# 13

# The finite element method for partial differential equations

## 13.1 Introduction

When we consider a partial differential equation, such as the ubiquitous Laplace equation

$$\nabla^2 \phi(\mathbf{r}) = 0, \tag{13.1}$$

together with some boundary condition(s), the obvious way of solving it that comes to mind is to discretise this equation on a regular grid, hoping that this grid can match the boundary in some way. Then we solve the discretised problem using, for example, iterative methods such as the Gauss–Seidel or conjugate gradients method (see Appendix A7.2). For many problems, this approach is adequate, but if the problem is difficult in the sense that it has a lot of structure on small scales in some region of the domain, or if the boundary has a complicated shape which is difficult to match with a regular grid, it might be useful to apply methods that allow for flexibility of the grid on which the solution is formulated. In this chapter we discuss such a method, the *finite element method*.

One way of looking at the finite element method (FEM) is by realising that many partial differential equations can be viewed as solution methods for variational problems. In the case of the Laplace equation with zero boundary condition, for example, finding the stationary solution of the functional

$$\int_D [\nabla \phi(\mathbf{r})]^2 \, \mathrm{d}^d r, \tag{13.2}$$

where the integral is over the $d$-dimensional domain $D$ and where we confine ourselves to functions $\phi(\mathbf{r})$ which vanish on the domain boundary, yields the same solution as that of the Laplace equation – in fact, the Laplace equation is the Euler equation for this functional (see the next section).

The integral can be discretised by dividing up the domain $D$ into *elements* of – in principle – arbitrary shape and size, and assuming a particular form of the solution within each element, a linear function for example, together with continuity conditions on the element boundaries. It turns out that finding the solution boils down to solving a sparse matrix problem, which can be treated by conjugate gradient methods, see (see Appendix A7.2).

In this chapter we discuss the finite element method, error estimation, and principles of local grid refinement. This will be done for two different problems: the Poisson/Laplace equation, and the equations for elastic deformation of a solid. Both problems will be considered in two dimensions only. The aim is to explain the ideas behind the finite element methods and adaptive refinement without going into too much detail. For a more rigorous and complete treatment, the reader is referred to the specialised literature [1–5].

Some special topics will be covered in the remaining sections: local adaptive grid refinement, dynamics, and, finally, the coupling of two descriptions, finite element and molecular dynamics, in order to describe phenomena at very different length scales occurring in one system.

Most of the sections describe implementation of FEM for standard problems. The reader is invited to try the implementation by him- or herself.

## 13.2 The Poisson equation

As mentioned in the previous section, the Laplace equation can easily be formulated in a variational way. The same holds for the Poisson equation:

$$\nabla^2 \phi(\mathbf{r}) = f(\mathbf{r}), \tag{13.3}$$

with appropriate boundary conditions. We assume Dirichlet boundary conditions on the edge of the domain, which we take as a simple square of size $L \times L$. The functional whose stationary solution satisfies this equation is

$$J[\phi(\mathbf{r})] = \int_D \{[\nabla \phi(\mathbf{r})]^2 + f(\mathbf{r})\phi(\mathbf{r})\}\, \mathrm{d}^d r, \tag{13.4}$$

as is easily verified using Green's first identity [6] together with the fact that $\phi$ vanishes on the boundary. From now on, we shall use $\mathrm{d}\Omega$ to denote a volume element occurring in integrals.

We now divide up the square into triangular elements, and assume that the solution $\phi(\mathbf{r})$ is linear within each element:

$$\phi(x, y) = a_i + b_i x + c_i y \tag{13.5}$$

within element $i$. Now consider a grid point. This will in general be a vertex of more than one triangle. Naturally, we want to assign a single value of the solution

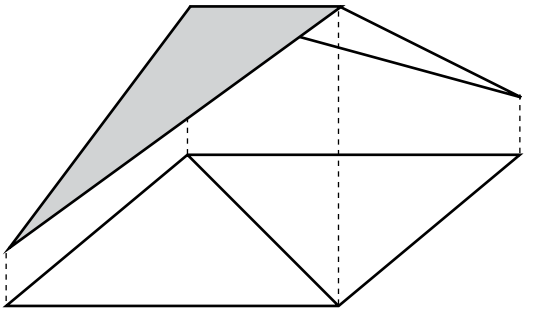

Figure 13.1. Two adjacent triangles on a square (ground plane) with a linear function $\phi(\mathbf{r})$ shown as the height (vertical) coordinate on both triangles. As $\phi$ is linear for each triangle, the requirement that the values of the two triangles are the same at their two shared vertices ensures continuity along their edges.

to that point, so we require the solution within each triangle sharing the same vertex to have the same value at that vertex. Linearity of the solution within the triangles then makes the solution continuous over each triangle edge (Figure 13.1). We see that for each triangle, the solution is characterised by three constants, $a_i$, $b_i$ and $c_i$. They can be fixed by the values of the solution at the three vertices of the triangle. It is also possible to use rectangles as elements. In that case, we must allow for one more degree of freedom of the solution (as there are now four vertices), and the form may then be

$$\phi(x, y) = a_i + b_i x + c_i y + d_i xy. \tag{13.6}$$

It is also possible to use quadratic functions on the triangles:

$$\phi(x, y) = a_i + b_i x + c_i y + d_i xy + e_i x^2 + f_i y^2, \tag{13.7}$$

requiring six conditions. In that case, we use the midpoints of the edges of the triangles as additional points where the solution must have a particular value. We shall restrict ourselves in this book to linear elements. In three dimensions, the linear solution requires four parameters to be fixed, and this can be done by using tetrahedra as elements (a tetrahedron has four vertices). The triangle and the tetrahedron are the elements with nonzero volume which are bounded by the *smallest possible* number of sides in two and three dimensions respectively. Such elements are called *simplices*. In one dimension, an element with this property is the line segment.

Now that we have a discrete representation of our solution by considering just its values on the vertices of the grid, we must find the expression for the integral within the approximations made (i.e. linear behaviour of the solution within the elements). To do this we digress a bit to introduce *natural coordinates*. For a triangle these are linear coordinates which have a value 1 at one of the vertices and zero at the two

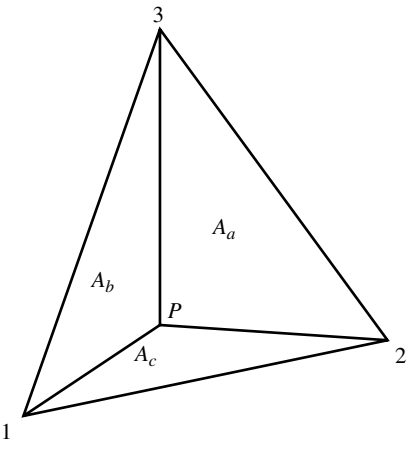


Figure 13.2. The areas $A_a$, $A_b$ and $A_c$ for any point $P$ within the triangle. The $A_i$ are used to define the natural coordinates $\xi_i$ of the point $P$. $A$ is the total surface area.

others. Any point $P$ within the triangle can be defined by specifying any two out of three natural coordinates, $\xi_a$, $\xi_b$ or $\xi_c$. These are defined by

$$\xi_i = \frac{A_i}{A}, \quad i = a, b, c, \tag{13.8}$$

where $A_i$, $A$ are the surface areas shown in Figure 13.2. The natural coordinates satisfy the requirement

$$\xi_a + \xi_b + \xi_c = 1. \tag{13.9}$$

The $x$- and $y$-coordinates of a point can be obtained from the natural coordinates by the linear transformation

$$\begin{pmatrix} 1 \\ x \\ y \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ x_a & x_b & x_c \\ y_a & y_b & y_c \end{pmatrix} \begin{pmatrix} \xi_a \\ \xi_b \\ \xi_c \end{pmatrix} \tag{13.10}$$

where $(x_a, y_a)$ are the Cartesian coordinates of vertex $a$ etc.

The reverse transformation

$$\begin{pmatrix} \xi_a \\ \xi_b \\ \xi_c \end{pmatrix} = \frac{1}{2A} \begin{pmatrix} x_b y_c - x_c y_b & y_{bc} & x_{cb} \\ x_c y_b - x_b y_c & y_{ca} & x_{ac} \\ x_a y_b - x_b y_a & y_{ab} & x_{ba} \end{pmatrix} \begin{pmatrix} 1 \\ x \\ y \end{pmatrix}, \tag{13.11}$$

with $2A = \det(A) = x_{ba} y_{ca} - x_{ca} y_{ba}$ and $x_{ab} = x_b - x_a$ etc., translates the $x, y$ coordinates into natural coordinates. All these relations can easily be checked.

Having natural coordinates, we can construct a piecewise linear approximation to the solution from the values of the solution at the vertices of the triangles. Calling

$\phi_a$, $\phi_b$ and $\phi_c$ these values at the corresponding vertices, the solution inside the triangle is given by

$$\tilde{\phi}(\mathbf{r}) = \phi_a \xi_a + \phi_b \xi_b + \phi_c \xi_c. \tag{13.12}$$

In order to evaluate integrals over the triangular element, we use the following formula:

$$\int_A \xi_a^k \xi_b^l \xi_c^m \, \mathrm{d}\Omega = 2A \frac{k!\, l!\, m!}{(2+k+l+m)!} \tag{13.13}$$

for non-negative integers $k$, $l$ and $m$, and for $a$, $b$ and $c$ assuming values 1, 2 and 3. Remember, $\mathrm{d}\Omega$ is the volume element.

We now have all the ingredients for solving the Laplace equation using triangular finite elements. First, note that the integral (13.4) now becomes a quadratic expression in the values $\phi_i$ at the grid points. This quadratic expression can be written in the form

$$J[\boldsymbol{\phi}] = -\boldsymbol{\phi}^{\mathrm{T}} \mathbf{K} \boldsymbol{\phi} - \mathbf{r}^{\mathrm{T}} \boldsymbol{\phi} \tag{13.14}$$

(we work out the specific form of the expressions below). Here, $\boldsymbol{\phi}$ is the vector whose elements are the values of the solutions at the grid points, $\mathbf{K}$ is a symmetric matrix, and $\mathbf{r}$ is a vector. Minimising this expression leads to the matrix equation

$$\mathbf{K}\boldsymbol{\phi} = \mathbf{r}. \tag{13.15}$$

The matrix–vector product on the left hand side can be evaluated as a sum over all triangles. Within a triangle, we deal with the values on its vertices.

To be more specific, let us calculate

$$\int_{\mathrm{elem}} (\nabla \phi)^2 \, \mathrm{d}\Omega. \tag{13.16}$$

Using (13.11) we have

$$\nabla \xi_a = \frac{1}{2A} (y_{bc}, x_{cb}) \tag{13.17}$$

and similar expressions for the other two natural coordinates. From these we have, with the parametrisation (13.12),

$$\nabla \tilde{\phi} = \frac{1}{2A} [\phi_a (y_{bc}, x_{cb}) + \phi_b (y_{ca}, x_{ac}) + \phi_c (y_{ab}, x_{ba})]. \tag{13.18}$$

Note that on the left and right hand side, we have two-dimensional vectors, which are given in row form on the right hand side. Also note that the components of the vector are constant over the triangle, which is natural as the solution is assumed to be linear within the triangle. The integral is the norm of this constant vector squared times the surface area of the triangle. Obviously this yields a quadratic expression in $\phi_a$, $\phi_b$ and $\phi_c$ of a form similar to (13.14), but now formulated for a single triangle.

This equation is defined by a matrix $\mathbf{k}$ which is called the *local stiffness matrix* for the triangle under consideration. Introducing the vectors

$$\mathbf{b} = \begin{pmatrix} y_{bc} \\ y_{ca} \\ y_{ab} \end{pmatrix} \tag{13.19}$$

and

$$\mathbf{c} = \begin{pmatrix} x_{cb} \\ x_{ac} \\ x_{bc} \end{pmatrix} \tag{13.20}$$

we see, after some calculation, that the stiffness matrix $\mathbf{k}$ can be evaluated as

$$\mathbf{k} = \mathbf{b}\mathbf{b}^{\mathrm{T}} + \mathbf{c}\mathbf{c}^{\mathrm{T}}, \tag{13.21}$$

which leads to the result

$$[\mathbf{k}] = \frac{1}{4A} \begin{pmatrix} y_{bc}^2 + x_{cb}^2 & y_{bc}y_{ca} + x_{cb}x_{ac} & y_{bc}y_{ab} + x_{cb}x_{ba} \\ y_{bc}y_{ca} + x_{cb}x_{ac} & y_{ca}^2 + x_{ac}^2 & y_{ca}y_{ab} + x_{ac}x_{ba} \\ y_{bc}y_{ab} + x_{cb}x_{ba} & y_{ca}y_{ab} + x_{ac}x_{ba} & y_{ab}^2 + x_{ba}^2 \end{pmatrix}. \tag{13.22}$$

We not only need the matrix representing the Laplacian operator, but we must also evaluate the integral containing the source term $f(\mathbf{r})$ in Eq. (13.4). The continuous function $f(\mathbf{r})$ is approximated by a piecewise linear function $\tilde{f}$ on the triangles – just like the solution $\phi(\mathbf{r})$. For a particular triangle with vertices $a$, $b$, and $c$, we have

$$\tilde{f}(\mathbf{r}) = f_a\xi_a + f_b\xi_b + f_c\xi_c. \tag{13.23}$$

We must multiply this by the linear approximation for $\phi$, Eq. (13.12), and then integrate over the element, using (13.13). The result must then be differentiated with respect to $\phi_a$, $\phi_b$, $\phi_c$, which results in a vector element

$$r_a = 2A\left(\frac{f_a}{12} + \frac{f_b}{24} + \frac{f_c}{24}\right), \tag{13.24}$$

and similar for $r_b$ and $r_c$.

The matrix–vector multiplication can be carried out as a loop over all triangular elements where for each element the stiffness matrix is applied to the three vertices of that triangle. Note that the stiffness matrix should always act on the *old* vector containing the field values $\phi_a$ and that the result should be added to the new vector (which initially is set to zero; see the next section). If we have a matrix–vector multiplication and a right hand side of the form (13.15), we can apply the conjugate gradients method to solve the matrix equation.

We have overlooked one aspect of the problem: if a triangle contains vertices on the boundary where the value of the solution is given (Dirichlet boundary condition), the corresponding values of $\phi(\mathbf{r})$ are known and therefore not included in the vector.

In that case we apply the stiffness matrix only to those points which are in the interior of the system. That is, we update only the interior points – the values at the boundary points remain unchanged.

### 13.2.1 Construction of a finite element program

The program should contain an array in which the equilibrium positions of the vertices are stored. Furthermore, we need a vector containing the displacements of each vertex. Note that locations and displacements are both two-dimensional in our case. Furthermore there is an array containing the relevant stiffness matrices for the triangles. For each triangle, we must know the indices of its three vertices. From this we can calculate

- The stiffness matrix for the triangle;
- The force vector of the triangle, which is the right hand side of the matrix equation to be solved.

The heart of the program is the multiplication of the field vector by the stiffness matrixes of the triangle. This can be done as follows.

```
Set the new global field vector to zero;
FOR each triangle DO
    Store the three old values of the field at the vertices
            in a local 3-vector;
    Multiply this vector by the stiffness matrix;
    Add the result to the appropriate entries of the new global
            field vector;
END FOR
```

You should now be able to write such a program. If you study the the problem of a point charge (delta function) on a $40 \times 40$ square grid, which is divided up into 3200 rectangular triangles with two $45^{\circ}$ angles, you need 118 conjugate gradient iterations to achieve convergence of the residue (the $L_2$ norm of the vector $[K][\phi] - [r]$) within $10^{-10}$. Obviously, this is the error in the solution of the matrix equation. The numerical error introduced by the discretisation of the grid may be (and will be) substantially larger.

## 13.3 Linear elasticity

### 13.3.1 The basic equations of linear elasticity

For many materials, deformations due to applied forces can to a good approximation be calculated using the equations of linear elasticity. These equations are valid in

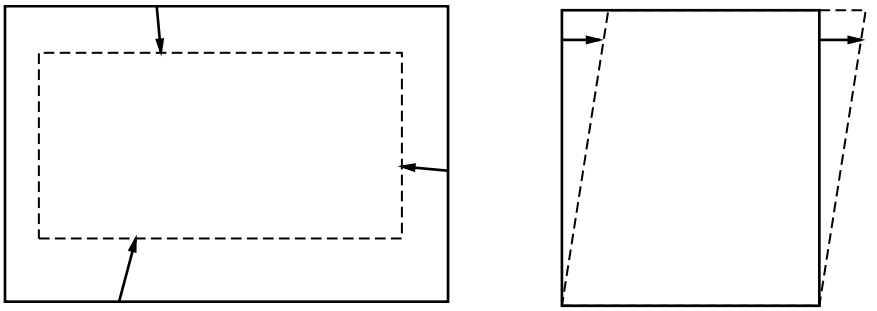

Figure 13.3. Two types of deformation, compression (left) and shear (right).

particular when the deformation is relatively small so that the total energy of the deformed system can be well approximated by a second order Taylor expansion.

There are two types of deformation. The first is compression or expansion of the system, and the second is shear. These effects are shown in Figure 13.3. We restrict ourselves to homogeneous isotropic systems in two dimensions. Then the resistance of a material to the two types of deformation is characterised in both cases by an elastic constant – in the literature either the Lamé constants $\lambda$ and $\mu$ are used, or the Young modulus $E$ and Poisson ratio $\nu$. They are related by

$$E = \frac{\mu(3\lambda + 2\mu)}{\lambda + \mu} \tag{13.25a}$$

$$\nu = \frac{\lambda}{2(\lambda + \mu)}. \tag{13.25b}$$

To formulate the equations of deformation, consider the displacement field $\mathbf{u}(\mathbf{r})$. This vector field is defined as the displacement of the point $\mathbf{r}$ as a result of external forces acting on the system. These forces may either be acting throughout the system (gravity is an example) or on its boundary, like pushing with a finger on the solid object. In the equilibrium situation, the forces balance each other inside the material. So, if we identify a small line (a planar facet in three dimensions) with a certain orientation somewhere inside the object, the forces acting on both sides of this line should cancel each other. These forces vary with the orientation of the line or facet, as can be seen by realising that in an isotropic medium and in the absence of external forces, the force is always normal to the line (it is due to the internal, isotropic pressure). Another way to see this is by considering gravity. This acts on a horizontal facet from above but not on a vertical facet. Therefore it is useful to define the *stress tensor* $\sigma_{ij}$ which gives the $j$th component of the force acting on a small facet with a normal along the $i$th Cartesian axis.

The stress plus the body forces results in the displacement. It is important that the actual value of the displacement matters less than its derivative: if we displace two points connected by a spring over a certain distance, the forces acting between the two points do not change. What matters is the difference in displacement of neighbouring points. Information concerning this is contained in the *strain* $\varepsilon_{ij}$. It is defined as

$$\varepsilon_{ij} = \frac{\mathrm{d}u_i}{\mathrm{d}x_j}. \tag{13.26}$$

For an isotropic, homogeneous material in two dimensions, only three components of stress and strain are important:

$$\sigma_{xx} \quad \text{and} \quad \varepsilon_{xx}; \tag{13.27a}$$

$$\sigma_{yy} \quad \text{and} \quad \varepsilon_{yy}; \tag{13.27b}$$

$$\sigma_{xy} \quad \text{and} \quad 2\varepsilon_{xy}. \tag{13.27c}$$

Stress and strain are related by Hooke's law:

$$\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\varepsilon}, \tag{13.28}$$

where $\boldsymbol{\sigma}$ is the vector $(\sigma_{xx}, \sigma_{yy}, \sigma_{xy})^{\mathrm{T}}$ and similarly for $\boldsymbol{\varepsilon}$. $\mathbf{C}$ is the elastic matrix:

$$\mathbf{C} = \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1}{2}(1-\nu) \end{pmatrix}. \tag{13.29}$$

The body is at rest in a state where all forces are in balance. The force balance equation reads

$$\mathbf{D}^{\mathrm{T}}\boldsymbol{\sigma} + \mathbf{f} = 0 \tag{13.30}$$

with

$$\mathbf{D} = \begin{pmatrix} \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{pmatrix} \tag{13.31}$$

This matrix can also be used to relate $\boldsymbol{\varepsilon}$ and $\mathbf{u}$:

$$\boldsymbol{\varepsilon} = \mathbf{D}\mathbf{u}. \tag{13.32}$$

There are two types of boundary conditions: parts of the boundary may be free to move in space, and other parts may be fixed. You may think of a beam attached to a wall at one end. In the example which we will work out below, we only include gravity as a (constant) force acting on each volume element of the system.

Just as in the case of the Laplace equation, we must find an integral formulation of the problem, and approximate the various relevant functions by some special form on the elements. As before, we will choose piecewise linear functions on the elements. Note that in this case we approximate each of the two components of the displacement field by these functions.

### *13.3.2 Finite element formulation*

The finite element formulation can be derived from the continuum equations if we can formulate the latter as a variational problem for a functional expression which is an integral formulation of the problem.

To find this formulation in terms of integrals, we introduce the so-called 'weak formulation', for the force balance equation, which has the form:

$$\int_\Omega (\delta\mathbf{u})^{\mathrm{T}}(\mathbf{D}^{\mathrm{T}}\boldsymbol{\sigma} + \mathbf{f})\, \mathrm{d}\Omega = 0. \tag{13.33}$$

Here, $\delta\mathbf{u}$ is an *arbitrary* displacement field satisfying the appropriate boundary conditions. Using (13.32) this integral equation is cast into the form

$$\int_\Omega (\delta\boldsymbol{\varepsilon})^{\mathrm{T}}\boldsymbol{\sigma}\, \mathrm{d}\Omega = -\int_\Omega (\delta\mathbf{u})^{\mathrm{T}}\mathbf{f}\, \mathrm{d}\Omega \tag{13.34}$$

We then can divide up the space $\Omega$ into $N$ elements (triangles for two dimensions) and write

$$\sum_{e=1}^{N}\int_{\Omega_e} (\delta\boldsymbol{\varepsilon}_e)^{\mathrm{T}}\boldsymbol{\sigma}_e\, \mathrm{d}\Omega_e = -\sum_{e=1}^{N}\int_{\Omega_e} (\delta\mathbf{u}_e)^{\mathrm{T}}\mathbf{f}_e\, \mathrm{d}\Omega_e. \tag{13.35}$$

From this we can derive the form of the stiffness matrix for the elastic problem. First note that the variables of the problem are the deformations $\mathbf{v}_n$ on the vertices $n$. This means that for each triangle we have six variables (two values at each of the three vertices). Therefore, the stiffness matrix is $6 \times 6$. The deformations do not enter as such into the problem but only through the strain tensor. We have

$$\boldsymbol{\varepsilon}_e = \begin{pmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ 2\varepsilon_{xy} \end{pmatrix} = \mathbf{D}\mathbf{u}_e = \begin{pmatrix} \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix}. \tag{13.36}$$

This tensor, however, is linearly related to the $\mathbf{v}_i$. We write the displacement field as

$$\mathbf{u} = \mathbf{v}_a\xi_a + \mathbf{v}_b\xi_b + \mathbf{v}_c\xi_c. \tag{13.37}$$

The $\xi_i$ depend on $x$ and $y$ – the relation is given in Eq. (13.11). From this, and from (13.17), we find

$$\boldsymbol{\varepsilon}_e = \begin{pmatrix} y_b - y_y & 0 & y_c - y_a & 0 & y_a - y_b & 0 \\ 0 & x_c - x_b & 0 & x_a - x_b & 0 & x_b - x_a \\ x_c - x_b & y_b - y_c & x_a - x_b & y_c - y_a & x_b - x_a & y_a - y_b \end{pmatrix} \begin{pmatrix} v_{ax} \\ v_{ay} \\ v_{bx} \\ v_{by} \\ v_{cx} \\ v_{cy} \end{pmatrix}. \tag{13.38}$$

We call the $3 \times 6$ matrix on the right hand side $B$. Using the relation

$$\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\varepsilon}, \tag{13.39}$$

we can rewrite the element integral of the left hand side of Eq. (13.35) as

$$\int_{\Omega_e} (\delta \mathbf{v})^{\mathrm{T}} \mathbf{B}^{\mathrm{T}} \mathbf{C} \mathbf{B} \mathbf{v} \, \mathrm{d}\Omega_e, \tag{13.40}$$

where $\mathbf{v}$ is a six-dimensional vector, $\mathbf{B}$ is a $3 \times 6$ matrix and $\mathbf{C}$ a $3 \times 3$ matrix.

Note that there is no dependence on the coordinates $x$ and $y$ in this expression. This can be traced back to the fact that we can express the integrand in terms of the strain, which contains derivatives of the deformation $\mathbf{u}$ which in turn is a *linear* function within the element. The integral is obtained by multiplying the constant integrand by the surface area $A$ of the integrand. The stiffness matrix $\mathbf{k}$ is therefore given by

$$\mathbf{k} = A\mathbf{B}^{\mathrm{T}}\mathbf{C}\mathbf{B}. \tag{13.41}$$

This is a $6 \times 6$ matrix which connects the six-dimensional vectors $\mathbf{v}$.

The right hand side of Eq. (13.35) also involves an integral expression. This contains the external force. Taking this to be gravity, it is constant. We must evaluate the integral

$$\mathbf{f}_e \cdot \int_{\Omega_e} [(\delta \mathbf{v})_a \xi_a + (\delta \mathbf{v})_b \xi_b + (\delta \mathbf{v})_c \xi_c] \, \mathrm{d}\Omega_e. \tag{13.42}$$

This can be written in the form

$$\mathbf{f}_e \cdot \mathbf{G}\mathbf{v}, \tag{13.43}$$

where $\mathbf{G}$ is the $2 \times 6$ matrix:

$$\frac{A}{3} \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}. \tag{13.44}$$

We have now reworked (13.35) to the form

$$\delta \mathbf{v}^{\mathrm{T}} \mathbf{K} \mathbf{v} = \delta \mathbf{v}^{\mathrm{T}} \mathbf{G} \mathbf{f}, \tag{13.45}$$

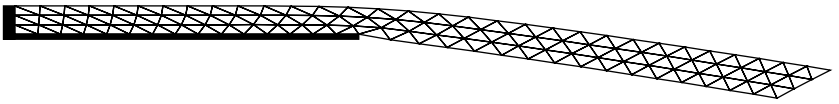

Figure 13.4. Deformation of a beam attached to a vertical wall, calculated with the finite element method. The beam is supported on half of its base.

where $\mathbf{v}$ now represents the vector of *all* displacements (that is, for the whole grid), $\mathbf{K}$ is the *full* stiffness matrix, which can be evaluated as a careful sum over the stiffness matrices for all triangles in the same spirit as described for the Laplace equation in Section 13.2.1, and the right hand side is a vector defined on the full grid. The dimension of the matrix problem is $2N$, where $N$ is the number of vertices. If points are subject to Dirichlet boundary conditions, they are excluded from the vectors and matrices, so that for actual problems the dimension is less than $2N$. The matrix equation found must hold for all $\delta\mathbf{v}$, which can only be true when

$$\mathbf{Kv} = \mathbf{Gf} \tag{13.46}$$

and this can be solved for using the conjugate gradients method. In Figure 13.4, the result of a deformation calculation is shown for a beam with the left end attached to a wall.

## 13.4 Error estimators

Like every numerical method, the finite element method is subject to errors of several kinds. Apart from modelling errors and errors due to finite arithmetic precision in the processor, the discretisation errors are important, and we will focus on these. Obviously the discretisation error can be made small by reducing the grid constant homogeneously over the lattice, but this can only be done at the cost of increasing the computer time needed to arrive at a stable solution. It might be that the error is due to only a small part of the system under consideration, and reducing the mesh size in those regions which are already treated accurately with a coarse mesh is unnecessary and expensive overkill.

It is therefore very useful to have available a *local estimator* of the error which tells us for a particular region or element in space what its contribution to the overall error is. In that case, we can refine the mesh only in those regions where it is useful. In this section, we first address the problem of formulating such a local error estimator and then describe a particular refinement strategy for triangular meshes.

One type of local error estimator is based on the notion that, unlike the displacement field, the *stress* usually is not continuous over the element boundaries. If a number of triangles meet at a particular mesh point, they will all have slightly different values of their stress components (recall that the stress is defined in terms of

first derivatives of the displacement fields). A more accurate solution would lead to continuous stresses and this can be achieved by some suitable averaging of the stress components at the mesh points. To be specific, the nodal stress at a mesh point $p$ would be given by

$$\sigma_p = \frac{1}{\sum_{\text{elems}} w_{\text{elem}}} \sum_{\text{elems}} w_{\text{elem}} \sigma_{\text{elem}} \tag{13.47}$$

where the stresses on the right hand side (in the sum) are the result of the finite element calculation; the weights $w_{\text{elem}}$ may be taken equal or related to the surface area of the elements sharing the vertex $p$. The error is then the difference between the 'old' stresses resulting from the calculation and the improved values based on the recipe above. We shall refer to the 'old' stress, resulting from the FEM calculation, as the FEM stress.

The question arises how the weights $w_{\text{elem}}$ can be chosen optimally. One answer to this question is provided by the *projection method* [7–10]. In this method we seek a *continuous*, piecewise linear stress field, which deviates to a minimal extent from the FEM stress. The deviation can be defined as the $L_2$-norm of the difference between the FEM stress and the continuous stress $\sigma_{\text{C}}$ which is a piecewise linear FEM-type expansion, based on the values $\boldsymbol{\sigma}_p$:

$$\Delta = \int_\Omega (\sigma_{\text{C}} - \sigma_{\text{FEM}})^{\text{T}} (\sigma_{\text{C}} - \sigma_{\text{FEM}}) \, \mathrm{d}\Omega. \tag{13.48}$$

We write the continuous stress within a particular triangle $(a, b, c)$, as usual, in the form

$$\sigma_{\text{C}} = \sigma_a \xi_a + \sigma_b \xi_b + \sigma_c \xi_c, \tag{13.49}$$

where $\sigma_a$ etc. are the values of the stresses at the three vertices (as the stress is continuous, it must be single-valued at the mesh points). The optimal approximation of the actual stress is defined by those values of $\sigma_{\text{C}}$ at the vertices for which the deviation $\Delta$ is minimal. This directly leads to the condition

$$\frac{\partial \Delta[\sigma_{\text{C}}]}{\partial \sigma_p} = 2 \int_\Omega \left( \frac{\partial \sigma_{\text{C}}}{\partial \sigma_p} \right)^{\text{T}} (\sigma_{\text{C}} - \sigma_{\text{FEM}}) \, \mathrm{d}\Omega = 0. \tag{13.50}$$

As the continuous stress field is a linear function of the values at the mesh points, we immediately obtain

$$\sum_q \int_\Omega \xi_p \xi_q \sigma_q \, \mathrm{d}\Omega - \int_\Omega \xi_p \sigma_{\text{FEM},p} \, \mathrm{d}\Omega = 0. \tag{13.51}$$

This expression needs some explanation. For the point $p$, the points $q$ run over $p$ and all its neighbours. The functions $\xi_p$ and $\xi_q$ are defined within the same triangle;

$\sigma_{\mathrm{FEM},p}$ is the (constant) stress in that triangle. Therefore we can evaluate the product of the matrix

$$\int_{\Omega} \xi_p \xi_q \, \mathrm{d}\Omega \tag{13.52}$$

with the vector $\sigma_p$ again as a sum over all triangular elements.

We can evaluate the resulting matrix equation in exactly the same way as the full matrix equation which we have solved in order to find the displacement. However, the present problem is usually solved within about 10% of the time needed for the full elasticity problem.

It is interesting to calculate the local error. For the problem we are focusing on, a beam attached to a wall, the corners where the beam is attached to the wall are the points where the error is maximal.

There exist other methods for calculating the local error. Superconvergent patch recovery (SPR) is based on the notion that the error oscillates throughout the elements – hence, there exist points where the error vanishes. Even if those points cannot be found, some points can be identified where the error is an order of magnitude better than average. These points usually are somewhere near the centre of the elements – the vertices are the worst possible points. Using the values at the superconvergent points, a much more accurate stress field can be constructed, and the difference between this field and the FEM field is used as the local error. For details see Refs. [8, 11, 12].

## 13.5 Local refinement

The local error can be used to decide which elements should be refined. Local refinement of triangles is a subtle problem mainly for two reasons. The first is that when a triangle is refined by dividing it up into two triangles as in Figure 13.5(a), the resulting triangles might have an awkward shape. The point is that narrow, long triangles are not suitable for FEM calculations because they give rise to large errors. Therefore, it is good practice to construct the new triangles by bisecting the longest edge of a triangle.

The second problem is that if we perform such a bisection, another triangle, sharing the same long edge, should be partitioned as well, as it would be impossible to have continuity of the solution otherwise (see Fig. 13.5(b)). Rivara therefore devised the following refinement procedure [13]:

- If a triangle needs refinement, we bisect its longest edge;
- If this edge is also the largest edge of the neighbouring triangle, this triangle should also be divided via bisection of the same edge;
- If the edge is not the longest edge of the neighbouring triangle, this triangle should be refined by bisecting its longest edge.

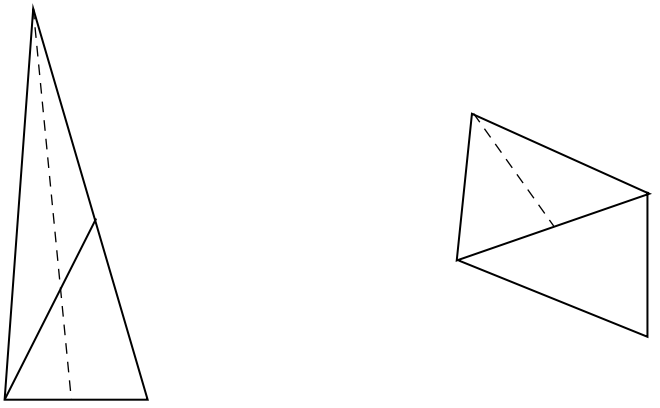


Figure 13.5. Refinement of triangular grid. (a) Two ways of partitioning a triangle – partitioning according to the dashed line is undesired. (b) It is not allowed to partition a triangle by bisecting an edge without partitioning its neighbour along the same edge.

This procedure is recursive in nature. It boils down to the following algorithm, starting from the triangle T which is to be refined:

```
ROUTINE RefineTriangle(T)
    Find the longest edge E of T;
    IF E is not the longest edge of the neighbouring triangle T′ THEN
        RefineTriangle(T′);
    END IF;
    Create a new mesh point by bisection of E;
END ROUTINE RefineTriangle.
```

Note that the routine does not generate triangles, but vertices. It is important to store the information concerning which vertices are neighbours. The new triangles can then be constructed from these data. In order to do this, we must make sure that, for each vertex, we have an array containing the neighbours of that vertex, ordered anticlockwise. If the vertex is a boundary point, the list starts with the leftmost neighbour and proceeds until the rightmost neighbour is reached. For vertices in the bulk, there is obviously no natural 'first' and 'last' neighbour: the first point is the right neighbour of the last point, and obviously the last point is then the left neighbour of the first.

For each vertex, all neighbouring triangles can be found by taking the vertex itself together with two subsequent neighbours. In this way, however, a triangle in the bulk would be counted three times. A way to define the triangle uniquely is by

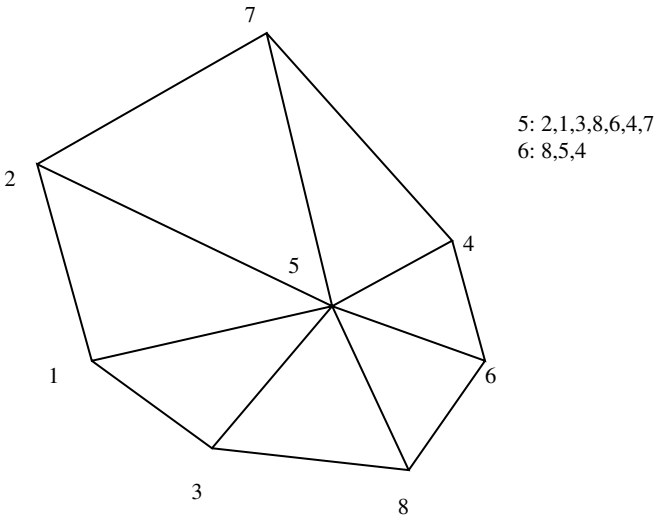


Figure 13.6. The data structure proposed by Rivara [13]. The points of the mesh are numbered in some way, and for each point, the neighbours are kept in a list. The list for a central point (number 5) is cyclic: the first point is connected to the last one, whereas a boundary point has a noncyclic neighbour list. Each triangle is counted only from the vertex with the lowest index possible.

requiring that the vertex we start from has a lower index than the two neighbours with which it forms the triangle.

The data structure is clarified in Figure 13.6.

Once we have a list of vertices with a list of neighbours for each vertex according to the rules specified above, the triangles can be generated straightforwardly:

```
FOR each vertex DO
    FOR each triangle spanned by the vertex
                    and two of its subsequent neighbours DO
        IF the central vertex has a lower index than the two neighbours THEN
            Add triangle to the list of triangles;
        END IF;
    END FOR;
END FOR
```

The line 'FOR each triangle spanned by the vertex and two of its subsequent neighbours DO' is different for edge points, where we only look at subsequent neighbour pairs *between the first and the last* neighbouring vertex, than for interior points, where we also include the pair formed by the first and the last neighbouring vertex.

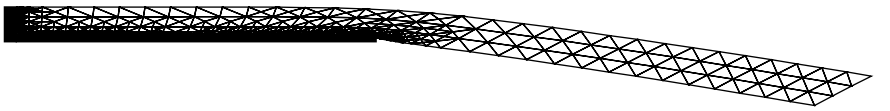

Figure 13.7. Deformed elastic beam which is attached to a vertical wall and supported over half its length. The difference from Figure 13.4, which shows the same beam, is the local refinement of the elements.

When the refinement procedure is carried out, we simply add the new vertices to the list of vertices. After the mesh has been refined, we construct the new list of triangles using the above algorithm.

The question is what the best measure of the error would be. We could take the $L_2$ norm of the difference between $\sigma$ and $\sigma_{\mathrm{FEM}}$. There are many other possibilities, and a very common one is the 'energy norm', defined as

$$e_{\mathrm{E}} = \int_{\Omega} (\sigma - \sigma_{\mathrm{FEM}})^{\mathrm{T}} \mathbf{C} (\sigma - \sigma_{\mathrm{FEM}}) \, \mathrm{d}^3 r. \tag{13.53}$$

Figure 13.7 shows the deformation of a beam which is attached to a wall and to a horizontal line over part of its lower edge. As is to be expected, the mesh is strongly refined near the sharp edge where the horizontal fixed line ends.

The use of adaptive refinement may give tremendous acceleration when a highly accurate solution is wanted for a heterogeneous problem.

## 13.6 Dynamical finite element method

In the previous sections we have assumed that dissipative forces remove all the kinetic energy so that an elastic object subject to forces will end up in a shape in which its potential energy is minimal. We may, however, also consider nondissipative dynamics in the elastic limit. We treat this case by formulating the total energy as a sum of the elastic energy, the work done by external forces and the kinetic energy:

$$H = \frac{1}{2} \int_{\Omega} \boldsymbol{\varepsilon}^{\mathrm{T}}(\mathbf{r}) \mathbf{C} \boldsymbol{\varepsilon}(\mathbf{r}) \, \mathrm{d}^3 r + \int_{\Omega} \mathbf{f}(\mathbf{r}) \cdot \mathbf{u}(\mathbf{r}) \, \mathrm{d}^3 r + \frac{1}{2} \int_{\Omega} \rho(\mathbf{r}) \dot{\mathbf{u}}^2(\mathbf{r}) \, \mathrm{d}^3 r. \tag{13.54}$$

We can perform the integrals as above, taking the mass density constant over a triangle, leading to

$$\mathbf{M}\ddot{\mathbf{v}} = -\mathbf{K}\mathbf{v} + \mathbf{G}\mathbf{f}. \tag{13.55}$$

The matrix $\mathbf{M}$ is the *mass matrix*. Putting the expressions for the natural coordinates in the integral containing the mass density, we find for the mass matrix $\mathbf{m}$ of a single triangle

$$m_{pq} = \frac{\rho A}{12}(1 + \delta_{pq}). \tag{13.56}$$

Here $\rho$ is the (average) mass density on the triangle. The global mass matrix is constructed from the local mass matrices in the same way in which the global stiffness matrix was found.

Adding dynamics to the program is a relatively small addition to the static program which was described in the previous sections. The solution of the equations of motion, however, is a bit more involved. This equation is not diagonal in the mass as is the case in the many-body dynamics of molecular dynamics simulations. Formulating the discrete solution using the midpoint rule

$$\mathbf{M}[\mathbf{u}(t+h) + \mathbf{u}(t-h) - 2\mathbf{u}(t)] = h^2(-\mathbf{K}\mathbf{v} + \mathbf{G}\mathbf{f}) \tag{13.57}$$

shows that, knowing the solution $\mathbf{u}$ at the times $t$ and $t-h$, we can predict its value at $t+h$ by solving an implicit equation. We can again use the conjugate gradient method for this purpose. This algorithm should be applied at each time step. As the solution to be found is close to the solution we had at the last time step, the conjugate gradient method will converge in general much faster than for a stationary state problem for which the initial solution is still far away from the final one (in the first case we speak of a *transient problem*). The difference between the two problems is the same as that between solving the diffusion equation (transient) and the Poisson/Laplace equation. It is also possible to add friction to the dynamics. A damping matrix is then introduced which has a shape similar to the mass matrix, but this is multiplied by the first time derivative of $\mathbf{u}$ rather than the second derivative. Obviously, the eigenvalues of the damping matrix must be negative (otherwise, there would be no damping).

A dynamical simulation shows an object wobbling as a result of external forces or of being released from a nonequilibrium state. In general, we see elastic waves propagating through the material.

## 13.7 Concurrent coupling of length scales: FEM and MD

If we exert strong forces on an object, there will be deviations from elastic behaviour due to the fact that a second order approximation of the potential energy in terms of the strain breaks down. New phenomena may then occur: in the first place, we see a change in speed of the elastic waves; moreover they start interacting, even in the bulk.[1] The most spectacular deviation from elastic behaviour occurs when we break the material. The elastic description fails completely in that case. In fact, when an object is broken or cut, the bonds between rows of atom pairs are broken and an accurate description should therefore include atomic details, preferably at the quantum level. The problem is that, although such a description is adequate for

[1] Elastic waves can also interact at the boundary of an object by coupling between the transverse and longitudinal components.

processes taking place near the fissure and far away from it, it becomes unfeasible when we want to include substantially large (parts of) objects. You may ask why we would bother about the processes far from a fissure, since the deviations of the atoms from their equilibrium positions are very small there. However, the energy released by breaking a bond will generate elastic waves into the bulk, which, when the bulk is small, will bounce back at the boundary and reinject energy to the fissure region. It is possible to couple an atomic description to an elastic medium which then carries the energy sufficiently far away . This is done by *concurrent coupling of length scales* [14, 15]. In this technique a quantum mechanical tight-binding description is applied to the region where the most essential physics is taking place: in our example this is the breaking of atomic bonds. The surrounding region is described by classical MD. Farther away, this description is then replaced by an elastic one, which is treated by finite elements. We shall not describe the full problem here – for this we refer to the papers by Broughton, Rudd and others [14, 15]. We shall, however, show that elastic FEM can be coupled to MD in a sensible way.

From the chapter on MD, it is clear that we would like to have dynamics described by a Hamiltonian. The dynamic FEM method has this property, and this is also the case for the MD method. We must ensure that this requirement is satisfied by the coupling regime. The coupling between FEM and MD is called *handshaking*. To show how this coupling is realised and to check that it gives sensible behaviour, we consider a 2D rectangular strip through which an elastic wave is travelling. The left hand side of the strip is treated using the FEM, the right hand side by MD. In order to realise the handshaking protocol, the finite element grid should approach atomic resolution near the boundary – grid points next to the boundary should coincide with equilibrium atomic positions of the MD system.

Within the MD, we use a Lennard–Jones potential as in Chapter 8. The equilibrium configuration with this potential in two dimensions is a triangular lattice. The situation is shown in Figure 13.8. The vertical dashed line in Figure 13.8 separates the FEM from the MD region. The Hamiltonian is composed of three parts: a FEM Hamiltonian for the points inside the FEM region, a MD Hamiltonian for the points in the MD region, and a handshaking Hamiltonian which contains the forces that the MD particles exert on the FEM points and vice versa. In order to have a smooth transition from one region to the other, this handshake Hamiltonian interpolates between a FEM and a MD Hamiltonian. It is built up as follows.

- The FEM triangles in the shaded region carry half of the FEM Hamiltonian; that is, we formulate the usual FEM Hamiltonian in this region, but multiply it by $1/2$.
- The points of the shaded region lying right of the dashed vertical line couple via a MD Hamiltonian to the points on the left, but this Hamiltonian is also multiplied by 1/2. Note that such couplings involve in general more than nearest neighbour points on the triangular grid – we neglect those here.

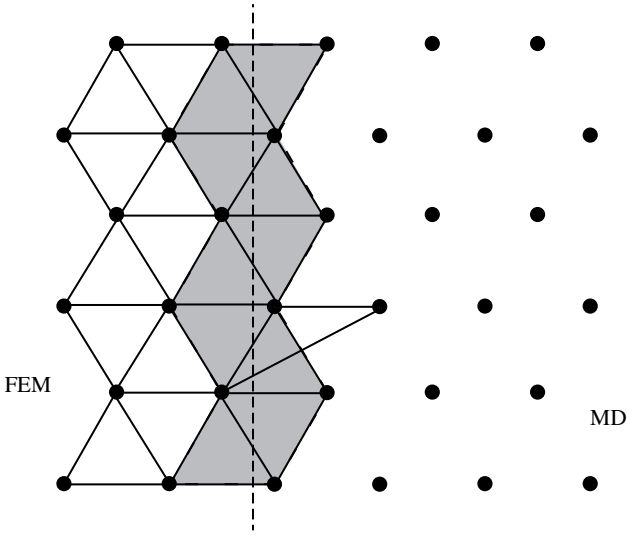


Figure 13.8. Strip modelled partly by finite elements and partly by molecular dynamics.

Three remarks are in place. In the original formulation [14], the MD region is three-dimensional, whereas the FEM region is only two-dimensional. The transition is made by averaging the MD points over the $z$-direction which is taken perpendicular to the FEM grid. Here we shall consider the strictly two-dimensional case for simplicity. The second remark concerns the treatment of the FEM masses. As we have seen above, the mass matrix couples the kinetic degrees of freedom at the vertices of the FEM triangles. However, in the handshake region, we strictly want to assign the mass of a real atom to the point. For this reason, we use the *lumped mass* approximation in the finite element description. In this formulation, we assign one-third of the mass of each triangle to each of its vertices. This means that the mass matrix has become diagonal, so that the numerical integration of the equations of motion has become much simpler as the solution of an implicit equation at each time step is avoided. The FEM mass is derived from the MD equilibrium by requiring that the same amount of mass is present per unit area in both descriptions.

The final remark is that the boundaries of the system in the MD and FEM description do not fit onto each other. In the FEM description, the triangles are taken uniform, but a MD system with a boundary will have a slightly smaller distance between the outermost layers than in the bulk, as a result of the fact that the next nearest-neighbour interactions pull the outermost particles slightly more towards the interior of the system. This deviation is minor so we do not correct for it.

Obviously, we could take periodic boundary conditions in the transverse direction, which implies a cylindrical description (this could be useful for describing a carbon nanotube). However, this is not compatible with longitudinal waves, as the Poisson ratio causes the system to expand where it is longitudinally compressed and vice versa. A periodic boundary condition would not allow for this to happen and cause unphysical, strong stresses to build up.

Once a FEM and a MD program are working, it is not so much work to couple them along the lines decribed above. We use the velocity Verlet algorithm which naturally splits into two steps, separated by a force calculation.
First step:

$$\mathbf{p}_i(t+h) = \mathbf{p}_i(t) + \frac{h}{2}\mathbf{F}_i[\mathbf{r}(t)]; \tag{13.58a}$$

$$\mathbf{r}_i(t+h) = \mathbf{r}_i(t) + h\mathbf{p}_i(t); \tag{13.58b}$$

Calculate $\mathbf{F}_i[\mathbf{r}(t)]$;

Second step:

$$\mathbf{p}_i(t+h) = \mathbf{p}_i(t+h) + \frac{h}{2}\mathbf{F}_i[\mathbf{r}(t+h)]$$

These steps must be kept in mind when setting up the algorithm for the full system. This algorithm looks as follows:

```
Calculate MD forces;
Calculate FEM forces;
Copy locations of leftmost MD points to a shadow array
                in the FEM region;
Copy locations of rightmost FEM points to a shadow array
                in the MD region;
FOR TimeStep = 1, MaxStep DO
    Set Initial values of boundary points;
    Do first integration step (see Eq. (13.58a));
    Copy locations of leftmost MD points to a shadow array
                in the FEM region;
    Copy locations of rightmost FEM points to a shadow array
                in the MD region;
    Calculate forces in FEM region, including those on the MD particles;
    Calculate forces in MD region, including those on the FEM particles;
    Add FEM forces acting on MD particles to MD forces;
    Add MD forces acting on FEM particles to FEM forces;
    Do second integration step [see Eq. (13.58b)];
END FOR.
```

Obviously, we should investigate which elasticity matrix should be used in the FEM domain. This is fully determined by the MD interaction, for which we take the pairwise Lennard-Jones interaction. We can evaluate the elastic constants by allowing the MD unit cell to deform, as is done in the Parrinello–Rahman method [16]. Another method is to measure the stretch resulting from a force applied to the left- and rightmost particles for a strip of atoms, fully described by MD. The lateral shrink as a result of the end-to-end stretch then gives us the Poisson ratio. For simplicity, we shall consider here the $T = 0$ limit, for which we can calculate the elasticity matrix analytically from the pair potential. The idea is that we can Taylor-expand the total energy per unit area with respect to the strain to second order, which corresponds precisely to how the elasticity matrix is defined: the change in energy per unit area resulting from a strain field $\boldsymbol{\varepsilon}$ is given by

$$\delta V = \frac{1}{2\Omega} \int_{\Omega} \boldsymbol{\varepsilon}^{\mathrm{T}} \mathbf{C} \boldsymbol{\varepsilon} \, \mathrm{d}^2 r. \tag{13.59}$$

In our case, we have for the total energy per unit area at small deviations, in the bulk:

$$\delta V = \frac{1}{2\Omega} \sum_{i \neq j} \left[ \frac{\partial V(\mathbf{R}_0)}{\partial r_{ij}^{\alpha}} (\delta r_i^{\alpha} - \delta r_j^{\alpha}) + \frac{1}{2} \frac{\partial^2 V(\mathbf{R}_0)}{\partial r_{ij}^{\alpha} \partial r_{ij}^{\beta}} (\delta r_i^{\beta} - \delta r_j^{\beta})(\delta r_i^{\alpha} - \delta r_j^{\alpha}) \right]. \tag{13.60}$$

Greek indices $\alpha$ and $\beta$ denote the Cartesian coordinates – they are summed over according to the Einstein summation convention. Equation (13.60) is nothing but a Taylor expansion to second order for the potential in terms of the coordinates. In equilibrium, the second term vanishes as the total force on each particle vanishes. We may write $\delta r_i^{\alpha} = u_i^{\alpha}$, where $u_i$ has precisely the same meaning as in the formulation of the finite element method: it is the deviation from equilibrium of coordinate $\alpha$ of particle $i$. Now we write $\mathbf{u}_{ij} = \mathbf{r}_{ij} - \mathbf{a}_{ij}$, where $\mathbf{a}_{ij}$ is the relative coordinate of particles $i$ and $j$ in equilibrium. We therefore have $u_{ij}^{\alpha} = a_{ij}^{\beta} \varepsilon_{\alpha\beta}$, so that we obtain

$$\delta V = \frac{1}{4\Omega} \sum_{i \neq j} a_{ij}^{\alpha} \varepsilon_{\alpha\beta} \frac{\partial^2 V(\mathbf{R}_0)}{\partial r_{ij}^{\beta} \partial r_{ij}^{\gamma}} \varepsilon_{\gamma\delta} a_{ij}^{\delta}, \tag{13.61}$$

and we can write

$$\tilde{C}_{\alpha\beta\gamma\delta} = \frac{1}{2\Omega} \sum_{i \neq j} a_{ij}^{\alpha} \frac{\partial^2 V(\mathbf{R}_0)}{\partial r_{ij}^{\beta} \partial r_{ij}^{\gamma}} a_{ij}^{\delta}. \tag{13.62}$$

For a pair potential, this can be worked out further to yield

$$\tilde{C}_{\alpha\beta\gamma\delta} = \frac{1}{2\Omega}\sum_{i\neq j}\left\{\frac{1}{r_{ij}^2}\left[V''(r_{ij}) - \frac{1}{r_{ij}}V'(r_{ij})\right]a_{ij}^{\alpha}a_{ij}^{\beta}a_{ij}^{\gamma}a_{ij}^{\delta}\right\}. \tag{13.63}$$

We have used the tilde ($\sim$) for the elasticity matrix because it is given in terms of the $xy$ components of the strain. The relation with the **C** matrix given above for two dimensions, which used $(\partial u_x/\partial y + \partial u_y/\partial x)/2$ as the third component, is given by

$$C_{11} = \tilde{C}_{xxxx}, \qquad C_{22} = \tilde{C}_{yyyy} \tag{13.64a}$$

$$C_{12} = \tilde{C}_{xxyy}, \qquad C_{21} = \tilde{C}_{yyxx} \tag{13.64b}$$

$$C_{13} = \frac{1}{2}(\tilde{C}_{xxxy} + \tilde{C}_{xxyx}), \qquad C_{23} = \frac{1}{2}(\tilde{C}_{yyxy} + \tilde{C}_{yyyx}) \tag{13.64c}$$

$$C_{33} = \frac{1}{4}(\tilde{C}_{xyxy} + \tilde{C}_{xyyx} + \tilde{C}_{yxxy} + \tilde{C}_{yxyx}) \tag{13.64d}$$

For a Lennard–Jones potential we find, in reduced units:

$$C = \begin{pmatrix} 76.8 & 25.5 & 0 \\ 25.6 & 76.8 & 0 \\ 0 & 0 & 25.6 \end{pmatrix}. \tag{13.65}$$

From this we find, for the case of plane stress: $\nu = 1/3$ and $E = 68$. The fact that $\nu = 1/3$ shows an important shortcoming of a pair potential: irrespective of the specific form of the potential, a pair potential always leads to $\nu = 1/3$.

## Exercises

13.1 In this problem, we study the natural coordinates for triangles. We consider an 'archetypical' triangle as shown in Figure 13.9. Now consider a mapping of this triangle to some other triangle, also shown in Figure 13.9. This can be obtained from the archetypical one by a translation over the vector $\mathbf{r}_{aa'}$, followed by a linear transformation. The matrix **U** of this linear transformation can be found as

$$\mathbf{U} = \begin{pmatrix} x'_b - x'_a & x'_c - x'_a \\ y'_b - y'_a & y'_c - y'_a \end{pmatrix}, \tag{13.66}$$

where $(x'_a, y'_a)$ are the Cartesian coordinates of the vector $\mathbf{r}_{a'}$ etc. We have

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \mathbf{U}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x_a \\ y_a \end{pmatrix} \tag{13.67}$$

Now we take for the natural coordinates in the archetypical triangle $x$, $y$ and $1 - (x + y)$. It is clear that these coordinates assume the value 1 on $a$, $b$ and $c$ respectively and vanish at the other points. We want the linear transformation of these coordinates to have the same property. We therefore consider the function

$$g(x', y') = f(x, y)$$

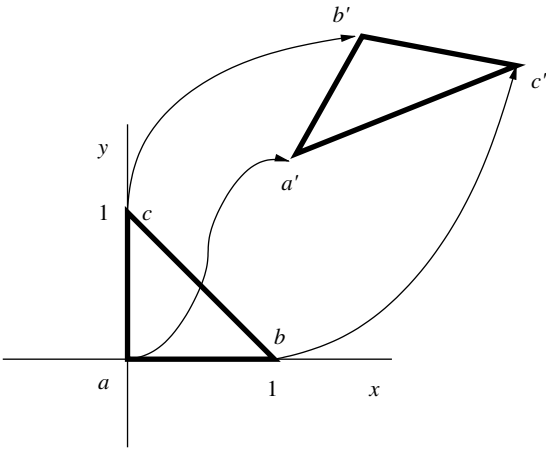


Figure 13.9. Archetypical triangle with two angles of 45° and sides 1, oriented along the $x$- and $y$-axes. Another triangle is shown, which can be obtained from the archetypical one through a linear transformation.

where $f(x, y) = x$, say, and where $x', y'$ are the images of $x, y$ under the transformation $\mathbf{U}$. It now is straightforward to verify that the expressions for the natural coordinates (13.11) are correct.

## References

[1] O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method: Its Basis and Fundamentals*, 6th edn. Oxford/Burlington (MA), Elsevier Butterworth-Heinemann, 2005.
[2] O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method for Solid and Structural Mechanics*, 6th edn. Oxford/Burlington (MA), Elsevier Butterworth-Heinemann, 2005.
[3] O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method for Fluid Dynamics*, 6th edn. Oxford/Burlington (MA), Elsevier Butterworth-Heinemann, 2005.
[4] K. J. Bathe, *Finite Element Procedures*. Upper Saddle River, NJ, Prentice Hall, 1996.
[5] J. Mackerle, *A Primer for Finite Elements in Elastic Structures*. New York, Wiley, 1999.
[6] W. Kaplan, *Advanced Calculus*, 4th edn. Reading (MA), Addison–Wesley, 1991.
[7] J. T. Oden and H. J. Brauchli, 'On the calculation of consistent stress distributions in finite element calculations,' *Int. J. Numer. Meth. Eng.*, **3** (1971), 317–25.
[8] E. Hinton and J. S. Campbell, 'Local and global smoothing of discontinuous finite element functions using a least squares method,' *Int. J. Numer. Meth. Eng.*, **8** (1974), 461–80.
[9] O. C. Zienkiewicz and J. Z. Zhu, 'A simple error estimator and adaptive procedures for practical engineering analysis,' *Int. J. Numer. Meth. Eng.*, **24** (1987), 337–57.
[10] O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method*, vol. I. London, McGraw-Hill, 1988.
[11] J. Barlow, 'Optimal stress locations in finite element models,' *Int. J. Numer. Meth. Eng.*, **10** (1976), 243–51.
[12] J. Barlow, 'More on optimal stress points, reduced integration, element distortions and error estimation,' *Int. J. Numer. Meth. Eng.*, **28** (1989), 1487–504.

[13] M.-C. Rivara, 'Design and data structure of fully adaptive, multigrid, finite element software,' *ACM Trans. Math. Software*, **10** (1984), 242–64.

[14] J. Q. Broughton, F. F. Abraham, N. Bernstein, and E. Kaxiras, 'Concurrent coupling of length scales: methodology and application,' *Phys. Rev. B*, **60** (1999), 2391–403.

[15] R. E. Rudd and J. Q. Broughton, 'Concurrent coupling of length scales in solid state systems,' *Phys. Stat. Sol.* (*b*), **217** (2000), 251–91.

[16] M. Parrinello and A. Rahman, 'Polymorphic transitions in single crystals: a new molecular dynamics method,' *J. Appl. Phys.*, **52** (1981), 7182–90.