Computer Assignment 4.1.1
MMP Finite Element Methods (WI4243AP-FE)

Shaikh Bechan 4146425
Zubin Ramlakhan 4170229

1 Introduction

The following is a report containing the evaluation of the heat equation with the finite element method. To solve the two-dimensional heat equation numerically, the finite element method is applied. To that end the minimization problem corresponding to the partial differential in question is derived. The solution of the minimization problem is approximated with the finite element method by deriving the element stiffness matrix and element stiffness vector using triangular and line elements.
The code used in the numerical solution can be found in the Appendix. Please note the theorems and notation are referenced to Numerical Methods in Scientific Computing by J. van Kan, A. Segal, F. Vermolen.

2 The Heat Equation

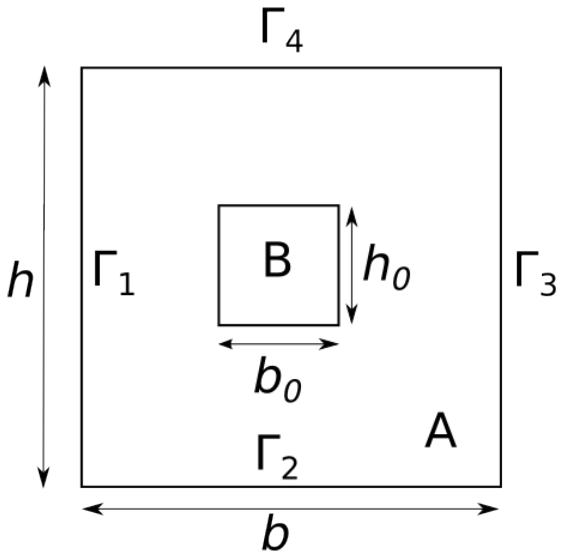


Figure 1: Rectangular plate made of two materials, A and B.

Consider the rectangular plate shown in figure 1, with width b and height h. In the center of the plate is a rectangular part, of a different material, with width, b_0 and height, h_0. A denotes the outer part of the plate while B denotes the inner part.

The situation is stationary and the effects of convection and sources/sinks are absent. Thus the heat equation is simply defined by the conduction (diffusion) term as shown in (1)

$$-\nabla \cdot (k\nabla T) = 0 \tag{1}$$

with heat conductivity k and temperature T. The thermal conductivity is different in each material, i.e. $k|_A = k_A$ and $k|_B = k_B$.

The boundary of A, ∂A, is divided in four segments such that $\partial A = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4$. On these boundaries the following conditions hold:

$$\begin{aligned} T|_{\Gamma_1 \cup \Gamma_3} &= T_0 & (2)\\ \nabla T|_{\Gamma_2} &= 0 & (3)\\ k\frac{\partial T}{\partial n}|_{\Gamma_4} &= -\,\alpha(T_w - T_\infty) & (4) \end{aligned}$$

In words, the temperature at boundaries Γ_1 and Γ_3 are held constant at $T = T_0$. The flux at Γ_2 is zero. Γ_4 obeys Newton's heat transfer relation with the thermal diffusivity α, the environment temperature T_∞ and the temperature at the wall T_w.

In this problem a symmetry plane is present. This plane runs along the midsection of the plate in the vertical direction. Here, to ensure symmetry, the gradient of the temperature must be zero.

3 Minimization problem

A partial differential equation can be related to an equivalent minimization problem with the following theorem:

Theorem 1 *Let L be a linear, symmetric, positive differential operator defined of space Σ and let*

$$Lu = f \tag{5}$$

Then the solution u minimizes the functional

$$I(u) = \int_\Omega \{\frac{1}{2}uLu - uf\}d\Omega \tag{6}$$

over the space Σ. On the other hand, if u minimizes (6) *then u satisfies* (5)

Another requirement u should obey is homogeneous boundary conditions, a requirement T fails. Even so, this theorem can still be utilized by making the boundary conditions homogeneous.

Let $Lu = f$ with non-homogeneous boundary equations and suppose that there is another smooth function w that also satisfies the same non-homogeneous boundary conditions. Now let

$$v = T - w \tag{7}$$

T and w satisfy the same non-homogeneous boundary conditions, therefore v satisfies homogeneous boundary conditions. Now the theorem can be applied to v as it meets all the requirements.

$$Lv = LT - Lw = f - Lw \tag{8}$$

so the corresponding minimization problem is given by

$$\underset{v \in \Sigma}{min} I(v) = \frac{1}{2}\int_{\Omega} vLvd\Omega - \int_{\Omega} v(f - Lw)d\Omega \tag{9}$$

Substituting (7) into (9) yields

$$\underset{v \in \Sigma}{min} I(v) = \frac{1}{2}\int_{\Omega} (T-w)L(T-w)d\Omega - \int_{\Omega} (T-w)(f - Lw)d\Omega \tag{10}$$

$f = 0$ and $L = -\nabla \cdot (k\nabla)$ according to (1), so (10) is rewritten as

$$\begin{aligned}
\underset{v \in \Sigma}{min} I(v) &= -\frac{1}{2}\int_{\Omega} (T-w)\nabla \cdot (k\nabla(T+w))d\Omega \\
&= \frac{1}{2}\int_{\Omega} \nabla(T-w)(k\nabla(T+w)) - \nabla \cdot \{(T-w)(k\nabla(T+w))\}d\Omega \\
&= \frac{1}{2}\int_{\Omega} \nabla(T-w)(k\nabla(T+w))d\Omega - \frac{1}{2}\int_{\Gamma} (T-w)(k\nabla(T+w)) \cdot \mathbf{\hat{n}}d\Gamma \\
&= \frac{1}{2}\int_{\Omega} \nabla T \cdot k\nabla T d\Omega - \frac{1}{2}\int_{\Omega} \nabla w \cdot k\nabla w d\Omega \\
&- \frac{1}{2}\int_{\Gamma} (T-w)(k\nabla(T+w)) \cdot \mathbf{\hat{n}}d\Gamma
\end{aligned} \tag{11}$$

where the product rule and subsequent Gauss theorem is applied to expand the integral in two volume integrals and one boundary integral. The second integral is independent of T and thus has no influence on the minimization and can be neglected.

The boundary integral is evaluated over four boundaries by taking in account the boundary conditions (2)−(4).

$$\frac{1}{2}\int\limits_{\Gamma_1\cup\Gamma_3}(T_0-T_0)(k\nabla(T+w))\cdot\mathbf{\hat{n}}d\Gamma=0 \tag{12}$$

$$\frac{1}{2}\int\limits_{\Gamma_2}(T-w)(k(0+0))\cdot\mathbf{\hat{n}}d\Gamma=0 \tag{13}$$

$$\begin{aligned}\frac{1}{2}\int\limits_{\Gamma_4}(T-w)(k\nabla(T+w))\cdot\mathbf{\hat{n}}d\Gamma &= \frac{1}{2}\int\limits_{\Gamma_4}Tk\frac{\partial T}{\partial n}d\Gamma-\frac{1}{2}\int\limits_{\Gamma_4}wk\frac{\partial w}{\partial n}d\Gamma\\ &= -\frac{1}{2}\int\limits_{\Gamma_4}\alpha T(T_w-T_\infty)d\Gamma\end{aligned} \tag{14}$$

where $\frac{1}{2}\int\limits_{\Gamma_4}wk\frac{\partial w}{\partial n}d\Gamma$ in (14) is neglected as it too is independent of T.

Finally, the general minimization problem is given by

$$\underset{T\in\Sigma}{min}I(v)=\frac{1}{2}\int\limits_{\Omega}k|\nabla T|^2d\Omega+\frac{1}{2}\int\limits_{\Gamma_4}\alpha T(T_w-T_\infty)d\Gamma \tag{15}$$

with $\Sigma:\{T|\quad T|_{\Gamma_1\cup\Gamma_3}=T_0\}$. The boundary conditions (2) and (3) are essential boundary conditions.

4 Element stiffness matrix and element stiffness vector

This problem can be solved by computing the element stiffness matrix and the element stiffness vector. To do this, the solution is approximated with piecewise linear functions $\phi_j(x,y)$ such that

$$T(x,y)\simeq T^n(x,y)=\sum_{j=1}^{n}c_j\phi_j(x,y) \tag{16}$$

One might derive the matrix and vector by substituting this approximation into the minimization problem $\underset{T\in\Sigma}{min}I(v)$ and then set the derivative of c_j to zero. This is called the Ritz method.

The second method is by deriving the Galerkin equations. To that end the partial differential equation is multiplied with a test function ϕ to derive these equations. Afterwards the approximation of the solution is substituted into the weak form which then results in the element stiffness matrix and element stiffness vector.

Both methods yield the same results. The following shows how the matrix and vector are acquired using Galerkin's method. First, the Galerkin equations:

$$
\begin{aligned}
-\int\limits_{\Omega} \phi \nabla \cdot (k \nabla T) d\Omega &= 0 \\
-\int\limits_{\Omega} \nabla \cdot (\phi k \nabla T) - \nabla \phi k \nabla T d\Omega &= 0 \\
-\int\limits_{\Gamma} \phi k \frac{\partial T}{\partial n} d\Gamma + \int\limits_{\Omega} \nabla \phi k \nabla T d\Omega &= 0 \\
\int\limits_{\Gamma_4} \phi \alpha \left(T_w - T_\infty \right) d\Gamma + \int\limits_{\Omega} \nabla \phi k \nabla T d\Omega &= 0 \qquad (17)
\end{aligned}
$$

where the boundary integral in the penultimate line is split up over the four boundaries like in (12) - (14) yielding (17) as a result.

The weak form is thus formulated as follows:

$$\int\limits_{\Gamma_4} \phi \alpha \left(T_w - T_\infty \right) d\Gamma + \int\limits_{\Omega} \nabla \phi k \nabla T d\Omega = 0$$

Find $T \in \Sigma\{$ *T sufficiently smooth* $|T_{\Gamma_1 \cup \Gamma_3} = T_0\}, \forall \phi \in \Sigma$

Now the approximation (16) is applied. ϕ is approximated in the same way i.e.

$$\phi \simeq \sum_{i=1}^{n} b_i \phi_i = \phi_i, i = 1 \ldots n$$

As ϕ is chosen arbitrarily, it is useful to make one of the of the coefficients b_i equal to one and the others to zero.

Substituting these approximations into the weak form results in

$$
\begin{aligned}
&\int\limits_{\Gamma_4} \alpha \phi_i \left(\sum_{j=1}^{n} c_j \phi_j - T_\infty \right) d\Gamma + \int\limits_{\Omega} \nabla \phi_i k \nabla \sum_{j=1}^{n} c_j \phi_j d\Omega = 0 \\
&\sum_{j=1}^{n} c_j \cdot \alpha \int\limits_{\Gamma_4} \phi_i \phi_j d\Gamma + \sum_{j=1}^{n} c_j \cdot \int\limits_{\Omega} \nabla \phi_i k \nabla \phi_j d\Omega = \sum_{j=1}^{n} c_j \cdot \alpha \int\limits_{\Gamma_4} \phi_i T_\infty d\Gamma \qquad (18)
\end{aligned}
$$

$$(i = 1 \ldots n)$$

(18) can be written more compactly as

$$\sum_{j=1}^{n} c_j S_{ij} = f_i$$

with the element stiffness vector

$$f_i = \alpha \int\limits_{\Gamma_4} \phi_i T_\infty d\Gamma$$

and the element stiffness matrix that consists of a boundary matrix

$$S_{ij}^B = \alpha \int\limits_{\Gamma_4} \phi_i \phi_j d\Gamma$$

and a volume matrix

$$S_{ij}^V = \int\limits_{\Omega} k \nabla \phi_i \nabla \phi_j d\Omega$$

The matrices and vector need to be evaluated and are done so using the theorem of Holland & Bell .

Theorem 2 Holland & Bell *Let S be a simplex in $\mathbb{R}$ and let Δ be the determinant defined by*

$$\Delta = \begin{vmatrix} 1 & x_0^1 & x_2^1 & \dots & x_n^1 \\ 1 & x_0^2 & x_2^2 & \dots & x_n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_0^{n+1} & x_2^{n+1} & \dots & x_n^{n+1} \end{vmatrix}$$

with $\mathbf{x}^1$, $\mathbf{x}^2$, $\cdots$ $\mathbf{x}^{n+1}$ the vertices of S.

Let $\lambda_i(\mathbf{x})$ be the linear basis functions over S defined by

$$\begin{array}{rl} \lambda_i(\mathbf{x}) & \textit{linear} \\ \lambda_i(\mathbf{x}^j) = \delta_{ij} & i, j = 1, 2 \cdots n+1 \end{array}$$

Then the following general integration rule holds:

$$\int\limits_S \lambda_1^{m_1} \lambda_2^{m_2} \cdots \lambda_{n+1}^{m_{n+1}} d\Omega = \frac{m_1! m_2! \cdots m_{n+1}!}{\left(\sum\limits_i m_i + n\right)!} |\Delta|,$$

for all $m_i \geq 0$.

$\sum\limits_i m_i$ represents the dimensionality of the elemeng i.e. 2 for triangular elements and 1 for line elements.

The set of functions ϕ_i are piecewise linear functions and therefore can be written as

$$\phi_i = \alpha_i + \beta_i x + \gamma_i y$$

The determinant of these basis functions are equal to

$$\Delta = \begin{vmatrix} 1 & x^1 & y^1 \\ 1 & x^2 & y^2 \\ 1 & x^3 & y^3 \end{vmatrix}$$

which is equal to twice the area of a triangular element.

According to Holland & Bell, f_i is evaluated as follows:

$$\begin{aligned} f_i &= \alpha T_\infty \int\limits_{\Gamma_4} \phi_i d\Gamma \\ &= \alpha T_\infty \|\mathbf{x}_{l1} - \mathbf{x}_{l2}\| \frac{1!}{(1+1)!} \\ &= \frac{1}{2} \alpha T_\infty \|\mathbf{x}_{l1} - \mathbf{x}_{l2}\| \end{aligned} \tag{19}$$

In this case, the integral is a boundary integral. Therefore, the elements along that boundary are one dimensional line elements with dimensionality 1. Hence instead of a determinant, the length $\|\mathbf{x}_{l1} - \mathbf{x}_{l2}\|$ of one element is needed.

S_{ij}^B is evaluated in almost the same way. The difference is that a distinction between the cases $i = j$ and $i \neq j$ is need as seen in (20)

$$S_{ij}^B = \alpha \int\limits_{\Gamma_4} \phi_i \phi_j d\Gamma = \alpha \|\mathbf{x}_{l1} - \mathbf{x}_{l2}\| \cdot \begin{cases} \frac{2!0!}{(1+2+0)!} = \frac{1}{3} & i = j \\ \frac{1!1!}{(1+1+1)!} = \frac{1}{6} & i \neq j \end{cases} \tag{20}$$

Before **theorem 2** is applied to S_{ij}^V the gradients of the basis functions are rewritten as

$$\nabla \phi_i \nabla \phi_j = (\beta_i \beta_j + \gamma_i \gamma_j)$$

This value is a constant and can be put in front of the integral. S_{ij}^V is equal to

$$S_{ij}^V = k\left(\beta_i \beta_j + \gamma_i \gamma_j\right) \int\limits_{\Omega} d\Omega = k\left(\beta_i \beta_j + \gamma_i \gamma_j\right) \frac{\|\Delta\|}{2} \tag{21}$$

The integral is essential the surface of the triangular element which is equal to half the determinant.
The plate can have two values for k depending on the elements being on area A or area B. Therefore S_{ij}^V must change accordingly i.e. k must assume the value corresponding to the material where the elements are situated.

5 Computation

5.1 Mesh

Before computing the element stiffness matrix and vector, a mesh has to be created. This mesh consists of a grid of triangular elements along the surface of the plates and line elements on the boundaries.

The data specific to the problem is given in table 1.
The problem is symmetrical along the y-axis, therefore the computational effort can be reduced by generating a mesh that covers half the domain. Such a mesh can be seen in the figure 2

T_0	$=$	1	h_0	$=$	0.4
k_A	$=$	0.1	h	$=$	1
k_B	$=$	1			
α	$=$	0.01	b_0	$=$	0.4
T_∞	$=$	0	b	$=$	1

Table 1: Dimensions of the plate and other parameters.

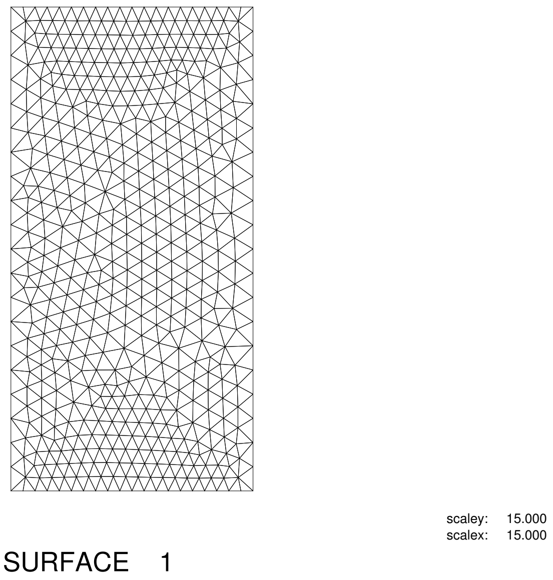


Figure 2: Only half of the domain of the plate is computed due to symmetry. Here the right hand side of the plate along with a number of elements can be seen.

5.2 Results

Due to the boundary conditions, the solution is expected to portray some sort of cooling at the top of the plate. The sides remain the same temperature, while the insulation at the bottom requires the isotherms to cross the plate perpendicular.

Figure 3 displays the problem's solution with increasing numbers of elements and at $\alpha = 0.01$ in the form of contour plots and color plots. The solution indeed behaves as predicted. The color bars next to the color plots confirm that the temperature of the plate never exceeds the temperature of the side and is positive everywhere, ruling out nonphysical behavior. The temperature is continuous is also across the interface of the two plates as expected. Also noticeable is the effect of the increase of elements. A rough mesh can lead to some crooked behavior in the isotherms. A finer mesh has a smoothing effect on this crookedness. Looking at the results, a mesh of 50 elements horizontal and vertical seems to suffice at yielding a trustworthy numerical solution to the heat equation.

The same problem is redone for a different value of α i.e. $\alpha = 10$, the result of which can be seen in figure 4. The overall profile of the isotherms remains the same, but the crucial difference is the fact that, due to the larger value of α, the cooling effect is much larger. That can be noticed from the fact that the cooling effect has penetrated the plate much further when figure 4 is compared to figure 3.

Again, the increase in number of elements has a smoothing effect, but like with the lower value for α, around 50 elements in both directions suffices.

6 Appendix

The following is the exact code used to create the mesh in mesh.msh, define the problem in mesh.prb and define the element stiffness matrices and vector in elemsubr.f90.

```
# mesh.msh
#
constants
reals
height = 1  # height of the square
width = 0.5 # half width of the square



integers
nelm_hor = 50           # number elements horizontal
nelm_vert = 50          # number elements vertical
shape_cur = 1           #
shape_sur = 3           #
end
#

#       Actual definition of the mesh
#
mesh2d
# Definition of the coordinates
points
p1=(0,0)
p2=(width,0)
p3=(width,height)
p4=(0,height)

#       Definition of the curves
curves
# Boundary of square
c1 = line shape_cur ( p1,p2,nelm= nelm_hor )
c2 = line shape_cur ( p2,p3,nelm= nelm_vert )
c3 = line shape_cur ( p3,p4,nelm= nelm_hor )
c4 = line shape_cur ( p4,p1,nelm= nelm_vert )
```

```
#        Definition of the surface

surfaces
s1 = triangle shape_sur (c1, c2, c3, c4 ) !B

plot
end
```

```
#  File: mesh.prb

#
#
#
#
problem
    types
        elgrp1 = (type=1)

        natboundcond
        bngrp1 = (type = 2)

        bounelements
        belm1 = curves(c3)

        essboundcond
        curves(c2)

end
#
# Define the structure of the main program, see Section
   4.7.9
#
structure
    matrix_structure: symmetric          # a symmetric
       profile matrix is used
    prescribe_boundary_conditions mesh = 1, curves(c2)
    solve_linear_system , mesh
    plot_contour text = 'Temperature a = 0.01'
    plot_coloured_levels mesh, text = 'Temperature a =
       0.01'

end
end_of_sepran_input
```

```
! File: elemsubr.f90
      subroutine elemsubr ( npelm, x, y, nunk_pel,
         elem_mat, &
```

```
                                   elem_vec , elem_mass ,
                                      prevsolution , itype )
!
!                               INPUT / OUTPUT PARAMETERS
!
      implicit none
      integer , intent(in) :: npelm, nunk_pel, itype
      double precision , intent(in) :: x(1:npelm), y(1:
         npelm), &
                                             prevsolution(1:
                                                nunk_pel)
      double precision , intent(out) :: elem_mat(1:
         nunk_pel,1:nunk_pel), &
                                              elem_vec(1:
                                                 nunk_pel), &
                                              elem_mass(1:
                                                 nunk_pel)

!
!!
!
!
!for example:
integer :: i, j
double precision :: beta(1:3), gamma(1:3), delta, h
real(8), parameter :: a = 0.01
real(8) :: k


delta = (x(2)-x(1))*(y(3)-y(1))-(y(2)-y(1))*(x(3)-x(1))
h = sqrt ( (x(2)-x(1))**2 + (y(2)-y(1))**2 )
beta(1) = (y(2)-y(3))/delta
beta(2) = (y(3)-y(1))/delta
beta(3) = (y(1)-y(2))/delta
gamma(1) = (x(3)-x(2))/delta
gamma(2) = (x(1)-x(3))/delta
gamma(3) = (x(2)-x(1))/delta

if ( itype==1 ) then
! statements to fill the arrays elem_mat and elem_vec
if (((x(1) < 0.2) .and. (y(1) < 0.7) .and. (y(1) > 0.3))
   .or. &
((x(2) < 0.2) .and. (y(2) < 0.7) .and. (y(2) > 0.3)) .or.
   &
((x(3) < 0.2) .and. (y(3) < 0.7) .and. (y(3) > 0.3)))
   then

k = 1d0

else
```

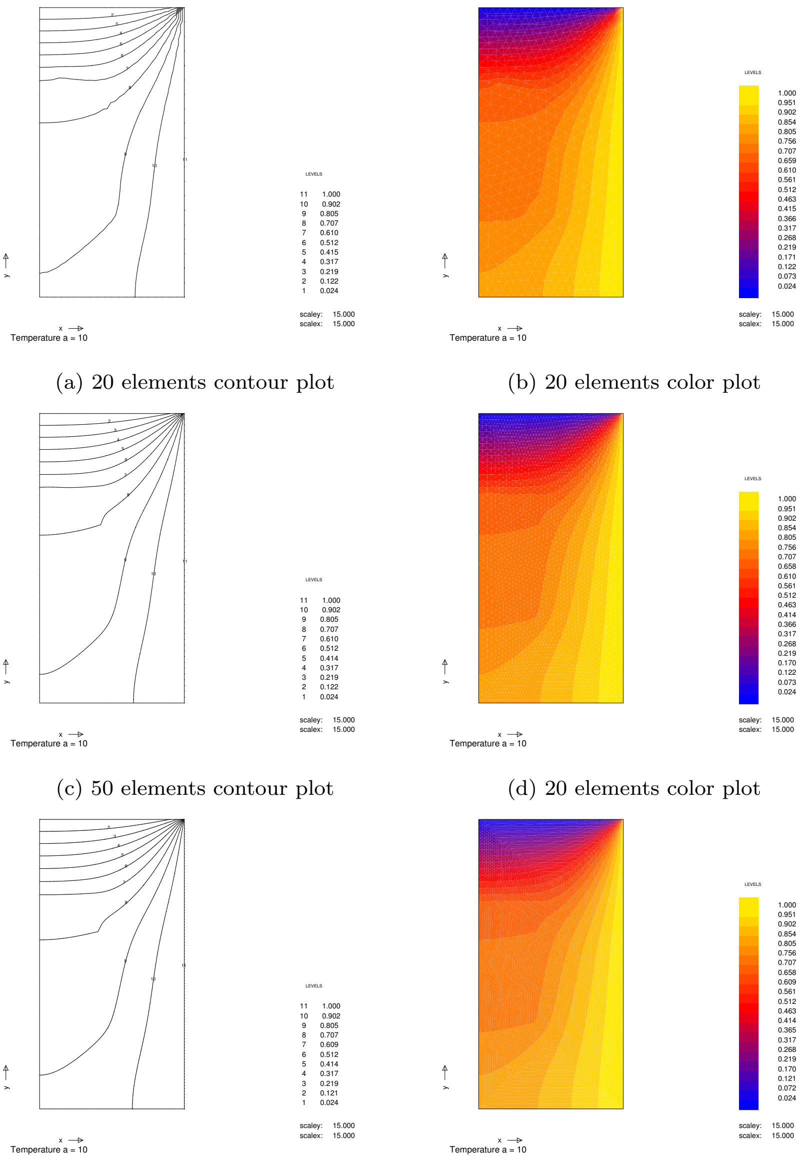
```
k = 0.1d0

endif
do j = 1, nunk_pel
do i = 1, nunk_pel
elem_mat(i,j) = 0.5d0 * k * abs(delta) * ( beta(i)*beta(j
    ) + gamma(i)*gamma(j) )
end do
elem_vec(j) = 0d0
end do

else if ( itype==2 ) then
!the same type of statements for itype = 2, etcetera
do j = 1, nunk_pel
do i = 1, nunk_pel
if ( i==j ) then
elem_mat(i,j) = h*a/3d0
else
elem_mat(i,j) = h*a/6d0
end if
end do
elem_vec(j) = 0d0
end do

end if

      end subroutine elemsubr
```



(a) 20 elements contour plot

(b) 20 elements color plot

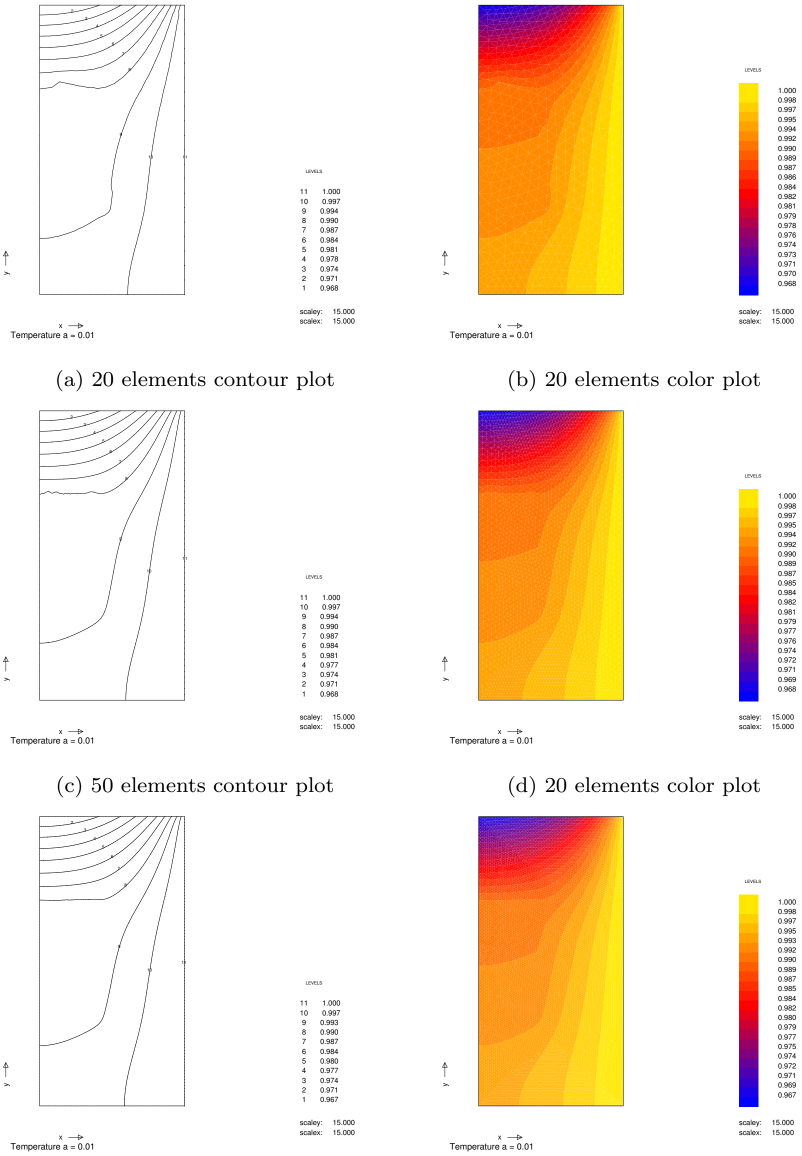
(c) 50 elements contour plot

(d) 20 elements color plot

(e) 99 elements contour plot

(f) 99 elements color plot

Figure 3: solution of the heat equation for a number of elements with $\alpha = 0.01$



(a) 20 elements contour plot

(b) 20 elements color plot

(c) 50 elements contour plot

(d) 20 elements color plot

(e) 99 elements contour plot

(f) 99 elements color plot

Figure 4: solution of the heat equation for a number of elements with $\alpha = 0.01$