Computer Implementation 8.1 *(Matlab)* *Transient heat flow example (p. 552)*

A transient analysis can be implemented in *Matlab* by a simple extension of the functions presented in earlier chapters. In this section the triangular element functions presented in Chapter 5 are extended to handle transient analysis of two dimensional boundary value problems.

MatlabFiles\Chap8\TransientBVPTriElement.m

```
function [km, ke, rq] = TransientBVPTriElement(kx, ky, p, q, m, coord)
% [km, ke, rq] = TransientBVPTriElement(kx, ky, p, q, m, coord)
% Generates for a triangular element for 2d BVP
% kx, ky, p, q, m = parameters defining the BVP
% coord = coordinates at the element ends

x1=coord(1,1); y1=coord(1,2);
x2=coord(2,1); y2=coord(2,2);
x3=coord(3,1); y3=coord(3,2);
b1 = y2 - y3; b2 = y3 - y1; b3 = y1 - y2;
c1 = x3 - x2; c2 = x1 - x3; c3 = x2 - x1;
f1 = x2*y3 - x3*y2; f2 = x3*y1 - x1*y3; f3 = x1*y2 - x2*y1;
A = (f1 + f2 + f3)/2;
kxx = 1/(4*A)*kx*[b1^2, b1*b2, b1*b3;
   b1*b2, b2^2, b2*b3; b1*b3, b2*b3, b3^2];
kyy = 1/(4*A)*ky*[c1^2, c1*c2, c1*c3;
   c1*c2, c2^2, c2*c3; c1*c3, c2*c3, c3^2];
kp = -(p*A/12)*[2, 1, 1; 1, 2, 1; 1, 1, 2];
ke = kxx + kyy + kp;
rq = 1/3*q*A*[1; 1; 1];
km = m*A/12 * [2,1,1; 1,2,1; 1,1,2];
```

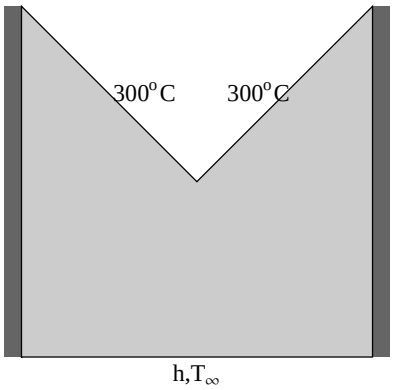
MatlabFiles\Chap8\BVPTriNBCTerm.m

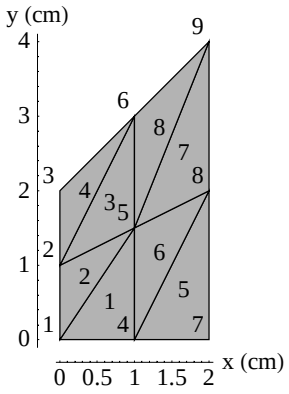
```
function [ka, rb] = BVPTriNBCTerm(side, alpha, beta, coord)
% [ka, rb] = BVPTriNBCTerm(side, alpha, beta, coord)
% Generates kalpha and rbeta when NBC is specified along a side
% side = side over which the NBC is specified
% alpha and beta = coefficients specifying the NBC
% coord = coordinates at the element ends

x1=coord(1,1); y1=coord(1,2);
x2=coord(2,1); y2=coord(2,2);
x3=coord(3,1); y3=coord(3,2);
switch (side)
case 1
   L = sqrt((x2-x1)^2+(y2-y1)^2);
```

```
    ka = -alpha*L/6 * [2,1,0; 1,2,0; 0,0,0];
    rb = beta *L/2 * [1; 1; 0];
case 2
    L = sqrt((x2-x3)^2+(y2-y3)^2);
    ka = -alpha*L/6 * [0,0,0; 0,2,1; 0,1,2];
    rb = beta *L/2 * [0; 1; 1];
case 3
    L = sqrt((x3-x1)^2+(y3-y1)^2);
    ka = -alpha*L/6 * [2,0,1; 0,0,0; 1,0,2];
    rb = beta *L/2 * [1; 0; 1];
end
```

Using these functions now we consider solution of the transient heat flow problem using an 8 element model. We assemble $\boldsymbol{M}$, $\boldsymbol{K}$, matrices and $\boldsymbol{R}$ vector from the corresponding element quantities in the usual manner. Finally the equations are expressed in a form suitable for the differential equation solver (ode23) in *Matlab.*





MatlabFiles\Chap8\HeatODE.m

```
% % Transient heat flow through V-groove
global Mf Kf Rf
h = 200; Tinf = 50; rho = 1600; cp = 800; t0=300;
nodes = (1/100)* [0, 0; 0, 1; 0, 2;
   1, 0; 1, 3/2; 1, 3; 2, 0; 2, 2; 2, 4];
kx = 3; ky = 3; p = 0; q = 0;
lmm = [1, 4, 5; 5, 2, 1; 2, 5, 6;
   6, 3, 2; 4, 7, 8; 8, 5, 4; 5, 8, 9; 9, 6, 5];
debc = [3,6,9]; ebcVals=t0*ones(length(debc),1);
dof=length(nodes); elems=size(lmm,1);
M=zeros(dof); K=zeros(dof); R = zeros(dof,1);
```

```
% Generate equations for each element and assemble them.
for i=1:elems
   lm = lmm(i,:);
   [m, k, r] = TransientBVPTriElement(kx, ky, p, q, rho*cp, nodes(lm,:));
   M(lm, lm) = M(lm, lm) + m;
   K(lm, lm) = K(lm, lm) + k;
   R(lm) = R(lm) + r;
end

% Compute and assemble NBC contributions
lm = lmm(1,:);
[k, r] = BVPTriNBCTerm(1, -h, h*Tinf, nodes(lm,:));
K(lm, lm) = K(lm, lm) + k;
R(lm) = R(lm) + r;

lm = lmm(5,:);
[k, r] = BVPTriNBCTerm(1, -h, h*Tinf, nodes(lm,:));
K(lm, lm) = K(lm, lm) + k;
R(lm) = R(lm) + r;
% Adjust for essential boundary conditions
dof = length(R);
df = setdiff(1:dof, debc);
Mf = M(df, df);
Kf = K(df, df);
Rf = R(df) - K(df, debc)*ebcVals;

% Setup and solve the resulting first order differential equations
d0 = t0*ones(length(Mf),1);
[t,d] = ode23('HeatODE',[0,300],d0);
plot(t,d(:,1),'-r',t,d(:,2),'-g',t,d(:,3),...
   '-c',t,d(:,4),'-m',t,d(:,5),'-y',t,d(:,6),'-k');
legend('T1', 'T2', 'T4', 'T5', 'T7', 'T8');
hold off
[t(1:10),d(1:10,:)]
```

MatlabFiles\Chap8\HeatODE.m

```
function ddot = HeatODE(t, d)
% ddot = HeatODE(t, d)
% function to set up equations for a transient heat flow problem
global Mf Kf Rf
ddot = inv(Mf)*(Rf - Kf*d);
```

```
>> VGrooveHeatEx

ans =
```

Columns 1 through 6

0	300	300	300	300	300
2.5083	282.52	300.31	281.59	303.27	278.91
6.2557	262.96	298.19	260	305.61	255.9
11.6	243.36	293.48	238.45	304.89	233.4
17.131	229.23	288.35	223.63	301.15	217.43
20.502	222.87	284.77	216.46	298.54	210.71
23.873	217.31	281.67	210.73	295.47	204.67
30.256	208.96	275.82	201.64	289.87	195.93
33.809	204.96	273.13	197.74	286.61	191.56
37.363	201.57	270.28	194	283.73	188.09

Column 7

300
307.69
313.87
316.24
314.47
312.01
309.33
303.65
300.6
297.47