

*Introduction to Finite Element  
Modelling in Geosciences:*  
**Diffusion Equation: From 1D to 2D**

D. A. May & M. Frehner  
ETH Zürich

July 9, 2014

## 1 Overview

One of the strengths of the finite element method is the relative ease with which it is possible to extend your code from one dimension to two (or three) dimensions. To demonstrate the finite element method in two-dimensions, we once again use the example of the diffusion equation. Remember that the diffusion equation is derived by combining a statement for the conservation of energy

$$\rho c_p \frac{\partial T}{\partial t} = -\nabla^T \mathbf{q} + s = - \begin{bmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix} \begin{bmatrix} q_x \\ q_y \end{bmatrix} + s, \quad (1)$$

with a general anisotropic constitutive relationship

$$\mathbf{q} = \begin{bmatrix} q_x \\ q_y \end{bmatrix} = - \begin{bmatrix} k_{xx} & k_{xy} \\ k_{yx} & k_{yy} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} T = -\mathbf{D}\nabla T, \quad (2)$$

to yield

$$\rho c_p \frac{\partial T}{\partial t} = \nabla^T (\mathbf{D}\nabla T) + s. \quad (3)$$

This equation governs transient heat conduction in two-dimensions with a source term  $s(x, y)$ . The dependent variable in this equation is the temperature  $T$ , the independent variables are time  $t$  and distance  $x$ , and  $k_{ij}(x, y)$  are the components of the thermal conductivity tensor,  $\rho(x, y)$  is density and  $c_p(x, y)$  heat capacity. If the thermal conductivity is isotropic and constant in space, Equation (3) in 2D reduces to

$$\frac{\partial T}{\partial t} = \kappa \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + s, \quad (4)$$

where  $\kappa = k/(\rho c_p)$  is the thermal diffusivity. For the isotropic case, we have that

$$\mathbf{D} = \begin{bmatrix} \kappa & 0 \\ 0 & \kappa \end{bmatrix}. \quad (5)$$

Nevertheless, as we will see, it is more natural and general to leave the equation in the form shown in Equation (3). In fact one of the benefits of the finite element method is that introducing an anisotropic diffusivity tensor does not introduce any additional complexity to the discretisation procedure.

## 2 Derivation of the element matrices

As in our one dimensional example, the first step of the finite element method is to choose an element type. We choose here one of the simplest two-dimensional elements which is the 4-node quadrilateral (see Figure 1) element. The four shape functions  $N_i(x, y)$  which go together with this type of element are polynomials of the form

$$N_i(x, y) = a_i + b_i x + c_i y + d_i xy,$$

where  $a_i, b_i, c_i, d_i$  are coefficients which relate to each node  $i$ . This element appears to be linear, but due to the presence of the cross term  $xy$  the quadrilateral basis functions in 2D are referred to as bilinear. For the moment we shall simply refer to the shape functions using the following notation

$$\mathbf{N}(x, y) = [N_1(x, y) \quad N_2(x, y) \quad N_3(x, y) \quad N_4(x, y)], \quad (6)$$

and we note that the shape functions are expressed in terms of the global coordinate system (i.e., they are a function of  $x$  and  $y$ ). The second step of the finite element method involves approximating the continuous variable  $T$  in Equation (3) in terms of nodal variables  $T_i$  using the shape functions just defined, i.e.,

$$\begin{aligned} T(x, y) &\approx [N_1(x, y) \quad N_2(x, y) \quad N_3(x, y) \quad N_4(x, y)] \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \\ &= \mathbf{N}\mathbf{T}. \end{aligned} \quad (7)$$

Substituting this approximation for  $T$  into Equation (3) yields the following equation

$$\rho c_p \frac{\partial}{\partial t} (\mathbf{N}\mathbf{T}) = \nabla^T (\mathbf{D}\nabla\mathbf{N})\mathbf{T} + s. \quad (8)$$

Multiplying this equation by the weighting functions (the shape functions) and integrating over the element volume  $\Omega^e$  leads to

$$\iint_{\Omega^e} \rho c_p \mathbf{N}^T \frac{\partial}{\partial t} (\mathbf{N}\mathbf{T}) \, dx dy = \iint_{\Omega^e} \mathbf{N}^T \nabla^T (\mathbf{D}\nabla\mathbf{N})\mathbf{T} \, dx dy + \iint_{\Omega^e} \mathbf{N}^T s \, dx dy. \quad (9)$$

Integration by parts over the domain  $\Omega^e$  and denoting the element boundaries lying on the domain boundary via  $\Gamma^e$  gives

$$\begin{aligned} \left[ \iint_{\Omega^e} \rho c_p \mathbf{N}^T \mathbf{N} \, dx dy \right] \left( \frac{\partial \mathbf{T}}{\partial t} \right) + \left[ \iint_{\Omega^e} \nabla(\mathbf{N})^T (\mathbf{D}\nabla\mathbf{N}) \, dx dy \right] \mathbf{T} \\ = \iint_{\Omega^e} \mathbf{N}^T s \, dx dy + \oint_{\Gamma^e} \hat{\mathbf{q}}^T \mathbf{n} \, dS, \end{aligned} \quad (10)$$

where  $\hat{\mathbf{q}}$  is an applied flux and  $\mathbf{n}$  is the outward point normal to the boundary  $\partial\Omega$ . Recall that the boundary integral term corresponds to the Neumann boundary condition. This equation can be simplified to

$$\mathbf{M}\mathbf{M} \left( \frac{\partial \mathbf{T}}{\partial t} \right) + \mathbf{K}\mathbf{M}\mathbf{T} = \mathbf{F}, \quad (11)$$

where

$$\mathbf{MM} = \iint_{\Omega^e} \rho c_p \mathbf{N}^T \mathbf{N} \, dx dy, \quad (12)$$

$$\mathbf{KM} = \iint_{\Omega^e} (\nabla \mathbf{N})^T \mathbf{D} \nabla \mathbf{N} \, dx dy \quad (13)$$

and

$$\mathbf{F} = \iint_{\Omega^e} \mathbf{N}^T s \, dx dy + \oint_{\Gamma^e} \hat{\mathbf{q}}^T \mathbf{n} \, dS. \quad (14)$$

Note that we leave  $p, c_p, \mathbf{D}$  and  $s$  inside the integrals as we have not assumed anything about how these coefficients vary within the domain, or within the element.

Note that the element matrices and vectors have a virtually identical form to those derived in session 1 in one dimension. In fact, there are only two significant differences. First, integration domain is now defined in two-dimensions. As we shall see, one can use the same numerical integration as used in the 1D problems. Second, the shape functions have derivatives in both the  $x$  and  $y$  directions.

Before continuing it is useful explicitly define the form (i.e., the dimensions) of some of the matrices defined above. For example, the shape functions  $\mathbf{N}$  are defined as

$$\mathbf{N}(x, y) = [N_1(x, y) \quad N_2(x, y) \quad N_3(x, y) \quad N_4(x, y)], \quad (15)$$

and the shape function derivatives are defined as

$$\nabla \mathbf{N}(x, y) = \begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial x} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \frac{\partial N_3}{\partial y} & \frac{\partial N_4}{\partial y} \end{bmatrix}. \quad (16)$$

The conductivity matrix is defined as <sup>1</sup>:

$$\mathbf{D} = \begin{bmatrix} k_{xx} & k_{xy} \\ k_{yx} & k_{yy} \end{bmatrix}. \quad (17)$$

Thus, one can see that the dimensions of the element matrices  $\mathbf{MM}$  and  $\mathbf{KM}$  are  $4 \times 4$  whereas the element vector  $\mathbf{F}$  has dimensions  $4 \times 1$ .

The next step is to discretise the time derivative in Equation (11) using finite differences. This is carried out in an identical manner to that already performed in 1D (see script 1) leading to

$$\mathbf{MM} \left( \frac{\mathbf{T}^{n+1} - \mathbf{T}^n}{\Delta t} \right) + \mathbf{KMT}^{n+1} = \mathbf{F}, \quad (18)$$

where  $\mathbf{T}^{n+1}$  is the new temperature at the nodes (i.e. the unknowns) and  $\mathbf{T}^n$  is the vector of the old (known) temperatures. Rearranging, one can write this as

$$\left( \frac{\mathbf{MM}}{\Delta t} + \mathbf{KM} \right) \mathbf{T}^{n+1} = \frac{\mathbf{MM}}{\Delta t} \mathbf{T}^n + \mathbf{F}, \quad (19)$$

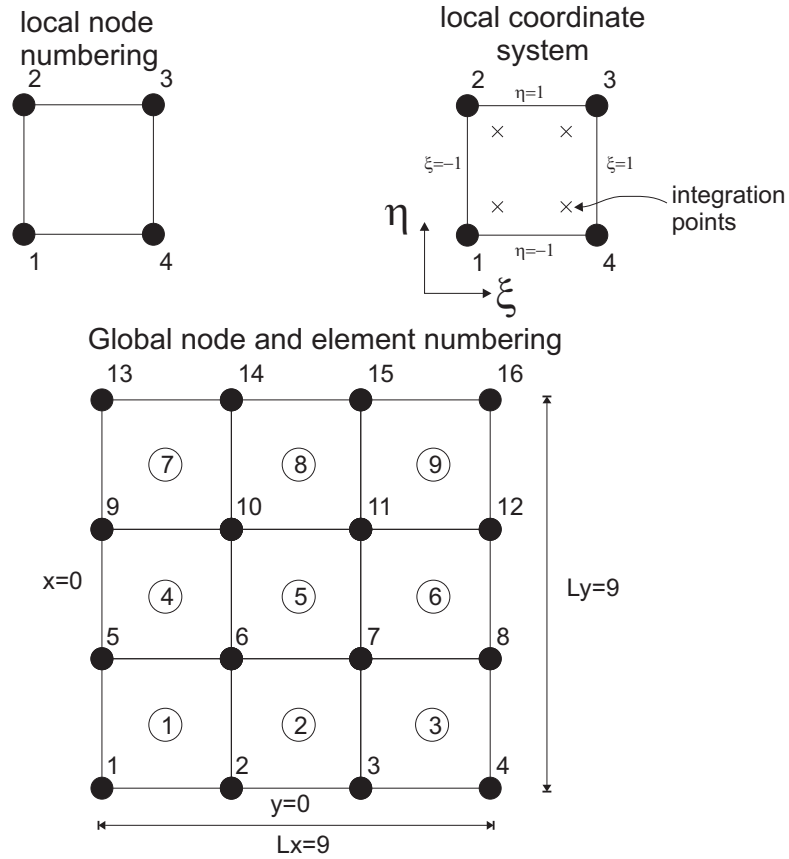
or more compactly as

$$\mathbf{LT}^{n+1} = \mathbf{RT}^n + \mathbf{F}, \quad (20)$$

where  $\mathbf{MM}$ ,  $\mathbf{KM}$  and  $\mathbf{F}$  have been previously defined.

<sup>1</sup>We note that to be physically valid we require that  $k_{xx}k_{yy} - k_{xy}k_{yx} > 0$  and that  $k_{xy} = k_{yx}$ .

2-D FEM MESH - one degree of freedom per node



$$g\_num = \begin{bmatrix} 1 & 2 & 3 & 5 & 6 & 7 & 9 & 10 & 11 \\ 5 & 6 & 7 & 9 & 10 & 11 & 13 & 14 & 15 \\ 6 & 7 & 8 & 10 & 11 & 12 & 14 & 15 & 16 \\ 2 & 3 & 4 & 6 & 7 & 8 & 10 & 11 & 12 \end{bmatrix}$$

Relationship between elements and global node numbers

Global coordinates of nodes

$$g\_coord = \begin{bmatrix} 0.0 & 3.0 & 6.0 & 9.0 & 0.0 & 3.0 & 6.0 & 9.0 & 0.0 & 3.0 & 6.0 & 9.0 & 0.0 & 3.0 & 6.0 & 9.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 3.0 & 3.0 & 3.0 & 3.0 & 6.0 & 6.0 & 6.0 & 6.0 & 9.0 & 9.0 & 9.0 & 9.0 \end{bmatrix}$$

Figure 1: A two-dimensional finite element mesh with one degree of freedom on each node.

### 3 Integration of the element matrices

The element matrices just derived contain integrals of the shape functions, or derivatives of the shape functions, both expressed in terms of global coordinates ( $x$  and  $y$ ). To evaluate such matrices two transformations are required. First, since we will provide shape functions (and their derivatives) in terms of local coordinates, it is necessary to devise some means of expressing the shape functions (and their derivatives) appearing in the integrals in terms of local coordinates. Second, the area over which the integration has to be carried out must be expressed in terms of local coordinates (with an appropriate change in the limits of integration).

As in one dimension, we will use Gauss-Legendre quadrature to approximate the integrals defining the element stiffness matrices. The formula for Gauss-Legendre quadrature in two-dimensions is

$$\int_{-1}^1 \int_{-1}^1 f(\xi, \eta) d\xi d\eta \approx \sum_{i=1}^{nx} \sum_{j=1}^{ny} f(\xi_i, \eta_j) w_i w_j = \sum_{n=1}^{nip} f(\xi_n, \eta_n) W_n \quad (21)$$

where  $nx$  and  $ny$  are number of integration points in each direction,  $\xi_i$  and  $\eta_j$  are the local spatial coordinates of the integration points, and  $w_i$  and  $w_j$ , or  $W_n$  are the weights. The quadrature weights and locations of the integration points are once again obtained from a table such as that provided in script 2. Note that the integration in two dimensions is very similar to the integration in one dimension. One simply sums over the total number of integration points  $nip (= nx \times ny)$  and multiplies by the weights  $W_n$ .

Derivatives are converted from one coordinate system to another by means of the chain rule, expressed in matrix form as

$$\begin{bmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix}, \quad (22)$$

where  $\mathbf{J}$  is the Jacobian matrix. The Jacobian matrix can be found by differentiating the global coordinates with respect to the local coordinates. In practice this requires one to multiply the coordinates of a particular element with the derivatives of the shape functions. For example, if we considered the 4 node quadrilateral element we would have;

$$\mathbf{J} = \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \frac{\partial N_3}{\partial \xi} & \frac{\partial N_4}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \frac{\partial N_3}{\partial \eta} & \frac{\partial N_4}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix}, \quad (23)$$

where  $(x_1, y_1)$  is the  $x, y$  coordinates of node 1, etc. The derivatives of the shape functions with respect to the global coordinates can thus be found via

$$\begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{bmatrix}, \quad (24)$$

where  $\mathbf{J}^{-1}$  is the inverse of the Jacobian matrix. The shape functions themselves do not have to be converted from the local to the global coordinate system. This is because we have chosen a special type of “isoparametric” element (which essentially means that the shape functions defining the geometry are the same as those interpolating the unknown function).

Transformation of integration from the global to local coordinate system is performed using the determinant of the Jacobian,  $\det(\mathbf{J})$ , according to the following relation

$$\iint f(x, y) \, dx dy = \int_{-1}^1 \int_{-1}^1 f(\xi, \eta) \det(\mathbf{J}(\xi, \eta)) \, d\xi d\eta. \quad (25)$$

We note that in contrast to the one dimensional example, the Jacobian  $\mathbf{J}$  is in general not a constant. Only under special conditions is  $\mathbf{J}$  a constant. Such conditions for bilinear basis functions include when the elements have edges which are parallel to the  $x$  and  $y$  axis, or the elements define parallelograms. Combining this result with Equation (21) yields the final integration formula

$$\iint f(x, y) \, dx dy \approx \sum_{i=1}^{nip} f(\xi_i, \eta_i) \det(\mathbf{J}(\xi_i, \eta_i)) W_i. \quad (26)$$

Again we emphasize that  $\det(\mathbf{J})$  should be evaluated at the quadrature point.

Finally, to complete the definition of the element stiffness matrices, we provide the definition for the shape functions associated with a bilinear, 2D quadrilateral element such as is shown in Figure 1. In local coordinates, they are defined via

$$N_1(\xi, \eta) = \frac{1}{4} (1 - \xi) (1 - \eta) \quad (27)$$

$$N_2(\xi, \eta) = \frac{1}{4} (1 - \xi) (1 + \eta) \quad (28)$$

$$N_3(\xi, \eta) = \frac{1}{4} (1 + \xi) (1 + \eta) \quad (29)$$

$$N_4(\xi, \eta) = \frac{1}{4} (1 + \xi) (1 - \eta), \quad (30)$$

and their derivatives with respect to  $\xi$  and  $\eta$  are (computed by hand, or with MAPLE);

$$\begin{aligned} \frac{\partial N_1(\xi, \eta)}{\partial \xi} &= -\frac{1}{4} (1 - \eta), & \frac{\partial N_1(\xi, \eta)}{\partial \eta} &= -\frac{1}{4} (1 - \xi) \\ \frac{\partial N_2(\xi, \eta)}{\partial \xi} &= -\frac{1}{4} (1 + \eta), & \frac{\partial N_2(\xi, \eta)}{\partial \eta} &= \frac{1}{4} (1 - \xi) \\ \frac{\partial N_3(\xi, \eta)}{\partial \xi} &= \frac{1}{4} (1 + \eta), & \frac{\partial N_3(\xi, \eta)}{\partial \eta} &= \frac{1}{4} (1 + \xi) \\ \frac{\partial N_4(\xi, \eta)}{\partial \xi} &= \frac{1}{4} (1 - \eta), & \frac{\partial N_4(\xi, \eta)}{\partial \eta} &= -\frac{1}{4} (1 + \xi). \end{aligned} \quad (31)$$

In summary, the following steps must be carried out to perform integration of the element matrices:

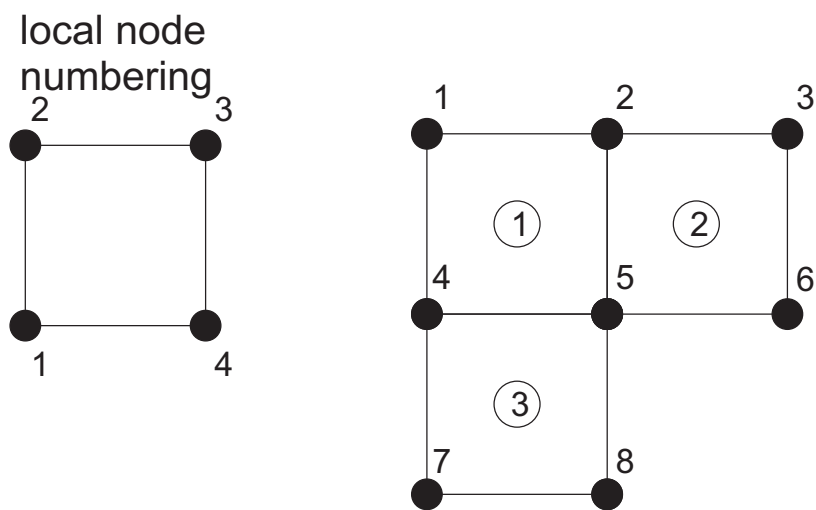
1. Define the number of integration points ( $n_{ip}$ ) and obtain and save the integration points and weights.
2. Do a loop over all elements and initialize the element matrices  $\mathbf{MM}$  and  $\mathbf{KM}$ .
3. Within the element loop, do a loop over each integration point. Evaluate the shape functions and their derivatives at the relevant integration point.
4. Calculate the Jacobian matrix by performing the operation in Equation (23).
5. One must convert derivatives from the local coordinates to the global coordinates by performing the operation in Equation (24).
6. Calculate the determinant of the Jacobian.
7. Perform the vector multiplication involving the shape functions or their derivatives (evaluated at integration points), multiplied by the relevant weights, which leads to the matrices  $\mathbf{MM}$  and  $\mathbf{KM}$ . The result should be added to the multiplication from proceeding integration points.
8. Once one has exited the loop over integration points, the integration is complete (i.e., the element matrices have been integrated).
9. Form the matrices  $\mathbf{L}$  and  $\mathbf{R}$  defined in Equation (20)
10. The element matrices can then be added to the global element stiffness matrix. This process is then continued for each element.

## 4 Assembly of the global stiffness matrix

The final step that needs to be performed is the node-by-node summing-up of the element matrices to form the global stiffness matrices. Once again, this is carried out in a manner very similar to that already done in one dimension. The only difference is that now one has to make more decisions in how the unknowns in the mesh are numbered globally. An example of a small 2D finite element mesh is shown in Figure 1. Note that each node represents one unknown (i.e. temperature) and is given a global index (and a global coordinate).

The global numbering of the equations used in your finite element is not unique. There are no strict rules as to how one should order the global indices. On structured grids with quadrilateral elements (like in our example) the traditional rule of thumb was to begin numbering in the direction with the fewest nodes in an effort to minimise the bandwidth of the global matrix and thus yield more efficient solves. In practice, when using modern factorisation techniques such as those employed in Matlab, this rule of thumb is no longer applicable. Furthermore it does not generalise to unstructured meshes. Here we advocate using the numbering scheme which is most convenient to implement. The relationship between the elements and global node numbers should be specified early in the program and saved in a matrix (for example in  $g\_num$ ). This matrix is then used to “steer” the entries within each element stiffness matrix into the correct position within the global matrix.

## 2-D FEM MESH - one degree of freedom per node



Relationship between elements  
and global node numbers

$$g\_num = \begin{bmatrix} 4 & 5 & 7 \\ 1 & 2 & 4 \\ 2 & 3 & 5 \\ 5 & 6 & 8 \end{bmatrix}$$

Figure 2: A small three element finite element mesh. In the text the diffusion equation is discretised for this mesh. The corresponding system of global equations are shown in Equation (33).



To illustrate this process with an example, we return to the 2D diffusion equation and consider assembling the global stiffness matrix which corresponds to a very small mesh which is illustrated in Figure 2. Referring to the element matrices defined in Equation (20), we introduce the following notation:

$$\mathbf{L} = \begin{bmatrix} L_{11} & L_{12} & L_{13} & L_{14} \\ L_{21} & L_{22} & L_{23} & L_{24} \\ L_{31} & L_{32} & L_{33} & L_{34} \\ L_{41} & L_{42} & L_{43} & L_{44} \end{bmatrix}.$$

Moreover, a superscript is introduced to represent the element number. Thus, for example, the term  $L_{11}^3$  is the term  $L_{11}$  from element 3. Similar notation is assumed for  $\mathbf{R}$ . The matrix  $\mathbf{g\_num}$  for the mesh in Figure 2 is given by:

$$\mathbf{g\_num} = \begin{bmatrix} 4 & 5 & 7 \\ 1 & 2 & 4 \\ 2 & 3 & 5 \\ 5 & 6 & 8 \end{bmatrix}. \quad (32)$$

Note that there are three columns, one for each element, while there are four rows, one for each node. As already stated,  $\mathbf{g\_num}$  is used to place the element matrices in the correct position in the global matrix. As an example, consider the term  $L_{13}^1$ . Since the term is from element 1, we must look at the first column of  $\mathbf{g\_num}$  where we see that the local node 13 (i.e., the first and third row) map to the global indices 42. This indicates that  $L_{13}^1$  appears in the global matrix  $\mathbf{L}_G$  at the position 42. Repeating this process for all terms leads to the global equations which will denote by

$$\mathbf{L}_G \mathbf{T}^{n+1} = \mathbf{R}_G \mathbf{T}^n + \mathbf{F}_G, \quad (33)$$

where

$$\mathbf{T}^k = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ T_7 \\ T_8 \end{bmatrix}^k, \quad (34)$$

$$\mathbf{L}_G = \begin{bmatrix} L_{22}^1 & L_{23}^1 & 0 & L_{21}^1 & L_{24}^1 & 0 & 0 & 0 \\ L_{32}^1 & L_{33}^1 + L_{22}^2 & L_{23}^2 & L_{31}^1 & L_{34}^1 + L_{21}^2 & L_{24}^2 & 0 & 0 \\ 0 & L_{32}^2 & L_{33}^2 & 0 & L_{31}^2 & L_{34}^2 & 0 & 0 \\ L_{12}^1 & L_{13}^1 & 0 & L_{11}^1 + L_{32}^3 & L_{14}^1 + L_{23}^3 & 0 & L_{21}^3 & L_{24}^3 \\ L_{42}^1 & L_{43}^1 + L_{12}^2 & L_{13}^2 & L_{41}^1 + L_{32}^3 & L_{44}^1 + L_{11}^2 + L_{33}^3 & L_{14}^2 & L_{31}^3 & L_{34}^3 \\ 0 & L_{42}^2 & L_{43}^2 & 0 & L_{41}^2 & L_{44}^2 & 0 & 0 \\ 0 & 0 & 0 & L_{12}^3 & L_{13}^3 & 0 & L_{11}^3 & L_{14}^3 \\ 0 & 0 & 0 & L_{42}^3 & L_{43}^3 & 0 & L_{41}^3 & L_{44}^3 \end{bmatrix} \quad (35)$$

and

$$\mathbf{R}_G = \begin{bmatrix} R_{22}^1 & R_{23}^1 & 0 & R_{21}^1 & R_{24}^1 & 0 & 0 & 0 \\ R_{32}^1 & R_{33}^1 + R_{22}^2 & R_{23}^2 & R_{31}^1 & R_{34}^1 + R_{21}^2 & R_{24}^2 & 0 & 0 \\ 0 & R_{32}^2 & R_{33}^2 & 0 & R_{31}^2 & R_{34}^2 & 0 & 0 \\ R_{12}^1 & R_{13}^1 & 0 & R_{11}^1 + R_{22}^3 & R_{14}^1 + R_{23}^3 & 0 & R_{21}^3 & R_{24}^3 \\ R_{42}^1 & R_{43}^1 + R_{12}^2 & R_{13}^2 & R_{41}^1 + R_{32}^3 & R_{44}^1 + R_{11}^2 + R_{33}^3 & R_{14}^2 & R_{31}^3 & R_{34}^3 \\ 0 & R_{42}^2 & R_{43}^2 & 0 & R_{41}^2 & R_{44}^2 & 0 & 0 \\ 0 & 0 & 0 & R_{12}^3 & R_{13}^3 & 0 & R_{11}^3 & R_{14}^3 \\ 0 & 0 & 0 & R_{42}^3 & R_{43}^3 & 0 & R_{41}^3 & R_{44}^3 \end{bmatrix}. \quad (36)$$

The process of summing the element matrix into the global matrix is easily achieved in a computer program. For example, once the matrix  $\mathbf{L}_e$  for a certain element  $e$  has been integrated, it is added to the global matrix using a command such as

```
>LG( g_num( :, iel ), g_num( :, iel ) ) = ...
    LG( g_num( :, iel ), g_num( :, iel ) ) + Le;
```

where  $iel$  is the current element index. Now that the global equations have been constructed all that remains is to apply boundary conditions and solve the system of equations. This is carried out in the same manner as the one dimensional problem considered earlier.

## 5 Exercises

1. Write a 2D FE code that solves the diffusion equation using bilinear, quadrilateral elements. Assume a constant, isotropic conductivity tensor. Apply constant temperature at the top and bottom boundaries, and zero flux at the side boundaries (i.e. set no boundary condition here).
2. Confirm that the code works when the grid is deformed (if not, it is likely that you made a mistake defining the Jacobian or your global derivatives). Define a mesh which has significant topography on the upper surface. Compute the temperature distribution within this domain.
3. Modify the code to take into account i) a spatially variable, anisotropic conductivity tensor, ii) variable grid spacing and iii) a spatially variable heat source.