Modelling elastic media with the wavelet transform

João Willy Corrêa Rosa, Fabbryccio Akkazzha C. M. Cardoso, Keiiti Aki,
Henrique S. Malvar, Fredy A. Villaorduña Artola, José Wilson Corrêa Rosa

Modelling elastic media with the wavelet transform

João Willy Corrêa Rosa,[1] Fabbryccio Akkazzha C. M. Cardoso,[2] Keiiti Aki,[3] Henrique S. Malvar,[4] Fredy A. Villaorduña Artola[1] and José Wilson Corrêa Rosa[1]

[1] *Instituto de Geociências, Universidade de Brasília, 70910-900 Brasília, DF Brazil. E-mail: jwilly@unb.br*
[2] *Departamento de Engenharia Elétrica, Universidade de Brasília, 70910-900 Brasília, DF Brazil*
[3] *Institut de Physique du Globe de Paris, 14 Route Nationale 3, 27ème km, 97418 La Plaine des Cafres, Isle de la Reunion, France*
[4] *Microsoft Research, Microsoft Corporation, One Microsoft Way, Redmond, WA 98052-6399, USA*

SUMMARY

We present a new method for modelling 2-D elastic media with the application of the wavelet transform, which is also extended to cases where discontinuities simulate geological faults between two different elastic media. The basic method consists of the discretization of the polynomial expansion for the boundary conditions of the 2-D problem involving the stress and strain relations for the media. This parametrization leads to a system of linear equations that should be solved for the determination of the expansion coefficients, which are the model parameters, and their determination leads to the solution of the problem. The wavelet transform is applied with two main objectives, namely to decrease the error related to the truncation of the polynomial expansion and to make the system of linear equations more compact for computation. This is possible due to the properties of this finite length transform. The method proposed here was tested for six different cases for which the analytical solutions are known. In all tests considered, we obtained very good matches with the corresponding known analytical solutions, which validate the theoretical and computational parts of the project. We hope that the new method is useful for modelling real media.

Key words: elastic media, modelling, wavelet transform.

1 INTRODUCTION

We propose, in this work, a new method of modelling 2-D elastic media. Associated with this new method, we use the wavelet transform to minimize truncation errors and to reduce the size of the system of linear equations that results from the application of the method. Despite the fact that the media we study are 2-D bodies, the result of the application of this new method can be viewed as an approximate solution to some specific 3-D problems.

Among the motivations for developing this method are possible geological applications (that is, the study of tectonic plates and geological faults) and simulations of the elastic behaviour of materials in several other fields of science.

We know that any given 2-D body may be subject to stress and displacements, which may be due to an external perturbation and whose behaviour depends on the geometry and the physical properties of the body. According to de Veubeke (1979), the stress field for such a plane body is related to the displacement components through a system of differential equations of two variables (x, y). The solution to this system can sometimes be obtained analytically. Once this system is solved, we obtain expressions for the stress and for the displacement components, which depend on two analytical functions, F and H, of the x and y variables.

The basic principle of the method is the discretization of the boundary conditions on the x and y space coordinates. These conditions are imposed on the stress and displacement components along a given boundary contour of the medium. In order to make the method feasible, the boundary conditions, which depend on the F and H analytical functions, have to be chosen in a way that allows the problem to be solved in the simplest manner. With this in mind, assume that F and H are given by

$$F(x, y) = \sum_{m=0}^{M} a_m f_m(x, y) \qquad \text{and} \qquad H(x, y) = \sum_{m=0}^{m} b_m g_m(x, y).$$

Note that F and H are linear functions of terms containing constants a_m and b_m, which are the model parameters. This means that the boundary conditions are also linear combinations involving these constants. When we make these boundary conditions discrete by taking sample points (x_n, y_n) along a contour, this results in a system of linear equations whose unknowns are the expansion coefficients of the F and H functions, namely f_m and g_m, respectively. Once we have solved this system of equations, we can determine which are the tension and the displacement components at any given point along the contour we are considering. This is useful to simulate how the medium will behave in terms of its stress field and its deformations when an external perturbation is applied.

However, we have not yet considered the whole modelling problem. It remains to decide which type of expansion we will use. We anticipate here (this will be discussed in detail below) that the expansion adopted will consider that F and H are polynomials. The main reason for this choice is the simplicity we have for the related analytical solutions to the stress and displacement components, although this is not a limitation to the use of other expansion functions in the method.

The digital signals we consider here, the stress and the displacement components are polynomial functions in the (x, y) 2-D space. In order to make the analysis simpler, assume first that these signals are only functions of x, and that the equation for the displacement component u has the following form:

$$u = \sum_{m=0}^{M} A_m x^m. \tag{1}$$

In order to determine the A_m ($m = 0, 1, \ldots M$) coefficients, we have to match, in the space or frequency domain, eq. (1) with another function at some points, say $\bar{U}(x_n)$, in the time domain or $\textbf{FT} \bar{U}_n$ (where $\textbf{FT}$ is the discrete Fourier transform operator) in the frequency domain, as shown below:

$$\bar{U}(x_n) = \sum_{m=0}^{M} A_m x_n^m, \qquad n = 0, 1, \ldots, N, \tag{2}$$

$$\textbf{FT} \bar{U}_n = \sum_{m=0}^{M} A_m X_{m,n}, \qquad n = 0, 1, \ldots, N. \tag{3}$$

Note that in both cases we have a system of $N+1$ linear equations such that we can have, at the most, $N+1$ unknowns. This means that, in both eq. (2) and eq. (3), we have to truncate the summation to a maximum of $N+1$ terms, which will generate a truncation error that will affect the solution to A_m. Since the signals considered here all have their energy mainly concentrated in the low-frequency components, the truncation error due to eq. (3) will be smaller than the error due to eq. (2), which will justify the application of the Fourier transform, which is the basis function transform commonly used in this kind of problem (e.g. Aki & Larner 1970).

According to our comments above for the determination of A_m, we can match the displacement component given by eq. (1) in the space or in the frequency domain with a known function. Assume now that we have a basis function transform available that combines both domains (space and frequency). It seems that there would be a greater flexibility in the matching described above because we would use both domains to match the two functions. The truncation error may thus be smaller than that due to the matching in the frequency domain alone. In fact, these transforms (which combine two domains) exist and are known as wavelet transforms (Daubechies 1992; Vetterli & Kovacevic 1995). Another reason for using the wavelet transform instead of the Fourier transform is the fact that the former concentrates more energy of the signal in a few basis functions than the latter, which can make the truncation error smaller.

We also have problems when we face cases where the polynomial function used is larger than necessary. This will not introduce errors, but will make the size of the matrix equation much larger than necessary, making the computation slower and the computer memory space required larger. From this, we get the idea of using the wavelet transform, which can concentrate the energy of a given signal in a few basis functions and make the system of equations more compact. This will allow the method to zero those coefficients that are much smaller than the largest one established for a given signal.

Another major source of error in this type of solution to the problem is that included in the pseudo-inverse that we will use here. This is an inversion technique based on the least-squares minimum of the difference between data and fitted results, on the basis of the singular value decomposition of the matrix that we want to invert. We cannot do much about avoiding this type of error. We can only try to estimate it, since this inversion method is among the methods available for dealing with problems of ill-conditioned matrices and for the solution of overdetermined systems of equations. Fortunately, despite the fact that the rank of the matrices obtained in the example cases we show here is always incomplete, it was always possible to establish one numerical rank without any ambiguity, which allowed us to have small error levels.

So, the main objective of this paper is the development of a new method of modelling that takes into account plane elastic bodies. We use wavelet transforms in order to reduce truncation errors and to make the resulting system of equations more compact.

In order to make the method useful for the study of geological models, or for the study of materials in other fields of science, it is necessary to validate the method through application to models with very simple geometry, whose analytical solutions are known and with more complex geometries, which take into account media with different elastic properties and include faults. The results for these cases prove that this is a very simple and useful method for modelling plane elastic media.
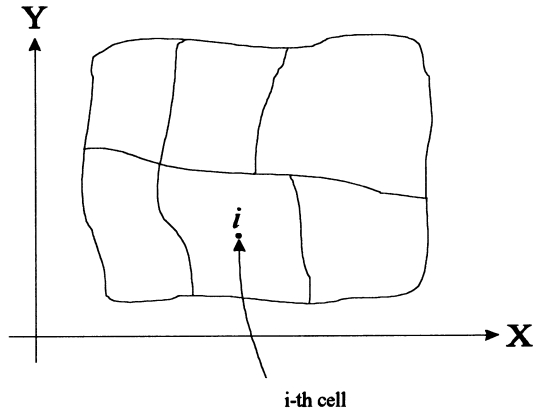
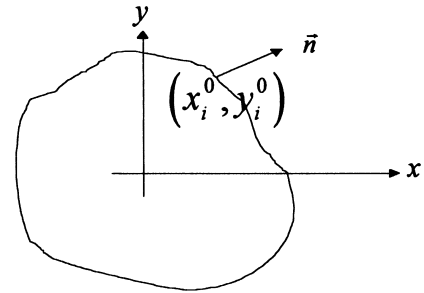Figure 1. Definition of the geometry of the general system.

Figure 2. Contour of a cell including the local coordinate system.

2 MODELLING THE GENERAL SYSTEM

Parts of the original development of this work are presented in the Appendix for the sake of clarity in the present text.

We define in Fig. 1 the general (arbitrary) system. The general system is composed of a region divided into I cells. These cells are named using an index, say i $(=1, 2, \ldots, I)$. Each cell is homogeneous and elastic, defined by coefficients G_i and v_i (rigidity and Poisson's ratio). They include a central point with coordinates (x_i^0, y_i^0) with respect to the main coordinate axis system (X, Y). We also introduce a specific coordinate system for each cell (Fig. 2),

$$\begin{cases} x = X - x_i^0 \\ y = Y - y_i^0 \end{cases},$$

(4)

and a vector of coefficients $\mathbf{x}_i$, which has the following components:

$$\mathbf{x}_i^{\mathrm{T}} = \left(a_0^{(i)}, b_0^{(i)}, p_0^{(i)}, q_0^{(i)}, a_1^{(i)}, b_1^{(i)}, p_1^{(i)}, q_1^{(i)}, \ldots a_M^{(i)}, b_M^{(i)}, p_M^{(i)}, q_M^{(i)} \right).$$

(5)

This vector contains the coefficients that describe the state of stress and displacement through the polynomial functions (A7), (A8), (A9), (A10) and (A11), where M is the order of the polynomial function used. Once we know these coefficients, we can calculate the stress (σ_x, σ_y, τ_{xy}) and the displacement (u, v) components.

According to the above, each cell has a set of coefficients that range from zero to order M. On the other hand, it is not possible to determine the coefficients for each cell separately since there is interaction among the cells determined by the internal contours of the model (that is, determined by those contours that separate each pair of neighbouring cells). Note that, according to eqs (5), (A7), (A8), (A9), (A10) and (A11), each cell has its own stress and displacement fields, indicating that each will be deformed in its own way. It is also important to remember that there may or may not be displacement of one cell with respect to another given cell of the model. This will depend on how we define the boundary conditions of the internal borders among the cells. If we allow displacement to exist between two given cells, the interface between these cells can be considered as a fault.

We now define the contour of each cell assuming that it encloses the centre of coordinates of the local coordinate system (x, y) (Fig. 2). We also define a vector $\mathbf{n}(n_x, n_y)$ normal to the contour line. Then, the traction components to the contour is given by eqs (A22) and (A23), which we rewrite as

$$\begin{cases} t_x = \sigma_x n_x + \tau_{xy} n_y \\ t_y = \tau_{xy} n_x + \sigma_y n_y \end{cases}.$$

The x-component of the traction, t_x, is defined as the force per unit of area in the x-direction that is applied by the side of the boundary that contains the vector normal to the other side, $\mathbf{n}$. The definition of t_y is analogous to the definition of t_x.

We then introduce the contour segments that divide the contours into appropriate sections. We number the contour segments using two indices: the first defines the number of the cell we are considering, while the second indicates the border along the contour in the counterclockwise direction (Fig. 3).

The starting point of segment ($i1$) is labelled $P(i, 1)$ and has coordinates (x_{i1}, y_{i1}), the starting point of segment ($i2$) [which is the endpoint of the segment ($i1$)] is called $P(i, 2)$ and has coordinates (x_{i2}, y_{i2}), and so on.

Some of the segments are along the external boundary of the system, and others are along the internal boundaries. If the segment we are considering is an internal one, the same segment is also part of a neighbouring cell. We apply a boundary condition to the external segments and one continuity condition (or fault condition, if this is the case) to the internal segments. The external boundary
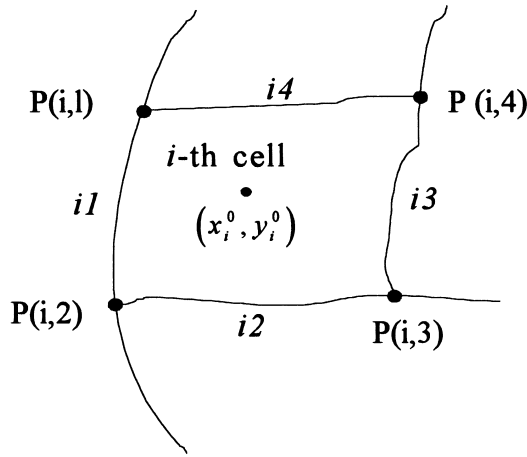
Figure 3. Parametrization of a contour line into segments (or borders) and the convention used to number these segments.
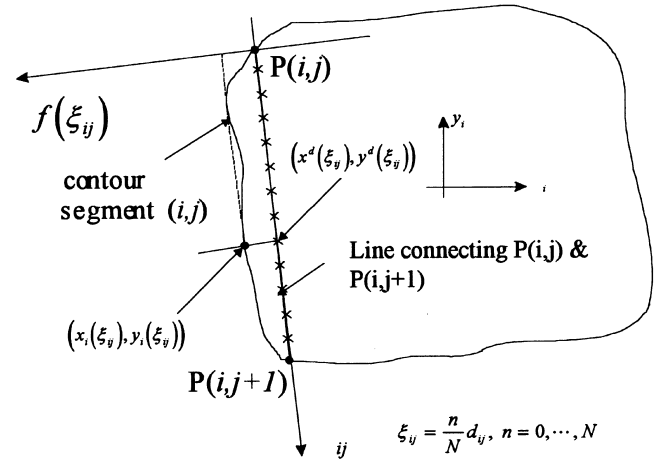
Figure 4. Parametrization of a given contour.

condition will be imposed on the traction, or on the displacement we observe, on the boundary we are considering. In cases of internal segments, the continuity boundary condition is imposed on the traction (t_x, t_y) and on the displacement (u, v). The fault boundary condition (when this is the case) consists of a discontinuity on the displacement and on the continuity of the traction.

Before we start analysing the boundary conditions, we should specify how we consider any given segment ($i1$) in this problem (Fig. 4).

2.1 Contour parametrization

For any given (i, j) segment we define the discrete variable ξ_{ij}, which has known values along the straight-line segment shown in Fig. 4. These values are taken in the counterclockwise direction with respect to the local axis of coordinates of the cell. Note in Fig. 4 that the real contour of the cell is different from the straight contour we consider. The difference between these two is given by the function $f_{ij}(\xi_{ij})$, which is measured in the direction of the normal to the cell, as shown in Fig. 4.

We need the locations of the points on the contour we are considering $(x_i(\xi_{ij}), y_i(\xi_{ij}))$ and the external normal to the real contour on each of these points. With this in mind, we define the following:

the starting point of the (i, j) segment, $P(i, j) = (x_{ij}, y_{ij})$;
the endpoint of the (i, j) segment, which is the starting point of the next (i, j) segment, $P(i, j+1) = (x_{i,j+1}, y_{i,j+1})$;
the distance between the starting and the endpoints, d_{ij} (which is the length of the straight line between the two points).

The procedure is as follows.

(i) First we draw a line connecting $P(i, j)$ and $P(i, j+1)$. The coordinates of each point on the straight line segment are

$$
\begin{cases}
x_i^d(\xi_{ij}) = x_{ij}^d + \xi_{ij} \dfrac{(x_{i,j+1} x_{ij})}{d_{ij}} \\[2mm]
y_i^d(\xi_{ij}) = y_{ij}^d + \xi_{ij} \dfrac{(y_{i,j+1} - y_{ij})}{d_{ij}}
\end{cases} . \tag{6}
$$

The corresponding normal vector to the real contour is given by

$$
\begin{cases}
n_{x_i}(\xi_{ij}) = \dfrac{dy_i(\xi_{ij})}{d\xi_{ij}} \Big/ \sqrt{\left(\dfrac{dx_i}{d\xi_{ij}}\right)^2 + \left(\dfrac{dy_i}{d\xi_{ij}}\right)^2} \\[4mm]
n_{y_i}(\xi_{ij}) = -\dfrac{dx_i(\xi_{ij})}{d\xi_{ij}} \Big/ \sqrt{\left(\dfrac{dx_i}{d\xi_{ij}}\right)^2 + \left(\dfrac{dy_i}{d\xi_{ij}}\right)^2}
\end{cases} . \tag{7}
$$

In this way we know the positions of the points and the normal to the (i, j) segment as a function of ξ_{ij}.

(ii) Now we have to find the coordinates of the points on the real contour. In order to do this, we draw a line normal to the segment, from $(x_i^d(\xi_{ij}), y_i^d(\xi_{ij}))$, and find the coordinates of the point of intersection of this line with the real contour. We write the coordinates $\{x_i(\xi_{ij}), y_i(\xi_{ij})\}$ of this point of intersection (Fig. 4).

After all the segments have been discretized and one segment (i, j) has been specified, we find the point on the contour $\{x_i(\xi_{ij}), y_i(\xi_{ij})\} = \{x(i, j, n), y(i, j, n)\}$. This point will be part of a table of coordinates (x, y) with three indices that has to be supplied by the user. The three indices are

$i = 1, 2 \ldots I$ (cell number),

$j = 1, 2 \ldots j(i)$ (number of the segment in the ith cell),

$n = 1, 2 \ldots N_i$ (number of the point on the jth segment).

Note that $\{x(i, j, n), y(i, j, n)\}$ is another way of representing $\{x(\xi_{ij}), y(\xi_{ij})\}$.

2.2 External boundary conditions when the displacement is given

In this case, we obtain the external boundary conditions with respect to the displacement. Consider one condition in one segment on the external border, where the displacement (u, v) is given. Consider the jth segment of the ith cell, and find the points on the contour going from $P(i, j)$ to $P(i, j+1)$,

$$(x(\xi_{ij}), y(\xi_{ij})), \qquad \xi_{ij} = \frac{n}{N_i} d_{ij} \qquad \text{and} \qquad n = 0, 1, \ldots, N_i.$$

The u and v displacement components are functions of x and y, but, in order to make the notation simpler, we will only use the variable ξ_{ij}. Suppose that the external boundary condition is given by

$$u(\xi_{ij}) = \bar{U}(\xi_{ij}),$$

$$v(xi_{ij}) = \bar{V}(\xi_{ij}),$$

where

$$\bar{U}(\xi_{ij}) = \sum_{m=0}^{M} \left[e_m^{(a)}(\xi_{ij})a_m^{(i)} + e_m^{(b)}(\xi_{ij})b_m^{(i)} + e_m^{(c)}(\xi_{ij})p_m^{(i)} + e_m^{(d)}(\xi_{ij})q_m^{(i)} \right], \tag{8}$$

$$\bar{V}(\xi_{ij}) = \sum_{m=0}^{M} \left[f_m^{(a)}(\xi_{ij})a_m^{(i)} + f_m^{(b)}(\xi_{ij})b_m^{(i)} + f_m^{(c)}(\xi_{ij})p_m^{(i)} + f_m^{(d)}(\xi_{ij})q_m^{(i)} \right]. \tag{9}$$

We then take the wavelet transform (operator **WT**) of both sides of eq. (8) with respect to ξ_{ij} (which is the variable that is used when we discretize a contour to parametrize the 2-D problem with respect to the x, y-coordinates) and we obtain

$$\mathbf{WT}\bar{U}(\xi_{ij}) = \sum_{m=0}^{M} \left[\mathbf{WT}e_m^{(a)}(\xi_{ij})a_m^{(i)} + \mathbf{WT}e_m^{(b)}(\xi_{ij})b_m^{(i)} + \mathbf{WT}e_m^{(c)}(\xi_{ij})p_m^{(i)} + \mathbf{WT}e_m^{(d)}(\xi_{ij})q_m^{(i)} \right]. \tag{10}$$

A similar equation can be obtained for the representation of (9).

2.3 External boundary conditions when the traction is known

We now consider the boundary conditions on the external border of the model, where the traction $\mathbf{t}(t_x, t_y)$ is given instead of the displacement conditions.

Assume that the jth segment of the ith cell is in an external contour, and that the external traction (T_x, T_y) is given. Following the discretization procedure of the contour described in Section (2.1), we sample the contour between points $P(i, j)$ and $P(i, j+1)$,

$$(x(\xi_{ij}), y(\xi_{ij})), \qquad \xi_{ij} = \frac{n}{N_i} d_{ij} \qquad \text{and} \qquad n = 0, 1, \ldots, N_i.$$

In this case, we also need the normal to each of these points,

$$(n_x(\xi_{ij}), n_y(\xi_{ij})), \qquad \xi_{ij} = \frac{n}{N_i} d_{ij} \qquad \text{and} \qquad n = 0, 1, \ldots, N_i.$$

Assume that the external boundary conditions are given,

$$t_x(\xi_{ij}) = T_x(\xi_{ij}),$$

$$t_y(\xi_{ij}) = T_y(\xi_{ij}).$$

In this case, we consider that the internal traction components, t_x and t_y, are known (equal to the external traction applied) on the (i, j) contour, and that they are given by T_x and T_y. Then,

$$T_x(\xi_{ij}) = \sum_{m=1}^{M} \left\{ p_m^{(a)}(\xi_{ij})a_m + p_m^{(b)}(\xi_{ij})b_m + p_m^{(c)}(\xi_{ij})p_m + p_m^{(d)}(\xi_{ij})q_m \right\}, \tag{11}$$

$$T_y(\xi_{ij}) = \sum_{m=1}^{M} \left\{ q_m^{(a)}(\xi_{ij})a_m + q_m^{(b)}(\xi_{ij})b_m + q_m^{(c)}(\xi_{ij})p_m + q_m^{(d)}(\xi_{ij})q_m \right\}, \tag{12}$$

where we have the following expressions for p_m and q_m:

$$p_m^{(a)}(\xi_{ij}) = n_x(\xi_{ij})g_m^{(a)}(\xi_{ij}) + n_y(\xi_{ij})r_m^{(a)}(\xi_{ij}),$$

$$q_m^{(a)}(\xi_{ij}) = n_x(\xi_{ij})r_m^{(a)}(\xi_{ij}) + n_y(\xi_{ij})h_m^{(a)}(\xi_{ij}).$$

We then take the wavelet transform, with respect to n, of both sides of eq. (11),

$$\mathbf{WT}T_x(\xi_{ij}) = \sum_{m=1}^{M} \left\{ \mathbf{WT}p_m^{(a)}(\xi_{ij})a_m + \mathbf{WT}p_m^{(b)}(\xi_{ij})b_m + \mathbf{WT}p_m^{(c)}(\xi_{ij})p_m + \mathbf{WT}p_m^{(d)}(\xi_{ij})q_m \right\}. \tag{13}$$

An equation similar to (13) is obtained for T_y.

2.4 Internal boundary conditions assuming continuity of the displacement

In this case we obtain the boundary condition equations for the internal segments, assuming that there is no displacement between two cells. The internal segment is divided into two contours, (i, j) and (k, l), which are the contours seen by the ith and kth cells, respectively. An example is shown in Fig. 5.

Assume that the segment defined by $P(i, j) \rightarrow P(i, j+1)$ is the same as that defined by $P(k, l) \rightarrow P(k, l+1)$. We consider the matching of u, v, t_x, t_y for each point along the segment.

We first consider the matching of u. In eq. (8) we express $u(i, j, n)$ of the jth segment of the ith cell as

$$u(\xi_{ij}) = \sum_{m=0}^{M} \left[e_m^{(a)}(\xi_{ij})a_m^{(i)} + e_m^{(b)}(\xi_{ij})b_m^{(i)} + e_m^{(c)}(\xi_{ij})p_m^{(i)} + e_m^{(d)}(\xi_{ij})q_m^{(i)} \right].$$

The same can be done for the lth segment of the kth cell,

$$u(\xi_{kl}) = \sum_{m=0}^{M} \left[e_m^{(a)}(\xi_{kl})a_m^{(k)} + e_m^{(b)}(\xi_{ij})b_m^{(k)} + e_m^{(c)}(\xi_{kl})p_m^{(k)} + e_m^{(d)}(\xi_{kl})q_m^{(k)} \right],$$

where $\xi_{kj} = d_{ij} - \xi_{ij}$.

The point specified by (i, j, n) is the same as the point specified by $(k, l, N(i, j) - n)$, that is, $\xi_{kj} = d_{ij} - \xi_{ij}$. In this way, the continuity of u along the contour can be expressed as

$$u(\xi_{ij}) = u(\xi_{kl}).$$

Taking the wavelet transform on both sides, we have

$$\sum_{m=0}^{M} \mathbf{WT}e_m^{(a)}(\xi_{ij})a_m^{(i)} + \sum_{m=0}^{M} \mathbf{WT}e_m^{(b)}(\xi_{ij})b_m^{(i)} + \sum_{m=0}^{M} \mathbf{WT}e_m^{(c)}(\xi_{ij})p_m^{(i)} + \sum_{m=0}^{M} \mathbf{WT}e_m^{(d)}(\xi_{ij})q_m^{(i)} - \sum_{m=0}^{M} \mathbf{WT}e_m^{(a)}(\xi_{kl})a_m^{(k)} - \sum_{m=0}^{M} \mathbf{WT}e_m^{(b)}(\xi_{kl})b_m^{(k)}$$

$$- \sum_{m=0}^{M} \mathbf{WT}e_m^{(c)}(\xi_{kl})p_m^{(k)} - \sum_{m=0}^{M} \mathbf{WT}e_m^{(d)}(\xi_{kl})q_m^{(k)} = 0. \tag{14}$$

Now we consider the continuity of the traction (t_x, t_y) along the internal contour, shared by segments $P(i, j) \rightarrow P(i, j+1)$ and $P(k, l) \rightarrow P(k, l+1)$. The continuity is expressed as $t_x(i, j, n) = t_x(k, l, n')$, where $n' = N(k, l) - n$, which means $\xi_{kj} = d_{ij} - \xi_{ij}$ (Fig. 6). On the other hand, $n_x(\xi_{ij}) = -n_x(d_{ij} - \xi_{ij})$ and $n_y(\xi_{ij}) = -n_y(d_{ij} - \xi_{ij})$, since the external normal of a segment is opposed to the external normal to another segment. The continuity condition for t_x will be, from eq. (11),

$$\sum_{m=0}^{M} p_m^{(a)}(\xi_{ij})a_m^{(i)} + \sum_{m=0}^{M} p_m^{(b)}(\xi_{ij})b_m^{(i)} + \sum_{m=0}^{M} p_m^{(c)}(\xi_{ij})p_m^{(i)} + \sum_{m=0}^{M} p_m^{(d)}(\xi_{ij})q_m^{(i)} - \sum_{m=0}^{M} p_m^{(a)}(\xi_{kl})a_m^{(k)} - \sum_{m=0}^{M} p_m^{(b)}(\xi_{kl})b_m^{(k)}$$

$$- \sum_{m=0}^{M} p_m^{(c)}(\xi_{kl})p_m^{(k)} - \sum_{m=0}^{M} p_m^{(d)}(\xi_{kl})q_m^{(k)} = 0. \tag{15}$$
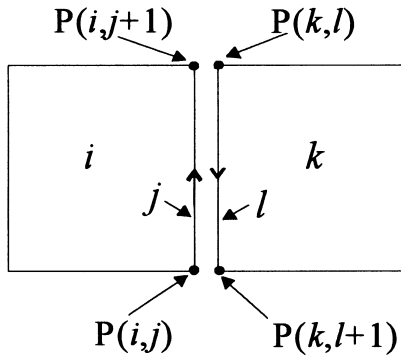
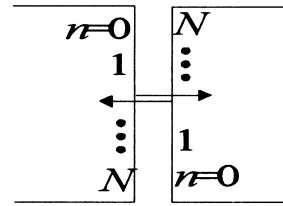**Figure 5.** Convention used for an internal segment of the problem.

Figure 6. Continuity condition for an internal boundary.

Taking the wavelet transform of the above equation, we obtain

$$\sum_{m=0}^{M} \mathbf{WT}p_m^{(a)}(\xi_{ij})a_m^{(i)} + \sum_{m=0}^{M} \mathbf{WT}p_m^{(b)}(\xi_{ij})b_m^{(i)} + \sum_{m=0}^{M} \mathbf{WT}p_m^{(c)}(\xi_{ij})p_m^{(i)} + \sum_{m=0}^{M} \mathbf{WT}p_m^{(d)}(\xi_{ij})q_m^{(i)} - \sum_{m=0}^{M} \mathbf{WT}p_m^{(a)}(\xi_{kl})a_m^{(k)} - \sum_{m=0}^{M} \mathbf{WT}p_m^{(b)}(\xi_{kl})b_m^{(k)}$$

$$- \sum_{m=0}^{M} \mathbf{WT}p_m^{(c)}(\xi_{kl})p_m^{(k)} - \sum_{m=0}^{M} \mathbf{WT}p_m^{(d)}(\xi_{kl})q_m^{(k)} = 0 \,. \tag{16}$$

2.5 Internal boundary conditions for cases of faults

The displacement discontinuity (Figs 7 and 8), that is, the effect of the fault itself, is imposed on the parallel displacement component u_p,

$$u_p(\xi_{ij}) + u_p(\xi_{kl}) = -kt_p \,, \tag{17}$$

where the constant k is inversely proportional to the friction between the two cells. This means that the higher the value of the constant k, the easier it is to have displacement along the fault, and that the lower the value of the constant k, the more difficult it is to have displacement along the fault.

The other boundary conditions are related to continuity on the medium,

$$u_n(\xi_{ij}) + u_n(\xi_{kl}) = 0 \,, \tag{18}$$

$$t_x(\xi_{ij}) + t_y(\xi_{kl}) = 0 \,, \tag{19}$$

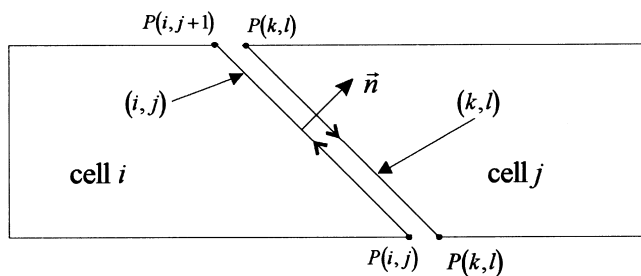$$t_y(\xi_{ij}) + t_y(\xi_{kl}) = 0 \,. \tag{20}$$

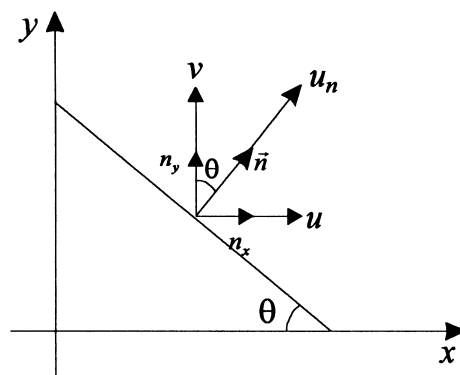**Figure 7.** Internal boundary segment for the case of a fault.

Figure 8. Normal displacement vector on a fault model.

3 MATRIX EQUATION FOR THE DETERMINATION OF THE MODEL PARAMETERS

We now show how to obtain the matrix equation that is used to calculate the parameters of the model. First, consider the vector $\mathbf{x}_i$, defined by eq. (5): we rewrite it as

$$\mathbf{x}_i = \begin{Bmatrix} a_0^{(i)} \\ b_0^{(i)} \\ p_0^{(i)} \\ q_0^{(i)} \\ \vdots \\ a_M^{(i)} \\ b_M^{(i)} \\ p_M^{(i)} \\ q_M^{(i)} \end{Bmatrix}.$$

We gather all the $\mathbf{x}_i$ vectors together in a single vector, defined as

$$\mathbf{x} = \begin{Bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_I \end{Bmatrix}.$$

The matrix equation for the determination of the $\mathbf{x}$ vector is written as

$$A\mathbf{x} = \mathbf{b}, \tag{21}$$

where A and $\mathbf{b}$ are known. As an example, we rewrite the set of eq. (11) in its matrix form,

$$T_x(\xi_{ij}) = \sum_{m=1}^{M} \left\{ p_m^{(a)}(\xi_{ij})a_m + p_m^{(b)}(\xi_{ij})b_m + p_m^{(c)}(\xi_{ij})p_m + p_m^{(d)}(\xi_{ij})q_m \right\}. \tag{22}$$

Recall that a set containing all coordinates $(x(\xi_{ij})y(\xi_{ij}))$, $\xi_{ij} = (n/N_i)d_{ij}$ and $n = 0, 1, \ldots, N_i$ is equivalent to $\{x(i, j, n), y(i, j, n)\}$. We can then represent eq. (22) by

$$T_x(i, j, n) = \sum_{m=1}^{M} \left\{ p_m^{(a)}(i, j, n)a_m + p_m^{(b)}(i, j, n)b_m + p_m^{(c)}(i, j, n)p_m + p_m^{(d)}(i, j, n)q_m \right\}. \tag{23}$$

The purpose of this change of notation is to improve our understanding of the set-up of the A_i matrix. If we hide the i and j indices, the A_i matrix can be written as

$$A_i = \begin{bmatrix} p_0^{(a)}(0) & p_0^{(b)}(0) & p_0^{(c)}(0) & p_0^{(d)}(0) & \cdots & p_M^{(a)}(0) & p_M^{(b)}(0) & p_M^{(c)}(0) & p_M^{(d)}(0) \\ p_0^{(a)}(1) & p_0^{(b)}(1) & p_0^{(c)}(1) & p_0^{(d)}(1) & \cdots & p_M^{(a)}(1) & p_M^{(b)}(1) & p_M^{(c)}(1) & p_M^{(d)}(1) \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ p_0^{(a)}(N) & p_0^{(b)}(N) & p_0^{(c)}(N) & p_0^{(d)}(N) & \cdots & p_M^{(a)}(N) & p_M^{(b)}(N) & p_M^{(c)}(N) & p_M^{(d)}(N) \end{bmatrix}. \tag{24}$$

On the other hand, the $\mathbf{b}_i$ vector is written as

$$\mathbf{b}_i = \begin{bmatrix} T_x(i, j, 0) \\ T_x(i, j, 1) \\ \vdots \\ T_x(i, j, N) \end{bmatrix}. \tag{25}$$

Note that the application of the wavelet transform to the above subsystem with respect to n is equivalent to applying the wavelet transform with N basis functions to the column vectors $(\mathbf{p}_0^{(a)}, \ldots, \mathbf{p}_M^{(d)})$ that make up the A_i matrix and to the right-hand side column vector $\mathbf{b}_i$.

Note that the boundary condition considered above was an external one, and that the components of the $\mathbf{b}_i$ vector may or may not be null. If the boundary conditions were internal ones, however, these conditions would all be null.

The total number of unknowns in the system of eq. (25) is $4(M+1) \times I$, where I is the number of cells. The number of equations of this system is given by the number of sample points multiplied by the number of boundary conditions: $N(i, j) \times [2(\text{number of external boundary segments}) + 4(\text{number of internal boundary segments})]$. Note that we have two boundary conditions for each external segment, and four boundary conditions for each internal segment.

4   COMPUTATION OF THE MODEL PARAMETERS

For the computation of the model parameters for a plane elastic medium, it is necessary, according to the modelling method presented above, to find a vector $\mathbf{x} \in \mathcal{R}$ such that $A\mathbf{x} = \mathbf{b}$, where $A \in \mathcal{R}^{m \times n}$ and $\mathbf{b} \in \mathcal{R}^m$ are given, and $m > n$ (such that we have an overdetermined system). In this section, the solution of this overdetermined system of linear equations is obtained in the least-squares (LS) sense. This solution is represented by $\mathbf{x}_{LS} = A^+ \mathbf{b}$, where A^+ is the pseudo-inverse of A. The pseudo-inverse of A is based on the SVD (singular value decomposition) of A. Furthermore, the calculation of the rank of matrix A is also performed using the SVD of A.

The algorithm we used is the one from Golub & Reinsch (1970). It first reduces the matrix A to an upper bidiagonal form using householder matrices. According to Golub & Van Loan (1985), this is the best way of treating problems of deficient rank throughout the computation of the SVD. Due to this, we emphasize here the ability of the SVD to manipulate the difficulties of deficient rank in least-squares problems, even if there are round-off approximations. Then, given $A = U\Sigma V^T$, the SVD of A is

$$\mathbf{x}_{LS} = \sum_{i=1}^{r} \left(\mathbf{u}_i^T \mathbf{b} / \sigma_i \right) \mathbf{v}_i ,$$

where $r = \text{rank}(A)$. We represent the computed versions of U, V and $\Sigma = \text{diag}(\sigma_i)$ by $\hat{U}$, $\hat{V}$ and $\hat{\Sigma} = \text{diag}(\hat{\sigma}_i)$. We can then show (Golub & Van Loan 1985) that

$$\hat{U} = W + \Delta U , \qquad \text{where} \qquad W^T W = I_m \qquad \text{and} \qquad \|\Delta U\|_2 \leq \varepsilon$$

$$\hat{V} = Z + \Delta V , \qquad \text{where} \qquad Z^T Z = I_n \qquad \text{and} \qquad \|\Delta V\|_2 \leq \varepsilon ,$$

$$\hat{\Sigma} = W^T (A + \Delta A) Z , \qquad \text{where} \|\Delta A\|_2 \leq \varepsilon \|A\|_2 ,$$

where ε is a small multiple of the machine precision limit. In this way, the SVD algorithm calculates the singular values of the approximated matrix $A + \Delta A$.

The $\hat{U}$ and $\hat{V}$ matrices are not necessarily approximated to their exact counterparts U and V. On the other hand, it can be shown that $\hat{\sigma}_k$ is similar to σ_k. This implies that near rank deficiency in A is detected when the SVD of A is computed.

We now discuss the condition number. Assume that the A matrix has incomplete rank ($r \leq n$). It is natural to least square problems fixing the last $n - r$ singular values as zero. We then have the question of how the calculated solution varies with respect to perturbations in A. This question can be answered in terms of the condition number,

$$\kappa_r(A) = \frac{\sigma_1}{\sigma_r} .$$

If $\kappa_r(A)$ is large, then the A matrix is said to be ill-conditioned, that is, there are singular values that should be treated as zero.

However, in the ill-conditioned least-squares problems, the observation of small singular values alone does not settle the decision about the value of rank(A). One way to avoid this difficulty is to use a parameter $\delta > 0$ and consider the hypothesis that A has numerical rank $\hat{r}$ if $\hat{\sigma}_i$ satisfies

$$\hat{\sigma}_1 \geq \ldots \geq \hat{\sigma}_{\hat{r}} > \delta \geq \hat{\sigma}_{\hat{r}+1} \geq \ldots \geq \hat{\sigma}_n .$$

Whenever this is the case, we can consider

$$\mathbf{x}_{\hat{r}} = \sum_{i=1}^{\hat{r}} \frac{\hat{\mathbf{u}}_i^T \mathbf{b}}{\hat{\sigma}_i} \hat{\mathbf{v}}_i$$

as an approximation of $\mathbf{x}_{LS}$.

The δ parameter would be consistent with the machine precision limit, for example, $\delta = \mathbf{u} \|A\|$. On the other hand, if the level of the relative error on the data is larger than $\mathbf{u}$, then δ would be correspondingly larger, for example, $\delta = 10^{-2} \|A\|$. Given that $\|\mathbf{x}_{\hat{r}}\|_2 \approx 1/\hat{\sigma}_{\hat{r}} \leq 1/\delta$, δ can be chosen with the intention of producing a least-squares solution with a conveniently small norm.

If $\hat{\sigma}_{\hat{r}} \gg \delta$, then we have reasons for not worrying about the $\mathbf{x}_{\hat{r}}$ solution, because the A matrix can be considered, without ambiguity, as a matrix of rank $\hat{r}$. On the other hand, $\{\hat{\sigma}_1, \ldots, \hat{\sigma}_n\}$ may not be clearly divided into subsets of large and small singular values, which will make the determination of $\hat{r}$ somewhat arbitrary. Fortunately, this is not the case for the problem we are considering.

5 APPLICATION OF THE METHOD TO SIMPLE CASES

In this section we apply the modelling method described above to three cases of a single cell (this means that there are no internal boundary conditions involved in the problem) for which the analytical solution is known. The analytical solutions for all three cases considered here were obtained from the use of mathematical equations presented by de Veubeke (1979).

5.1 Geometry 1

5.1.1 *System definition and expected results*

Consider the problem shown in Fig. 9. The theoretical solution is easily obtained when we compare the boundary conditions to the general expressions for σ_x, σ_y and τ_{xy}, which are given by eqs (A9) (A10) and (A11), respectively. In this way, we obtain the theoretical solutions

$$\begin{cases} \sigma_x = \sigma_y = a_1 \\ \tau_{xy} = 0 \end{cases}$$

and $a_1 = P/2$. Since the only non-zero coefficient is $a_1 = P/2$, the equations for u and v (eqs A7 and A8, respectively) result in

$$\begin{cases} Gu = \dfrac{(1-\mu)}{1+\mu}\, a_1 x \\ Gv = \dfrac{(1-\mu)}{1+\mu}\, a_1 y \end{cases}$$

which will yield the deformed shape of the body shown in Fig. 10.

5.1.2 *Application of the method*

The first step in the application of the method is the definition of the system, which was given in Section 5.1.1. It is important to highlight some points of that definition. First, there is only one system of local coordinates, which is the same as the main coordinate axis, because there is a single cell in our model. There is only one coefficient vector, which is given by

$$\mathbf{x}_1^{\mathrm{T}} = \left(a_0^{(1)}, b_0^{(1)}, p_0^{(1)}, q_0^{(1)}, \ldots, a_M^{(1)}, b_M^{(1)}, p_M^{(1)}, q_M^{(1)} \right).$$

Given that the model has a square shape, the segmentation of the contour of the body is obvious. There are four contour segments, which correspond to the four square sides.
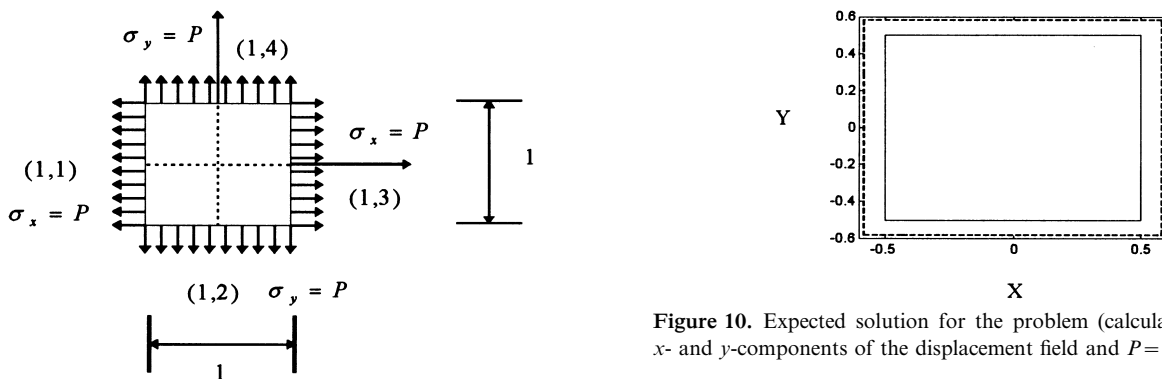

Figure 9. Problem geometry, including the boundary conditions. The elastic constants are $G_1 = 1$ and $\mu = 1/3$.

Figure 10. Expected solution for the problem (calculated using the x- and y-components of the displacement field and $P = 1$).

In order to discretize the contour, the procedure is also very simple since the segments are all straight lines. In this way, we obtain

$\{x(1, 1, n), y(1, 1, n)\} = \{-0.5, 0.5 - n/N\}$ for the $(1, 1)$ contour, with $n = 0, 1, \ldots, N$,

$\{x(1, 2, n), y(1, 2, n)\} = \{-0.5 + n/N, -0.5\}$ for the $(1, 2)$ contour, with $n = 0, 1, \ldots, N$,

$\{x(1, 3, n), y(1, 3, n)\} = \{0.5, -0.5 + n/N\}$ for the $(1, 3)$ contour, with $n = 0, 1, \ldots, N$,

$\{x(1, 4, n), y(1, 4, n)\} = \{0.5 - n/N, 0.5\}$ for the $(1, 4)$ contour, with $n = 0, 1, \ldots, N$.

Regarding the above boundary conditions, we know the tractions that are applied to the external contours. We have the following.

For contour $(1, 1)$

$$\begin{cases} t_x(1, 1, n) = -P \\ t_y(1, 1, n) = 0 \end{cases} \quad \text{and} \quad \begin{cases} n_x(1, 1, n) = -1 \\ n_y(1, 1, n) = 0 \end{cases}.$$

Then, from eq. (11) we obtain

$$\sum_{m=1}^{M} \left[g_m^{(a)}(1, 1, n)a_m^{(1)} + g_m^{(b)}(1, 1, n)b_m^{(1)} + g_m^{(c)}(1, 1, n)p_m^{(1)} + g_m^{(d)}(1, 1, n)q_m^{(1)} \right] = P \tag{26}$$

and from eq. (12) we have

$$\sum_{m=1}^{M} \left[r_m^{(a)}(1, 1, n)a_m^{(1)} + r_m^{(b)}(1, 1, n)b_m^{(1)} + r_m^{(c)}(1, 1, n)p_m^{(1)} + r_m^{(d)}(1, 1, n)q_m^{(1)} \right] = 0. \tag{27}$$

For contour $(1, 2)$

$$\begin{cases} t_x(1, 2, n) = 0 \\ t_y(1, 2, n) = -P \end{cases} \quad \text{and} \quad \begin{cases} n_x(1, 2, n) = 0 \\ n_y(1, 2, n) = -1 \end{cases}.$$

Then,

$$\sum_{m=1}^{M} \left[r_m^{(a)}(1, 2, n)a_m^{(1)} + r_m^{(b)}(1, 2, n)b_m^{(1)} + r_m^{(c)}(1, 2, n)p_m^{(1)} + r_m^{(d)}(1, 2, n)q_m^{(1)} \right] = 0, \tag{28}$$

$$\sum_{m=1}^{M} \left[h_m^{(a)}(1, 2, n)a_m^{(1)} + h_m^{(b)}(1, 2, n)b_m^{(1)} + h_m^{(c)}(1, 2, n)p_m^{(1)} + h_m^{(d)}(1, 2, n)q_m^{(1)} \right] = P. \tag{29}$$

For contour $(1, 3)$

$$\begin{cases} t_x(1, 3, n) = P \\ t_y(1, 3, n) = 0 \end{cases} \quad \text{and} \quad \begin{cases} n_x(1, 3, n) = 1 \\ n_y(1, 3, n) = 0 \end{cases}.$$

Then,

$$\sum_{m=1}^{M} \left[g_m^{(a)}(1, 3, n)a_m^{(1)} + g_m^{(b)}(1, 3, n)b_m^{(1)} + g_m^{(c)}(1, 3, n)p_m^{(1)} + g_m^{(d)}(1, 3, n)q_m^{(1)} \right] = P, \tag{30}$$

$$\sum_{m=1}^{M} \left[r_m^{(a)}(1, 3, n)a_m^{(1)} + r_m^{(b)}(1, 3, n)b_m^{(1)} + r_m^{(c)}(1, 3, n)p_m^{(1)} + r_m^{(d)}(1, 3, n)q_m^{(1)} \right] = 0. \tag{31}$$

For contour $(1, 4)$

$$\begin{cases} t_x(1, 4, n) = 0 \\ t_y(1, 4, n) = P \end{cases} \quad \text{and} \quad \begin{cases} n_x(1, 3, n) = 1 \\ n_y(1, 3, n) = 0 \end{cases}.$$

Then,

$$\sum_{m=1}^{M} \left[r_m^{(a)}(1, 4, n)a_m^{(1)} + r_m^{(b)}(1, 4, n)b_m^{(1)} + r_m^{(c)}(1, 4, n)p_m^{(1)} + r_m^{(d)}(1, 4, n)q_m^{(1)} \right] = 0, \tag{32}$$

$$\sum_{m=1}^{M} \left[h_m^{(a)}(1, 4, n)a_m^{(1)} + h_m^{(b)}(1, 4, n)b_m^{(1)} + h_m^{(c)}(1, 4, n)p_m^{(1)} + h_m^{(d)}(1, 4, n)q_m^{(1)} \right] = P. \tag{33}$$

The next stage is the construction of the matrix equation from eqs (26)–(33),

$$
A = \begin{bmatrix}
\mathbf{g}_0^{(a)}(1,1) & \mathbf{g}_0^{(b)}(1,1) & \mathbf{g}_0^{(c)}(1,1) & \mathbf{g}_0^{(d)}(1,1) & \ldots & \mathbf{g}_M^{(a)}(1,1) & \mathbf{g}_M^{(b)}(1,1) & \mathbf{g}_M^{(c)}(1,1) & \mathbf{g}_M^{(d)}(1,1) \\
\mathbf{r}_0^{(a)}(1,1) & \mathbf{r}_0^{(b)}(1,1) & \mathbf{r}_0^{(c)}(1,1) & \mathbf{r}_0^{(d)}(1,1) & \ldots & \mathbf{r}_M^{(a)}(1,1) & \mathbf{r}_M^{(b)}(1,1) & \mathbf{r}_M^{(c)}(1,1) & \mathbf{r}_M^{(d)}(1,1) \\
\mathbf{r}_0^{(a)}(1,2) & \mathbf{r}_0^{(b)}(1,2) & \mathbf{r}_0^{(c)}(1,2) & \mathbf{r}_0^{(d)}(1,2) & \ldots & \mathbf{r}_M^{(a)}(1,2) & \mathbf{r}_M^{(b)}(1,2) & \mathbf{r}_M^{(c)}(1,2) & \mathbf{r}_M^{(d)}(1,2) \\
\mathbf{h}_0^{(a)}(1,2) & \mathbf{h}_0^{(b)}(1,2) & \mathbf{h}_0^{(c)}(1,2) & \mathbf{h}_0^{(d)}(1,2) & \ldots & \mathbf{h}_M^{(a)}(1,2) & \mathbf{h}_M^{(b)}(1,2) & \mathbf{h}_M^{(c)}(1,2) & \mathbf{h}_M^{(d)}(1,2) \\
\mathbf{g}_0^{(a)}(1,3) & \mathbf{g}_0^{(b)}(1,3) & \mathbf{g}_0^{(c)}(1,3) & \mathbf{g}_0^{(d)}(1,3) & \ldots & \mathbf{g}_M^{(a)}(1,3) & \mathbf{g}_M^{(b)}(1,3) & \mathbf{g}_M^{(c)}(1,3) & \mathbf{g}_M^{(d)}(1,3) \\
\mathbf{r}_0^{(a)}(1,3) & \mathbf{r}_0^{(b)}(1,3) & \mathbf{r}_0^{(c)}(1,3) & \mathbf{r}_0^{(d)}(1,3) & \ldots & \mathbf{r}_M^{(a)}(1,3) & \mathbf{r}_M^{(b)}(1,3) & \mathbf{r}_M^{(c)}(1,3) & \mathbf{r}_M^{(d)}(1,3) \\
\mathbf{r}_0^{(a)}(1,4) & \mathbf{r}_0^{(b)}(1,4) & \mathbf{r}_0^{(c)}(1,4) & \mathbf{r}_0^{(d)}(1,4) & \ldots & \mathbf{r}_M^{(a)}(1,4) & \mathbf{r}_M^{(b)}(1,4) & \mathbf{r}_M^{(c)}(1,4) & \mathbf{r}_M^{(d)}(1,4) \\
\mathbf{h}_0^{(a)}(1,4) & \mathbf{h}_0^{(b)}(1,4) & \mathbf{h}_0^{(c)}(1,4) & \mathbf{h}_0^{(d)}(1,4) & \ldots & \mathbf{h}_M^{(a)}(1,4) & \mathbf{h}_M^{(b)}(1,4) & \mathbf{h}_M^{(c)}(1,4) & \mathbf{h}_M^{(d)}(1,4)
\end{bmatrix}_{8(N+1) \times 4(M+1)} .
$$

Note that each of the column vectors of the A matrix has N components. There will thus be $8N$ lines in the A matrix, which means, $8N+8$ equations. The number of unknowns is given by the number of columns of A, which is $4(M+1)$. The vector $\mathbf{b}$ has the following form:

$$
\mathbf{b} = \begin{bmatrix}
[P]_{N \times 1} \\
[0]_{N \times 1} \\
[0]_{N \times 1} \\
[P]_{N \times 1} \\
[P]_{N \times 1} \\
[0]_{N \times 1} \\
[0]_{N \times 1} \\
[P]_{N \times 1}
\end{bmatrix}_{8N \times 1} ,
$$

and finally,

$$
\mathbf{x}_1 = \begin{bmatrix}
a_0^{(1)} \\
b_0^{(1)} \\
p_0^{(1)} \\
q_0^{(1)} \\
\vdots \\
a_M^{(1)} \\
b_M^{(1)} \\
p_M^{(1)} \\
q_M^{(1)}
\end{bmatrix}_{4(M+1) \times 1} .
$$

Now we have to apply the wavelet transform with N basis functions to each of the vectors that comprise A and to the $\mathbf{b}$ right-hand-side vector.

After we have applied the method and found the equation $A\mathbf{x}=\mathbf{b}$, we calculate the model parameters. We make the number of equations greater than or equal to the number of unknowns in the problem [$8(N+1) \geq 4(M+1)$ or $N \geq (M-1)/2$]. For this case, as well as for the others treated here, we use $M=7$ and $N=8$. The solution to the matrix problem is then $\mathbf{x}=A^+\mathbf{b}$, where A^+ is the pseudo-inverse of A. This is the solution of the system of linear equations in the least-squares sense considering the SVD of the A matrix.
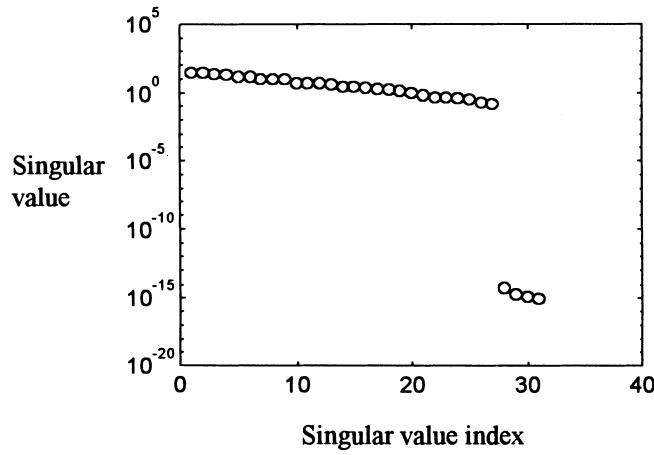
Figure 11. Singular values obtained in the solution of the problem.

Performing the singular value decomposition of the A matrix, we obtain $k_r(A) = \sigma_1/\sigma_r$ is of order 10^{16}. The A matrix is thus of incomplete rank. Since the last singular values are much smaller than the others (Fig. 11), we can say that the rank of this matrix is 28, and not 32 ($=4M+4$). This matrix will thus be inverted using the pseudo-inverse with a δ value (the tolerance value) equal to 10^{-10}, so that the small singular values are treated as zero. After the A matrix has been inverted and the vector $\mathbf{x} = A^+\mathbf{b}$ has been computed, we obtain the coefficients shown in Fig. 12.

We have considered coefficients up to seventh order, but they were all nulls, except the coefficient $a_1 = 0.5$. Since P was fixed as 1, then the result obtained is the same as the theoretical result. We have also computed the deformation values (through the displacement values) using the general equations (A7) and (A8) and the above coefficients, resulting in the geometry shown in Fig. 13. If the result obtained with the application of the method developed here is compared with the expected result (Fig. 10), we see that they are identical.

We also tried to make the A matrix compact in order to check the results. For this, we recall that the wavelet transform was applied to each of the vectors comprising A and $\mathbf{b}$. For each of these transformed component vectors we zeroed those coefficients for which the absolute values were smaller than tol $\times$ max(component vector), where tol is a tolerance value, which is selected by the user. The results presented in Figs (12) and (13) were obtained for tol $=0$. The results shown in Figs 14 and 15, however, were obtained with a tolerance value of tol $=0.9$.
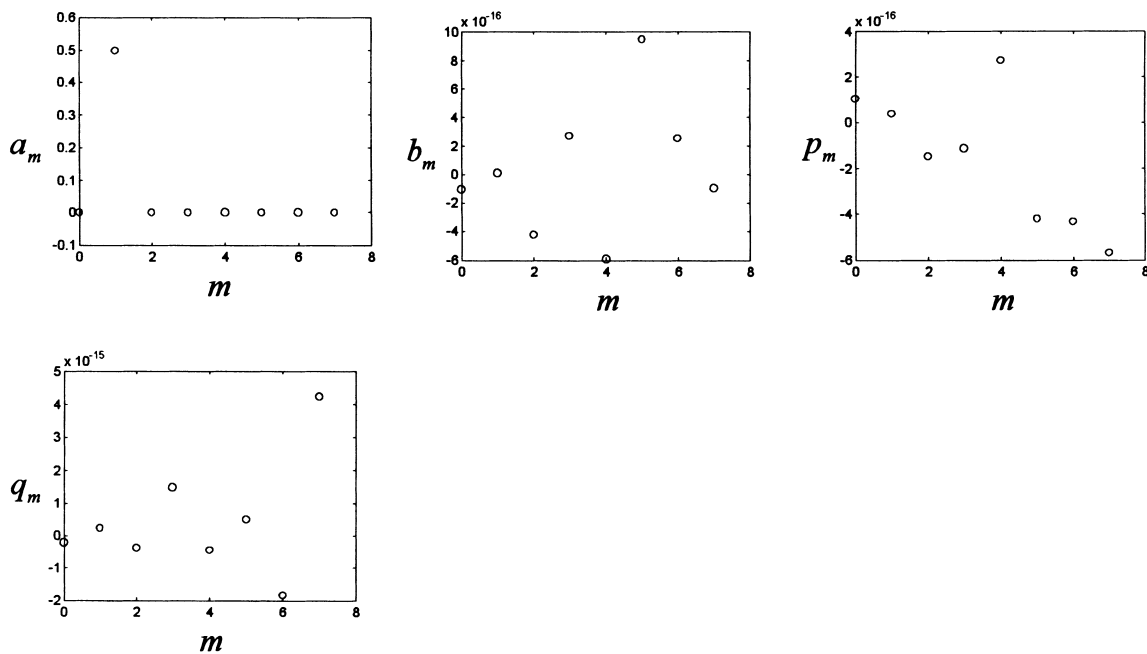

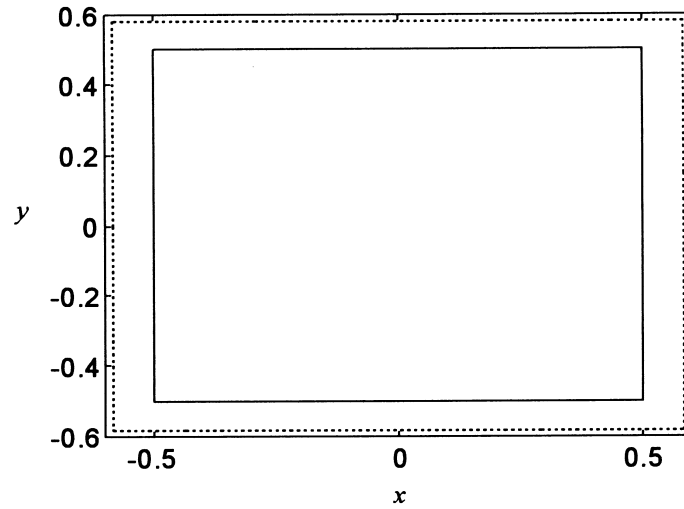**Figure 12.** Coefficients obtained in the solution of the problem.

Figure 13. Result obtained from the application of the method. The body represented by the solid line is the original one, and that represented by the dotted line is the deformed one.

Before the compression of the matrix, the percentage of non-zero elements of the matrix was 75 per cent. After the compression, this rate dropped to 9.8 per cent. If we superpose Figs 15 and 13 we obtain Fig. 16. Note that the two results (compact and non-compact systems) are basically the same and comparable to the analytical solution of the problem, which makes the compact version valid for this case.

5.2 Geometry 2

5.2.1 Definition of the system and expected results

Consider now the problem shown in Fig. 17. The theoretical solution is obtained when we compare the boundary conditions for σ_x, σ_y and τ_{xy} with eqs (A9), (A10) and (A11), respectively. We obtain

$$
\begin{cases}
\sigma_x = 2a_2 x \\
\sigma_y = 6a_2 x \\
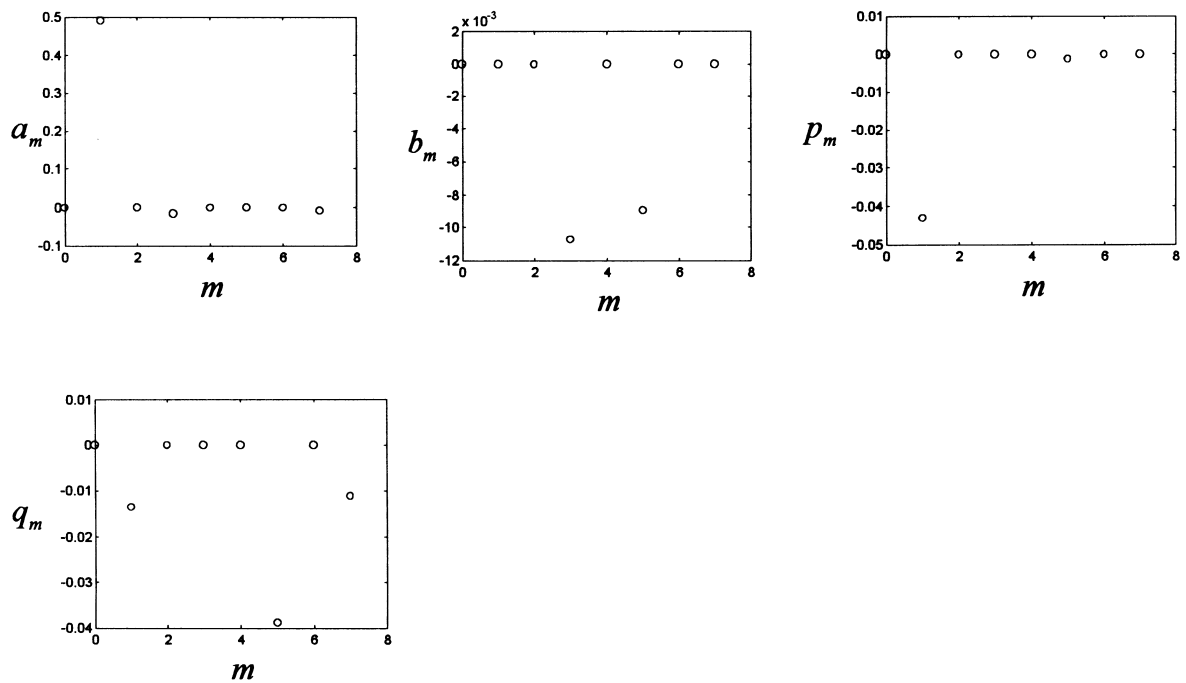\tau_{xy} = -2a_2 y
\end{cases} .
$$

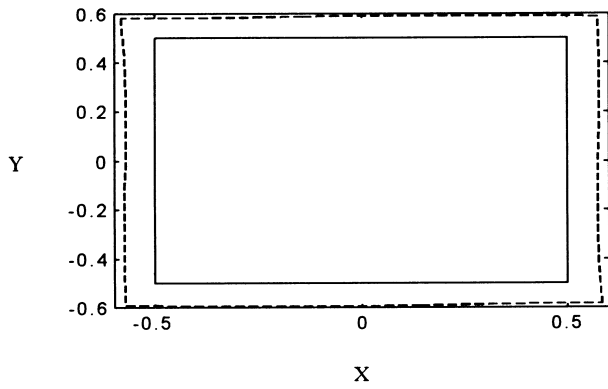**Figure 14.** Coefficients obtained with the compact matrix.

Figure 15. Deformation result obtained when we use the compact *A* matrix and the compact form of the **b** vector.
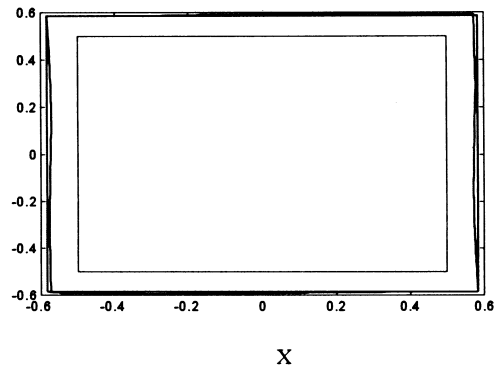

Figure 16. Superposition of the results presented in Fig. 13 (normal system) and those from Fig. 15 (corresponding to the compact system solution).

With respect to the displacement, the expected result will be

$$\begin{cases} Eu = a_2(1-3\mu)x^2 - a_2(5+\mu)y^2 \\ Ev = 2(3-\mu)a_2xy \end{cases},$$

which will produce the deformed body shown in Fig. 18.

5.2.2 *Application of the method*

The application of the method is identical to that presented in Section 5.1.2. We will thus only emphasize the calculation of the coefficients and the results.

After the *A* matrix is built, we calculate the its singular value decomposition (Fig. 19). We fix the value 10^{-10} as the limit for the same reasons as given in the previous case. Inverting the matrix using this tolerance value, we obtain the coefficients shown in Fig. 20. According to the coefficients obtained, we can consider that only the coefficient $a_2 = 0.5$ is different from zero. Using this we obtain the result shown in Fig. 21.

With the compact system, the percentage of non-zero elements dropped to 10.6 per cent in this case, and the computer memory size dropped to 2768 bytes. Before we made the system compact, the percentage of non-zero elements was 64.8 per cent, and the size of the matrix in the computer memory was 16 384 bytes. The results with and without the use of the compact version of the problem are shown in Fig. 22, together with the original (undeformed) body. Note that again we have a very good match of the two results.
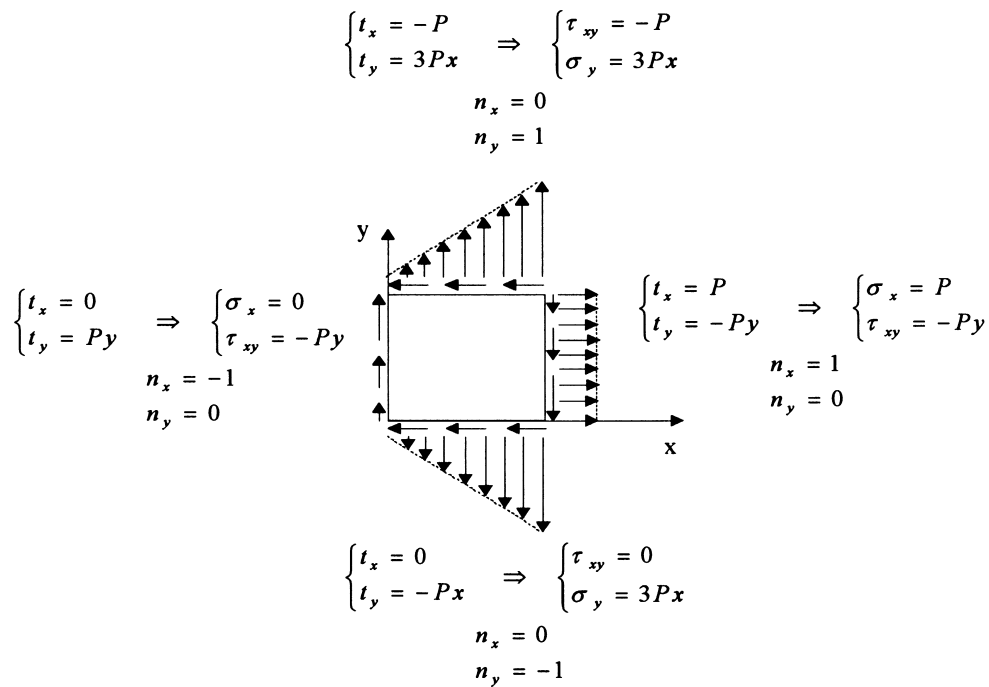
$$\begin{cases} t_x = -P \\ t_y = 3Px \end{cases} \Rightarrow \begin{cases} \tau_{xy} = -P \\ \sigma_y = 3Px \end{cases}$$
$$n_x = 0$$
$$n_y = 1$$

$$\begin{cases} t_x = 0 \\ t_y = Py \end{cases} \Rightarrow \begin{cases} \sigma_x = 0 \\ \tau_{xy} = -Py \end{cases}$$
$$n_x = -1$$
$$n_y = 0$$

$$\begin{cases} t_x = P \\ t_y = -Py \end{cases} \Rightarrow \begin{cases} \sigma_x = P \\ \tau_{xy} = -Py \end{cases}$$
$$n_x = 1$$
$$n_y = 0$$

$$\begin{cases} t_x = 0 \\ t_y = -Px \end{cases} \Rightarrow \begin{cases} \tau_{xy} = 0 \\ \sigma_y = 3Px \end{cases}$$
$$n_x = 0$$
$$n_y = -1$$

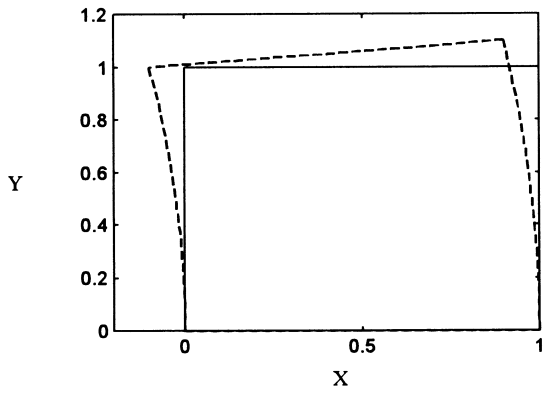**Figure 17.** Geometry of the problem and associated boundary conditions.

Figure 18. Analytical solution of the problem compared with the original square form.
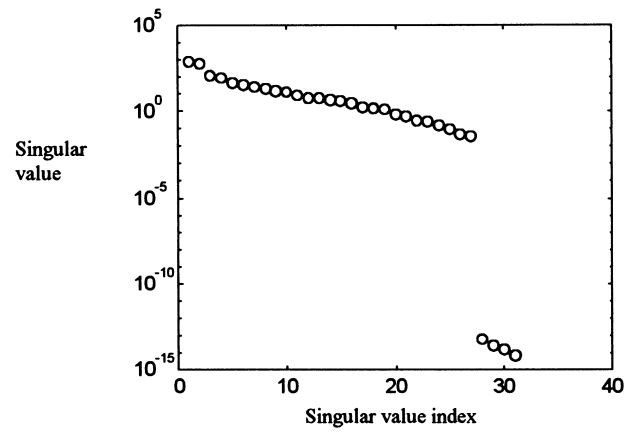

Figure 19. Singular values from the decomposition of the A matrix.
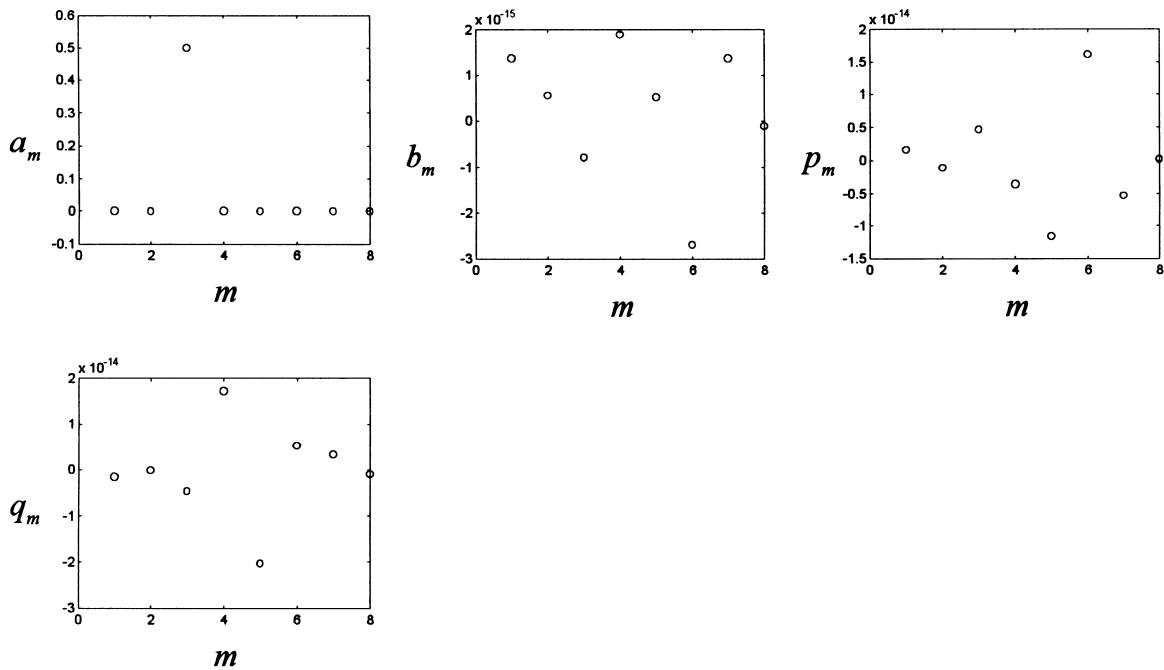

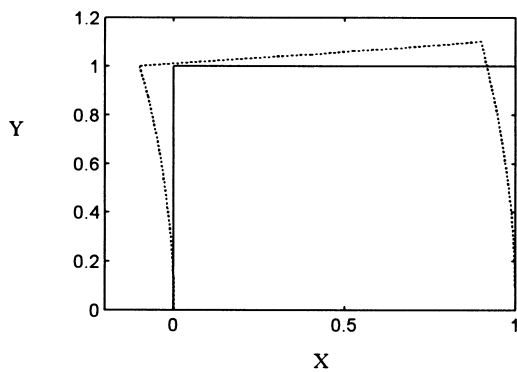**Figure 20.** Coefficients obtained in the solution of the problem.

Figure 21. Result obtained. The body represented by the solid line is the original and that represented by the dotted line is the deformed one.
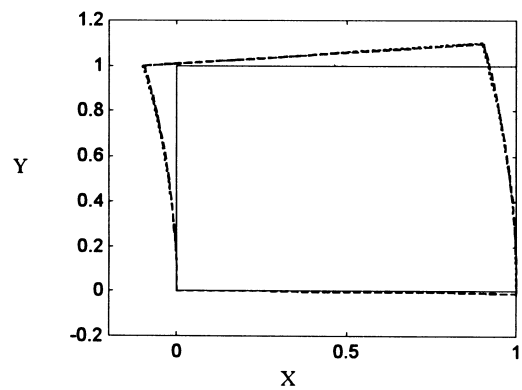

Figure 22. Result obtained with and without the use of the compact matrix and right-hand side vector of the problem compared with the expected analytical solution.
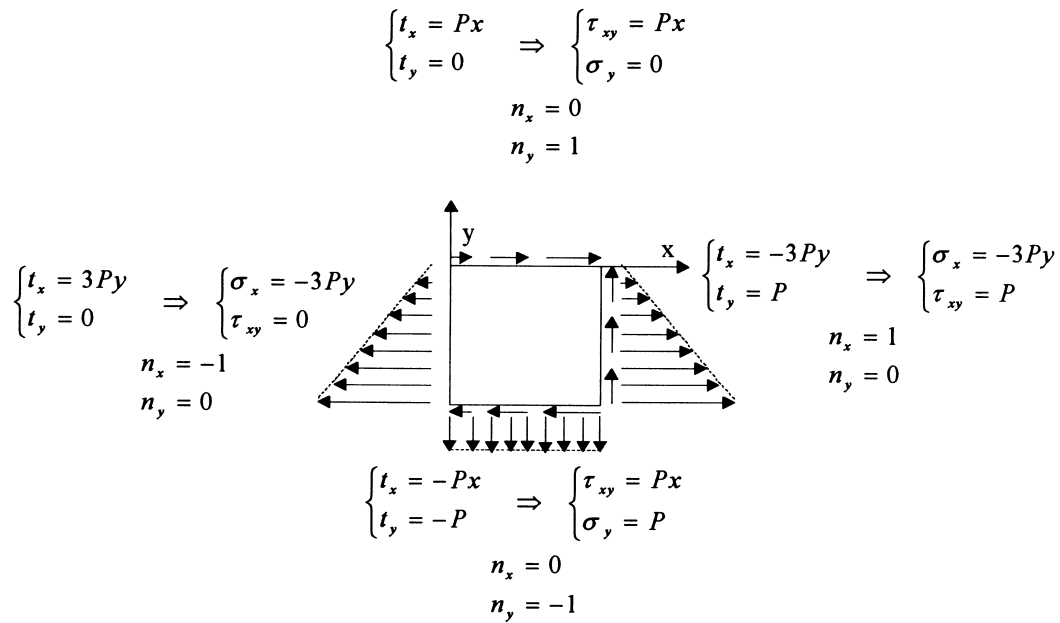
© 2001 RAS, *GJI* **146**, 454–488

$$\begin{cases} t_x = Px \\ t_y = 0 \end{cases} \Rightarrow \begin{cases} \tau_{xy} = Px \\ \sigma_y = 0 \end{cases}$$

$$n_x = 0$$
$$n_y = 1$$

$$\begin{cases} t_x = 3Py \\ t_y = 0 \end{cases} \Rightarrow \begin{cases} \sigma_x = -3Py \\ \tau_{xy} = 0 \end{cases}$$

$$n_x = -1$$
$$n_y = 0$$

$$\begin{cases} t_x = -3Py \\ t_y = P \end{cases} \Rightarrow \begin{cases} \sigma_x = -3Py \\ \tau_{xy} = P \end{cases}$$

$$n_x = 1$$
$$n_y = 0$$

$$\begin{cases} t_x = -Px \\ t_y = -P \end{cases} \Rightarrow \begin{cases} \tau_{xy} = Px \\ \sigma_y = P \end{cases}$$

$$n_x = 0$$
$$n_y = -1$$

Figure 23. Geometry of the problem and associated boundary conditions.

5.3   Geometry 3

5.3.1   *Definition of the system and expected results*

Consider the case shown in Fig. 23. The theoretical solutions to this system are

$$\begin{cases} \sigma_x = -6b_2y \\ \sigma_y = -2b_2y \\ \tau_{xy} = 2b_2x \end{cases}$$

and $b_2 = P/2$. The only non-zero coefficient is $b_2 = P/2$ so the expressions for the displacement components u and v are

$$\begin{cases} Gu = -\dfrac{(3-\mu)}{(1+\mu)}\,b_2xy \\ Gv = b_2\,\dfrac{(5+\mu)}{(1+\mu)}\,x^2 - b_2\,\dfrac{(1-3\mu)}{1+\mu}\,y^2 \end{cases}.$$

The solution, in terms of the displacement components, is shown in Fig. 24.

5.3.2   *Application of the method*

Using the same procedure as in Section 5.1.2, we obtained the results shown in Fig. 25. There are four small singular values; this means that the rank can be considered deficient and equal to 28. After calculating the pseudo-inverse, the coefficients obtained from the solution of the problem are those shown in Fig. 26. In terms of deformation, the solution is that of Fig. 27. The result calculated by this method is again identical to the theoretical result.

Using the compact version of the method, the percentage of non-zero elements dropped to 10.7 per cent, and the computer memory size of the matrix dropped to 2768 bytes. The corresponding values for these two variables before the application of the compact version of the method were 64.9 per cent and 16 384 bytes. Note that, once again (Fig. 28), there is a very good match between the theoretical and computed solutions.

6   MODELLING ELASTIC MEDIA WITH FAULTS

We now consider systems that are better suited to demonstrate and test the method, based on the initial theory of de Veubeke (1979), for models with internal contours. The contours considered in the previous cases were somewhat limited because they were all straight lines and they were all external boundaries to the bodies considered. Those cases were intended to prove the validity of the method.
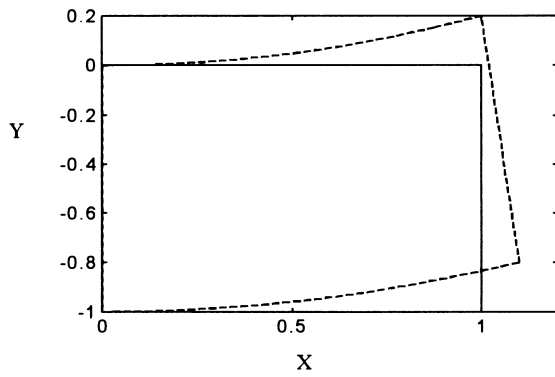
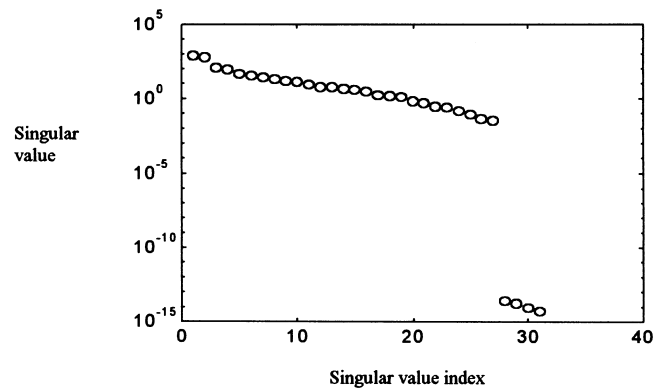**Figure 24.** Expected result from the known analytical solution of the problem.

Figure 25. Singular values obtained from the singular value decomposition of the A matrix.

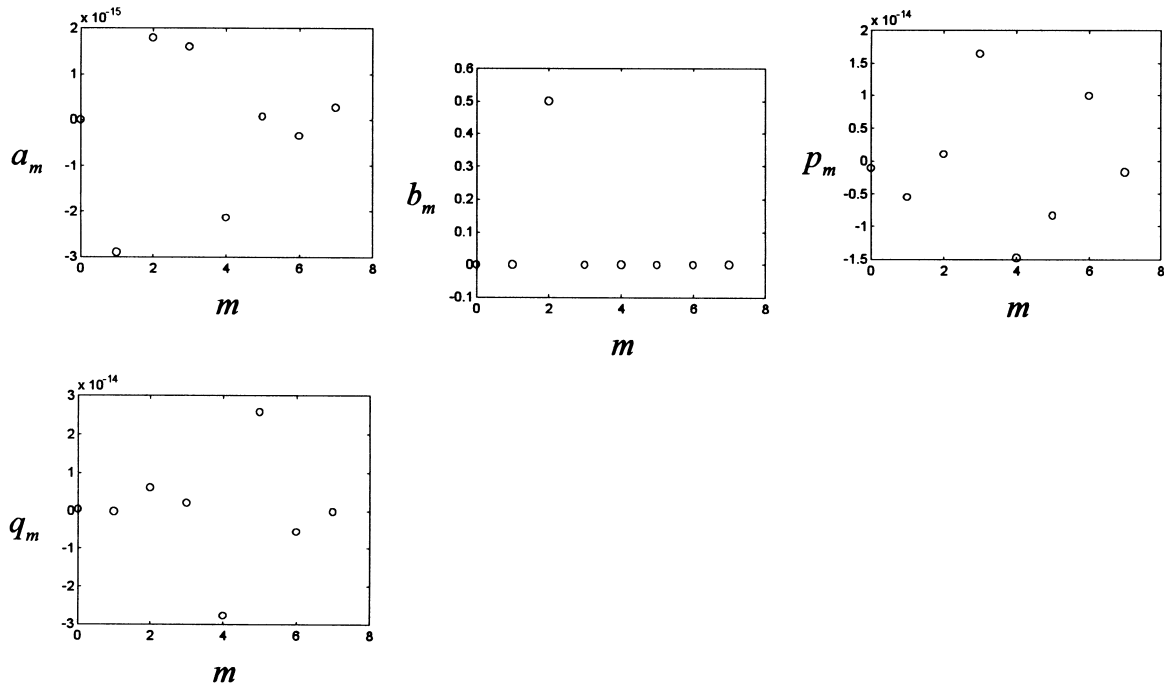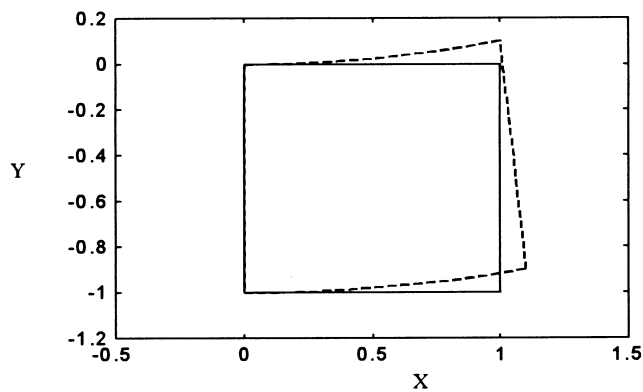**Figure 26.** Coefficients obtained from the solution of the problem.

Figure 27. Result obtained from the application of the method. The body represented by the solid line is the original system and that represented by the dashed line is the deformed body.
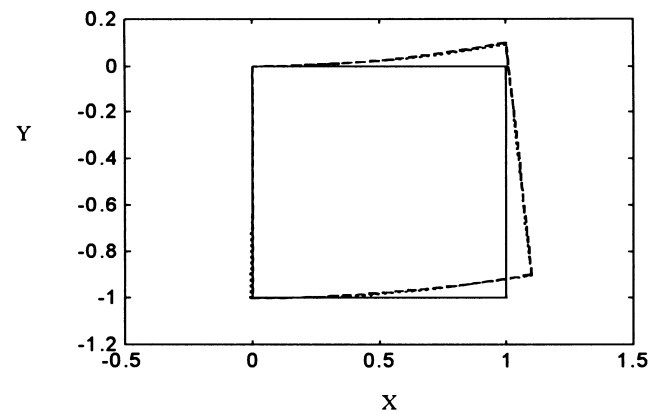

Figure 28. Result obtained for the deformation of the body compared with the deformation computed from the known analytical solution of the problem.

We will now treat cases that involve two cells whose geometries (Fig. 29) are the same. The differences between the cases we now consider are in the elastic properties of the cells and the amplitude of the sinusoidal boundary that represents the internal contour. We treat three cases, which are as follows.

(1) Sinusoidal maximum amplitude at the boundary $F = 0.1$, the same elastic properties for both cells ($G_1 = G_2 = 1$ and $\mu_1 = \mu_2 = 0.5$) and displacement continuity for the internal contour. This means that the friction between the two cells is infinite.
(2) Same as case (1), but with different elastic properties for the two cells ($G_1 = 10$, $G_2 = 1$, $\mu_1 = \mu_2 = 0.5$).
(3) Sinusoidal amplitude is null, $F = 0$ (the fault is a straight line), the same elastic properties for both cells ($G_1 = G_2 = 1$ and $\mu_1 = \mu_2 = 0.5$) and continuity of the displacement components on the internal contour of the model.

6.1   Definition of the system

Consider the system defined as shown in Fig. 29. Since there are two cells, there will be two local coordinate axis systems and we will have two vectors of coefficients, given by

$$\mathbf{x}_1^{\mathrm{T}} = \left(a_0^{(1)}, b_0^{(1)}, p_0^{(1)}, q_0^{(1)}, \ldots, a_M^{(1)}, b_M^{(1)}, p_M^{(1)}, q_M^{(1)} \right),$$

$$\mathbf{x}_2^{\mathrm{T}} = \left(a_0^{(2)}, b_0^{(2)}, p_0^{(2)}, q_0^{(2)}, \ldots, a_M^{(2)}, b_M^{(2)}, p_M^{(2)}, q_M^{(2)} \right).$$

6.2   The boundary conditions and their discretization

The second step of the modelling method is the identification of the contour segments. There are six contours in the system, three for each cell. The sinusoidal interface separating the two cells is treated as the (1, 3) contour for cell 1 and as the (2, 3) contour for cell 2. In this way, we can require displacement between the two cells. The displacement will be controlled by a sliding factor k. If the cells are welded together, then $k = 0$. If $k \neq 0$, then the internal contour segment is a fault.

The discretization of the external contours is simple because these are straight lines. For the internal contours, the discretization is not so straightforward, but it is also simple. The discretization of the contours results in sets of coordinates $\{x(\xi_{ij}), y(\xi_{ij})\}$ (where i represents the cell we are considering, j represents the contour and ξ_{ij} is the variable that is used for the discretization and for the parametrization of the (i, j) contour). These sets of coordinates, as well as the corresponding boundary conditions, are shown below for each case.

Contour (1, 1)

The boundary coordinates are

$$\begin{cases} x(\xi_{11}) = -0.25 \\ y(\xi_{11}) = 0.75 - \xi_{11} \\ n_x(\xi_{11}) = -1 \\ n(\xi_{11}) = 0 \\ \xi_{11} = \dfrac{n}{N}\,1, \qquad n = 0, 1, \ldots, N \end{cases} \tag{34}$$

The boundary conditions are

$$\begin{cases} t_x = 0 \\ t_y = 0 \end{cases} \tag{35}$$

Contour (1, 2)

The boundary coordinates are

$$\begin{cases} x(\xi_{12}) = \xi_{12} - 0.25 \\ y(\xi_{12}) = -0.25 \\ n_x = 0 \\ n_y = -1 \\ \xi_{12} = \dfrac{n}{N}\,1, \qquad n = 0, 1, \ldots, N \end{cases} \tag{36}$$

The boundary conditions are

$$
\begin{cases} t_x = 0 \\ t_y = -p \end{cases},
$$

(37)

where t_x and t_y are given by

$$
t_x = \sigma_x n_x + \tau_{xy} n_y,
$$

$$
t_y = n_x \tau_{xy} + n_y \sigma_y.
$$

Contour (2, 1)

The boundary coordinates are

$$
\begin{cases} x(\xi_{21}) = 0.25 \\ y(\xi_{21}) = -0.75 + \xi_{21} \\ n_x = 1 \\ n_y = 0 \\ \xi_{21} = \dfrac{n}{N}\,1, \qquad n = 0, 1, \ldots, N-1 \end{cases}.
$$

(38)

The boundary conditions are

$$
\begin{cases} t_x = 0 \\ t_y = 0 \end{cases}.
$$

(39)

Contour (2, 2)

The boundary coordinates are

$$
\begin{cases} x(\xi_{22}) = 0.25 - \xi_{22} \\ y(\xi_{22}) = 0.25 \\ n_x(\xi_{22}) = 0 \\ n_y(\xi_{22}) = 1 \\ \xi_{22} = \dfrac{n}{N}\,1, \qquad n = 0, 1, \ldots, N-1 \end{cases}.
$$

(40)

The boundary conditions are

$$
\begin{cases} t_x = 0 \\ t_y = -p \end{cases}.
$$

(41)

Contour (1, 3)

We have

$$
\begin{cases} n_x^{ij} = \dfrac{y_{i,j+1} - y_{ij}}{d_{ij}} \\ n_y^{ij} = \dfrac{x_{i,j+1} - x_{ij}}{d_{ij}} \end{cases}.
$$

The coordinates of a point on the real contour are given by

$$
\begin{cases} x_i(\xi_{ij}) = x_i^d(\xi_{ij}) + n_x^{ij} f_{ij}(\xi_{ij}) \\ y_i(\xi_{ij}) = x_i^d(\xi_{ij}) + n_y^{ij} f_{ij}(\xi_{ij}) \end{cases}.
$$

© 2001 RAS, *GJI* **146**, 454–488

If we use $f(\xi_{13}) = F \sin[(2\pi\xi_{13})/\sqrt{2}]$ and $F = 0.1$, we obtain

$$
\begin{cases}
x_1(\xi_{13}) = 0.75 - \dfrac{\xi_{13}}{\sqrt{2}} + \dfrac{f(\xi_{13})}{\sqrt{2}} \\[2mm]
y_1(\xi_{13}) = -0.25 - \dfrac{\xi_{13}}{\sqrt{2}} + \dfrac{f(\xi_{13})}{\sqrt{2}} \\[2mm]
\xi_{13} = \dfrac{n}{N}\sqrt{2}, \qquad n = 0, 1, \ldots, N-1
\end{cases}
\tag{42}
$$

Substituting eq. (42) into

$$
\begin{cases}
n_{x_i}(\xi_{ij}) = \dfrac{dy_i(\xi_{ij})}{d\xi_{ij}} \bigg/ \sqrt{\left(\dfrac{dx_i}{d\xi_{ij}}\right)^2 + \left(\dfrac{dy_i}{d\xi_{ij}}\right)^2} \\[3mm]
n_{y_i}(\xi_{ij}) = -\dfrac{dx_i(\xi_{ij})}{d\xi_{ij}} \bigg/ \sqrt{\left(\dfrac{dx_i}{d\xi_{ij}}\right)^2 + \left(\dfrac{dy_i}{d\xi_{ij}}\right)^2}
\end{cases},
$$

we obtain

$$
\begin{cases}
n_{x_1} = \dfrac{1 + F\pi\sqrt{2}\cos\left(\dfrac{2\pi\xi_{13}}{\sqrt{2}}\right)}{\sqrt{2\left(1 + 2F^2\pi^2\cos^2\left(\dfrac{2\pi\xi_{13}}{\sqrt{2}}\right)\right)}} \\[6mm]
n_{y_1} = \dfrac{1 - \sqrt{2}\pi F\cos\left(\dfrac{2\pi\xi_{13}}{\sqrt{2}}\right)}{\sqrt{2\left(1 + 2F^2\pi^2\cos^2\left(\dfrac{2\pi\xi_{13}}{\sqrt{2}}\right)\right)}}
\end{cases}.
\tag{43}
$$

Contour (2, 3)

It is not necessary to use

$$
\begin{cases}
n_x^{ij} = \dfrac{y_{i,j+1} - y_{ij}}{d_{ij}} \\[3mm]
n_y^{ij} = \dfrac{x_{i,j+1} - x_{ij}}{d_{ij}}
\end{cases}
$$

and the fact that the coordinates of a point on the real contour are given by

$$
\begin{cases}
x_i(\xi_{ij}) = x_i^d(\xi_{ij}) + n_x^{ij} f_{ij}(\xi_{ij}) \\[2mm]
y_i(\xi_{ij}) = x_i^d(\xi_{ij}) + n_y^{ij} f_{ij}(\xi_{ij})
\end{cases}
$$

to obtain the points on the sinusoidal contour seen by cell (2), because the local coordinate axis for cell (2) is translated from the local coordinate axis of cell (1) by 0.5 in both the x- and y-directions,

$$
\begin{cases}
x_2(\xi_{23}) = x_1(\xi_{23}) - 0.5 \\[2mm]
y(\xi_{23}) = y_1(\xi_{23}) - 0.5
\end{cases}.
\tag{44}
$$

Regarding the components of the external normal, it is not necessary to apply

$$
\begin{cases}
n_{x_i}(\xi_{ij}) = \dfrac{dy_i(\xi_{ij})}{d\xi_{ij}} \bigg/ \sqrt{\left(\dfrac{dx_i}{d\xi_{ij}}\right)^2 + \left(\dfrac{dy_i}{d\xi_{ij}}\right)^2} \\[3mm]
n_{y_i}(\xi_{ij}) = -\dfrac{dx_i(\xi_{ij})}{d\xi_{ij}} \bigg/ \sqrt{\left(\dfrac{dx_i}{d\xi_{ij}}\right)^2 + \left(\dfrac{dy_i}{d\xi_{ij}}\right)^2}
\end{cases}
$$

to obtain the components of the normal direction because they are opposed to those from the (1,3) contour when the same point is considered,

$$
n_{x_2}(\xi_{13}) = -n_{x_1}(\sqrt{2} - \xi_{13}),
$$
$$
n_{x_2}(\xi_{13}) = -n_{y_1}(\sqrt{2} - \xi_{13}).
\tag{45}
$$

The boundary conditions for (1, 3) and (2, 3) are

$$t_x^{(1)}(x_1, y_1) + t_x^{(2)}(x_2, y_2) = 0, \tag{46}$$

$$t_y^{(1)}(x_1, y_1) + t_y^{(2)}(x_2, y_2) = 0, \tag{47}$$

$$u_n^{(1)}(x_1, y_1) + u_n^{(2)}(x_2, y_2) = 0, \tag{48}$$

$$u_p^{(1)}(x_1, y_1) + u_p^{(2)}(x_2, y_2) + k t_p^{(1)}(x_1, y_1) = 0, \tag{49}$$

Boundary conditions (46) and (47) correspond to the continuity of t_x and t_y. Boundary condition (48) corresponds to the displacement discontinuity for the normal component and boundary condition (49) corresponds to the constitutive relation of the fault for the internal contour. Recall that, for this case, $k = 0$.

The next step is to write the boundary conditions in their polynomial form. We need the following equations:

$$u(x, y) = \sum_{m=0}^{M} \left[e_m^{(a)}(x, y)a_m + e_m^{(b)}(x, y)b_m + e_m^{(c)}(x, y)p_m + e_m^{(d)}(x, y)q_m \right],$$

$$v(x, y) = \sum_{m=0}^{M} \left[f_m^{(a)}(x, y)a_m + f_m^{(b)}(x, y)b_m + f_m^{(c)}(x, y)p_m + f_m^{(d)}(x, y)q_m \right],$$

$$\sigma_x(x, y) = \sum_{m=0}^{M} \left[g_m^{(a)}(x, y)a_m + g_m^{(b)}(x, y)b_m + g_m^{(c)}(x, y)p_m + g_m^{(d)}(x, y)q_m \right],$$

$$\sigma_y(x, y) = \sum_{m=0}^{M} \left[h_m^{(a)}(x, y)a_m + h_m^{(b)}(x, y)b_m + h_m^{(c)}(x, y)p_m + h_m^{(d)}(x, y)q_m \right],$$

$$\tau_{xy}(x, y) = \sum_{m=0}^{M} \left[r_m^{(a)}(x, y)a_m + r_m^{(b)}(x, y)b_m + r_m^{(c)}(x, y)p_m + r_m^{(d)}(x, y)q_m \right].$$

Then, for contour (1, 1)

$$\sum_{m=1}^{M} \left\{ g_m^{(a)}(\xi_{11})a_m^{(1)} + g_m^{(b)}(\xi_{11})b_m^{(1)} + g_m^{(c)}(\xi_{11})p_m^{(1)} + g_m^{(d)}(\xi_{11})q_m^{(1)} \right\} = 0, \tag{50}$$

$$\sum_{m=1}^{M} \left\{ r_m^{(a)}(\xi_{11})a_m^{(1)} + r_m^{(b)}(\xi_{11})b_m^{(1)} + r_m^{(c)}(\xi_{11})p_m^{(1)} + r_m^{(d)}(\xi_{11})q_m^{(1)} \right\} = 0. \tag{51}$$

For contour (1, 2)

$$\sum_{m=1}^{M} \left\{ r_m^{(a)}(\xi_{12})a_m^{(1)} + r_m^{(b)}(\xi_{12})b_m^{(1)} + r_m^{(c)}(\xi_{12})p_m^{(1)} + r_m^{(d)}(\xi_{12})q_m^{(1)} \right\} = 0, \tag{52}$$

$$\sum_{m=1}^{M} \left\{ h_m^{(a)}(\xi_{12})a_m^{(1)} + h_m^{(b)}(\xi_{12})b_m^{(1)} + h_m^{(c)}(\xi_{12})p_m^{(1)} + h_m^{(d)}(\xi_{12})q_m^{(1)} \right\} = p. \tag{53}$$

For contour (2, 1)

$$\sum_{m=1}^{M} \left\{ g_m^{(a)}(\xi_{21})a_m^{(2)} + g_m^{(b)}(\xi_{21})b_m^{(2)} + g_m^{(c)}(\xi_{21})p_m^{(2)} + g_m^{(d)}(\xi_{21})q_m^{(2)} \right\} = 0, \tag{54}$$

$$\sum_{m=1}^{M} \left\{ r_m^{(a)}(\xi_{21})a_m^{(2)} + r_m^{(b)}(\xi_{21})b_m^{(2)} + r_m^{(c)}(\xi_{21})p_m^{(2)} + r_m^{(d)}(\xi_{21})q_m^{(2)} \right\} = 0. \tag{55}$$

For contour (2, 2)

$$\sum_{m=1}^{M} \left\{ r_m^{(a)}(\xi_{22})a_m^{(2)} + r_m^{(b)}(\xi_{22})b_m^{(2)} + r_m^{(c)}(\xi_{22})p_m^{(2)} + r_m^{(d)}(\xi_{22})q_m^{(2)} \right\} = 0, \tag{56}$$

$$\sum_{m=1}^{M} \left\{ h_m^{(a)}(\xi_{22})a_m^{(2)} + h_m^{(b)}(\xi_{22})b_m^{(2)} + h_m^{(c)}(\xi_{22})p_m^{(2)} + h_m^{(d)}(\xi_{22})q_m^{(2)} \right\} = -p. \tag{57}$$

For contours (1, 3) and (2, 3)

$$\sum_{m=1}^{M} \left\{ \alpha_m^{(a)}(\xi_{13})a_m^{(1)} + \alpha_m^{(b)}(\xi_{13})b_m^{(1)} + \alpha_m^{(c)}(\xi_{13})p_m^{(1)} + \alpha_m^{(d)}(\xi_{13})q_m^{(1)} + \alpha_m^{(a)}(\xi_{23})a_m^{(2)} + \alpha_m^{(b)}(\xi_{23})b_m^{(2)} + \alpha_m^{(c)}(\xi_{23})p_m^{(2)} + \alpha_m^{(d)}(\xi_{23})q_m^{(2)} \right\} = 0, \quad (58)$$

where,

$$\alpha_m^{(a)}(\xi_{ij}) = n_{x_i}(\xi_{ij})g_m^{(a)}(\xi_{ij}) + n_{y_i}(\xi_{ij})r_m^{(a)}(\xi_{ij}),$$

$$\sum_{m=1}^{M} \left\{ \beta_m^{(a)}(\xi_{13})a_m^{(1)} + \beta_m^{(b)}(\xi_{13})b_m^{(1)} + \beta_m^{(c)}(\xi_{13})p_m^{(1)} + \beta_m^{(d)}(\xi_{13})q_m^{(1)} + \beta_m^{(a)}(\xi_{23})a_m^{(2)} + \beta_m^{(b)}(\xi_{23})b_m^{(2)} + \beta_m^{(c)}(\xi_{23})p_m^{(2)} + \beta_m^{(d)}(\xi_{23})q_m^{(2)} \right\} = 0 \quad (59)$$

and

$$\beta_m^{(a)}(\xi_{ij}) = n_{x_i}(\xi_{ij})r_m^{(a)}(\xi_{ij}) + n_{y_i}(\xi_{ij})h_m^{(a)}(\xi_{ij}),$$

$$\sum_{m=1}^{M} \left\{ \gamma_m^{(a)}(\xi_{13})a_m^{(1)} + \gamma_m^{(b)}(\xi_{13})b_m^{(1)} + \gamma_m^{(d)}(\xi_{13})p_m^{(1)} + \gamma_m^{(d)}(\xi_{13})q_m^{(1)} + \gamma_m^{(a)}(\xi_{23})a_m^{(2)} + \gamma_m^{(b)}(\xi_{23})b_m^{(2)} + \gamma_m^{(c)}(\xi_{23})p_m^{(2)} + \gamma_m^{(d)}(\xi_{23})q_m^{(2)} \right\} = 0, \quad (60)$$

$$\gamma_m^{(a)}(\xi_{ij}) = n_{x_i}(\xi_{ij})e_m^{(a)}(\xi_{ij}) + n_{y_i}(\xi_{ij})f_m^{(a)}(\xi_{ij}),$$

$$\sum_{m=0}^{M} \left\{ \rho_m^{(a)}(\xi_{13})a_m^{(1)} + \rho_m^{(b)}(\xi_{13})b_m^{(1)} + \rho_m^{(c)}(\xi_{13})p_m^{(1)} + \rho_m^{(d)}(\xi_{13})q_m^{(1)} + \rho_m^{(a)}(\xi_{23})a_m^{(2)} + \rho_m^{(b)}(\xi_{23})b_m^{(2)} + \rho_m^{(c)}(\xi_{23})p_m^{(2)} + \rho_m^{(d)}(\xi_{23})q_m^{(2)} \right.$$

$$\left. + k \left[\psi_m^{(a)}(\xi_{13})a_m^{(1)} + \psi_m^{(b)}(\xi_{13})b_m^{(1)} + \psi_m^{(c)}(\xi_{13})p_m^{(1)} + \psi_m^{(d)}(\xi_{13})q_m^{(1)} \right] \right\} = 0. \quad (61)$$

Finally,

$$\rho_m^{(a)}(\xi_{ij}) = n_{x_i}(\xi_{ij})f_m^{(a)}(\xi_{ij}) + n_{y_i}(\xi_{ij})e_m^{(a)}(\xi_{ij}),$$

$$\psi_m^{(a)}(\xi_{ij}) = n_{x_i}(\xi_{ij})\beta_m^{(a)}(\xi_{ij}) - n_{y_i}(\xi_{ij})\alpha_m^{(a)}(\xi_{ij}).$$

6.3 The matrix equation for the determination of the model parameters

The boundary conditions (eqs 50–61) are linear with respect to the coefficients (which are the model parameters). As a result, the system of equations, which is composed of the boundary conditions, is linear and can be represented by a matrix equation whose solution is a vector, where the elements are the parameters we are interested in. The matrix equation is given by

$$A\mathbf{x} = \mathbf{b} \quad (62)$$

where the matrix A is

$$A = \begin{bmatrix}
\mathbf{g}_0^{(a)}(\xi_{11}) & \mathbf{g}_0^{(b)}(\xi_{11}) & \mathbf{g}_0^{(c)}(\xi_{11}) & \mathbf{g}_0^{(d)}(\xi_{11}) & \cdots & \mathbf{g}_M^{(d)}(\xi_{11}) & 0 & 0 & 0 & 0 & \cdots & 0 \\
\mathbf{r}_0^{(a)}(\xi_{11}) & \mathbf{r}_0^{(b)}(\xi_{11}) & \mathbf{r}_0^{(c)}(\xi_{11}) & \mathbf{r}_0^{(d)}(\xi_{11}) & \cdots & \mathbf{r}_M^{(d)}(\xi_{11}) & 0 & 0 & 0 & 0 & \cdots & 0 \\
\mathbf{r}_0^{(a)}(\xi_{12}) & \mathbf{r}_0^{(b)}(\xi_{12}) & \mathbf{r}_0^{(c)}(\xi_{12}) & \mathbf{r}_0^{(d)}(\xi_{12}) & \cdots & \mathbf{r}_M^{(d)}(\xi_{12}) & 0 & 0 & 0 & 0 & \cdots & 0 \\
\mathbf{h}_0^{(a)}(\xi_{12}) & \mathbf{h}_0^{(b)}(\xi_{12}) & \mathbf{h}_0^{(c)}(\xi_{12}) & \mathbf{h}_0^{(d)}(\xi_{12}) & \cdots & \mathbf{h}_M^{(d)}(\xi_{12}) & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & \mathbf{g}_0^{(a)}(\xi_{21}) & \mathbf{g}_0^{(b)}(\xi_{21}) & \mathbf{g}_0^{(c)}(\xi_{21}) & \mathbf{g}_0^{(d)}(\xi_{21}) & \cdots & \mathbf{g}_M^{(d)}(\xi_{21}) \\
0 & 0 & 0 & 0 & \cdots & 0 & \mathbf{h}_0^{(a)}(\xi_{21}) & \mathbf{h}_0^{(b)}(\xi_{21}) & \mathbf{h}_0^{(c)}(\xi_{21}) & \mathbf{h}_0^{(d)}(\xi_{21}) & \cdots & \mathbf{h}_M^{(d)}(\xi_{21}) \\
0 & 0 & 0 & 0 & \cdots & 0 & \mathbf{r}_0^{(a)}(\xi_{22}) & \mathbf{r}_0^{(b)}(\xi_{22}) & \mathbf{r}_0^{(c)}(\xi_{22}) & \mathbf{r}_0^{(d)}(\xi_{22}) & \cdots & \mathbf{r}_M^{(d)}(\xi_{22}) \\
0 & 0 & 0 & 0 & \cdots & 0 & \mathbf{h}_0^{(a)}(\xi_{22}) & \mathbf{h}_0^{(b)}(\xi_{22}) & \mathbf{h}_0^{(c)}(\xi_{22}) & \mathbf{h}_0^{(d)}(\xi_{22}) & \cdots & \mathbf{h}_M^{(d)}(\xi_{22}) \\
\mathbf{a}_0^{(a)}(\xi_{13}) & \mathbf{a}_0^{(b)}(\xi_{13}) & \mathbf{a}_0^{(c)}(\xi_{13}) & \mathbf{a}_0^{(d)}(\xi_{13}) & \cdots & \mathbf{a}_M^{(d)}(\xi_{13}) & \mathbf{a}_0^{(a)}(\xi_{23}) & \mathbf{a}_0^{(b)}(\xi_{23}) & \mathbf{a}_0^{(c)}(\xi_{23}) & \mathbf{a}_0^{(d)}(\xi_{23}) & \cdots & \mathbf{a}_M^{(d)}(\xi_{23}) \\
\boldsymbol{\beta}_0^{(a)}(\xi_{13}) & \boldsymbol{\beta}_0^{(b)}(\xi_{13}) & \boldsymbol{\beta}_0^{(c)}(\xi_{13}) & \boldsymbol{\beta}_0^{(d)}(\xi_{13}) & \cdots & \boldsymbol{\beta}_M^{(d)}(\xi_{13}) & \boldsymbol{\beta}_0^{(a)}(\xi_{23}) & \boldsymbol{\beta}_0^{(b)}(\xi_{23}) & \boldsymbol{\beta}_0^{(c)}(\xi_{23}) & \boldsymbol{\beta}_0^{(d)}(\xi_{23}) & \cdots & \boldsymbol{\beta}_M^{(d)}(\xi_{23}) \\
\boldsymbol{\gamma}_0^{(a)}(\xi_{13}) & \boldsymbol{\gamma}_0^{(b)}(\xi_{13}) & \boldsymbol{\gamma}_0^{(c)}(\xi_{13}) & \boldsymbol{\gamma}_0^{(d)}(\xi_{13}) & \cdots & \boldsymbol{\gamma}_M^{(d)}(\xi_{13}) & \boldsymbol{\gamma}_0^{(a)}(\xi_{23}) & \boldsymbol{\gamma}_0^{(b)}(\xi_{23}) & \boldsymbol{\gamma}_0^{(c)}(\xi_{23}) & \boldsymbol{\gamma}_0^{(d)}(\xi_{23}) & \cdots & \boldsymbol{\gamma}_M^{(d)}(\xi_{23}) \\
\boldsymbol{\tau}_0^{(a)}(\xi_{13}) & \boldsymbol{\tau}_0^{(b)}(\xi_{13}) & \boldsymbol{\tau}_0^{(c)}(\xi_{13}) & \boldsymbol{\tau}_0^{(d)}(\xi_{13}) & \cdots & \boldsymbol{\tau}_M^{(d)}(\xi_{13}) & \boldsymbol{\rho}_0^{(a)}(\xi_{23}) & \boldsymbol{\rho}_0^{(b)}(\xi_{23}) & \boldsymbol{\rho}_0^{(c)}(\xi_{23}) & \boldsymbol{\rho}_0^{(d)}(\xi_{23}) & \cdots & \boldsymbol{\rho}_M^{(d)}(\xi_{23})
\end{bmatrix}, \quad (63)$$

where

$$\tau_m^{(a)}(\xi_{13}) = \rho_m^{(a)}(\xi_{13}) + k\psi_m^{(a)}(\xi_{13}).$$

The vector **b** is

$$\begin{bmatrix} \mathbf{0}_{N\times 1} \\ \mathbf{0}_{N\times 1} \\ \mathbf{0}_{N\times 1} \\ [-p]_{N\times 1} \\ \mathbf{0}_{N\times 1} \\ \mathbf{0}_{N\times 1} \\ \mathbf{0}_{N\times 1} \\ [-p]_{N\times 1} \\ \mathbf{0}_{N\times 1} \\ \mathbf{0}_{N\times 1} \\ \mathbf{0}_{N\times 1} \\ \mathbf{0}_{N\times 1} \end{bmatrix}. \tag{64}$$

Given A and **b**, we can calculate the vector of coefficients

$$\mathbf{x} = \begin{bmatrix} a_0^{(1)} \\ b_0^{(1)} \\ p_0^{(1)} \\ q_0^{(1)} \\ \vdots \\ q_M^{(1)} \\ a_0^{(2)} \\ b_0^{(2)} \\ p_0^{(2)} \\ q_0^{(2)} \\ \vdots \\ q_M^{(2)} \end{bmatrix}.$$

Each component vector of A contains $N+1$ elements. There are thus $12(N+1)$ lines in matrix A, which means $12(N+1)$ equations in the system. The number of unknowns is $8(M+1)$. The system has to be solved for $12(N+1) \geq 8(M+1)$ unknowns.

If we examine eqs (A7) and (A8) for the components of the displacement in the x- and y-directions, respectively, we can conclude that the a_0 and b_0 coefficients are related to the displacement of a rigid body, b_1, which is related to the rotation, and both p_0 and q_0 will mean simple translations. Then, from the systems presented here, $a_0 = b_0 = 0$ for both cells. Furthermore, we can say that $b_1 = 0$ for all cases where the elastic properties of the two cells are the same.

In order to finish the application of the method to the cases that we consider in this paper, we modify the system of equations in a similar fashion to what was first used in seismology by Aki & Larner (1970) with the Fourier transform by applying the wavelet transform (Daubechies 1992; Vetterli & Kovacevic 1995), with $N+1$ basis functions, with the boundary conditions with respect to ξ. This is equivalent to applying the wavelet transform with $N+1$ basis functions to each vector component of A and **b**.

7 ANALYSIS CRITERIA

The faults modelled in this work present such complexity that there is no analytical solution for them. For these cases, we will use two ways of looking at the results. The first criteria is a tentative estimate of the degree of contamination of the solution by errors introduced by the inversion method. We assume that the system is described by the matrix equation $A\mathbf{x} = \mathbf{b}$, where A and $\mathbf{b}$ are known and A is ill-conditioned. The least-squares solution for this system, using the singular value decomposition of A, is given by

$$\mathbf{x}_{LS} = A^+ \mathbf{b},$$

where A^+ is the pseudo-inverse of A. Bearing in mind that with the least-squares method we obtain a solution x_{LS} that minimizes $\rho_{LS} = \|A\mathbf{x}_{LS} - \mathbf{b}\|$, we can use, as a measure of the error, the value of ρ_{LS}. However, this is an absolute measure; a relative measure of the error would be more convenient for us.

Define an error vector relative to the external applied force, P:

$$\mathbf{e}_p = \frac{|A\mathbf{x}_{LS} - \mathbf{b}|}{P}.$$

Define also the error vector relative to the lateral length of the square:

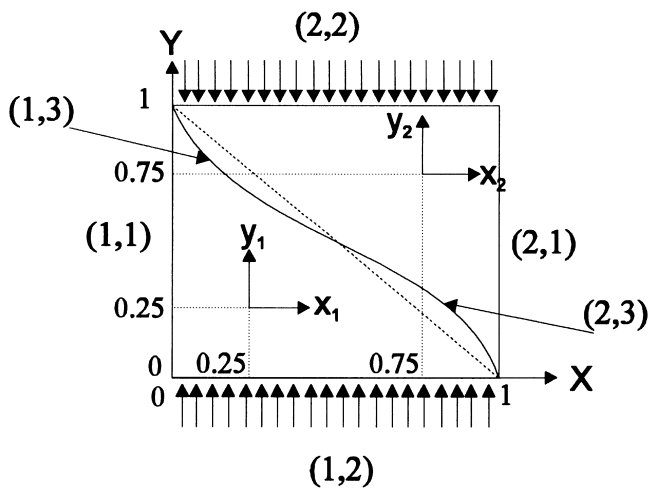$$\mathbf{e}_U = \frac{|A\mathbf{x}_{LS} - \mathbf{b}|}{U}.$$

Figure 29. Definition of the system geometry, which will also be used to validate the modelling method.

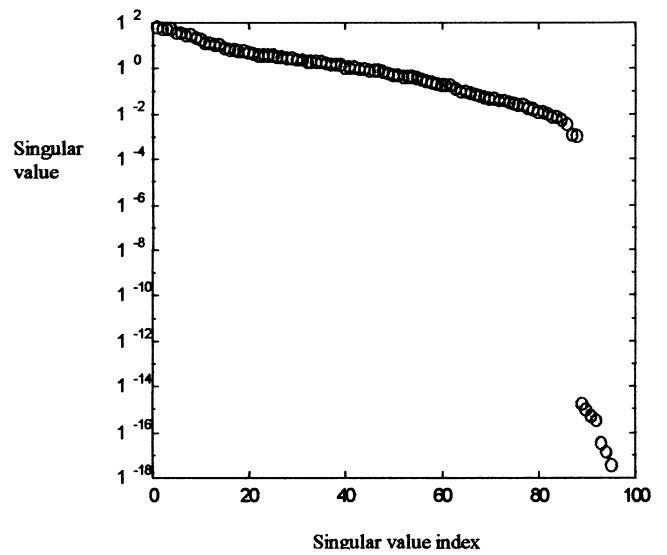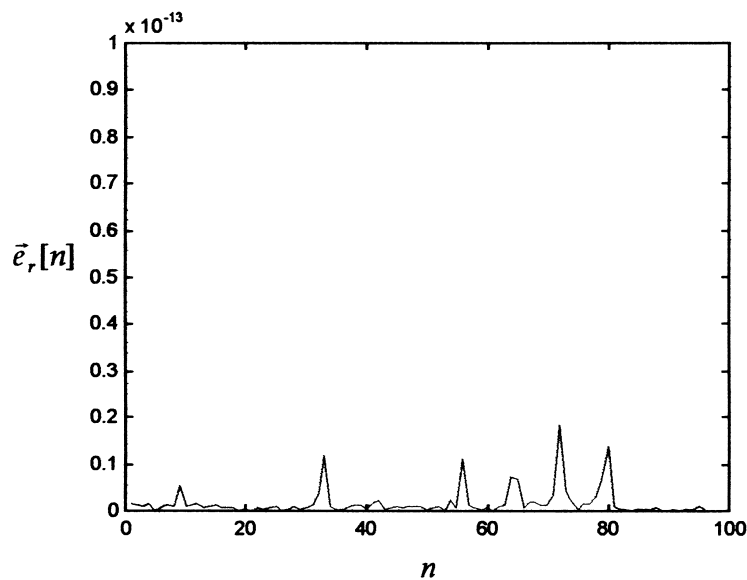**Figure 30.** Singular value decomposition of A

Figure 31. Relative error in the traction and displacement components.
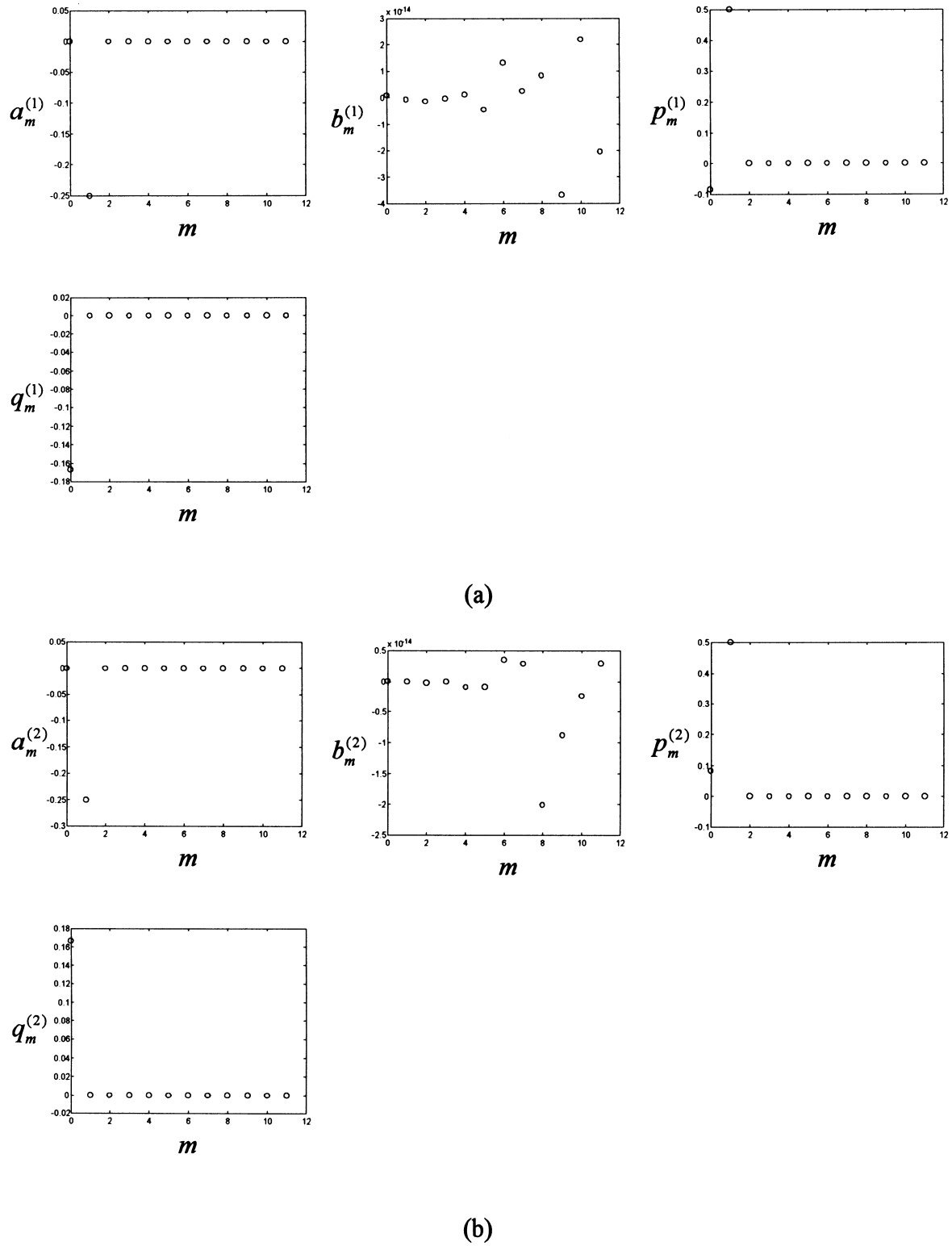
(a)

(b)

Figure 32. Model parameters for (a) cell 1 and (b) cell 2.

Considering that both P and U are unity, we can define a relative error vector,

$$\mathbf{e}_r = \frac{|A\mathbf{x}_{\mathrm{LS}} - \mathbf{b}|}{1}.$$

Since the only boundary conditions that are related to the displacement are the last two elements in the vector, the $2(N+1)$ lines of the vector $\mathbf{e}_r$ contain the normal and the parallel displacement errors relative to the length of the square. The other lines contain errors in

the traction determined along the segments of the contour relative to the applied external traction. We then fixed the order of the maximum relative displacement and the traction errors to be 10^{-3}. If the error introduced by the inversion method is not significant or acceptable ($e_r < 10^{-3}$), then we use the second criterion, which is the subjective analysis of the results.

However, the inversion method used is not the only source of errors in the problem. There are also errors due to the truncation of the polynomial order. This problem can be avoided if we combine two methods. The first is the use of the wavelet transform to minimize such errors. The second consists of remodelling the system with a polynomial of higher order. If the solution obtained with the model that uses a higher-order polynomial, say M_2, is the same as that obtained with the model that uses polynomials of order M_1 ($M_1 < M_2$), then we can consider that the method that uses polynomials of order M_1 describes the system considered totally, that is, the truncation error is zero. Note that this strategy does not say that using polynomials of order M_1 is best; that is, there can be polynomials of even lower order that totally represent the system considered. We highlight the fact that the results obtained by using higher-order polynomials are not presented because we would have a redundancy of data, given that the solutions obtained by higher-order polynomials, would be similar to those shown here.

8   FIRST FAULT MODEL

After we have built the system of equations $A\mathbf{x} = \mathbf{b}$, where A and $\mathbf{b}$ are known, we calculate the vector of coefficients $\mathbf{x}$ (model parameters). The system is that defined by Fig. 29 with $G_1 = G_2 = 1$, $\mu_1 = \mu_2 = 0.5$ and the fault amplitude $F = 0.1$. Modelling the system with polynomials of order $M = 11$ and with $N = 7$ discretization points, we obtain a matrix A with 96 lines ($12N + 12$) and 96 columns ($8M + 8$). From the singular value decomposition of A (Fig. 30), we note that matrix A has an incomplete numerical rank equal to 89.

Note that a few singular values (treated here as zero) are much smaller than most of the singular values considered. According to the above discussion, we can thus trust the solution obtained, because the A matrix can be considered, without ambiguity, as having a rank equal to 89. In fact, the relative error on the determined traction and displacement components (Fig. 31) is almost of the same order as the double precision used in the computer program for the solution of the problem (i.e. 10^{-16}). The solution, in terms of coefficients, is shown in Fig. 32.

Using the coefficients presented in Fig. 32, we calculated the deformation (using the displacement values) of the system. This result is shown in Fig. 33. In this case, note the symmetry on the deformations suffered by the system. This was expected intuitively because the elastic properties of the two cells are identical and both are welded together, that is, the system works as if it was a single body.

When we compress matrix A, we reduce the computer memory size required to store it. In this case, it was reduced from 73 728 to 25 488 bytes. The percentage of non-zero elements was reduced from 59.03 to 22.7 per cent. Although the result is not shown here, the deformation sustained by the system, when we use the compressed version of the matrix, is very close to that shown in Fig. 33.

9   SECOND FAULT MODEL

The system modelled in this section is also that shown in Fig. 29, considering the elastic properties $G_1 = 10$, $G_2 = 1$ and $\mu_1 = \mu_2 = 1/3$. Furthermore, the maximum amplitude of the sinusoidal interface is $F = 0.1$. The system described above was modelled using
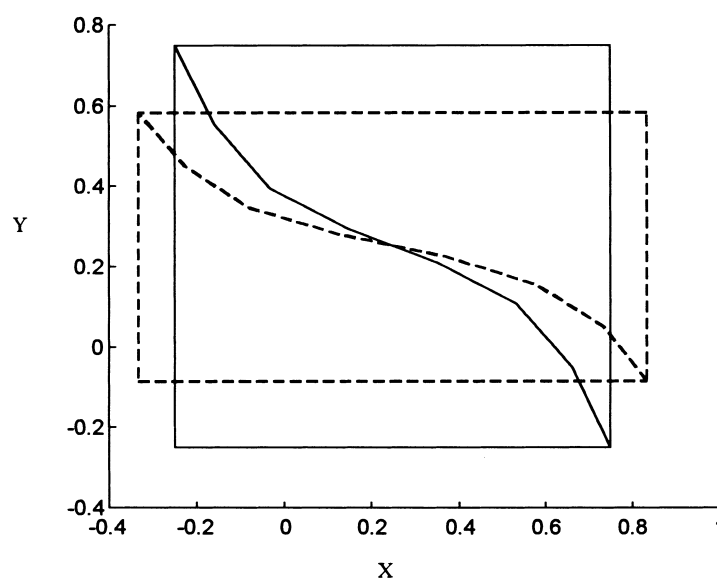

Figure 33. Solution of the problem. The dashed line represents the deformed body, while the full line shows the original body.

polynomials of order $M = 23$ and $N = 15$ discretization points for each contour segment. Having done this, we obtained a system of linear equations represented by the matrix equation $A\mathbf{x} = \mathbf{b}$. When solving such a system using the pseudo-inverse, we obtain the parameters shown in Fig. 34.

Using such coefficients in the computation of the displacement and of the traction components, we obtained the relative error vector (see Section 7 for more details) shown in Fig. 35. Keeping in mind that the errors are tolerable, we used the model parameters to determine the deformation sustained by the system. The result is shown in Fig. 36. Qualitatively, the deformation experienced by the system is very consistent. Note that cell 2 (the upper cell), was deformed more than cell 1. This was expected, since the rigidity of cell 1
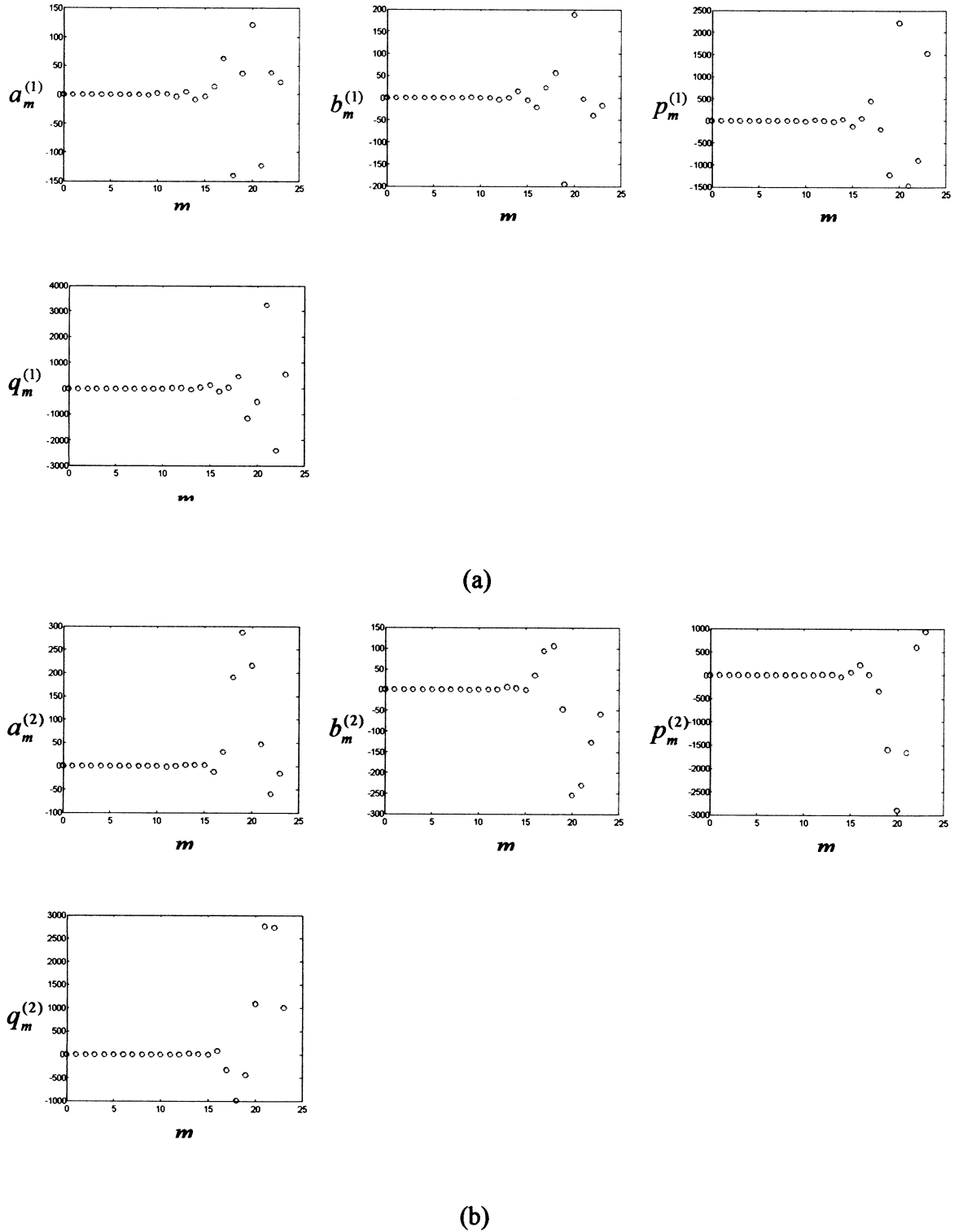

(a)

(b)

Figure 34. Coefficients of the model calculated using the pseudo-inverse. (a) Coefficients of cell 1; (b) coefficients of cell 2.
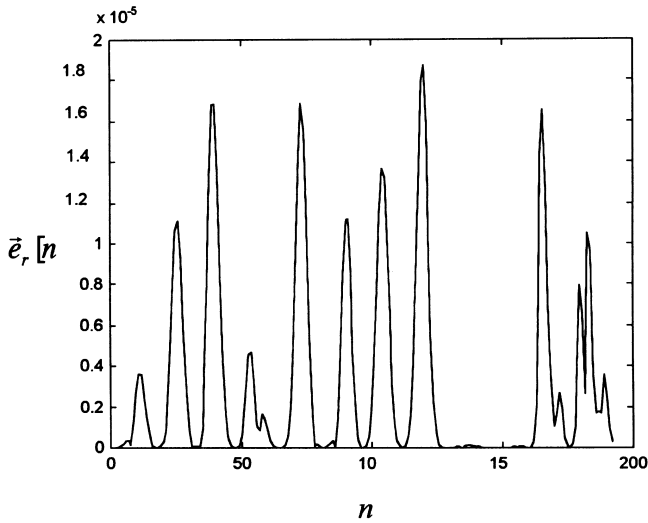
Figure 35. Relative error as a function of the traction and displacement components relative to the external traction applied and the lateral length of the square, respectively.
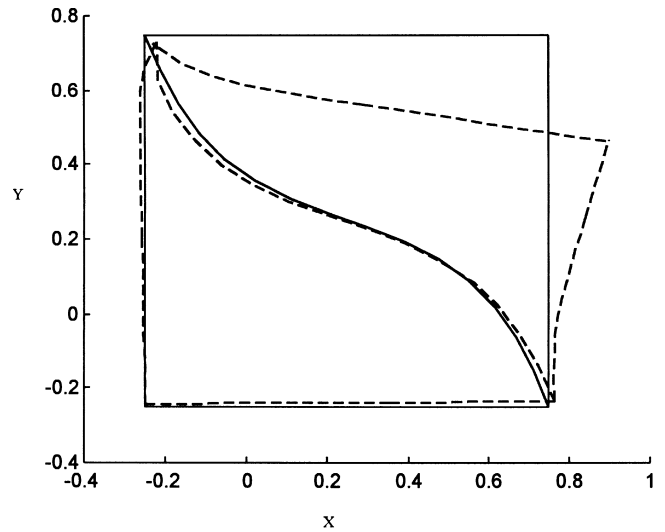

Figure 36. Deformation undergone by the system considered due to the application of an external traction, calculated using the model parameters.

is 10 times greater than the rigidity of cell 2. Note also that cell 2 undergoes a rotation when deformed; that is, the right-hand side of this cell is deformed more than the left-hand side. In fact, the right-hand side of the system consists mainly of elements corresponding to cell 2, whose rigidity is smaller (Fig. 36).

10   THIRD FAULT MODEL

The third case consists of modelling the system presented in Fig. 29 with a sinusoidal internal segment with amplitude $F=0$. The elastic properties of the two cells are the same and are given by $G_1=G_2=1$ and $\mu_1=\mu_2=0.5$. The system was modelled using polynomials of order $M=8$ and $N=7$ discretization points, so we obtained a system of 96 equations ($12N+12$) and 72 unknowns ($8M+8$). The rank of the matrix is incomplete, but can be chosen, without any ambiguity, as being rank(A) $=64$. Since the numerical rank is very well determined (Fig. 37), we expect the relative errors on the traction and on the displacement components to be very small. In fact, the numerical order of the error is equal to the precision used in the calculation of the pseudo-inverse (10^{-16}). The relative error is shown in Fig. 38.

Once we have computed the pseudo-inverse, we obtain the model parameters shown in Fig. 39. Note that the coefficients of order higher than 2 all have null value, suggesting that polynomials of order 1 could have being used to model this system. Note that
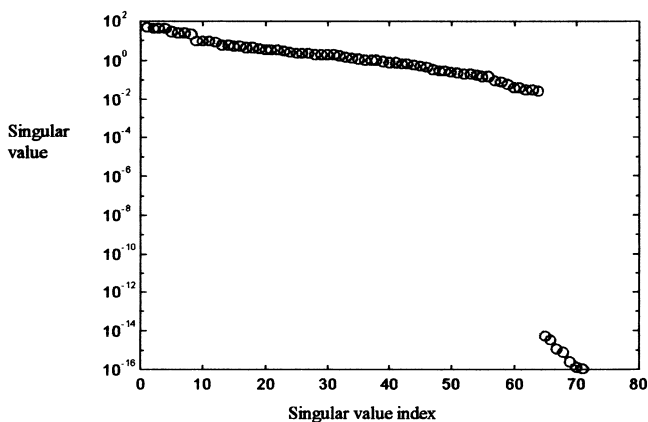

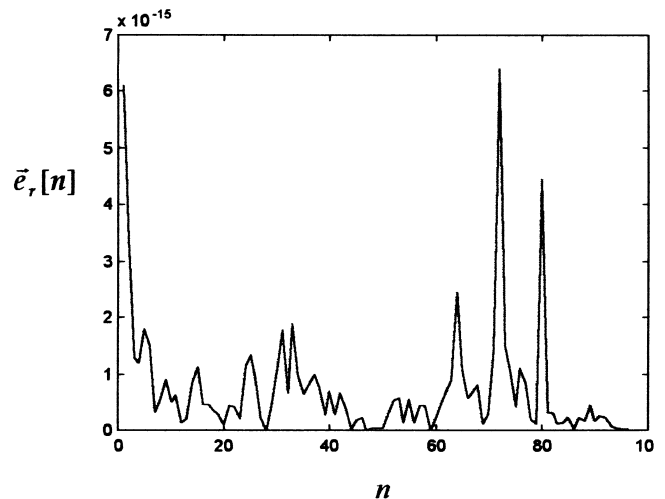**Figure 37.** Singular value decomposition of the A matrix.

Figure 38. Error related to the traction and the displacement (relative to the external applied traction and to the lateral length of the square, respectively). Both the original lateral length and the external traction have values equal to one.
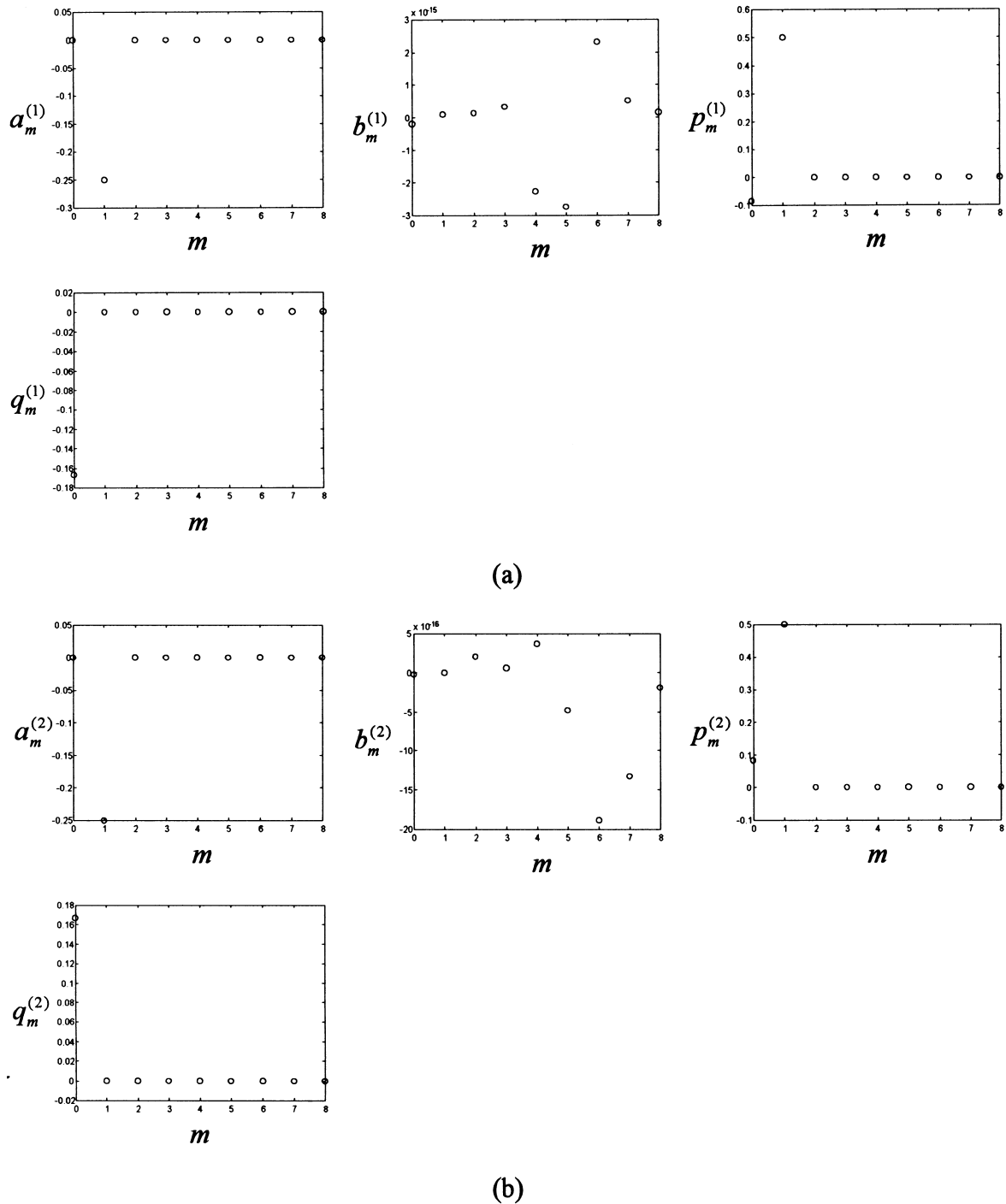
Figure 39. Coefficients of the model computed using the pseudo-inverse. (a) Cell 1 coefficients; (b) cell 2 coefficients.

all contours are straight and that the traction, applied to the system on the upper and lower borders, is constant along the entire segment. On the other hand, if we consider a system where the interface is sinusoidal, we need polynomials of higher order to describe the sinusoidal behaviour, which is reflected in the observed traction and displacement components.

After we have obtained the model parameters, we determine the model deformation, which we show in Fig. 40. Note that the system deforms in the same way as a single body, as in the first case (Fig. 33). In fact, the solutions obtained for the two cases are identical.

This system was made compact and we obtained good results. With the compaction procedure, the percentage of non-null elements changed from 56.48 to 15.89, and the size of the computer memory used was reduced from 55 296 bytes to 13 464 bytes. The solution corresponding to the compact equation problem is superposed on the solution of the non-compact matrix problem in Fig. 41, revealing a very close similarity between the two.
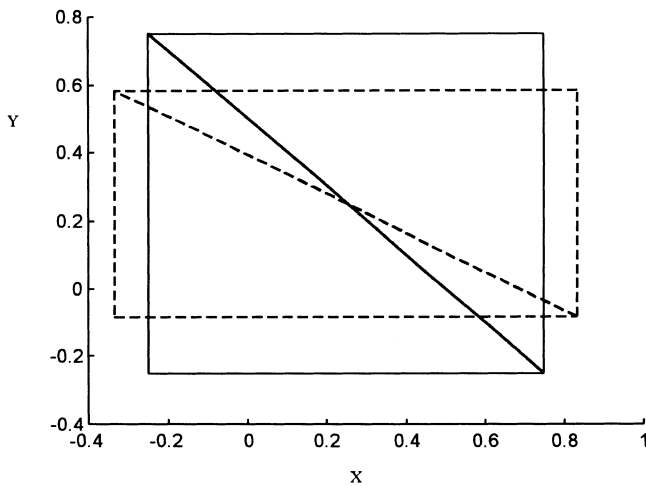
Figure 40. Deformed body solution compared to the original body. The dashed line represents the deformed body, while the full line represents the original body.
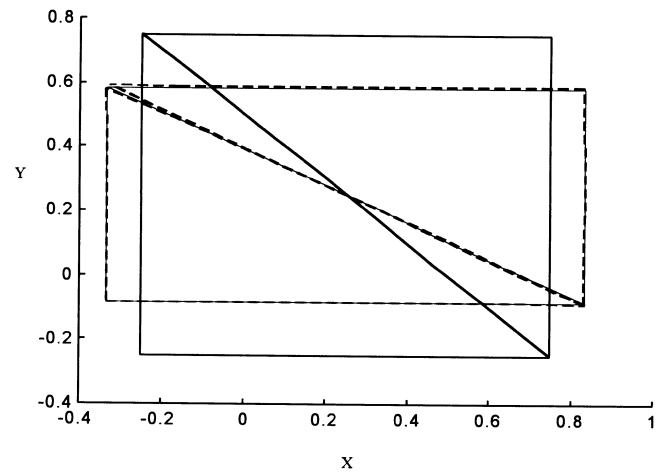

Figure 41. Comparison of the solutions obtained for the system when matrix *A* is in its compact form (dashed line) and when it is not compact (full line) for the determination of the solution of the problem.

Why did this system show good results when the compact matrix was used when the solution of other systems (with a sinusoidal interface, such as the first case, where the observed deformation is identical to that obtained for this case) did not? The reason for this is related to the degree of complexity of the two systems. The system for which the internal interface is sinusoidal is more complex than that with a straight line interface. As a consequence, the more complex system (with the sinusoidal interface) will show less redundancy of information in the matrix than the less complex system (that with a straight line interface).

11 CONCLUSIONS

We introduced a new method for modelling 2-D elastic media with the application of the wavelet transform. The method proved to be reliable and fast when we applied it to a set of six models for which we knew the analytical solution. Usage of the wavelet transform reduces the associated computational errors when we truncate the polynomial expansion and makes the system of linear equations more compact for the computation.

We expanded the method for modelling elastic media using the wavelet transform to cases composed of more than one elastic medium, including the simulation of faults separating two distinct elastic media. The results of six different tests, involving cases with relatively complex fault structures, proved that the method is suitable for modelling 2-D cases where the elastic constants of the media involved are known. Although we cannot compare the results with analytical solutions to the problems including faults, the results we obtain prove that the method is also effective when modelling such cases. We expect that this method will be useful in modelling real cases with great success, due to the fast and compact solution to the problem we obtain when using the approach to model 2-D elastic models.

REFERENCES

Aki, K. & Larner, K., 1970. Surface motion of a layered medium having an irregular interface due to incident plane SH waves, *J. geophys. Res.,* **75,** 933–954.

Daubechies, I., 1992. *Ten Lectures on Wavelets,* Society for Industrial and Applied Mathematics, Philadelphia, PA.

de Veubeke, B.M.F., 1979. *A Course in Elasticity,* Springer Verlag, New York.

Golub, G.H. & Reinsch, C., 1970., Singular value decomposition and least squares solutions, *Num. Math.,* **14,** 403–420.

Golub, G.H. & Van Loan, C.F., 1985. *Matrix Computation,* Johns Hopkins University Press, Baltimore, MD.

Vetterli, M. & Kovacevic, J., 1995. *Wavelets and Subband Coding,* Prentice Hall, Englewood Cliffs, NJ.

APPENDIX A: GENERAL PLANE STRESS PROBLEM

Using the theory developed by de Veubeke (1979), we can express the stress and displacement components as functions of F and H, which are analytical functions,

$$u = \frac{1}{2G} \mathcal{R}e\{(3 - \mu)F - (1 + \mu)\varsigma\overline{F'} + \overline{H'}\},$$ (A1)

$$v = \frac{1}{2G} \mathcal{I}m\{(3 - \mu)F - (1 + \mu)\varsigma\overline{F'} + \overline{H'}\},$$ (A2)

$$\sigma_x = \mathcal{R}e\{2(1 + v)F' - (1 + v)\overline{\zeta}F'' + H''\},$$ (A3)

$$\sigma_y = \mathcal{R}e\{2(1 + v)F' + (1 + v)\overline{\zeta}F'' - H''\},$$ (A4)

$$\tau_{xy} = \mathcal{I}m\{(1 + v)\overline{\zeta}F'' - H''\}.$$ (A5)

The determination of functions F and H is obviously related to the boundary conditions of the problem. We can usually distinguish the case in which the surface stress components are specified along the borders (called the first fundamental problem) from the case where the displacement components are specified there (called the second fundamental problem).

A1   Applications to the case where $F(\zeta)$ and $H(\zeta)$ are polynomials

The first analytical function that we look at is the polynomial function, which is a linear relationship including coefficients that allows the set-up of a system of linear equations for the determination of these coefficients:

$$\begin{cases} F = \dfrac{1}{1 + \mu} \displaystyle\sum_{m=0}^{M} (a_m + ib_m)(x + iy)^m \\[4mm] H' = \displaystyle\sum_{m=0}^{M} (p_m + iq_m)(x + iy)^m \end{cases}.$$ (A6)

From eqs (A1) and (A6), we obtain

$$2Gu(x, y) = \frac{3 - \mu}{1 + \mu} a_0 + p_0 + 2x \frac{1 - \mu}{1 + \mu} a_1 - \frac{4y}{1 + \mu} b_1 + xp_1ya_1 + \sum_{m=2}^{M} \mathcal{R}e\left\{\left[\frac{3 - \mu}{1 + \mu}(x + iy)^m - m(x^2 + y^2)(x - iy)^{m-2}\right]a_m\right.$$

$$\left. + i\left[\frac{3 - \mu}{1 + \mu}(x + iy)^m + m(x^2 + y^2)(x - iy)^{m-2}\right]b_m + [(x - iy)^m]p_m - [i(x - iy)^m]q_m\right\}.$$ (A7)

From eqs (A2) and (A6), we obtain

$$2Gv(x, y) = \frac{3 - \mu}{1 + \mu} b_0 - q_0 + 2y \frac{1 - \mu}{1 + \mu} a_1 + \frac{4x}{1 + \mu} b_1 - yp_1xa_1 + \sum_{m=2}^{M} \mathcal{I}m\left\{\left[\frac{3 - \mu}{1 + \mu}(x + iy)^m - m(x^2 + y^2)(x - iy)^{m-2}\right]a_m\right.$$

$$\left. + i\left[\frac{3 - \mu}{1 + \mu}(x + iy)^m + m(x^2 + y^2)(x - iy)^{m-2}\right]b_m + [(x - iy)^m]p_m - [i(x - iy)^m]q_m\right\}.$$ (A8)

From eqs (A3) and (A6), we obtain

$$\sigma_x(x, y) = 2a_1 + p_1 + 2xa_2 - 6yb_2 + 2xp_2 - 2yq_2 + \sum_{m=3}^{M} \mathcal{R}e\{m[(3 - m)x^2 - (1 + m)y^2 + 4ixy](x + iy)^{m-3}a_m$$

$$+ im[(3 - m)x^2 - (1 + m)y^2 + 4ixy](x + iy)^{m-3}b_m + m(x + iy)^{m-1}p_m + im(x + iy)^{m-1}q_m\}.$$ (A9)

From eqs (A4) and (A6), we obtain

$$\sigma_y(x, y) = 2a_1 - p_1 + 6xa_2 - 2yb_2 - 2xp_2 + 2yq_2 + \sum_{m=3}^{M} \mathcal{R}e\{m[(m + 1)x^2 + (m - 3)y^2 + 4ixy](x + iy)^{m-3}a_m$$

$$+ im[(m + 1)x^2 + (m - 3)y^2 + 4ixy](x + iy)^{m-3}b_m - m(x + iy)^{m-1}p_m - im(x + iy)^{m-1}q_m\}.$$ (A10)

From eqs (A5) and (A6), we obtain

$$\tau_{xy}(x, y) = -q_1 - 2ya_2 + 2xb_2 - 2yp_2 - 2xq_2 + \sum_{m=3}^{M} \mathcal{I}m\{m(m - 1)(x^2 + y^2)(x + iy)^{m-3}a_m$$

$$+ im(m - 1)(x^2 + y^2)(x + iy)^{m-3}b_m - m(x + iy)^{m-1}p_m - im(x + iy)^{m-1}q_m\}.$$ (A11)

We then define the following equations as a function of Cartesian coordinates:

$$u(x, y) = \sum_{m=0}^{M} \left[e_m^{(a)}(x, y)a_m + e_m^{(b)}(x, y)b_m + e_m^{(c)}(x, y)p_m + e_m^{(d)}(x, y)q_m \right], \tag{A12}$$

$$v(x, y) = \sum_{m=0}^{M} \left[f_m^{(a)}(x, y)a_m + f_m^{(b)}(x, y)b_m + f_m^{(c)}(x, y)p_m + f_m^{(d)}(x, y)q_m \right], \tag{A13}$$

$$\sigma_x(x, y) = \sum_{m=0}^{M} \left[g_m^{(a)}(x, y)a_m + g_m^{(b)}(x, y)b_m + g_m^{(c)}(x, y)p_m + g_m^{(d)}(x, y)q_m \right], \tag{A14}$$

$$\sigma_y(x, y) = \sum_{m=0}^{M} \left[h_m^{(a)}(x, y)a_m + h_m^{(b)}(x, y)b_m + h_m^{(c)}(x, y)p_m + h_m^{(d)}(x, y)q_m \right], \tag{A15}$$

$$\tau_{xy}(x, y) = \sum_{m=0}^{M} \left[r_m^{(a)}(x, y)a_m + r_m^{(b)}(x, y)b_m + r_m^{(c)}(x, y)p_m + r_m^{(d)}(x, y)q_m \right]. \tag{A16}$$

From eqs (A12) and (A7), we obtain the following set of equations:

$$\begin{cases} e_0^{(a)} = \dfrac{1}{2G}\dfrac{3-v}{1+v} \\[2mm] e_1^{(a)} = \dfrac{1}{G}\dfrac{1-v}{1+v}\,x \\[2mm] e_m^{(a)} = \dfrac{1}{2G}\dfrac{3-v}{1+v}\,\mathscr{Im}\{(x+iy)^m\} - \dfrac{m}{2G}(x^2+y^2)\mathscr{Re}\{(x-iy)^{m-2}\}, \quad m\geq 2 \\[2mm] e_1^{(b)} = 0 \\[2mm] e_1^{(b)} = -\dfrac{2}{G(1+v)}\,y \\[2mm] e_m^{(b)} = -\dfrac{1}{2G}\dfrac{3-v}{1+v}\,\mathscr{Im}\{(x+iy)^m\} - \dfrac{m}{2G}(x^2+y^2)\mathscr{Im}\{(x-iy)^{m-2}\}, \quad m\geq 2 \\[2mm] e_m^{(c)} = \dfrac{1}{2G}\,\mathscr{Re}\{(x-iy)^m\}, \quad m\geq 0 \\[2mm] e_m^{(d)} = \dfrac{1}{2G}\,\mathscr{Im}\{(x-iy)^m\}, \quad m\geq 0 \end{cases} \tag{A17}$$

From eqs (A13) and (A8), we obtain

$$\begin{cases} f_0^{(a)} = 0 \\[2mm] f_1^{(a)} = \dfrac{1}{G}\dfrac{1-v}{1+v}\,y \\[2mm] f_m^{(a)} = \dfrac{1}{2G}\dfrac{3-v}{1+v}\,\mathscr{Im}\{(x+iy)^m\} - \dfrac{m}{2G}(x^2+y^2)\mathscr{Im}\{(x-iy)^{m-2}\}, \quad m\geq 2 \\[2mm] f_0^{(b)} = \dfrac{1}{2G}\dfrac{3-v}{1+v} \\[2mm] f_1^{(b)} = -\dfrac{2}{G(1+v)}\,x \\[2mm] f_m^{(b)} = \dfrac{1}{2G}\dfrac{3-v}{1+v}\,\mathscr{Re}\{(x+iy)^m\} - \dfrac{m}{2G}(x^2+y^2)\mathscr{Re}\{(x-iy)^{m-2}\}, \quad m\geq 2 \\[2mm] f_m^{(c)} = \dfrac{1}{2G}\,\mathscr{Im}\{(x-iy)^m\}, \quad m\geq 0 \\[2mm] e_m^{(d)} = -\dfrac{1}{2G}\,\mathscr{Re}\{(x-iy)^m\}, \quad m\geq 0 \end{cases} \tag{A18}$$

From eqs (A14) and (A9), we get the set of equations

$$
\begin{cases}
g_1^{(a)} = 2 \\[6pt]
g_2^{(a)} = 2x \\[6pt]
g_m^{(a)} = m\,\mathcal{R}e\{[(3-m)x^2 - (1+m)y^2 + 4ixy][x+iy]^{m-3}\}, \qquad m \geq 3 \\[6pt]
g_1^{(b)} = 0 \\[6pt]
g_2^{(b)} = -6y \\[6pt]
g_m^{(b)} = -m\,\mathcal{I}m\{[(3-m)x^2 - (1+m)y^2 + 4ixy][x+iy]^{m-3}\}, \qquad m \geq 3 \\[6pt]
g_1^{(c)} = 1 \\[6pt]
g_2^{(c)} = 2x \\[6pt]
g_m^{(c)} = m\,\mathcal{R}e\{(x+iy)^{m-1}\}, \qquad m \geq 3 \\[6pt]
g_1^{(d)} = 0 \\[6pt]
g_2^{(d)} = -2y \\[6pt]
g_m^{(d)} = -m\,\mathcal{I}m\{(x+iy)^{m-1}\}, \qquad m \geq 3
\end{cases}
\tag{A19}
$$

Comparing eqs (A15) and (A10), we obtain the set of equations

$$
\begin{cases}
h_1^{(a)} = 2 \\[6pt]
h_2^{(a)} = 6x \\[6pt]
h_m^{(a)} = m\,\mathcal{R}e\{[(1+m)x^2 - (m-3)y^2 + 4ixy][x+iy]^{m-3}\}, \qquad m \geq 3 \\[6pt]
h_1^{(b)} = 0 \\[6pt]
h_2^{(b)} = -2y \\[6pt]
h_m^{(b)} = -m\,\mathcal{I}m\{[(1+m)x^2 - (m-3)y^2 + 4ixy][x+iy]^{m-3}\}, \qquad m \geq 3 \\[6pt]
h_1^{(c)} = -1 \\[6pt]
h_2^{(c)} = -2x \\[6pt]
h_m^{(c)} = m\,\mathcal{R}e\{(x+iy)^{m-1}\}, \qquad m \geq 3 \\[6pt]
h_1^{(d)} = 0 \\[6pt]
h_2^{(d)} = 2y \\[6pt]
h_m^{(d)} = m\,\mathcal{I}m\{(x+iy)^{m-1}\}, \qquad m \geq 3
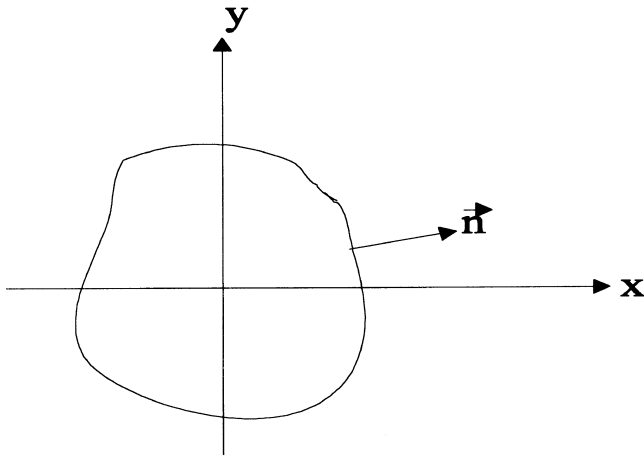\end{cases}
\tag{A20}
$$

Figure A1. Definition of the normal reference vector to the contour used.
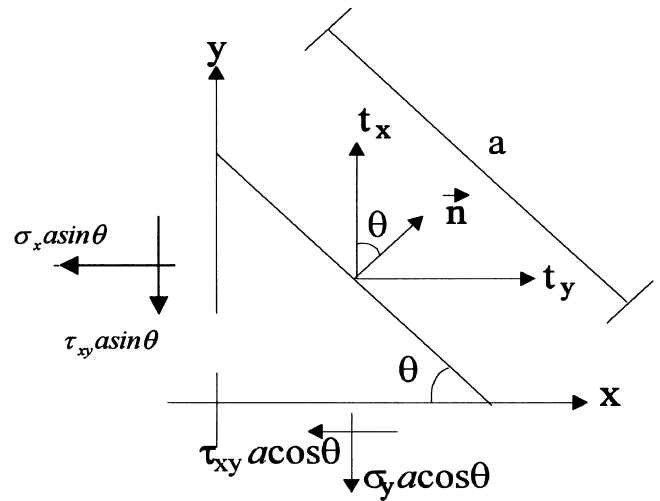

Figure A2. Definition of the *x*- and *y*-components of the traction vector.

From eqs (A16) and (A11), we obtain the set of equations

$$
\begin{cases}
r_1^{(a)} = 0 \\
r_2^{(a)} = -2y \\
r_m^{(a)} = m(m-1)\mathscr{Im}\{(x^2+y^2)(x+iy)^{m-3}\}, \quad m \geq 3 \\
r_1^{(b)} = 0 \\
r_2^{(b)} = 2x \\
r_m^{(b)} = (m-1)\mathscr{Re}\{[x^2+y^2][x+iy]^{m-3}\}, \quad m \geq 3 \\
r_1^{(c)} = 0 \\
r_2^{(c)} = -2y \\
r_m^{(c)} = -m\,\mathscr{Im}\{(x+iy)^{m-1}\}, \quad m \geq 3 \\
r_1^{(d)} = -1 \\
r_2^{(d)} = -2x \\
r_m^{(d)} = -m\,\mathscr{Re}\{(x+iy)^{m-1}\}, \quad m \geq 3
\end{cases}
\tag{A21}
$$

Finally, we obtain the equations for the case of the internal stress field. Define $\mathbf{n}(n_x, n_y)$ as the normal to the external contour (Fig. A1). The traction, $\mathbf{t}(t_x, t_y)$, is an internal force per unit area in the direction of the normal vector $\mathbf{n}$ (Fig. A2). We thus obtain

$$
t_x = \sigma_x n_x + \tau_{xy} n_y
\tag{A22}
$$

$$
t_y = n_x \tau_{xy} + n_y \sigma_y
\tag{A23}
$$

Substituting eqs (A14) and (A16) into eq. (A22), we obtain

$$
t_x = \sum_{m=1}^{M}\{(n_x g_m^{(a)} + n_y r_m^{(a)})a_m\} + \sum_{m=1}^{M}\{(n_x g_m^{(b)} + n_y r_m^{(b)})b_m\} + \sum_{m=1}^{M}\{(n_x g_m^{(c)} + n_y r_m^{(c)})p_m\} + \sum_{m=1}^{M}\{(n_x g_m^{(d)} + n_y r_m^{(d)})q_m\}.
\tag{A24}
$$

Substituting eqs (A15) and (A16) into eq. (A23), we obtain

$$
t_y = \sum_{m=1}^{M}\{(n_x r_m^{(a)} + n_y h_m^{(a)})a_m\} + \sum_{m=1}^{M}\{(n_x r_m^{(b)} + n_y h_m^{(b)})b_m\} + \sum_{m=1}^{M}\{(n_x r_m^{(c)} + n_y h_m^{(c)})p_m\} + \sum_{m=1}^{M}\{(n_x r_m^{(d)} + n_y h_m^{(d)})q_m\}.
\tag{A25}
$$