

An efficient multi-layer planar 3D fracture growth algorithm using a fixed mesh approach[†]

Eduard Siebrits^{1,*†} and Anthony P. Peirce²

¹Schlumberger, P.O. Box 4610, Houston, TX 77210-4610, U.S.A.

²Department of Mathematics, University of British Columbia, Vancouver, BC, Canada

SUMMARY

We present a planar three-dimensional (3D) fracture growth simulator, based on a displacement discontinuity (DD) method for multi-layer elasticity problems. The method uses a fixed mesh approach, with rectangular panel elements to represent the planar fracture surface. Special fracture tip logic is included that allows a tip element to be partially fractured in the tip region. The fracture perimeter is modelled in a piece-wise linear manner. The algorithm can model any number of interacting fractures that are restricted to lie on a single planar surface, located orthogonal to any number of parallel layers. The multiple layers are treated using a Fourier transform (FT) approach that provides a numerical Green's function for the DD scheme. The layers are assumed to be fully bonded together. Any fracture growth rule can be postulated for the algorithm. We demonstrate this approach on a number of test problems to verify its accuracy and efficiency, before showing some more general results. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: fracture growth simulation; layered elastic media; Fourier transforms; displacement discontinuity

1. INTRODUCTION

In the oil and gas industry, a hydraulic fracturing process is routinely used to increase production from oil and gas reservoirs. Many hydraulic fractures are injected into layered reservoirs, and care must be taken in the design of these treatments to avoid fracture growth into non-producing or water-bearing layers. Hydraulic fracturing simulators are used to design pumping schedules, monitor treatments in real time, and to evaluate the success of pumped treatments. For design purposes, hydraulic fractures are generally assumed to be planar in form, even though inhomogeneities in the reservoir can severely test this assumption. Hydraulic fractures are injected at depth and as a result the predominant fracture orientation is vertical. Reservoir layers are of sedimentary origin, and are thus usually, although not always, horizontally oriented.

Most hydraulic fracturing simulators use a single value for Young's modulus and Poisson's ratio, obtained by averaging elastic properties across the layers of the layered reservoir that are hydraulically fractured. Some simulators use various approximate techniques (e.g. References [1, 2]) to account for the layered nature of the reservoir. The most rigorous approach [3] currently used in the industry deals with multiple layered materials in an approximate manner by 'stacking up' a series of fully bonded single interface analytical solutions together. This approach (and other less accurate ones) can lead to significant errors in fracture width prediction in cases where elastic properties vary from one layer to the next. Errors are compounded by greater modulus contrasts, and by thinner layers.

In this paper, we present a multi-layered elastic model which provides highly accurate fracture width and stress predictions in a layered reservoir environment. We allow any number of parallel layers, with all layer interfaces restricted to being fully bonded. Layer thicknesses can range from 0.2 to 1000 m, i.e. over three orders of magnitude (scaling would also allow us to also study very thin layered materials, for example, such as are found in electronic components). Very thick and very thin layers can be located adjacent to each other. Layers can be assigned elastic properties that differ by more than an order of magnitude. The fracture plane can intersect any number of layer interfaces but is currently restricted to lie orthogonal to them; however, the method is quite general and will allow for the introduction of multiple inclined fractures. We neglect all fluid coupling effects because in this paper we deal exclusively with the multi-layer elasticity scheme.

The multi-layer elasticity scheme uses a FT-based approach, documented separately in References [4, 5], to generate a set of numerical Green's functions on a fixed parent mesh that spans the entire region of interest. Essential Green's function data are stored in memory, and embedded in a 3D DD method to allow solution of the fracture width profile given any set of applied traction loads that can vary on an element by element basis. Equation solution is performed iteratively using a highly efficient iterative solver. By prescribing appropriate boundary conditions in Fourier space, the problem domain may be made infinite or semi-infinite with a traction-free half-space surface.

In Section 2, we describe the planar 3D fracture geometry in a layered material. We outline the FT procedure which is used to construct the multi-layer Green's function in Section 3. In Section 4, we discuss discretization details and the assemblage of the influence coefficient matrix, followed by the equation solution procedure in Section 5. In Section 6, we discuss the method employed to track the moving fracture boundary, and the growth criterion used is summarized in Section 7. Section 8 describes three verification problems involving a partially loaded penny crack in a homogeneous material, a penny crack in a three-layer system, and a penny crack located near the interface of a two-material composite. In Section 9, we present results of a complex fracture growth simulation that demonstrates the generality of this approach in solving planar fracture growth problems in layered materials. Conclusions are drawn in Section 10.

2. PROBLEM DESCRIPTION

Figure 1 schematically depicts a typical problem of interest to us. A multi-layered elastic material, with all layer interfaces parallel and fully bonded, contains a planar fracture, in this case shown symmetric about $x = 0$. The top layer ends at a traction-free surface, and the

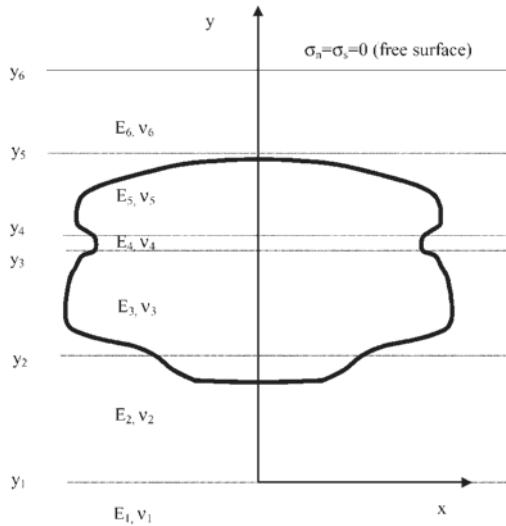


Figure 1. Multi-layered elastic material containing planar 3D fracture.

bottom layer is assumed to be of infinite thickness. The fracture plane is oriented at 90° to the layer interfaces, which means that all shear DDs on the fracture surface are zero. The normal tractions on the fracture surface are the known boundary conditions, and the fracture width is unknown. We will use the DD method [6], which is a boundary integral method designed to solve crack problems, to solve for the fracture width given a known fracture pressure.

3. BOUNDARY INTEGRAL EQUATION FOR A PLANAR CRACK IN A LAYERED MEDIUM

In this section, we describe the boundary integral equation that determines the width w of a planar crack which is associated with a prescribed normal pressure p applied to the bounding surfaces of the crack. Here the width w is defined to be the DD between the two surfaces which represent the crack. For a vertical crack subjected to a normal pressure, the boundary integral equation can be written in the form

$$p(x, y) = \int_A C(x, y; \xi, \eta) w(\xi, \eta) d\xi d\eta \quad (1)$$

where A is the region occupied by the planar fracture and $C(x, y; \xi, \eta)$ is the layer Green's function, which represents the horizontal stress $\sigma_{zz} = p$ at the point (x, y) due to a unit point normal DD $D_z = w$ located at the point (ξ, η) . Since we only need to track one stress component σ_{zz} and one DD component D_z for this problem, we choose, for notational convenience, to represent these by the symbols p and w , respectively. In the following subsections we will outline the FT procedure which is used to construct the layer Green's function in an efficient and accurate fashion.

3.1. Governing equations

Consider a linear elastic material that occupies a region in 3D space and which is in a state of equilibrium. In this case the stresses σ_{ij} and the strains $\varepsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i})$, which are defined in terms of the displacement gradients $u_{i,j} = \partial u_i / \partial x_j$ at any point within the body, are related by

$$\sigma_{ij} = \lambda \varepsilon_{kk} \delta_{ij} + 2G \varepsilon_{ij} \quad (2)$$

where λ and G are Lamé's constants that can be expressed in terms of the Young's modulus E and Poisson's ratio ν of the material by the formulae: $\lambda = Ev/[(1+\nu)(1-2\nu)]$ and $G = E/[2(1+\nu)]$. It is convenient to introduce the constants a , b , and f that are defined by: $a = \lambda + 2G$, $b = \lambda$ and $f = 2G$.

We assume that the elastic medium is in equilibrium so that the stresses satisfy the equilibrium equations

$$\sigma_{ij,j} + f_i = 0 \quad (3)$$

where f_i are the applied body forces.

It is also useful in this context, in which the layer properties do not change in the x and z (measured out of the page) directions but do vary in the y direction (see Figure 1), to rewrite the system equations (2) and (3) in the form of a system in which the y derivatives have been separated from the x and z derivatives:

$$\partial_y T = \mathcal{A}T + F \quad (4)$$

where T represents the vector of stresses and displacements defined by

$$T = [\sigma_{yy} \ \sigma_{xy} \ \sigma_{yz} \ u_y \ u_x \ u_z]^T$$

the body force vector is given by $F = [-f_y \ -f_x \ -f_z \ 0 \ 0 \ 0]^T$, and \mathcal{A} is the differential operator involving only x and z derivatives that is defined by

$$\mathcal{A} = \begin{bmatrix} 0 & -\partial_x & -\partial_z & 0 & 0 & 0 \\ -\frac{b}{a}\partial_x & 0 & 0 & 0 & \frac{(b^2 - a^2)}{a}\partial_{xx} - \frac{f}{2}\partial_{zz} & \left(\frac{(b^2 - ab)}{a} - \frac{f}{2}\right)\partial_{xz} \\ -\frac{b}{a}\partial_z & 0 & 0 & 0 & \left(\frac{(b^2 - ab)}{a} - \frac{f}{2}\right)\partial_{xz} & \frac{(b^2 - a^2)}{a}\partial_{zz} - \frac{f}{2}\partial_{xx} \\ \frac{1}{a} & 0 & 0 & 0 & -\frac{b}{a}\partial_x & -\frac{b}{a}\partial_z \\ 0 & \frac{2}{f} & 0 & -\partial_x & 0 & 0 \\ 0 & 0 & \frac{2}{f} & -\partial_z & 0 & 0 \end{bmatrix} \quad (5)$$

3.2. Reduction of layer partial differential equations to ordinary differential equations using the FT

In this section, we make use of the FT pair defined by

$$\hat{g}(m, n) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{i(mx+nz)} g(x, z) dx dz \quad (6)$$

$$g(x, z) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-i(mx+nz)} \hat{g}(m, n) dm dn \quad (7)$$

to reduce the governing system of partial differential equations into a system of layer ordinary differential equations.

By taking the FT of the operator equations (5), and introducing the change of variables (see Reference [7]):

$$\begin{aligned} \hat{u}_s &= -i(m\hat{u}_x + n\hat{u}_z)/k \\ \hat{u}_t &= -i(n\hat{u}_x - m\hat{u}_z)/k \end{aligned} \quad (8)$$

and

$$\begin{aligned} \hat{\tau}_s &= -i(m\hat{\sigma}_{xy} + n\hat{\sigma}_{yz})/k \\ \hat{\tau}_t &= -i(n\hat{\sigma}_{xy} - m\hat{\sigma}_{yz})/k \end{aligned} \quad (9)$$

we obtain

$$\partial_y \hat{T} = \hat{\mathcal{A}} \hat{T} + \hat{F} \quad (10)$$

where

$$\hat{\mathcal{A}} = \begin{bmatrix} 0 & -k & 0 & 0 & 0 & 0 \\ \frac{b}{a}k & 0 & 0 & \frac{(a^2 - b^2)}{a}k^2 & 0 & 0 \\ \frac{1}{a} & 0 & 0 & -\frac{b}{a}k & 0 & 0 \\ 0 & \frac{2}{f} & k & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{f}{2}k \\ 0 & 0 & 0 & 0 & \frac{2}{f} & 0 \end{bmatrix} \quad (11)$$

where $k = (m^2 + n^2)^{1/2}$ and the elements of \hat{T} and \hat{F} have been arranged as follows:

$$\hat{T} = [\hat{\sigma}_{yy}, \hat{\tau}_s, \hat{u}_y, \hat{u}_s, \hat{\tau}_t, \hat{u}_t]^T$$

and $\hat{F} = [-\hat{f}_y, -\hat{f}_s, 0, 0, -\hat{f}_t, 0]^T$.

We observe that unknowns involving $\hat{\sigma}_{yy}$, $\hat{\tau}_s$, \hat{u}_y and \hat{u}_s (the s -sub-system) are completely decoupled from the unknowns involving $\hat{\tau}_t$ and \hat{u}_t (the t -sub-system). The s -sub-system is sufficient to determine boundary value problems for 2D plane strain, while the autonomous t -sub-system is the only additional part that needs to be added to the plane strain equations in order to determine boundary value problems in 3D. A similar decoupling of the spectral ordinary differential equations also occurs if the Hankel transformation is applied to the layered elasticity problem (see, for example, References [8, 9]).

3.3. Exact solution to the layer ordinary differential equations and spectral coefficients

Considering the wavenumber k as a parameter, we can now determine the homogeneous solution to the system of ordinary differential equations (10) (see Reference [7]), which can be expressed in terms of solutions to the s -sub-system and the t -sub-system as follows:

$$\begin{bmatrix} T_s \\ T_t \end{bmatrix} = \begin{bmatrix} Z_s & 0 \\ 0 & Z_t \end{bmatrix} \begin{bmatrix} A_s \\ A_t \end{bmatrix} \quad (12)$$

where

$$\begin{aligned} T_s &= [\hat{\sigma}_{yy}^l/k \quad \hat{\tau}_s^l/k \quad \hat{u}_y^l \quad \hat{u}_s^l]^T \quad \text{and} \quad T_t = [\hat{\tau}_t^l/k \quad \hat{u}_t^l]^T \\ A_s &= [A_1 \quad A_2 \quad A_3 \quad A_4]^T \quad \text{and} \quad A_t = [A_5 \quad A_6]^T \end{aligned}$$

and

$$Z_s = \begin{bmatrix} -f e^{-ky} & (l_4 - fky)e^{-ky} & f e^{ky} & (l_4 + fky)e^{ky} \\ -f e^{-ky} & (l_5 - fky)e^{-ky} & -f e^{ky} & -(l_5 + fky)e^{ky} \\ e^{-ky} & kye^{-ky} & e^{ky} & kye^{ky} \\ e^{-ky} & (ky - l_2)e^{-ky} & -e^{ky} & -(ky + l_2)e^{ky} \end{bmatrix} \quad (13)$$

and

$$Z_t = \begin{bmatrix} -\frac{f}{2} e^{-ky} & \frac{f}{2} e^{ky} \\ e^{-ky} & e^{ky} \end{bmatrix} \quad (14)$$

The constants l_j in (12) are defined as follows:

$$l_2 = \frac{\lambda + 3G}{\lambda + G}, \quad l_4 = \frac{2G^2}{\lambda + G}, \quad l_5 = \frac{2G(\lambda + 2G)}{\lambda + G} \quad (15)$$

It is important to note that the spectral coefficients required to define the primary variables can be expressed entirely in terms of the single wavenumber parameter $k = \sqrt{m^2 + n^2}$. This property is a result of the invariance of the solution to rotations about the y -axis and can be

exploited to reduce the FT inversion problem from one which involves sampling the integrand at points throughout the (m, n) plane to what amounts to a 1D sampling of the wavenumber parameter k . From such rotationally invariant solutions we would not expect to be able to construct special solutions such as that for a vertical DD D_z , which is highly directional and certainly does not share the same rotational invariance. However, we will see later in this section that the solution for a point vertical DD can be expressed in terms of a linear combination of three separate stress–traction discontinuities each of which are rotationally invariant. The coefficients of these three elemental stress–traction discontinuities are functions of the directional wavenumbers m and n , which build in the required directional dependence of the solutions.

The unknown coefficients $A_j(k)$ depend on the parameter k and we will refer to them as the spectral coefficients throughout this paper. It will be seen that the spectral coefficients provide a useful representation of the solution as they separate the exponentially decaying part of the solution from the exponentially growing part of the solution. Once the spectral coefficients in any one layer are known, then using (12) it is possible to determine the stresses and displacements at any desired point within that layer.

The spectral coefficients $A_j(k)$ are determined by solving a system of algebraic equations for each wavenumber k . The algebraic equations are obtained by requiring that the tractions and displacements are continuous across each of the interfaces in the layer, defining boundary values for the problem, and prescribing the appropriate sources which represent a point DD. A normal point vertical DD with a displacement jump Δu in the z direction can be represented by stress–traction discontinuities of the following form (see Reference [9]):

$$[\hat{T}(y_s)] = \begin{bmatrix} 0 \\ \frac{\Delta u(b^2 - a^2)}{a} \\ \frac{\Delta u b}{a} \\ 0 \\ 0 \\ 0 \end{bmatrix} + \frac{m^2}{(m^2 + n^2)} \begin{bmatrix} 0 \\ \Delta u(a - b) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \frac{mn}{(m^2 + n^2)} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \Delta u(a - b) \end{bmatrix} \quad (16)$$

The stress–traction discontinuities $[\hat{T}(y_s)]$ can thus be decomposed into a linear combination of three discontinuity vectors, each of which is independent of the directional wavenumbers m and n . Thus it is possible to solve for the spectral coefficients for each of these discontinuity vectors for all the required wavenumbers k . These spectral coefficients are invariant to rotations about the vertical y -axis. The directional dependence required to construct the influences due to a vertical DD are therefore encapsulated in the above coefficients, which depend on the directional wave numbers m and n .

In order to model vertical fractures that run perpendicular to the layers, it is necessary to have an expression for the stress normal to the fracture surface. In the co-ordinate systems defined in Figure 3 it is necessary to determine the stress component $\hat{\sigma}_{zz}$, which can be defined

in terms of the spectral coefficients as follows:

$$\begin{aligned} k\hat{\sigma}_{zz} = & fn^2A_1e^{-ky} + (-l_6n^2 - l_7m^2 + fn^2ky)A_2e^{-ky} - fmnA_5e^{-ky} \\ & - fn^2A_3e^{ky} + (-l_6n^2 - l_7m^2 - fn^2ky)A_4e^{ky} - fmnA_6e^{ky} \end{aligned} \quad (17)$$

where we have defined the new constants $l_6 = 2G(2\lambda + 3G)/(\lambda + G)$ and $l_7 = 2\lambda G/(\lambda + G)$.

Once the values of the spectral coefficients $A_j(k)$ in each layer have been determined (this will be dealt with in more detail in the next section) and the FT of the displacements $\hat{u}_i(k)$ and stresses $\hat{\sigma}_{ij}(k)$ within each layer have been determined, then the displacements and stresses within each layer can be determined by applying formula (7) for the inversion of the FT.

3.4. Solution of the algebraic equations using the flexibility matrix method

In this section we describe the technique that is used to determine the spectral coefficients $A_j^l(k)$ which define the solution to a given boundary value problem. We first establish the equations that determine the stiffness properties of each of the layers in terms of the degrees of freedom of the model. For the spectral methods that we use in this paper, the degrees of freedom in the model are represented by the unknown spectral coefficients, that are parameterized by the wavenumber k . Once we have established equations for the stresses and displacements within each of the layers in terms of the internal degrees of freedom, we stitch all the layers involved in the problem together at their common interfaces by imposing conditions of continuity in displacements and tractions across the interfaces. Discontinuous sources (such as force discontinuities or displacement discontinuities) can be represented by introducing the appropriate jump conditions across pseudo-interfaces introduced for this purpose. Finally, the whole mechanical problem is completed into a well-posed system of equations by introducing the appropriate conditions at the boundaries of the pack of layers, e.g. specified tractions, specified displacements, or a complementary combination of tractions and displacements. From Equations (12)–(14), we observe that for very thick layers or large wavenumbers constructing the system of layer equations directly from the solution (12) will lead to overflow problems. The scheme we use is based on the flexibility matrix method (see References [10–14]), which we describe below.

Rather than determining the spectral coefficients $A_j^l(k)$ directly, the spectral coefficients are eliminated and the solutions for the displacements $[u_s] = [\hat{u}_y \hat{u}_s]^T$ at the top and bottom of a layer are expressed in terms of the solution for the stresses $[p] = [\hat{\sigma}_{yy}/k \hat{\tau}_s/k]^T$ at the top and bottom of the layers. In what follows we will, for the sake of brevity, not use the hats to denote the fact that u and p are in fact the FTs of the displacements and stresses. (Note that we only give the details here for the s -sub-system while those for the t -sub-system can be obtained by following a similar sequence of steps).

In what follows the index l refers to the layer number while the subscripts t and b refer to the quantities defined at the top and bottom of the layer. Equations (12) can be used to express the displacements in terms of stresses using the flexibility matrix R^l as follows:

$$\begin{bmatrix} u_t^l \\ u_b^l \end{bmatrix} = \begin{bmatrix} R_{tt}^l & R_{tb}^l \\ R_{bt}^l & R_{bb}^l \end{bmatrix} \begin{bmatrix} p_t^l \\ p_b^l \end{bmatrix} \quad (18)$$

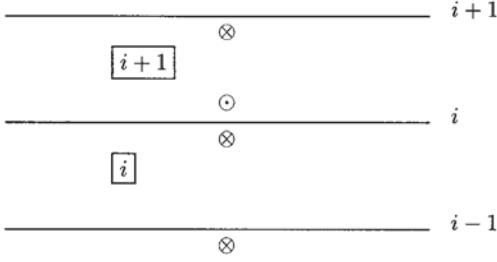


Figure 2. Flexibility matrix sample points.

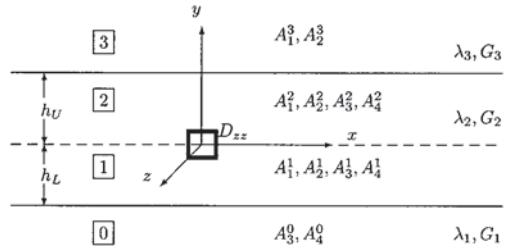


Figure 3. A vertical DD in the middle layer of three bonded elastic layers.

where the flexibility sub-matrices for the s -sub-system are as follows:

$$R_{tt} = \frac{1}{D} \begin{bmatrix} -l_5(\text{th} + kd \text{se}^2) & -(l_4\text{th}^2 + fk^2d^2\text{se}^2) \\ -(l_4\text{th}^2 + fk^2d^2\text{se}^2) & -l_5(\text{th} - kd \text{se}^2) \end{bmatrix} \quad (19)$$

$$R_{bb} = \frac{1}{D} \begin{bmatrix} l_5(\text{th} + kd \text{se}^2) & -(l_4\text{th}^2 + fk^2d^2\text{se}^2) \\ -(l_4\text{th}^2 + fk^2d^2\text{se}^2) & l_5(\text{th} - kd \text{se}^2) \end{bmatrix} \quad (20)$$

$$R_{bt} = \frac{l_5}{D} \begin{bmatrix} -(th + kd)se & -k d th se \\ k d th se & -(th - kd)se \end{bmatrix} \quad (21)$$

$$R_{tb} = \frac{l_5}{D} \begin{bmatrix} (th + kd)se & -k d th se \\ k d th se & (th - kd)se \end{bmatrix} \quad (22)$$

where, for the sake of brevity the superscript l identifying the layer number has been omitted, we have used the notation $\text{th} = \tanh(kd)$, and $\text{se} = \operatorname{sech}(kd)$, and $D = f^2[(1 + k^2d^2)\text{se}^2 - 1]$.

If we assume that the i th interface (see Figure 2) is fully bonded then the jump in the stresses $\Delta p^i = p_b^{i+1} - p_t^i$ or the displacements $\Delta u^i = u_b^{i+1} - u_t^i$ are zero. If there are prescribed jump conditions they can be expressed in the following form:

$$p_b^{i+1} = p_t^i + \Delta p^i, \quad u_b^{i+1} = u_t^i + \Delta u^i \quad (23)$$

Using conditions (23) in (18) we obtain the following equations for the displacements either side of the i th interface that lies at the top of the i th layer:

$$\otimes : u_t^i = R_{tt}^i p_t^i + R_{tb}^i p_b^i = R_{tt}^i p_t^i + R_{tb}^i (p_t^{i-1} + \Delta p^{i-1}) \quad (24)$$

$$\odot : u_b^{i+1} = R_{bt}^{i+1} p_t^{i+1} + R_{bb}^{i+1} p_b^{i+1} = R_{bt}^{i+1} p_t^{i+1} + R_{bb}^{i+1} (p_t^i + \Delta p^i) \quad (25)$$

Subtracting (24) from (25) and using the jump condition for the displacements given in (23) we obtain

$$\Delta u^i = R_{bt}^{i+1} p_t^{i+1} + (R_{bb}^{i+1} - R_{tt}^i) p_t^i - R_{tb}^i p_t^{i-1} + R_{bb}^{i+1} \Delta p^i - R_{tb}^i \Delta p^{i-1} \quad (26)$$

Since the quantities Δu^i , Δp^i , and Δp^{i-1} are specified in a typical problem, separating the known from the unknown quantities, we can re-write (26) in the form of a set of vector recurrence relations or difference equations for the tractions p_t at the top of three successive layers (these points are marked by the \otimes symbol in Figure 2):

$$B^i p_t^{i+1} + C^i p_t^i + A^i p_t^{i-1} = \Delta^i \quad (27)$$

where $B^i = R_{bt}^{i+1}$, $C^i = (R_{bb}^{i+1} - R_{tt}^i)$, $A^i = -R_{tb}^i$, and $\Delta^i = \Delta u^i - R_{bb}^{i+1} \Delta p^i + R_{tb}^i \Delta p^{i-1}$. Once the traction vectors p_t^i at the tops of each of the layers have been determined, the traction vectors below each of the layers can be readily determined from (23). An efficient recursive procedure to solve the block tri-diagonal system (27) and be obtained by using a block *LU* decomposition (see Reference [13]). The flexibility relation (18) can be used to determine the displacements on either side of each of the interfaces. The desired spectral coefficients with reference to any given co-ordinate system can be determined using (12).

Because the flexibility matrix approach involves tanh and sech functions, which are bounded as $kd \rightarrow \infty$, the flexibility matrix method is well suited to treating problems for which the products kd are large. However, as $kd \rightarrow 0$ certain coefficients in the flexibility matrices R_{tt} and R_{bb} have an asymptotic behaviour which is $O((kd)^{-3})$. A refinement of the flexibility matrix method, which makes use of asymptotic rescaling to enable it to deal with both very small and very large wavenumbers, is given in Reference [5].

3.5. The high wavenumber limit—the uniform asymptotic solution

In the case of a layered material with multiple layers it is not possible to obtain a simple closed-form solution for the spectral coefficients and therefore for the stresses and displacements due to a DD for example. Indeed, for a problem that has three layers and two interfaces or more, the analytic solutions will involve an infinite series of image DDs that are required to impose the appropriate boundary conditions on the interfaces, which is cumbersome for numerical computations. Thus, for a more complex problem, one inevitably has to resort to a numerical solution of the system of algebraic equations for the spectral coefficients and then a numerical inversion. This numerical inversion process, as we will see later in this paper, is essentially equivalent to inverting a Hankel transform. The numerical inversion process is complicated by the presence of highly singular sources in the problem that increase the high-frequency content of the spectral coefficients. Indeed, the only feasible way to obtain any sort of numerical inversion in the presence of a concentrated DD, is to first subtract off the singular DD coefficients for a DD in an infinite medium and to then invert the remaining low frequency components using numerical integration (see, for example, Reference [7]). However, for problems in which the DD comes very close to an interface, which occurs when modelling a crack touching or intersecting an interface, the asymptotic solution in the limit $k \rightarrow \infty$ provided by a DD in an infinite medium no longer removes all the high frequency components that prove to be troublesome for numerical integration. If the DD is a small but finite distance from the interface, then *eventually* as $k \rightarrow \infty$, the spectral coefficients will tend to the infinite space values. However, if the source DD is really close to the interface, the actual spectral coefficients will differ significantly from the infinite space DD solution for wavenumbers up to $k = O(h^{-1})$, where h is the distance between the source DD and the interface. For this reason we make use of the analytic spectral coefficients for a vertical DD which is in one of two bonded half-planes that can have different material properties. This is

the appropriate asymptotic limit for a source DD that is situated close to an interface. It allows us to significantly reduce the range of wavenumbers that need to be calculated—in some cases by two orders of magnitude. But perhaps more importantly it substantially reduces the high-frequency content of the spatial influences that need to be integrated numerically in order to obtain an integrated Green's function—this becomes particularly important when trying to arrive at Green's functions for elements that come into contact with the interface between two materials.

For this purpose we make use of special solutions which are the exact spectral coefficients for vertical DDs in one of two bonded half-planes. However, the typical situation will involve a DD element that finds itself sandwiched between two interfaces. These interfaces separate the layer, in which the DD element falls, from the outside world. The ‘outside world’ might comprise two or more distinct layers. Thus, the next step in finding useful asymptotic solutions would be to determine the solution for a DD element that falls in the middle layer of three elastic layers, i.e. within a layer that is sandwiched between two bonded elastic half-planes. As mentioned earlier, the solution to this problem will involve an infinite series of image DDs that are distributed about an infinite periodic sequence of image interfaces that are located further and further away from the source as the terms of the series progress. These image DDs will result in features that alter the spectral coefficients that decay at a rate of e^{-2kH} , where H is the distance from the image to the original source DD. Thus, the effects of the remote DD will die very rapidly, leaving spectral coefficients whose behaviour is dominated by the source and the nearest images. In this section, we outline the construction of the uniformly valid leading order asymptotic approximation to the spectral coefficients for the three-layer problem. These asymptotic spectral coefficients are then used to eliminate the high wavenumber components from the spectral coefficients. Since only the remaining low wavenumber components need to be inverted, this procedure significantly reduces the computational cost of numerical inversion and integrated kernel evaluation. This uniform asymptotic solution makes it possible to efficiently determine the effect of a single DD element that touches two interfaces simultaneously.

Consider a three layer elastic medium comprising two half-planes that are bonded to a finite strip with the moduli and spectral coefficients defined as shown in Figure 3. If the vertical DD falls in the region where $h_L \ll h_U$, the three-layer solution will tend for large k values to the solution for a vertical DD in the upper part of two bonded half-planes, while if the DD falls in the region where $h_L \gg h_U$, the three-layer solution will tend for large k values to the solution for a vertical DD in the lower part of two bonded half-planes. If on the other hand, the DD is not much closer to one interface than the other so that $h_L \sim h_U$, then as $k \rightarrow \infty$ the ultimate asymptotic solution is the solution for a vertical DD in an infinite medium to which both the upper and lower solutions tend. Thus, we have a typical situation that occurs in asymptotic analysis (see, for example, Reference [15]), in which two different asymptotic solutions are valid in different regions but they are both valid in a finite overlap region that they both share. In this case, it is possible to obtain an asymptotic approximation that is uniformly valid over the three regions by superimposing the two asymptotic solutions and subtracting the solution in the match region

$$A_j^{l,\mu}(k) \xrightarrow{k \rightarrow \infty} A_j^{l,U}(k) + A_j^{l,L}(k) - A_j^{l,\infty}(k) \quad (28)$$

where $A_j^{l,\mu}(k)$ is used to represent the uniformly valid asymptotic solution, $A_j^{l,U}(k)$ represents the corresponding bonded half-plane spectral coefficient in which the interface is above the

source DD, $A_j^{l,L}(k)$ represents the corresponding bonded half-plane spectral coefficient in which the interface is below the source DD, and $A_j^{l,\infty}(k)$ represents the spectral coefficient for a point vertical DD in an infinite medium with material properties λ_2 and G_2 for the middle layer. Complete details of this uniform asymptotic solution are given in Reference [4].

3.6. Inversion of the FT

In this section, we discuss the process of inversion of the FTs of the influences. In order to make it possible to use the spectral method to determine the spatial influences of the singular case in which a crack intersects the interface between two layers, it is necessary to first subtract off the uniform asymptotic spectral coefficients $A_j^{l,\mu}(k)$ (discussed in the previous sub-section) for a vertical DD in the middle layer of a three-layer elastic medium from the numerical spectral coefficients $A_j^l(k)$ determined using the techniques described in Section 3.4, i.e.

$$A_j^{l,\mathcal{L}}(k) = A_j^l(k) - A_j^{l,\mu}(k) \quad (29)$$

Since $A_j^l(k) \xrightarrow{k \rightarrow \infty} A_j^{l,\mu}(k)$, it follows that $A_j^{l,\mathcal{L}}(k) \xrightarrow{k \rightarrow \infty} 0$. If the uniform asymptotic approximation closely mimics the true solution, then $A_j^{l,\mathcal{L}}(k)$ will only be non-zero for relatively low-frequencies. Thus after peeling off the asymptotic solution, the remaining spectral coefficients $A_j^{l,\mathcal{L}}(k)$ that need to be inverted contain only relatively low-frequency components. As a result, it is possible to invert the low-frequency spectra $A_j^{l,\mathcal{L}}(k)$ very efficiently using numerical integration. The high-frequency components, that are associated with the uniform asymptotic solution $A_j^{l,\mu}(k)$, are numerically intractable. However, these uniform spectral coefficients can be inverted analytically to yield approximate spatial stress and displacement components due to a point vertical DD in a three-layer material (see Reference [4]).

The point kernels must be multiplied by the appropriate basis functions to obtain the desired elemental DD influences. In this paper, we shall only consider the case of piecewise constant basis functions, but any of the standard DD discretizations can be obtained for multi-layered media using the spectral method by implementing the appropriate choice of basis functions. In order to complete the process, the low-frequency components also need to be integrated against the appropriate basis functions. Since we do not have an explicit expression for the low-frequency spectral components that have been inverted numerically, it is not possible to obtain an analytic expression for the integrated low-frequency components. The spatial functions associated with these low-frequency spectra are smooth functions since they only involve low-frequency Fourier modes all of which are smoothly varying functions in space. It is therefore possible to integrate the contribution of these low-frequency spectral components very accurately with a low-order Gauss integration scheme in order to determine a set of integrated influence coefficients.

Finally, in order to obtain the stress components in the l th layer of a multilayered elastic medium due to a vertical DD in one of the layers, we superimpose the low-frequency stress components $\sigma_{ij}^{l,\mathcal{L}}$ that are associated with the spectral coefficients $A_j^{l,\mathcal{L}}(k)$, and the singular stress components $\sigma_{ij}^{l,\mu}$ due to the uniform asymptotic spectral coefficients $A_j^{l,\mu}(k)$, i.e.

$$\sigma_{ij}^l = \sigma_{ij}^{l,\mathcal{L}} + \sigma_{ij}^{l,\mu}$$

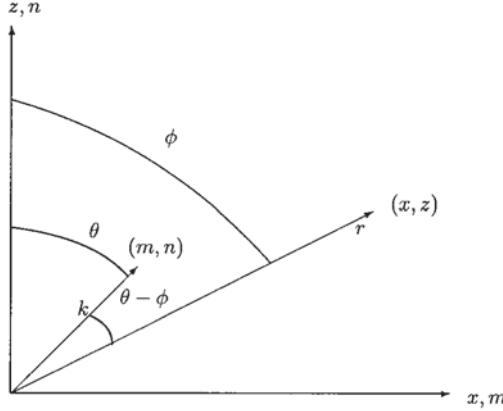


Figure 4. Polar co-ordinates for the FT inversion.

A similar procedure can be followed to obtain the displacement components due to a vertical DD in a multi-layered elastic medium, i.e.

$$u_i^l = u_i^{l,\mathcal{L}} + u_i^{l,\mu}$$

3.6.1. Reduction to Hankel transforms. Applying the inverse FT defined in (7) to $\hat{\sigma}_{zz}$ we obtain

$$\sigma_{zz}(x, z) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-i(mx+nz)} \hat{\sigma}_{zz}(m, n) dm dn \quad (30)$$

We now express the integration variables m and n in (30) in terms of polar co-ordinates (k, θ) in which $m = k \sin \theta$ and $n = k \cos \theta$. We also express x and z in terms of the polar co-ordinates (r, ϕ) so that $x = r \sin \phi$ and $z = r \cos \phi$ (see Figure 4).

In terms of these new variables

$$mx + nz = kr \cos(\theta - \phi)$$

so that (30) can be written in the form

$$\sigma_{zz}(r \sin \phi, r \cos \phi) = \frac{1}{(2\pi)^2} \int_0^{\infty} \int_{-\pi}^{\pi} e^{-ikr \cos(\theta - \phi)} \hat{\sigma}_{zz}(k \sin \theta, k \cos \theta) d\theta k dk \quad (31)$$

By introducing the notation

$$K_{pq}(k; r, \phi) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-ikr \cos(\theta - \phi)} \cos^p \theta \sin^q \theta d\theta \quad (32)$$

we can write the terms in $\hat{\sigma}_{zz}(k \sin \theta, k \cos \theta)$ as a linear combination of integrals of the following form:

$$I_{spq} = \frac{1}{2\pi} \int_0^{\infty} e^{-ky} A_j(k) k^s K_{pq} dk \quad (33)$$

Making use of the identity (see Reference [16])

$$J_n(\rho) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-in\theta} e^{i\rho \sin \theta} d\theta \quad (34)$$

where J_n is the n th-order Bessel function, it is possible to represent the integrals K_{pq} in the form

$$K_{pq} = \sum_{j=0}^{p+q} \left(\sum_{i=0}^j \alpha_i \left(\frac{z}{r} \right)^{2i} \right) J_{2j}(kr) \quad (35)$$

Thus by making use of (31)–(35) it is possible to invert $\hat{\sigma}_{zz}$ by evaluating Hankel transforms of the form

$$H_n(r, y) = \int_0^\infty h(k; r, y) J_n(kr) dk$$

for which there exist efficient algorithms (see, for example, Reference [17]). The process is made even more efficient by the fact that only the low frequency components $\hat{\sigma}_{zz}^{l,\mathcal{L}}$ need to be inverted, because all the high-frequency components have been accounted for by means of the uniform asymptotic solution for a three-layer material.

4. ASSEMBLING THE INFLUENCE MATRIX

The procedure described above can be used to determine the influence coefficients of any shape of DD element that we may use to tessellate a planar fracture as well as any choice of shape function. However, the algorithm we propose makes use of a rectangular Eulerian grid that is fixed in space. In addition, we assume a constant DD variation over each element. The rectangular element shape allows us to employ analytical integrations (see Section 3), which are considerably more efficient than numerical integrations.

Typically the set of all possible influence coefficients are pre-calculated before the fracture propagation process begins. We observe that due to the horizontal translational invariance of the problem, only the influence of a single sending element for each horizontal row of elements on all the possible elements in the mesh needs to be stored. This is depicted schematically in Figure 5, where the influence between a source element and a receiver defined by either dashed arrow is identical to that defined by the solid arrow. These stress influences are determined by adding the integrated uniform stress components $\sigma_{zz}^{l,\mu}$ and the integrated low-frequency stress components $\sigma_{zz}^{l,\mathcal{L}}$ at receiving points located at the centres of the receiving elements. Exploiting this invariance property leads to a substantial reduction in the memory requirements as well as the computational cost of calculating the influence coefficients.

As the fracture evolves through the pre-defined rectangular grid, more elements are activated and their influences are assembled into an influence matrix which defines the elasticity matrix for the current growth step. Assembling all possible send-receive pairs of influences and storing them in a matrix C , we obtain the following discrete form

$$\sum_{j=1}^J C_{ij} w_j = p_i \quad (36)$$

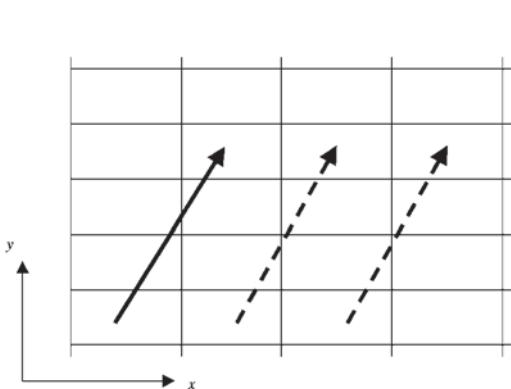


Figure 5. Shift property of layer-parallel elasticity matrix influences.

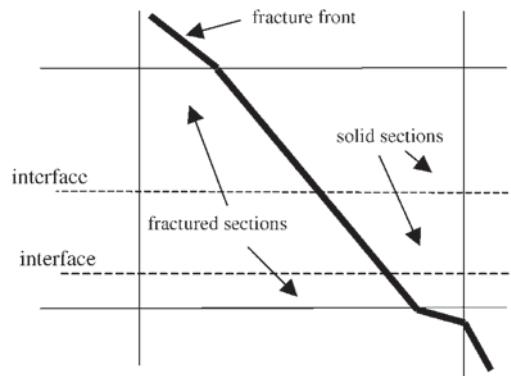


Figure 6. Partially fractured element schematic, showing three sub-elements on a parent tip element.

of the crack integral equation (1), where J is the total number of active elements, w_j is the unknown width (DD) value within the crack at the centre of the j th element, and p_i is the prescribed pressure within the crack at the centre of the i th element.

A regular mesh has an advantage over a moving mesh in that ‘book-keeping’ becomes much easier to administer. In addition, interpolation errors that arise in the moving mesh case due to re-meshing are avoided. Such errors can accumulate with each growth step. The fixed mesh approach allows us to pre-calculate, for a given layered problem, all elastic influences. We can then extract in an ‘on-the-fly’ fashion the influences from the parent set that are required for the current fracture extent at the current growth step.

Interfaces will, in general, not align with the element edges of the regular mesh. We thus allow any row of parent elements to be sub-divided into rectangular sub-elements (to within a specified tolerance) so that interfaces exactly match sub-element edges. This is depicted in Figure 6, where a tip element is cut by two interfaces into three sub-elements. The elastic solution is obtained using the sub-element mesh, whereas tracking of the fracture front (see Section 6) is performed on the regular parent mesh. This sub-division of elements ensures that we obtain an accurate elastic response.

A major potential drawback of a regular mesh approach is that the fracture tip is not captured accurately. We employ special ‘partially fractured’ elements along the fracture tip to ensure that an accurate width profile is obtained. The self-effect influence coefficients, C_{ii} , of each tip element i contain special correction factors that ensure that an accurate solution is obtained using the piecewise constant elements. This is summarized in References [18, 19]. We have extended this work, based on square elements, to rectangular elements. The correction factors ensure that the width solution, obtained from the constant DD elements, is comparable to linear elements without incurring any extra computational overhead.

5. SOLUTION OF ELASTICITY EQUATIONS

The elasticity equations are most efficiently solved by means of an iterative solver. We use an L1D iterative solver [20] that is efficient for this type of problem involving full matrices

with diagonal dominance. For example, on a 200MHz PC, a L1D scheme will solve a 64×64 or 4096 element system in 152 s, using 15 iterations. A direct solver requires 139 min of CPU time (55 times slower).

As noted in Section 4, the algorithm has been structured to make use of the translational invariance property so that only the essential influence coefficients are stored, allowing large problems to be solved very efficiently. A 4096 element system would normally require 134 MB of storage just for the influence matrix (assuming that double precision is used). Our algorithm requires only about 2 MB of storage for the same sized problem.

6. FRACTURE FRONT-TRACKING PROCEDURE

The logic required to keep track of the fracture front location is complicated. Special care must be taken to ensure that ‘close shave’ situations, that arise when the front is in close proximity to an element corner and/or edge, are handled correctly; that large or small fracture growth jumps relative to the element size are handled correctly; that merging fronts (from the same fracture or from separate fractures) are handled correctly; and that front segment connectivity is tracked correctly. However, because the major part of the front-tracking logic only needs to track along the tip elements, this process is efficient. A typical CPU update time for a 50×50 parent mesh is around 1–5 s on a 200 MHz PC, depending on the complexity of the fracture front. A 100×100 parent mesh typically requires about 15 s of CPU time per crack growth update. Most of the CPU time for larger problems is used to check that a closed contour has been obtained for each fracture front, and is not expended in the front-tracking process. Unfortunately, every single element in the parent mesh is currently used to perform this check, which uses the property that the sum of the angles from an interior reference point (i.e. element centre) to all the vertices of a polygon (i.e. fracture front) equals 2π .

The fracture geometry is stored in the form of a vector, containing numbers with values 0, 1, 2, 3, 4, 5, or 6, depending on the local geometry of the fracture front. Recall that all elements are rectangular in shape, so a mesh can be set up as shown in Figure 7, where ‘0’ indicates a solid (unfractured) element, and all other numbers indicate fractured elements. In order to define the fracture surface in Figure 7, ‘2’ indicates a straight edge, ‘3’ indicates an obtuse-angled corner, ‘4’ indicates an acute-angled corner, the combination ‘5’ and ‘6’ indicates a ‘peninsula’-type configuration, and ‘1’ indicates a fractured interior element.

The fracture front is automatically assigned on an element by element basis, as shown in Figure 8. In this figure, the mined element corners are indicated by the ‘+’ symbols, and the fracture front segments are the solid lines connected by small squares and diamonds at all element cut points. An element cut point is simply the location where the fracture front segment cuts the current element—there are always two cut points assigned per tip element.

In Figure 9, we schematically illustrate the front advance procedure in the front-tracking algorithm. The dashed arrows are the growth vectors from the previous growth step. In order to represent the advanced fracture front on the rectangular mesh, the arrowheads are joined together by straight dashed lines. The locations where the dashed lines cut element edges are termed cut points. All cut points are then joined together to form new fracture front segments. The new growth vectors, indicated by the solid arrows, are assumed to originate from the centre of each front segment (one per element). The logic will only allow one growth vector per element, one growth segment per element, and two cut points per element. Notice that

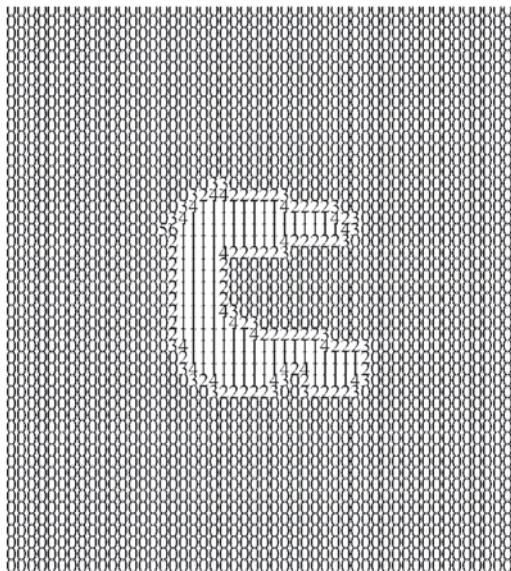


Figure 7. Initial fracture pattern in ‘binary’ form as used by simulator.

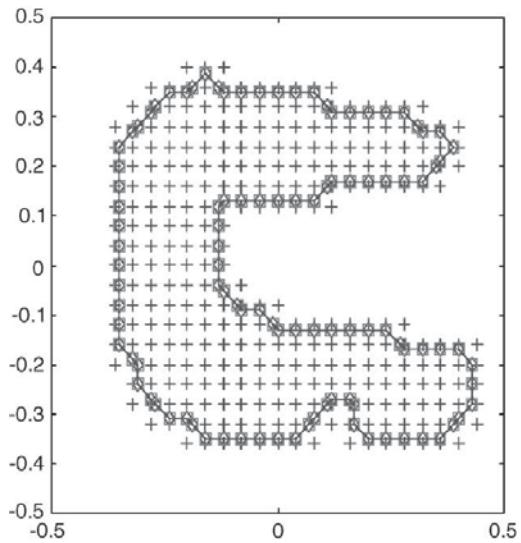


Figure 8. Initial fracture front pattern, showing cut points, front segments, and fractured elements.

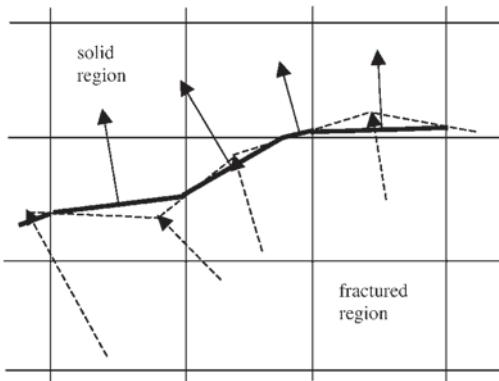


Figure 9. Front-tracking control layout.

this type of growth logic introduces a natural smoothing of the fracture front because of the combination of dashed pseudo and solid actual fronts that are generated as part of the process.

Whenever the fracture front is concave, multiple growth vectors may enter a new element. If this situation occurs, we delete all but one of the multiple growth vectors. Whenever the fracture front is convex, or if large growth increments are allowed, we can generate situations where potential new tip elements do not contain growth vectors. In these cases, extra front segments are added as needed so that each tip element will contain a front segment, so that the front is unbroken.

Whenever high curvature exists along the front, situations may arise where only one side of a potential tip element is cut by the dashed lines. Since this situation would imply that we are trying to resolve features at the sub-mesh level, these ‘peninsula’-type situations are not allowed to occur, and the associated growth vectors are deleted. The neighbours of each deleted growth vector are joined up instead to form a smoother front. Deletion of these ‘peninsula’-type situations is consistent with the mesh resolution. The front will automatically adjust during the next growth increment, in accordance with the fracture growth calculations, should a peninsula that has a resolution of more than one element in size persist.

At all times, connectivity must be maintained along the front. All front segment nodes are made aware of their neighbours via a connectivity array. The algorithm performs automatic internal checks to ensure that connectivity is always maintained. This type of front control is quite general. It is possible to grow any number of separate multiple fronts. In addition, there is logic that allows different parts of the same front or separate fronts to merge once they intersect each other. The key to controlling such a system is local autonomy. Each front segment only cares about its neighbor on each side, and is also aware of any potential clash with other approaching fronts.

7. FRACTURE GROWTH CONTROL

The mechanism of fracture growth depends on the characteristics of the material in which the fracture is being propagated. In the case of hydraulic fracture propagation, the host rock may offer little or no resistance to the fracture propagation process. In such cases, fracture growth is dominated by the growth of the expanding fluid front, and the host rock can be assumed to have zero toughness. In this paper, we do not treat this situation. At the other extreme, the fracture toughness is large enough so that it dominates the fracture growth process and viscous effects are negligible. We could also consider the case where a combination of rock toughness and viscous behaviour determines the rate of fracture growth.

In the absence of fluid flow, the fracture growth criterion is simple to encode. Our algorithm is general enough to incorporate any fracture growth rule. We use the approach of Mastrojannis *et al.* [21], where the normal growth increment for front propagation of a planar fracture is dictated by the local stress intensity strength as

$$\Delta L = \Delta L_{\max} \left(\frac{K_I - K_{Ic}}{K_I^{\max} - K_{Ic}} \right)^\alpha \quad \text{for } K_I > K_{Ic}$$

$$\Delta L = 0 \quad \text{for } K_I < K_{Ic}$$
(37)

where, according to Reference [21], α is a material-dependent constant, K_I^{\max} is the maximum stress intensity computed before fracture growth along the front and K_{Ic} is the rock toughness. In this work, we choose $\alpha = 1$ (we view α as a numerical parameter), and assume ΔL_{\max} to be a user-defined constant. In the absence of fluid flow coupling, the growth rate is unimportant. Typically, we choose a maximum growth increment ΔL_{\max} along the front of the order of the element size. The simulator can also handle growth increments larger or smaller than the element size.

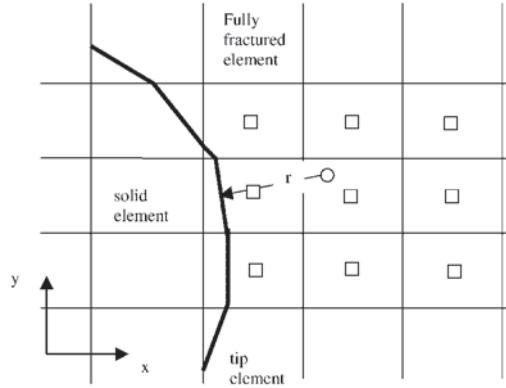


Figure 10. Nine-noded patch used to sample width for stress intensity calculation.

The stress intensity K_I is calculated at a finite number of points along the fracture front. This is accomplished using a local nine-noded patch of fracture widths, as depicted in Figure 10. The elements containing small squares constitute the current nine-noded patch. The arrowhead indicates the centre of the current fracture front segment, where we wish to calculate a stress intensity factor. The arrow base, indicated by the small circle, indicates the position at which an averaged width value is determined, based on bi-quadratic interpolation from the surrounding nine elements in the patch. The length of the arrow is the distance r behind the fracture front at which the width value is computed, according to the standard plane strain formula

$$K_I = \lim_{r \rightarrow 0} \frac{Gw}{2(1-v)} \sqrt{\frac{\pi}{2r}} \quad (38)$$

where w is the width, G the local shear modulus and v the Poisson's ratio. In the application of (38), we sample the width at two points, r_0 and r_1 to obtain discrete widths w_0 and w_1 , along the same normal arm from the current fracture tip location, and apply a two-term asymptotic expansion of form

$$w \stackrel{r \rightarrow 0}{=} c_0 r^{1/2} + c_1 r^{3/2} + O(r^{5/2}) \quad (39)$$

The leading term

$$c_0 = \left(\frac{w_0 r_1^{3/2} - w_1 r_0^{3/2}}{r_0^{1/2} r_1^{3/2} - r_1^{1/2} r_0^{3/2}} \right)$$

is used in (38) to obtain an accurate value of stress intensity given by

$$K_I = \frac{Gc_0}{2(1-v)} \sqrt{\frac{\pi}{2}} \quad (40)$$

The distances r_0 and r_1 are user-defined, but a value of $r_0 = z\Delta x$, and $r_1 = r_0 + \Delta x$ where Δx is the element length and $z = 2-5$, generally yields accurate results. In the case of rectangular elements of size Δx by Δy , we allow r_0 to vary with direction as

$$r_0 = z(\Delta x \cos^2 \beta + \Delta y \sin^2 \beta)$$

where β is the angle between the local search radius r_0 and the x -axis.

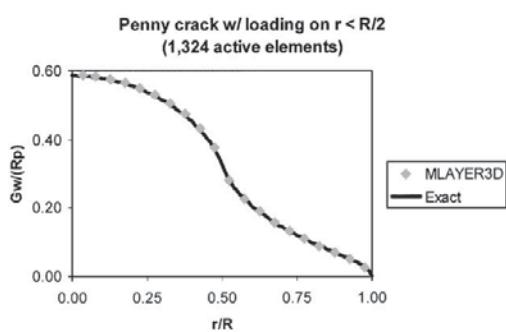


Figure 11. Width profile along radius of partially loaded penny crack.

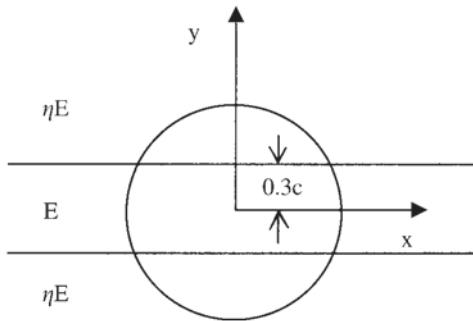


Figure 12. Geometry of penny crack spanning three elastic layers.

8. VERIFICATION OF ALGORITHM

8.1. Partially loaded penny crack

In order to test the accuracy of the basic algorithm and the ability of the mesh to handle a curved boundary, we model a partially loaded penny crack which has a known analytical solution. Figure 11 shows the width profile along the radius of the partially loaded penny crack of radius R , obtained using the algorithm. In this problem, the central section is loaded over radius $0 < r < a$, where $a = R/2$, by a constant pressure p . The analytical width solution is given by [22] as

$$w(r) = \frac{4(1-v)Rp}{\pi G} \left(\sqrt{1 - (r/R)^2} + \frac{a}{R} E(\varphi_1, r/a) - \sqrt{\frac{R^2 - a^2}{R^2 - r^2}} \right) \quad \text{if } r \leq a$$

$$w(r) = \frac{4(1-v)Rp}{\pi G} \left(\sqrt{1 - (r/R)^2} + \frac{r}{R} E(\varphi_2, a/r) - \sqrt{\frac{R^2 - r^2}{R^2 - a^2}} - \frac{r^2 - a^2}{rR} F(\varphi_2, a/r) \right) \quad \text{if } r > a \quad (41)$$

where G is the shear modulus, v is Poisson's ratio, $F(\bullet, \bullet)$ and $E(\bullet, \bullet)$ are the incomplete elliptic integrals [23] of the first and second kind, respectively, and

$$\varphi_1 = \arcsin \sqrt{\frac{R^2 - a^2}{R^2 - r^2}}$$

$$\varphi_2 = \arcsin \sqrt{\frac{R^2 - r^2}{R^2 - a^2}}$$

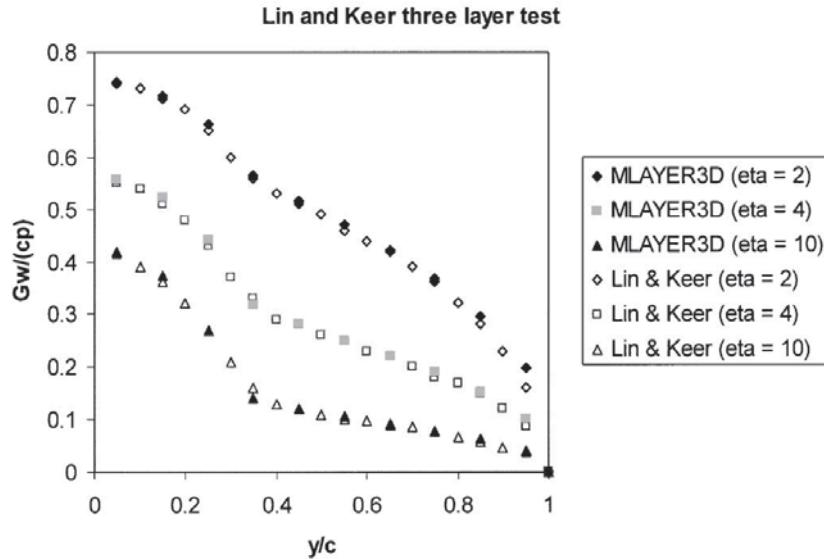


Figure 13. Normalized width vs radial distance for three layer case.

The penny crack was modelled using 1324 elements, with all tip elements partially fractured so as to match the penny crack perimeter in a piecewise linear fashion. The total CPU time was 18 s on a 200 MHz Pentium PC, using a 48×48 parent mesh encapsulating the penny crack. The numerical results (MLAYER3D) are clearly very accurate.

8.2. Fully loaded penny crack in three-layer material

A penny crack of radius c straddling three layers is shown in Figure 12. The outer layers are stiffer than the inner layer by a factor η , and the penny crack is loaded under constant pressure p . Our width results along the y -axis compare very well with those of Lin and Keer [24] (obtained from their paper by digitization), as shown in Figure 13. Each problem was solved in 41 s using a 200 MHz Pentium PC, based on a 30×30 parent mesh encapsulating the crack.

8.3. Stress intensity calculations along front perimeter

Kuo and Keer [25] have published results for stress intensity factors along the perimeter of a penny crack located orthogonal to the interface of a two-material composite, as shown in Figure 14. The penny crack is loaded by a constant pressure, p , and the elastic properties are defined by a constant Poisson's ratio of $\nu_1 = \nu_2 = 0.3$ and a ratio between the shear moduli, given by $\Gamma = G_2/G_1$. Figure 15 shows a comparison between our results and those obtained by Kuo and Keer (extracted from their paper by digitization) for various choices of Γ . Their results match ours except for a small constant vertical shift. However, we know that the exact stress intensity factor for the uniform material case ($\Gamma = 1$) is given by [25]

$$\frac{K_I}{p\sqrt{\pi c}} = \frac{2}{\pi} \simeq 0.6366 \quad (42)$$

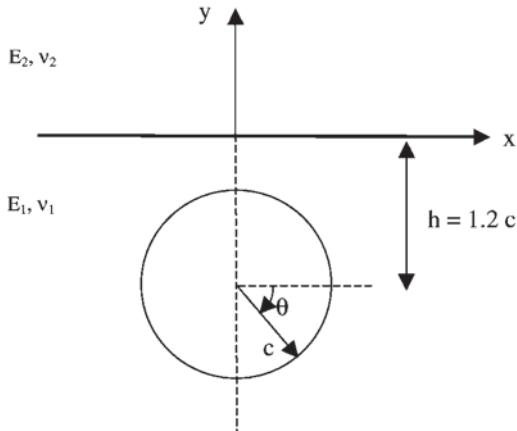


Figure 14. Geometry of a penny crack located near the interface of a two-material composite.

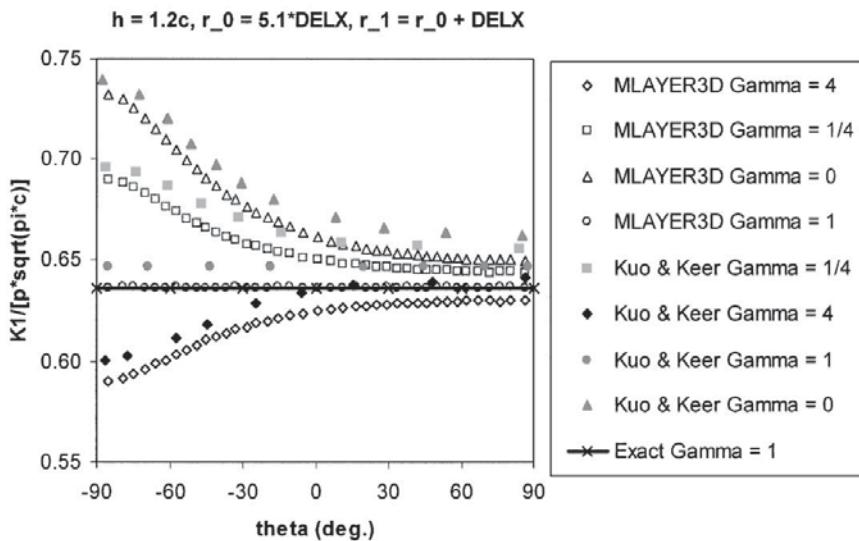


Figure 15. Stress intensity factors around the perimeter of a penny crack located near the interface of a two-material composite.

This value of 0.6366 matches our result almost exactly, where we have employed a parent mesh of 96×96 elements.

The effect of reducing the search radius in the asymptotic stress intensity computation is shown in Figure 16. Here, we have used a 48×48 parent mesh to encapsulate the penny crack in a homogeneous material (i.e. $\Gamma = 1$). Scatter in the numerical results increases as the search radius is decreased because the nine-noded patch starts to include collocation points outside the crack. The use of an interpolation scheme which discounts the spurious contributions from outside the crack would reduce these errors considerably. In addition, the tip elements

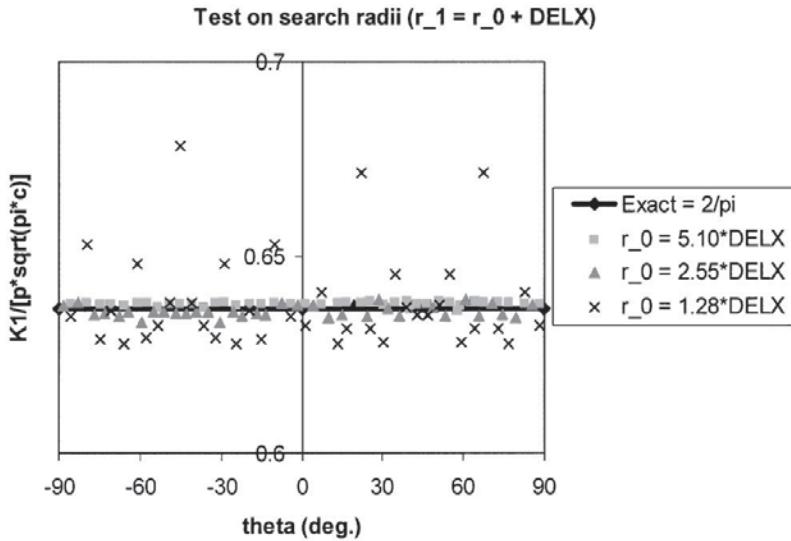


Figure 16. Effect of reducing search radius on accuracy of asymptotic stress intensity computation.

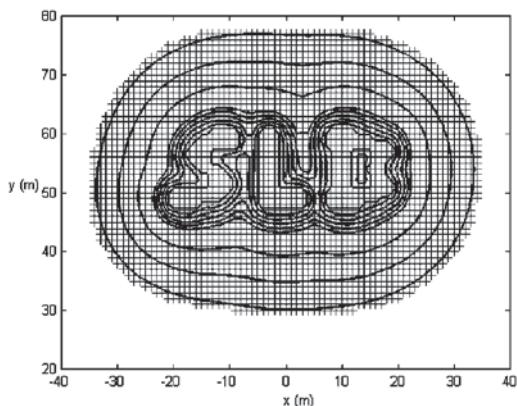


Figure 17. Fracture fronts obtained when three separate fractures combine into a single one.

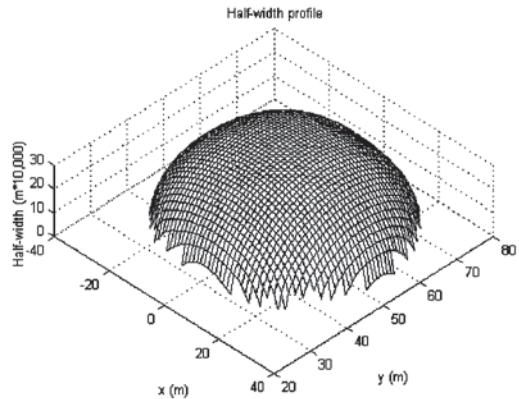


Figure 18. 3D surface plot of half-width on fracture surface at 20th growth step.

do exhibit larger errors than the interior ones, which contributes to the errors in the stress intensity calculation determined by the smaller search radii.

9. GENERAL APPLICATION OF MULTI-LAYER SCHEME

9.1. Multiple interacting fracture growth under uniform conditions

As an effective demonstration of the front-tracking logic, we show in Figure 17 the fracture pattern at various growth steps, based on three starting fractures in a homogeneous material

Table I. Multi-layer case properties.

Layer j	Layer thick, d_j (m)	y_j top of layer j	E (GPa)	ν	σ_v (MPa)	K_{lc} (MPa)
10	1000	1161	20	0.3	0.1	0.2
9	100	161	10	0.2	0.2	0.1
8	20	61	50	0.1	0.5	0.5
7	3	41	20	0.4	0.4	0.2
6	2	38	10	0.2	0.2	0.1
5	20	36	40	0.1	0.1	0.4
4	5	16	30	0.3	0.1	0.3
3	1	11	20	0.2	0.1	0.2
2	10	10	70	0.1	0.1	0.7
1	1000	0	30	0.1	0.1	0.3

($E = 10$ GPa, $\nu = 0.1$). Each fracture is kept at the same constant pressure of $p = 1$ MPa throughout, and the element size is 1 m squared. Zero toughness and zero confining stress are assumed to act on the fracture. This example shows a number of features of the algorithm. First, it demonstrates the ability of the algorithm to cope with multiple convex- and concave-shaped fractures. Second, it demonstrates the ability of the front-tracking logic to cope with two separate fronts merging, as well as two sections of the same front merging. Third, it demonstrates the ability of the logic to cope with islands that vanish after a number of growth steps. Fourth, it demonstrates that the fracture grows into a more circular front shape at later growth steps, which is the expected final configuration based on energy conservation principles. Figure 17 shows the fracture fronts superimposed on each other from growth steps 0–5, 10, 15 and 20 (where the 0th step is the starter geometry). Figure 18 is a 3D surface plot of fracture half-width over the entire fracture surface at the 20th growth step. The total CPU time for all 20 growth steps was 793 s on a 200 MHz Pentium PC, using a 100×100 parent mesh.

9.2. Multi-layer multiple interacting fracture growth

In this sub-section, we repeat the same experiment of Section 9.1, but assume that a planar fracture propagates in a *layered* material. Table I summarizes layer properties for the same three starter fractures as used in Section 9.1. Higher layer numbers are located at shallower depths. Layer interfaces are denoted by variable y_j , thicknesses by d_j , confining stresses by σ_v and toughnesses by K_{lc} . Our parent mesh extends over the vertical range $0 < y < 100$ m. The thick top layer (see Table I) is added to avoid free surface effects (which the simulator automatically models) at the top of the stack of layers.

Figure 19 shows the front locations at growth steps 0–5, 10, 15, 20, as before. The influence of the different elastic layers is clearly evident on the growth pattern. Figure 20 shows a 3D surface plot of the fracture half-width at the 20th growth step. Again, the effect of modulus variation is clearly evident in the result. The three starter fractures are located in the high modulus/toughness region $49 \text{ m} < y < 60 \text{ m}$. Downward vertical growth is restricted at $y = 36 \text{ m}$ by layer $j = 5$ which has higher toughness. Upward vertical growth is initially restricted into the low modulus layer $j = 9$ because the drop in elastic modulus produces a drop in stress intensity as the fracture moves across the interface. This layer also has lower

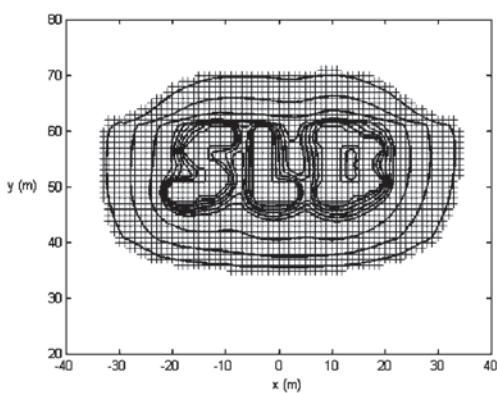


Figure 19. Fracture fronts obtained when three separate fractures combine into a single one in a layered material.

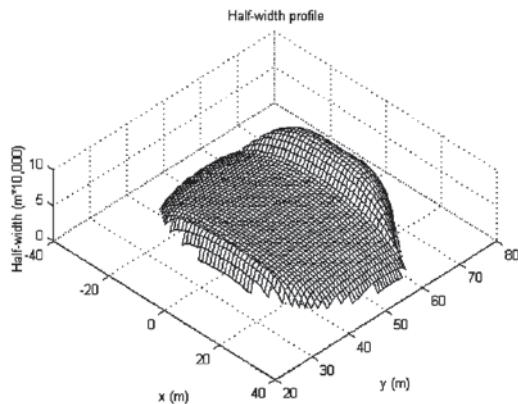


Figure 20. 3D surface plot of half-width on fracture surface at 20th growth step.

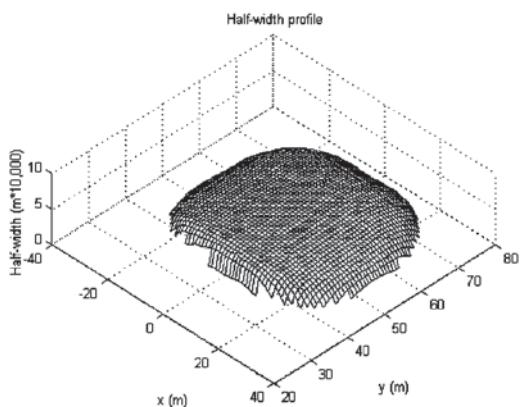


Figure 21. 3D surface plot of half-width on fracture surface at 20th growth step for $E(j=9)=50$ GPa.

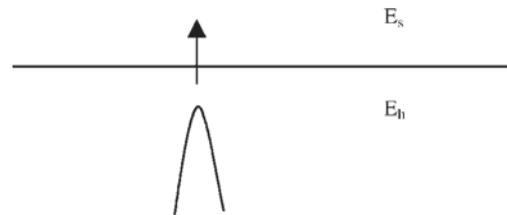


Figure 22. Fracture tip approaching an interface.

toughness, so we might expect more front advance into the lower toughness region, but there is a delicate interaction between toughness and modulus. Layer $j=9$ exhibits greater fracture width due to its lower modulus. The overall extent of fracture growth is less in the multi-layered case, because toughness reduces the growth potential as compared to the control problem of Section 9.1. Figure 21 shows the fracture half-width for the case where all data remains the same, except that we increase the Young's modulus in layer $j=9$ to 50 GPa. Fracture growth into this layer is now promoted, but the maximum width in this layer is reduced. The total CPU time for all 20 growth steps, including generation of the multilayer influence coefficients, was 1021 s on a 200 MHz Pentium PC, using a 100×100 parent mesh.

Whether or not fracture growth is favoured in the low or high modulus/toughness regions can be explained by examination of (38). Assume that a fracture approaches an interface

that separates two layers with higher (E_h) and lower (E_s) Young's moduli, as shown in Figure 22. We assume, for argument's sake, that the materials have the same toughness. As the fracture approaches the interface, its width profile near the tip remains approximately unchanged, because it is partly governed by the bulk of the fracture. When the tip is located in the material with higher modulus it will thus experience higher stress intensity due to (38), and will thus fracture more easily. As the fracture crosses into the lower modulus region, fracture growth becomes more restricted because stress intensity has dropped (for the same approximate fracture width). Once the fracture enters the lower modulus material, it will, however, start to experience more fracture width, as shown in Figure 20.

Note that our model ignores all effects related to potential branching, etc., of the fracture front as it approaches and intersects a material interface. Such effects can be significant (e.g. Reference [26]), but are not dealt with in our current simplified model.

10. CONCLUSIONS

We have described a planar 3D fracture growth algorithm, capable of modelling the growth of a planar fracture in an elastic multi-layer material. The fracture growth algorithm uses a DD approach, with numerical Green's functions for layered materials generated using a FT method. The numerical scheme is based on a fixed mesh consisting of rectangular elements. A front-tracking algorithm is used to trace the fracture tip in a piecewise linear fashion. Special tip elements are employed that allow an element to be partially fractured. This ensures that accurate widths and stresses are obtained even though piecewise constant rectangular elements are used in the discretization process. Verification and general examples were presented that demonstrate that the numerical scheme is both accurate and efficient.

ACKNOWLEDGEMENTS

The authors thank Schlumberger for permission to publish this work. In addition, the authors thank John Napier of CSIR, and Steven Crouch of the University of Minnesota for their invaluable technical advice, and help in providing benchmark data from other DD schemes. Finally, special thanks to John Napier and John Ryder (latter retired from Chamber of Mines Research Organization) for their discussions on partially fractured DD tip elements.

REFERENCES

1. *WinGOHFER simulator, Version 6.0*. STIM-LAB Inc.: Duncan, OK and Marathon Oil Co.: Denver, CO, 1998.
2. *TerraFrac simulator, Version 6.0*. TerraTek Inc.: Salt Lake City, UT, 1998.
3. Wang J-J, Clifton RJ. Numerical modelling of hydraulic fracturing in layered formations with multiple elastic moduli. In *Rock Mechanics Contributions and Challenges*, Hustrulid W, Johnson GA. (eds). Balkema: Rotterdam, 1990; 303–309.
4. Peirce AP, Siebrits E. Uniform asymptotic solutions for accurate modelling of cracks in layered elastic media. *International Journal of Fracture* 2001, to appear.
5. Peirce AP, Siebrits E. The scaled flexibility matrix method for the efficient solution of boundary value problems in 2D and 3D layered elastic media. *Computer Methods in Applied Mechanics and Engineering* 2001, to appear.
6. Crouch SL, Starfield AM. *Boundary Element Methods in Solid Mechanics*. Unwin Hyman: London, 1990.
7. Wardle LJ. Stress analysis of multilayered anisotropic elastic systems subject to rectangular loads. *Technical Paper no. 33*, Division of Applied Geomechanics, CSIRO Institute of Earth Resources, Australia, 1980.
8. Singh SJ. Static deformation of a multilayered half-space by internal sources. *Journal of Geophysical Research* 1970; **75**:3257–3263.

9. Kennett BLN. *Seismic Wave Propagation in Stratified Media*. Cambridge University Press: Cambridge, 1983.
10. Buffler H. Der Spannungszustand in einem geschichteten Körper bei axialsymmetrischer Belastung. *Ingenieur-Archiv* 1961; **30**(6):417–430.
11. Buffler H. Die Bestimmung des Spannungs und Verschiebungszustandes eines geschichteten Körpers mit Hilfe von Übertragungsmatrizen. *Ingenieur-Archiv* 1962; **31**(1):229–240.
12. Buffler H. Theory of elasticity of a multilayered medium. *Journal of Elasticity* 1971; **1**(2):125–143.
13. Linkov A, Filippov N. Difference equations approach to the analysis of layered systems. *Meccanica* 1991; **26**:195–209.
14. Maier G, Novati G. On boundary element-transfer matrix analysis of layered elastic systems. *Proceedings of the 7th International Conference on Boundary Elements in Engineering*, Como, Italy, 1985; 1–28.
15. Bender CM, Orszag SA. *Advanced Mathematical Methods for Scientists and Engineers*. McGraw-Hill: New York, 1978.
16. Olver FWJ. *Asymptotics and Special Functions*. Academic Press: New York, 1974.
17. Chave AD. Numerical integration of related Hankel transforms by quadrature and continued fraction expansion. *Geophysics* 1983; **48**(12):1671–1686.
18. Ryder JA, Napier JAL. Error analysis and design of a large-scale tabular mining stress analyzer. *Proceedings of the 5th International Conference on Numerical Methods in Geomechanics*, Nagoya, 1985; 1549–1555.
19. Napier JAL, Stephansen SJ. Analysis of deep-level mine design problems using the MINSIM-D boundary element program. *Proceedings of the 20th International Symposium on the Application of Computers and Mathematics in the Mineral Industries*, vol. 1, Mining Journal SAIMM: Johannesburg, 1987; 3–19.
20. Ryder JA. Optimal iteration schemes suitable for general non-linear boundary element modelling applications. In *Proceedings of the 7th International Conference on Computer Methods and Advances in Geomechanics*, Cairns, Beer G, Booker JR, Carter JP (eds). Balkema: Rotterdam, 1991; 1079–1084.
21. Mastrojannis EN, Keer LM, Mura T. Growth of planar fractures induced by hydraulic fracturing. *International Journal for Numerical Methods in Engineering* 1980; **15**:41–54.
22. Desroches J. Personal communications, 2000.
23. Gradshteyn IS, Ryzik IM. *Tables of Integrals, Series, and Products*. Academic Press: New York, 1980; 904–905.
24. Lin W, Keer LM. Analysis of a vertical crack in a multilayered medium. *Journal of Applied Mechanics* 1989; **56**:63–69.
25. Kuo CH, Keer LM. Three-dimensional analysis of cracking in a multilayered composite. *Journal of Applied Mechanics* 1995; **62**:273–281.
26. He M-Y, Hutchinson JW. Crack deflection at an interface between dissimilar elastic materials. *International Journal of Solids Structures* 1989; **25**(9):1053–1067.