [MASS]\{\Delta Q\}_{n,n+1}^{p} + \Delta t\left(\Theta\{RES\}_{n+1}^{p} + (1-\Theta)\{RES\}_n\right)}$$

$$\boxed{[JAC]_{n,n+1}^{p} = \mathrm{S}_e\left(\frac{\partial\{FQ\}_e}{\partial\{Q\}_e}\right)_{n,n+1}^{p}}$$

Thus, to advance to a new time level, a guess must be made for the new value of $\{Q\}_{n+1}$. Newton will then iterate this solution towards a converged value of $\{Q\}_{n+1}$ using the previous values of $\{Q\}_n$ to form $\{RES\}_n$ and $\{\delta Q\}_{n,n+1}$.

The plan of attack is thus

1. Have an old/previous time value of $\{Q\}_n$
2. Guess a new time value of $\{Q\}_{n+1}^{p}$

   ITERATION LOOP

   a. Form $\{FQ\}_{n,n+1}^{p}$ using the values of $\{Q\}_n$ and $\{Q\}_{n+1}^{p}$

   b. Form $[JAC]_{n,n+1}^{p}$ using the values of $\{Q\}_n$ and $\{Q\}_{n+1}^{p}$

   c. Solve for $\{\delta Q\}_{n+1}^{p+1}$

   d. Get new iterate value of the new time value of $\{Q\}_{n+1}^{p+1} = \{Q\}_{n+1}^{p} + \{\delta Q\}_{n+1}^{p+1}$

   e. Recover new iterate value of the new time value of $\{\Delta Q\}_{n,n+1}^{p+1} = \{Q\}_{n+1}^{p+1} - \{Q\}_n$

   f. Repeat until $\max(\left|\{\delta Q\}_{n+1}^{p+1}\right|)$ is within the convergence criteria giving the final new time value of $\{Q\}_{n+1}$

3. Go to the next time step and repeat

HOMEWORK:

For transient heat transfer with no source and Dirichlet boundary conditions, apply GWS + ΘTS, hence verify (10). Then apply Newton and verify the jacobian and residual terms in the attached code.

```
Q_n = T_ini;                    % Initialize Qn with IC data
Q_np1 = T_ini;                  % Initialize Qnp1 with guess of IC data
DELTA_Q = Q_np1 - Q_n;          % Form temporal change in Q
delta_Q = pi*ones(nnodes,1);    % Initialize delta Q
num_its = 1;

%OUTER TIME STEPPING LOOP
for t_loop=1:nsteps
   %INNER ITERATION LOOP
   while ((max(abs(delta_Q))>newt_crit)&(num_its <= max_its))

      % Form temporal MASS contribution to total residual FQp using DELTA_Q
            RES_MASS = asres1D([],[],[],1,A200L,DELTA_Q);
      % Form spatial residual contributions to total residual FQp
            RES_np1 = asres1d(alpha,[],[],-1,A211L,Q_np1);
            RES_n = asres1d(alpha,[],[],-1,A211L,Q_n);

      % Form total residual term FQp
            FQp = RES_MASS + DELTA_T*(theta*RES_np1 + (1-theta)*RES_n);

      %*******************************************************************
      % Form temporal MASS contribution to jacobian JACp
            JAC_MASS =  asjac1d([],[],[],1,A200L,[]);
      % Form spatial residual contributions to jacobian JACp
            JAC_np1 =   asjac1d(alpha,[],[],-1,A211L,[]);

      % Form total jacobian term JACp
      JACp = JAC_MASS + DELTA_T*theta* JAC_np1;

      %*******************************************************************
      % Modify JACp for dirichlet data
      JACp(1,:) = zeros(1,nnodes);
      JACp(1,1) = 1;
      JACp(nnodes,:) = zeros(1,nnodes);
      JACp(nnodes,nnodes) = 1;

      % Modify FQp for Dirichlet data
      FQp(1,1) = 0;
      FQp(nnodes,1) = 0;

      % Solve for incremental change in Q_np1
      delta_Q = JACp \ -FQp;

      % Update Q_np1 with incremental change
      Q_np1 = Q_np1 + delta_Q;

      % Recover new value of DELTA_Q_np1
      DELTA_Q_np1 = Q_np1 - Q_n;

      % increment numits counter
      num_its = num_its + 1;
   end
   % Moving to new time station, current Q_np1 now becomes the previous Q_n
   Q_n = Q_np1;
   DELTA_Q = Q_np1 - Q_n;          % Form temporal change in Q
   delta_Q = pi*ones(nnodes,1);    % Initialize delta Q
   num_its = 1;
end
```