INTERNATIONAL JOURNAL FOR NUMERICAL AND ANALYTICAL METHODS IN GEOMECHANICS, VOL. 15, 000–000 (1991)

EFFICIENT MULTIGRID SOLUTION OF BOUNDARY ELEMENT MODELS OF CRACKS AND FAULTS

A. P. PEIRCE

Department of Mathematics and Statistics, McMaster University, Hamilton, Ont L8S 4K1, Canada

SUMMARY

We consider the multigrid solution of a discretized singular integral equation whose solution represents the displacement discontinuity distribution across a pressurized crack or the slip on a fault subjected to a prescribed shear stress. The multigrid technique reduces the operation count for a crack model having N degrees of freedom from $O(N^3)$ operations for standard stationary iterative methods to $O(N^2)$ operations. In the numerical simulations performed the multigrid approach proves to be extremely efficient even for small values of N. We use Fourier analysis to determine the spectral properties of the coarse grid correction process and the effect of a number of different interpolation operators on the multigrid algorithm. We show that the multigrid technique can be combined with the process of lumping remote influences to yield an algorithm that involves $O(N)$ operations. The performance of multigrid iteration in a nonlinear environment is explored by considering seams filled with nonlinear material. For this purpose a segmented multigrid algorithm is developed that allows for different seam constitutive relations to be used along different line segments. The $O(N^2)$ operation count characteristic of linear multigrid iteration is shown to persist in this nonlinear environment and the segmented multigrid approach is shown to provide significant computational savings for values of N well within the range for typical problems that occur in practice.

1. INTRODUCTION

In the numerical modelling of mining excavations in rockmasses that are assumed to be linear elastic, boundary element (BE) methods have distinct advantages over domain discretization methods such as finite differences and finite elements. Since such excavation problems typically involve infinite or semi-infinite domains, BE models, which only have unknowns on the boundaries of the excavations, are very efficient in terms of the number of elements they require. Another feature of BE methods is that they provide convenient and accurate representations of singular physical phenomena such as jumps in the displacement and stress fields within the elastic body. For example, one form of the indirect BE technique[1] that provides an elegant representation of cracks and faults in two- and three-dimensional elastic bodies is the displacement discontinuity (DD) method.[2] Representing such singular phenomena using domain discretization methods typically requires specialized treatment such as the use of crack elements and severe mesh deformations in the vicinity of the crack. Using the DD method the effect of these singular phenomena in geomechanics (e.g. tabular mining excavations, backfill in tabular excavations, slip on faults and joints, soft seams, and parting planes between various rock strata) are easily modelled. In addition to modelling these localized nonlinearities in an otherwise linear elastic medium, the DD method has also been extended to model nonlinear material behavior (e.g. elasto-plastic behavior) in the vicinity of underground excavations.[3]

In principle, the DD method provides for the modelling of a large number of interacting tabular excavations, parting planes, faults, mining-induced fractures, and continuum rock damage.

0363–9061/91/000000–00$00.00

Received 4 February 1991
Revised 1991

The ultimate engineering objective of these models is the design of safer excavations and the possible exploitation of fractures in the excavation process. However, comprehensive modelling of the pre-existing geological discontinuities and the autonomous generation of mining-induced fractures by the models require that a large number of algebraic equations need to be solved, which constrains the size of the problems that can be addressed. In this paper we investigate the use of multigrid (MG)[4-7] techniques to reduce the computational effort required to solve the discretized integral equations associated with DD models. Our approach is to investigate a model problem that is sufficiently simple to allow analytic methods to be used to explore the convergence properties of the MG algorithm while still retaining most of the important properties that the DD method displays when it is applied to practical mining problems. Indeed, the model problem we have chosen has a number of physical interpretations: (i) the displacements on the boundary of a single tabular (slit-like in 2D) excavation in an infinite elastic medium or (ii) the slip produced on a fault in an infinite elastic medium when subjected to a prescribed shear stress.

As with all BE formulations, the influence matrices associated with the DD method are fully populated. Direct solution methods involve $O(N^3)$ operations for such problems, while iterative solution methods for these problems typically require that the number of iterations grow in proportion to the number of degrees of freedom and therefore also involve $O(N^3)$ operations. This computational disadvantage of the BE technique is compensated by the fact that essentially only the boundaries of the problem need to be discretized, which reduces the number of degrees of freedom required in the DD models. However, since we wish to model problems with large numbers of interacting excavations and geological discontinuities, and given that both direct and iterative solution methods require $O(N^3)$ operations, the computational effort will rapidly become prohibitive without an acceleration procedure such as MG. Moreover, since iterative solution techniques provide considerable flexibility in the modelling of nonlinear behavior within faults and cracks, which is not provided by direct solution methods, it is important to develop efficient iterative algorithms such as MG to solve the discretized nonlinear equations.

Two different approaches have been used to reduce the computational effort for iterative solution of the discretized DD equations. Both these techniques aim to speed up the evaluation of the convolution sums that have to be evaluated in the iteration process. The first approach[8] which is referred to as 'lumping', involves approximating far-field influences by averaged influences, while near-neighbor influences are calculated directly. The second appraoch[9] which only applies to problems on a rectangular mesh, uses the fast Fourier transform (FFT) to speed up the evaluation of the convolution sums. In the case of lumping we can reduce the operation count from $O(N^3)$ to $O(N^2)$ operations and using the FFT we can reduce the operation count to $O(N^2 \log N)$ operations. The MG technique investigated in this paper is aimed at reducing an entirely different component of the computational effort—the number of iterations—by improving the convergence of the iterative scheme. It will be seen that the MG approach also reduces the operation count for iteration from $O(N^3)$ to $O(N^2)$—the number of operations required to perform the convolution sums directly. Since the MG approach reduces a component of the computational effort that is different from that reduced by lumping or FFT, there is potential for reducing the computational effort even further by combining MG with one of these convolution acceleration procedures.

The MG technique has recently enjoyed extensive success in the solution of algebraic equations arising from the domain discretizations of linear and nonlinear partial differential equations.[4-6] In addition, MG methods have been applied to solving BE discretizations of fluid dynamic problems.[10,11] In this paper we investigate the performance of the MG method in solving a model boundary integral equation commonly found in modelling geological discontinuities. A variety of MG methods are applied to the model problem. In addition, we explore the

combination of MG with lumping to achieve a faster algorithm and consider the effect of simple material nonlinearities on the convergence of MG iteration.

In Section 2 we describe the linear and nonlinear model problems and discuss the discretization procedure that is used in this paper. In Section 3 we first consider the convergence characteristics of a wide class of stationary iterative methods applied to the solution of the discretized model problem. Having established the shortcomings of the class of stationary iterative methods we introduce the linear MG procedure and analyze its convergence characteristics when applied to solving the discretized linear model problem. We then discuss the efficiency of the MG algorithm applied to the model problem and describe an algorithm combining MG and lumping. In the remainder of Section 3 we discuss the adaptation of the FAS multigrid approach to the modelling of nonlinear material behavior and give a physical interpretation to the procedure. In Section 4 we compare the performance of various MG algorithms to that of the corresponding Gauss–Seidel algorithm for both linear and nonlinear model problems. In Section 5 we summarize the results, make some concluding remarks, and discuss the importance of this technique for analyzing engineering problems.

2. MODEL PROBLEMS AND DISCRETIZATION

If a collection of tabular crack-like excavations located in the plane $z = 0$ in $\mathbb{R}^3$ is subjected to a normal stress $p(x, y)$ then the convergence DD distribution $U(\xi, \eta)$ is given by the following integral equation:[12]

$$\frac{E}{3\pi(1-\nu^2)} \iint \frac{U(\xi,\eta)}{[(x-\xi)^2+(y-\eta)^2]^{3/2}}\, d\xi\, d\eta + p(x,y) = 0$$

Here E is the Young's modulus and ν the Poisson's ratio for the elastic medium. This equation is used extensively in the engineering design of tabular mining excavations in the vicinity of fault planes.

The model problem arises as a special case of the above equation when the tabular excavations are long rectangular strips in the x–y plane. In this case the excavations can be idealized as a one-dimensional problem by integrating the above equation to yield a collection of crack-like excavations located in the line $y = 0$ in $\mathbb{R}^2$ and subjected to a normal stress distribution $p(x)$. In this case the convergence DD distribution $U(\xi)$ is governed by:[2]

$$\frac{E}{4\pi(1-\nu^2)} \int_{-b}^{b} \frac{U(\xi)}{(x-\xi)^2}\, d\xi + p(x) = 0 \tag{1a}$$

Equation (1a) expresses the fact that the superposition of the mining-induced stresses, represented by the integral in (1a), and the pre-existing stresses $p(x)$ should leave the boundary of the excavation stress free.

We discretize (1a) by dividing the interval $[-b, b]$ into $2N + 1$ equal subintervals or elements of length h and by assuming that over each of these subintervals $U(\xi)$ is constant. The equations for the $2N + 1$ unknown DD values associated with the piecewise constant approximation to $U(\xi)$ are obtained by collocation at the midpoints $x_m = mh$, $m = -N, \ldots, N$ of the elements:

$$\sum_{m=-N}^{N} A_{nm} u_m = -p_n = f_n \tag{2}$$

where

$$A_{mn} = \frac{\gamma}{4(m-n)^2 - 1}; \quad \gamma = \frac{E}{h\pi(1-\nu^2)}$$

and u_m is used to denote the numerical approximation to $\bar{U}(x_m)$. The convolution property $A_{mn} = A_{|m-n|}$ has been exploited in the FFT algorithm[9] to speed up the iterative solution of (2) and will be used in Section 3 to determine the spectral properties of A.

In order to test the performance of the MG technique for a nonlinear problem, we consider the mining to be performed in a soft seam having a nonlinear constitutive relationship. The effect of the soft seam is incorporated into the model problem by including the term $F[x, U(x)]$ on the right side of (1a):

$$\frac{E}{4\pi(1-\nu^2)}\int_{-b}^{b}\frac{U(\xi)}{(x-\xi)^2}\,d\xi + p(x) = F[x, U(x)] \tag{1b}$$

Here $F[x, U(x)]$ represents the load carried by the soft seam when it has been compressed by an amount $U(x)$. A stress free crack-like excavation on a subinterval $[c, d] \subset [-b, b]$ can be obtained within this framework by letting $F[x, U(x)] = 0$ for $x \in [c, d]$.

3. ITERATIVE SOLUTION OF MODEL PROBLEM

3.1. Stationary iterative methods

The basis for the MG method is the use of stationary iterative techniques to solve (2). We introduce a class of relaxation operators[4] by decomposing the matrix A as follows:

$$\mathbf{A} = \mathbf{A}^+ + \mathbf{A}^0 + \mathbf{A}^-$$

where $\mathbf{A}^0$ acts on those points that are solved for simultaneously in the relaxation process, $\mathbf{A}^-$ acts on those points where previous values are used within the relaxation process and $\mathbf{A}^+$ acts on those points where new values are already available. The class of relaxation operators can now be expressed in the form:

$$(\mathbf{A}^0 + \omega\mathbf{A}^+)\mathbf{u}^{k+1} = \omega\mathbf{f} + [(1-\omega)\mathbf{A}^0 - \omega\mathbf{A}^-]\mathbf{u}^k \tag{3}$$

Here $\mathbf{u}^k$ denotes the approximate solution to (2) before the relaxation step and $\mathbf{u}^{k+1}$ denotes the solution after the relaxation step. If we choose $\mathbf{A}^0$ to comprise the diagonal elements of A, $\mathbf{A}^-$ to contain the off-diagonal parts of A and $\mathbf{A}^+ = 0$, then we obtain the weighted Jacobi method. If we choose $\mathbf{A}^0$ to comprise the diagonal elements of A, $\mathbf{A}^-$ to contain the strictly upper triangular part of A and $\mathbf{A}^+$ to contain the strictly lower triangular part of A then we obtain the successive over relaxation (SOR) method. By choosing $\omega = 1$ in the SOR method we obtain the Gauss–Seidel (GS) method.

If we consider the minimization of the function $V(\mathbf{u}) = \mathbf{f}^T\mathbf{u} - \frac{1}{2}\mathbf{u}^T\mathbf{A}\mathbf{u}$, then the necessary condition for a minimum is that $\nabla V^T(\mathbf{u}) = \mathbf{f} - \mathbf{A}\mathbf{u} = 0$, which is just (2). It is interesting to note that the weighted Jacobi method can be expressed in the form of a gradient method for the minimization of $V(\mathbf{u})$:

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \omega\tilde{\mathbf{A}}^{-1}\mathbf{r}^k \tag{4}$$

where the residual r is defined to be:

$$\mathbf{r}^k = \mathbf{f} - \mathbf{A}\mathbf{u}^k$$

and $\tilde{\mathbf{A}}^{-1}$ is an approximate inverse of A. If we choose $\tilde{\mathbf{A}}$ to consist of the diagonal elements of A then we obtain the weighted Jacobi method. If we let $\tilde{\mathbf{A}} = \mathbf{I}$ and choose ω_k to be a nonnegative scalar minimizing $V[\mathbf{u}^k - \omega\nabla V^T(\mathbf{u}^k)]$ then we obtain the method of steepest descents.[13]

We now use Fourier analysis to investigate the convergence properties of the class of iteration schemes (3) when applied to the solution of (2). It is convenient to assume that we are dealing with an infinite grid $X_h = \{x_m = mh, m \in \mathbb{Z}\}$. Physically this assumption can be justified by considering two different models. In the first we consider iteration on an infinite seam which lies in the line $y = 0$. In the second we consider a finite crack modelled by a large but finite number of DD elements, which is approximated by the infinite grid.

On the infinite grid X_h the operator A is an infinite Toeplitz matrix which has eigenfunctions:

$$\phi(\theta, x) = e^{i\theta x/h}; \quad -\pi < \theta \leq \pi$$

Operating on $\phi(\theta, x)$ with A we obtain the Fourier representation $\hat{A}(\theta, h)$ of A:

$$\hat{A}(\theta, h) = -\frac{\gamma\pi}{2}\sin\left(\frac{|\theta|}{2}\right) \tag{5}$$

where the identity[14]

$$\sum_{m=1}^{\infty} \frac{\cos(m\theta)}{4m^2 - 1} = \frac{1}{2} - \frac{\pi}{4}\sin\left(\frac{|\theta|}{2}\right); \quad -\pi \leq \theta \leq \pi \tag{6}$$

has been used. We note that by letting $\theta = \xi h$ and taking the limit as $h \to 0$ in (5) we obtain the Fourier transform of the kernel in (1a):

$$\frac{E}{4\pi(1-\nu^2)}\int_{-\infty}^{\infty} \frac{e^{(i\xi x)}}{x^2}\,dx = -\frac{E}{4(1-\nu^2)}|\xi|$$

This result can be derived directly using distribution theory.[15]

Let $\mathbf{u}^*$ denote the exact solution of (2), which also satisfies (3) identically. Since A is a linear operator, the equation governing the evolution of the error vector $\mathbf{e} = \mathbf{u}^* - \mathbf{u}$ can be obtained by subtracting (3) from (2) in which u has been replaced by $\mathbf{u}^*$. Following this process we see that the evolution of the error e is governed by (3) with u replaced by e and without the inhomogeneous term f. Expanding the error vectors $\mathbf{e}^{k+1}$ and $\mathbf{e}^k$ in terms of the eigenfunctions $\phi(\theta, x)$ we obtain the following Fourier representation of the action of the relaxation operator (3):

$$\hat{\mathbf{e}}^{k+1}(\theta, h) = R(\theta, h; \omega)\hat{\mathbf{e}}^k(\theta, h) \tag{7}$$

where

$$R(\theta, h; \omega) = \frac{(1-\omega)\hat{A}^0(\theta, h) - \omega\hat{A}^-(\theta, h)}{\hat{A}^0(\theta, h) + \omega\hat{A}^+(\theta, h)}$$

and $\hat{A}^0(\theta, h)$, $\hat{A}^-(\theta, h)$ and $\hat{A}^+(\theta, h)$ are Fourier representations of the infinite Toeplitz matrices A^0, A^- and A^+.

It is possible in the case of the weighted Jacobi method to obtain a simple expression for the damping factor in (7):

$$R_J(\theta, h; \omega) = 1 - \frac{\omega\pi}{2}\sin\left(\frac{|\theta|}{2}\right) \tag{8}$$

To ensure that the errors are damped from one iteration to the next we require that the convergence factor:

$$\rho_J = \max_{|\theta| < \pi} |R_J(\theta, h; w)|$$

is less than 1. This requirement will be met provided $0 < \omega < 4/\pi$. We notice from (8) that, for

values of ω away from 0 or $4/\pi$, it is the low-frequency error components (i.e. those for which $\theta \approx 0$) that persist after a number of iterations and therefore dominate the convergence of the iteration scheme. In Figure 1 the damping factor $R_J(\theta, h; \omega = 1)$ is plotted for $0 < \theta \leq \pi$. The damping factor is clearly largest for the lowest frequencies. For a particular mesh with $2N + 1$ elements the lowest frequency $\theta_{\min} \propto N^{-1}$, thus from (8) it follows that the asymptotic behavior of the convergence factor is:

$$\rho_J = 1 - O(N^{-1}), \quad N \to \infty \tag{9}$$

Over-relaxation (i.e. choosing $1 < \omega < 4/\pi$) in this case will accelerate the reduction of low frequency errors slightly by increasing the $O(N^{-1})$ term in (9) by a factor of at most $4/\pi$. As was stated in the introduction it can be seen from (9) that the number of iterations required to reduce the error in the solution of (2) to within a prescribed tolerance will grow in proportion to the number of degrees of freedom.

It is interesting to note that the asymptotic convergence rate for the weighted Jacobi scheme when applied to finite difference (and finite element) equations typically degrades as $O(N^{-2})$. This is due to the weak coupling of the sparse finite difference equations that transmit low frequency (i.e. long range) effects extremely slowly. Since the BE equations (2) are all coupled (because A is

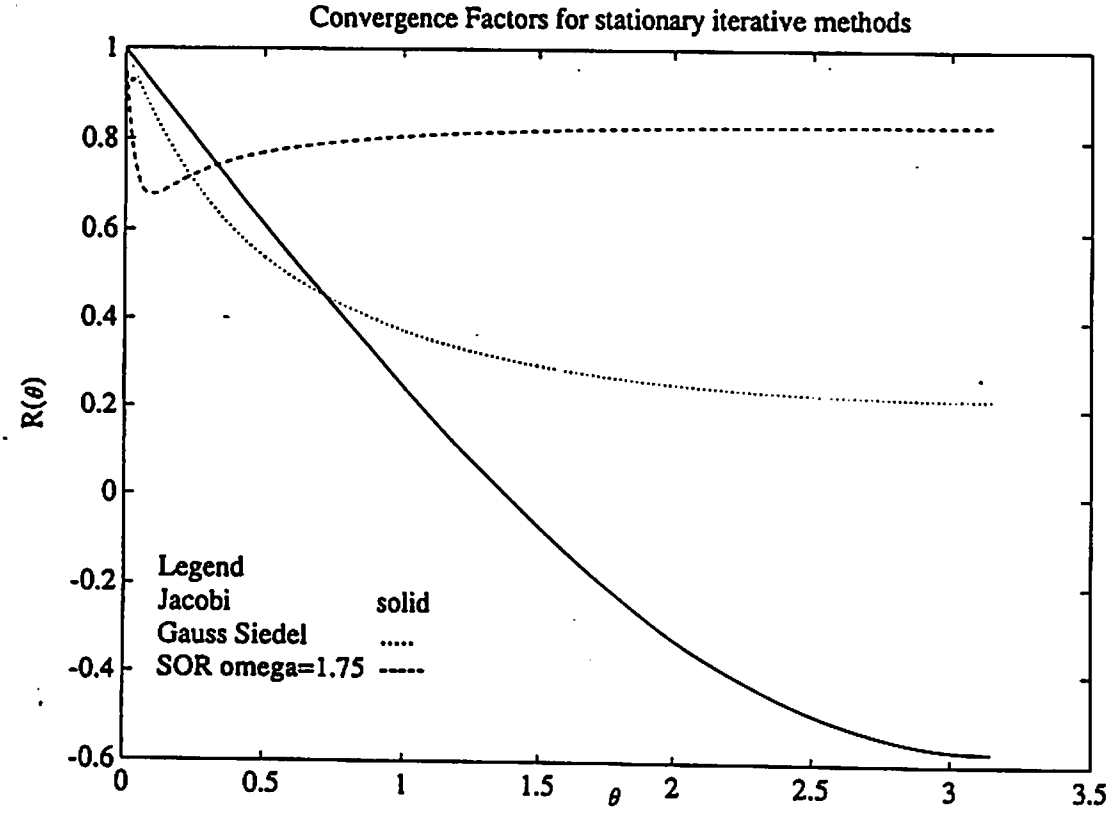

Figure 1. The damping factors $R_J(\theta, h; \omega = 1)$, $R_{GS}(\theta, h)$, and $R_{SOR}(\theta, h; \omega = 1{\cdot}75)$ defined in (7). The damping factors are largest for the lowest frequencies $\theta \approx 0$ for all methods while both Jacobi and GS methods rapidly smooth high frequency errors $\pi/2 < \theta \leq \pi$. SOR reduces the lower frequency errors in the range $0{\cdot}05 < \theta < 0{\cdot}25$ at the expense of the smoothing of the high frequency errors. However as $\theta \to 0$ the low frequency errors dominate the SOR algorithm once again.

fully populated) it is not surprising that the long-range effects are transmitted more rapidly for the BE equations.

In the case of SOR it is not easy to obtain a simple expression for the damping factor $R_{SOR}(\theta, h; \omega)$ defined in (7) so it has been evaluated numerically and is plotted in Figure 1 for GS iteration (i.e. $\omega = 1$) and SOR with $\omega = 1{\cdot}75$. As was the case with weighted Jacobi iteration, the damping factors are largest for the lowest frequencies. From these two curves it can be seen that over-relaxation reduces the lower frequency errors $0{\cdot}05 < \theta < 0{\cdot}25$ at the expense of the smoothing of the high frequency errors. If we increase N then $\theta_{min} \to 0$ and the low frequency errors start to dominate once more. Unfortunately the performance of SOR depends crucially on the choice of ω, which depends on the problem. For nonlinear problems the use of SOR can lead to unphysical solutions and the choice of the appropriate ω becomes even more uncertain. The MG techniques we discuss in Section 3 aim to achieve robust problem-independent improvement of the convergence of the iterative schemes.

It is possible to determine the asymptotic behavior of $R_{SOR}(\theta, h; \omega)$ in the low frequency limit $\theta \to 0$:

$$R_{SOR}(\theta, h; \omega) \approx \frac{1 - \dfrac{\omega\pi|\theta|}{4(2-\omega)}}{1 + \dfrac{\omega\pi|\theta|}{4(2-\omega)}} \tag{10}$$

Here (6) has been used and the imaginary terms in $R_{SOR}(\theta, h; \omega)$ have been ignored since they are subdominant. From (10) it follows that in order to ensure that $\rho_{SOR} < 1$ we must require that $0 < \omega < 2$. Equation (10) does not provide sufficient information to determine the optimal value of ω since the asymptotic relation is not uniformly valid in the range $-\pi < \theta \leq \pi$.

In the case of GS iteration, $\omega = 1$, and (10) reduces to:

$$R_{GS}(\theta, h) \approx 1 - \frac{\pi}{2}|\theta| \tag{11}$$

Comparing (8) and (11) we see that GS iteration can be expected to converge twice as fast as weighted Jacobi iteration with $\omega = 1$. However, the number of iterations required by GS iteration to reduce the error in the solution of (2) to within a prescribed tolerance also grows in proportion to the number of degrees of freedom.

From Figure 1 it can also be seen that both Jacobi and SOR methods rapidly smooth high-frequency errors $\pi/2 < \theta \leq \pi$. This is due to the strong near-neighbor coupling of the BE equations (2) which cause local/high frequency adjustments to be performed rapidly.

We have seen that the class of relaxation schemes (3) is efficient at smoothing high frequency errors while it is inefficient at reducing low-frequency errors—particularly when the mesh is refined. These are the essential features that we will require to construct a MG algorithm in Section 3.2. Since iterative algorithms to solve the BE equations degrade much less rapidly when the mesh is refined than do stationary iterative methods when used to solve finite difference equations, there may not be any advantage in improving the convergence rate for the BE equations by MG. In this paper we examine this issue.

3.2. *Linear multigrid iteration*

Since there is an abundance of literature describing the MG method (e.g. see References 4–7), we briefly describe the MG process using only two levels of 'gridding'. Consider a fine mesh X_h

and a coarse mesh X_{2h} and denote quantities and operators on each mesh by the subscripts h and $2h$ respectively. The MG method involves the following steps:

1. Smooth high-frequency errors by relaxing q_1 times on the finer mesh:

$$\mathbf{u}_h \leftarrow \mathbf{R}_h^{q_1}\mathbf{u}_h$$

2. Transfer the residual $\mathbf{r}_h = \mathbf{f}_h - \mathbf{A}_h\mathbf{u}_h$ to a coarser mesh by 'injection':

$$\mathbf{r}_{2h} = \mathbf{I}_h^{2h}\mathbf{r}_h$$

3. Solve (2) exactly on the coarser mesh using the injected residual as the inhomogeneous term:

$$\mathbf{u}_{2h} = \mathbf{A}_{2h}^{-1}\mathbf{r}_{2h}$$

4. Transfer the correction back to the finer grid by interpolation and update the approximate solution:

$$\mathbf{u}_h \leftarrow \mathbf{u}_h + \mathbf{I}_{2h}^h\mathbf{u}_{2h}$$

5. Smooth high-frequency errors by relaxing q_2 times on the finer mesh as in step 1.

Steps 2–4 serve to reduce the low frequency errors and are collectively known as coarse grid CG correction. The action of CG correction on the error $\mathbf{e}_h$ can be represented by the action of a single operator CG:[6]

$$\mathbf{e}_h \leftarrow \mathbf{R}_h^{q_2}[\mathbf{I} - \mathbf{I}_{2h}^h\mathbf{A}_{2h}^{-1}\mathbf{I}_h^{2h}\mathbf{A}_h]\mathbf{R}_h^{q_1}\mathbf{e}_h = \mathbf{CG}\mathbf{e}_h \tag{12}$$

Comparing (12) with (4) we see that the coarse grid correction procedure can be viewed formally as the construction of an approximate inverse $\mathbf{I}_{2h}^h\mathbf{A}_{2h}^{-1}\mathbf{I}_h^{2h}$. In practice steps 2–4 are repeated recursively on a coarsening sequence of meshes such as $X_h, X_{2h}, X_{4h}, \ldots$ and the exact inversion only takes place on the coarsest mesh.

It is not possible to obtain the Fourier representation of CG directly by operating on the functions $\phi(\theta, x)$ since they are not the eigenfunctions of CG. Instead a careful account has to be kept of the action of each of the component operators in CG. The first step in this process is to establish the relationship between the modes $\phi(\theta, x)$ on the fine mesh X_h and the modes on X_{2h}, which we shall denote by $\Phi(\theta, x)$. We divide the modes $\phi(x, h)$ into low-frequency modes $(-\pi/2 < \theta \le \pi/2)$ and high frequency modes $(\pi/2 < |\theta| \le \pi)$. The low-frequency fine mesh modes have the following relationship with the coarse mesh modes:

$$\phi(\theta, x) = \Phi(\theta, x), \quad -\frac{\pi}{2} < \theta \le \frac{\pi}{2}, x \in X_{2h}$$

The high-frequency modes can be expressed in terms of values of $\theta \in (-\pi/2, \pi/2]$ as follows:

$$\phi(\theta', x); \quad \theta' = \theta - \mathrm{sgn}(\theta)\pi; \quad -\frac{\pi}{2} < \theta \le \frac{\pi}{2}$$

These high-frequency modes are precisely those that are aliased when represented on the coarse mesh X_{2h}.

The action of the coarse grid operator $\mathbf{A}_{2h}$ on the low frequency modes is:

$$\mathbf{A}_{2h}\phi(\theta, x) = -\frac{\gamma\pi}{4}\sin(|\theta|)\phi(\theta, x); \quad -\frac{\pi}{2} < \theta \le \frac{\pi}{2} \tag{13}$$

The action of the fine mesh operator $\mathbf{A}_h$ on the low and high frequency modes can be expressed as

follows:

$$A_h\phi(\theta, x) = -\frac{\gamma\pi}{2}\sin\left(\frac{|\theta|}{2}\right)\phi(\theta, x); \quad -\frac{\pi}{2} < \theta \le \frac{\pi}{2} \tag{14a}$$

and

$$A_h\phi(\theta', x) = -\frac{\gamma\pi}{2}\cos\left(\frac{\theta}{2}\right)\phi(\theta', x); \quad -\frac{\pi}{2} < \theta \le \frac{\pi}{2} \tag{14b}$$

where (6) has been used to obtain (14) and the identity[14]

$$\sum_{m=1}^{\infty} \frac{(-1)^m \cos(m\theta)}{4m^2 - 1} = \frac{1}{2} - \frac{\pi}{4}\cos\left(\frac{\theta}{2}\right); \quad -\pi \le \theta \le \pi$$

has been used to obtain (14b).

3.2.1 The injection operator I_h^{2h}. We construct the injection operator by linear interpolation. In Figure 2 we show how the residual values at the fine mesh points marked $2j$ and $2j + 1$ are interpolated with weights 1/2 to give the injected residual at the point marked j on the coarse mesh:

$$r_{2h,j} = \{\tfrac{1}{2} \quad \tfrac{1}{2}\} \begin{Bmatrix} r_{h,2j} \\ r_{h,2j+1} \end{Bmatrix} \tag{15}$$

We now consider the action of I_h^{2h} on the fine mesh modes:
Low Frequency:

$$I_h^{2h} e^{i(2j)\theta} = \cos\left(\frac{\theta}{2}\right) e^{i(2j+1/2)\theta}; \quad -\frac{\pi}{2} < \theta \le \frac{\pi}{2} \tag{16}$$

High Frequency:

$$I_h^{2h} e^{i(2j)\theta'} = -i\sin\left(\frac{\theta}{2}\right) e^{i(2j+1/2)\theta}; \quad -\frac{\pi}{2} < \theta \le \frac{\pi}{2} \tag{17}$$

Note that the index $2j + \frac{1}{2}$ in (16) and (17) represents the fine mesh index of the coarse mesh point j that lies midway between the fine mesh points $2j$ and $2j + 1$ (see Figure 2).

3.2.2 The interpolation operators I_{2h}^h. We consider a number of different interpolation operators. One fundamental requirement of the interpolation operators is that the volume of closure

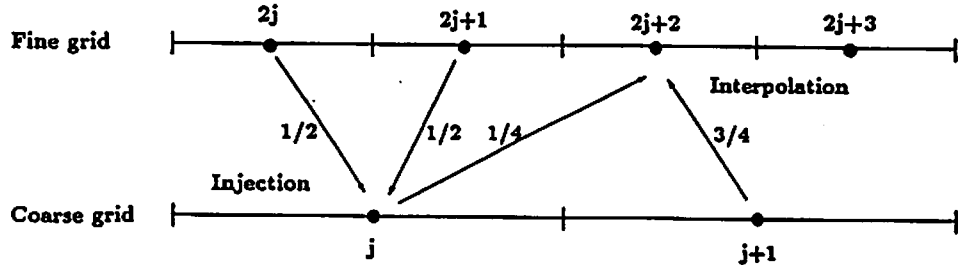

Figure 2. The residual values at the fine mesh points marked $2j$ and $2j + 1$ are interpolated with weights $\frac{1}{2}$ to give the injected residual at the point marked j on the coarse mesh. Linear interpolation between the values $u_{2h,j}$ and $u_{2h,j+1}$ on the coarse grid is used to obtain the solution on the fine grid at points located at $2j + 1$ and $2j + 2$

should be conserved when the solution is transferred from the coarse mesh to the fine mesh, i.e.

$$h \sum_{m=-N}^{N} u_{h,m} = 2h \sum_{m=-N/2}^{N/2} u_{2h,m}$$

This requirement can be conveniently tested by writing the interpolation operator in matrix form and ensuring that each column has a sum of 2. Another fundamental requirement of the interpolation operators is that they should at least transfer rigid body displacements exactly between meshes, which means that each of the rows should sum to 1.

Piecewise constant interpolation (PWC): In this case we transfer the solution $u_{2h,j}$ on the coarse grid directly to the fine grid points located at $2j$ and $2j+1$ (see the relative locations of these mesh points in Figure 2). This interpolation process can be expressed as follows:

$$\begin{Bmatrix} u_{h,2j} \\ u_{h,2j+1} \end{Bmatrix} = \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} [u_{2h,j}] \tag{18}$$

The local interpolation operator given in (18) can be used to construct a global interpolation operator that satisfies the volume of closure and rigid body requirements.

The action of the interpolation operator $I^h_{\mathrm{PWC},2h}$ on the coarse-mesh mode $e^{i(j+\frac{1}{2})2\theta}$ is:

$$I^h_{\mathrm{PWC},2h}\, e^{i(j+1/4)2\theta} = \cos\left(\frac{\theta}{2}\right) e^{i(2j)\theta} + i \sin\left(\frac{\theta}{2}\right) e^{i(2j)\theta}; \quad -\frac{\pi}{2} < \theta \leq \frac{\pi}{2} \tag{19}$$

Here $j + \frac{1}{4}$ is the coarse mesh index of the grid point located at $2j+1$ on the fine mesh. The interpolation operator projects the coarse mesh mode onto the low frequency fine mesh mode with weight $\cos(\theta/2)$ and the high-frequency fine-mesh mode with weight i $\sin(\theta/2)$. The complex weight i $\sin(\theta/2)$ reflects the fact that the coarse and fine meshes are shifted with respect to each other. This mesh shift manifests itself as a phase shift that applies only to the high frequency components.

Piecewise linear interpolation (PWL): In this case we use linear interpolation between the values $u_{2h,j}$ and $u_{2h,j+1}$ on the coarse grid to obtain the solution on the fine grid at points located at $2j+1$ and $2j+2$ (the appropriate weights for linear interpolation are shown in Figure 2). This interpolation process can be expressed as follows:

$$\begin{Bmatrix} u_{h,2j-1} \\ u_{h,2j} \\ u_{h,2j+1} \\ u_{h,2j+2} \end{Bmatrix} = \begin{Bmatrix} 1/4 \\ 3/4 \\ 3/4 \\ 1/4 \end{Bmatrix} [u_{2h,j}] \tag{20}$$

The global interpolation operators constructed from (20) satisfy both the rigid body and volume of closure requirements for mesh points away from the boundaries. Using piecewise constant interpolation at the boundary points it is possible to construct a global interpolation operator that satisfies both these requirements:

$$\begin{Bmatrix} u_{h,0} \\ u_{h,1} \\ u_{h,2} \\ \vdots \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots \\ 3/4 & 1/4 & 0 & \cdots \\ 1/4 & 3/4 & 0 & \cdots \\ 0 & 3/4 & 1/4 & \cdots \\ 0 & 1/4 & 3/4 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{Bmatrix} u_{2h,0} \\ u_{2h,1} \\ \vdots \end{Bmatrix} \tag{21}$$

where 0 is the index of the mesh point closest to the edge on both the coarse and fine meshes. In practice the matrix element 1 representing a piecewise constant approximation of the solution can be replaced by 1/2, which represents a linear approximation of the solution at the edge of the crack (assuming that at the tip of the crack $U = 0$). Numerical studies indicate that this adjustment [being a local (i.e. high frequency) effect] does not have a noticeable influence on the convergence properties of the MG scheme.

In this case the action of the interpolation operator defined by (20) $I^{2h}_{\mathrm{PWL},h}$ on the coarse mesh mode $e^{i(J+\frac{1}{2})2\theta}$ is given by:

$$I^{2h}_{\mathrm{PWL},h} e^{i(J+\frac{1}{2})2\theta} = \frac{1}{4}[e^{i(2J-3/2)\theta} + 3e^{i(2J+1/2)\theta}]$$
$$= \cos^3\left(\frac{\theta}{2}\right)e^{i(2J)\theta} + i\sin^3\left(\frac{\theta}{2}\right)e^{i(2J)\theta'}; \quad -\frac{\pi}{2} < \theta \le \frac{\pi}{2} \tag{22}$$

The similarities between the action of the PWL and PWC interpolation operators can be seen by comparing (22) with (19).

Piecewise quadratic interpolation (PWQ): In this case we use quadratic interpolation between the values $u_{2h,j-1}$, $u_{2h,j}$ and $u_{2h,j+1}$ on the coarse grid to obtain the solution on he fine grid at points located at $2j + 1$ and $2j + 2$. This interpolation process can be expressed as follows:

$$\begin{Bmatrix} u_{h,2j-2} \\ u_{h,2j-1} \\ u_{h,2j} \\ u_{h,2j+1} \\ u_{h,2j+2} \\ u_{h,2j+3} \end{Bmatrix} = \frac{1}{32}\begin{Bmatrix} -3 \\ 5 \\ 30 \\ 30 \\ 5 \\ -3 \end{Bmatrix}[u_{2h,j}] \tag{23}$$

The global interpolation operators constructed from (20) satisfy both the rigid body and volume of closure requirements for mesh points away from the boundaries. Analogous edge correction can be performed in this case.

In this case the action of the interpolation operator defined by (23) $I^{h}_{\mathrm{PWQ},2h}$ on the coarse mesh mode $e^{i(J+1/4)2\theta}$ is given by:

$$I^{h}_{\mathrm{PWQ},2h} e^{i(J+1/4)2\theta} = \cos^3\left(\frac{\theta}{2}\right)\left(1 + \frac{3}{2}\sin^2\left(\frac{\theta}{2}\right)\right)e^{i(2J)\theta} + i\sin^3\left(\frac{\theta}{2}\right)\left(1 + \frac{3}{2}\cos^2\left(\frac{\theta}{2}\right)\right)e^{i(2J)\theta'};$$
$$-\frac{\pi}{2} < \theta \le \frac{\pi}{2} \tag{24}$$

3.2.3 The spectral properties of the coarse grid correction operator CG. From (17) it can be seen that the action of the injection operator I^{2h}_{h} on the high frequency modes $\phi(\theta', x)$ results in a multiple of the corresponding low frequency mode $\phi(\theta, x)$. In addition from (19) and (21) we observe that the action of I^{h}_{2h} on the low frequency mode $\phi(\theta, x)$ results in a linear combination of the low frequency $\phi(\theta, x)$ and high frequency $\phi(\theta', x)$ modes. Thus the operator CG will not leave each of the modes invariant. However, it does leave the subspace, span $\{\phi(\theta, x), \phi(\theta', x)\}$, invariant. Thus the damping characteristic of CG for each frequency θ can be obtained by finding the eigenvalues of a 2×2 matrix. In particular:

$$\mathrm{CG}\begin{bmatrix} e^{i(2J)\theta} \\ e^{i(2J)\theta'} \end{bmatrix} = \begin{bmatrix} (1 - \mathscr{C}(\theta))R^{q_1+q_2}(\theta) & -i\mathscr{S}(\theta)R^{q_1}(\theta)R^{q_2}(\theta') \\ i\mathscr{C}(\theta)\mathrm{sgn}(\theta)R^{q_1}(\theta')R^{q_2}(\theta) & (1 - |\mathscr{S}(\theta)|)R^{q_1+q_2}(\theta') \end{bmatrix}\begin{Bmatrix} e^{i(2J)\theta} \\ e^{i(2J)\theta'} \end{Bmatrix} \tag{25}$$

Here equations (13) (14a), (14b), (16), (17), (19), (22) and (24) have been used to determine the action of each of the component operators in (14a). In addition, $R(\theta)$ is defined in (7) and $\mathscr{S}(\theta)$ and $\mathscr{C}(\theta)$ are given by:

$$\mathscr{S}(\theta) = \begin{cases} \sin\left(\frac{\theta}{2}\right) & \text{for piecewise constant interpolation} \\ \sin^3\left(\frac{\theta}{2}\right) & \text{for piecewise linear interpolation} \\ \sin^3\left(\frac{\theta}{2}\right)\left[1 + \frac{1}{2}\cos^2\left(\frac{\theta}{2}\right)\right] & \text{for piecewise quadratic interpolation} \end{cases}$$

and

$$\mathscr{C}(\theta) = \begin{cases} \cos\left(\frac{\theta}{2}\right) & \text{for piecewise constant interpolation} \\ \cos^3\left(\frac{\theta}{2}\right) & \text{for piecewise linear interpolation} \\ \cos^3\left(\frac{\theta}{2}\right)\left[1 + \frac{1}{2}\sin^2\left(\frac{\theta}{2}\right)\right] & \text{for piecewise quadratic inerpolation} \end{cases}$$

We note that for small θ the factor $[1 - \mathscr{C}(\theta)]$ in the first diagonal term in (25) represents roughly the factor by which the low frequency errors are reduced by coarse grid correction. For the PWC and PWL interpolation schemes this factor is $O(\theta^2)$ whereas for the PWQ scheme this factor is $O(\theta^4)$. The factor $(1 - |\mathscr{S}(\theta)|)$ in the second diagonal term represents the (typically small) amount by which the high frequency errors are reduced by coarse grid correction.

By calculating the eigenvalues of the matrix in (25) we obtain the following expression for the error damping characteristic of CG as a function of θ:

$$\widehat{CG}(\theta) = \frac{1}{2}[\alpha[1 - \mathscr{C}(\theta)] + \beta[1 - |\mathscr{S}(\theta)|] + \sqrt{\{[\alpha(1 - \mathscr{C}(\theta)) - \beta(1 - |\mathscr{S}(\theta)|)]^2 + 4\alpha\beta|\mathscr{S}(\theta)|\mathscr{C}(\theta)\}}] \quad (26)$$

where $\alpha = R^{q_1+q_2}(\theta)$ and $\beta = R^{q_1+q_2}(\theta')$. We notice from (26) that $\widehat{CG}(\theta)$ depends on the total number $q_1 + q_2$ of smoothing relaxations and not on the order in which they are applied, therefore we will use the single parameter $q = q_1 + q_2$ in the remainder of the paper. If coarse grid correction were performed without smoothing relaxations (i.e. $q = 0$) then it follows from (26) that $\widehat{CG}(\theta) = 1$ for all θ. This is because coarse grid correction alone does nothing to reduce the high frequency errors.

Using a solid line in Figure 3 the function $\widehat{CG}(\theta)$ for piecewise constant interpolation is plotted for different values of q. It has been assumed that Gauss–Seidel iteration has been used to perform the smoothing relaxations. The reduction of low frequency errors can be seen clearly from these plots. In the case $q = 1$ the intermediate frequency $\theta = \pi/4$ dominates the convergence of the CG correction scheme. The plots in Figure 3 become clearer when compared to the error damping characteristic of GS iteration plotted in Figure 1. The error damping characteristic of GS iteration in Figure 1 for the high frequencies $(\pi/2 < |\theta| \leq \pi)$ has a minimum of (0·222) and maximum of (0·288). From Figure 3 we observe that the piecewise constant CG damping factor

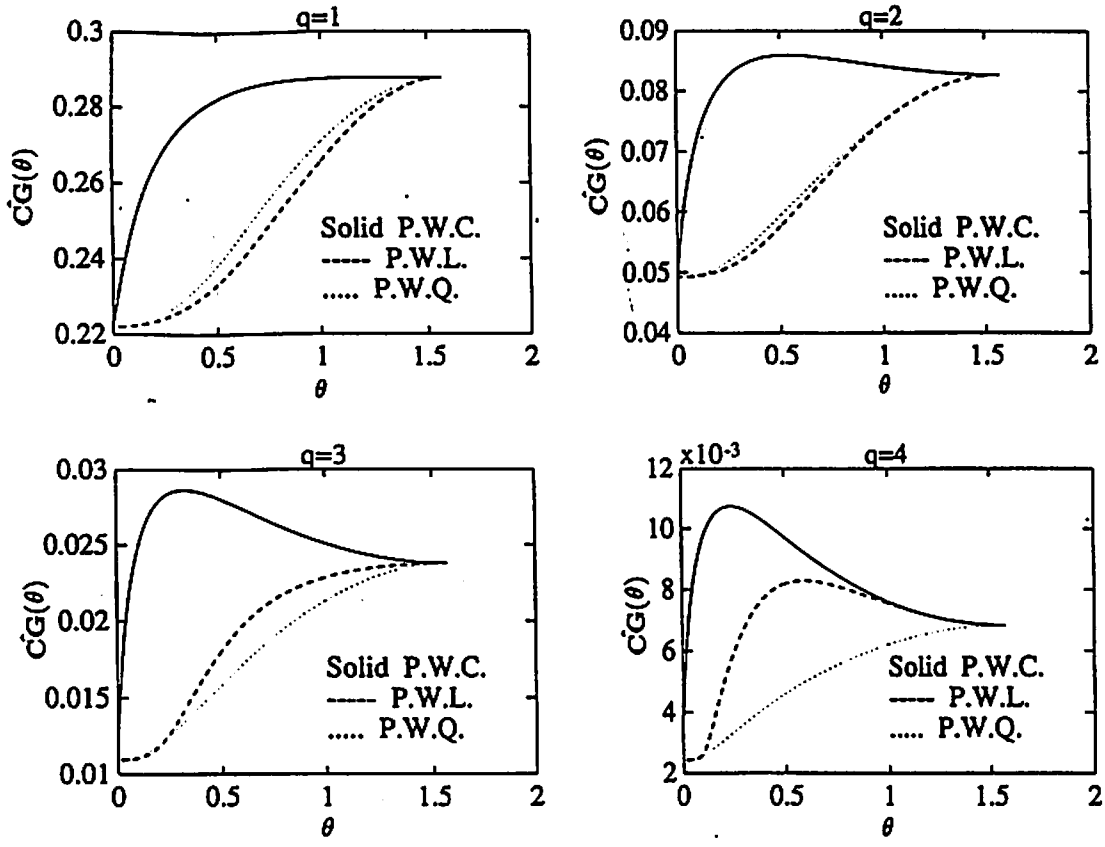

Figure 3. The function $\widehat{CG}(\theta)$ for PWC (solid), PWL (dashed) and PWQ (dotted) interpolation is plotted for different values of q. It has been assumed that Gauss–Seidel iteration has been used to perform the smoothing relaxations. For PWC interpolation the function $\widehat{CG}(\theta)$ is monotonic only for $q = 1$, for PWL interpolation the function $\widehat{CG}(\theta)$ is monotonic when $1 \leq q \leq 3$, while for PWQ interpolation the function $\widehat{CG}(\theta)$ is monotonic for all values of q considered

for $q = 1$ has a minimum of (0·224) and a maximum of (0·288). Therefore, the CG correction scheme has essentially inherited the desirable high frequency damping characteristic from the GS scheme. The undesirable low frequency damping characteristic of the GS scheme has been made subdominant to the high frequency damping by the CG correction process. What is most significant, from an efficiency point of view, is that the upper bound $\widehat{CG}_{max} = 0{\cdot}288$ on the damping characteristic is independent of the number of mesh points $2N + 1$ used in the discretization of the problem. Thus this desirable damping characteristic will not degrade as the mesh is refined. In the case $q = 2$ shown in Figure 3, the damping characteristic is no longer dominated by the GS iteration process. In this case extra smoothing associated with $q = 2$ has reduced the intermediate frequency (i.e. $\theta = \pi/4$) errors below those introduced by interpolation in the CG correction process. For GS iteration to dominate the CG damping characteristic, the curve should be monotonic like the GS damping characteristic — which is the case for PWC when $q = 1$ but not for $q = 2$, 3, and 4.

Using a dashed line in Figure 3, the function $\widehat{CG}(\theta)$ for piecewise linear interpolation is plotted for different values of q. In this case, the maximum and minimum values of $\widehat{CG}(\theta)$ for $q = 1$ are almost identical to the damping factors of GS iteration (see Figure 1) in the high frequency range

$(\pi/2 < |\theta| \leq \pi)$. In this case monotonicity persists for $q = 2$, and $q = 3$, after which (i.e. for $q \geq 4$) the errors due to the linear interpolation component of the CG process becomes significant.

In order to validate the theoretical convergence factors given by (26) we measured the rate at which the L_2 norm of the residual was reduced when a PWL CG correction scheme was used to solve the discretized crack integral equation (2). Let ρ_2 denote the amount by which the L_2 norm of the residual decreased between successive coarse-grid corrections. The following results were obtained:

$$\rho_{2,q=1} = 0{\cdot}213,\ \rho_{2,q=2} = 0{\cdot}078, \qquad \rho_{2,q=3} = 0{\cdot}017, \qquad \rho_{2,q=4} = 0{\cdot}044$$

In order to compare this with the theoretical values of $\widehat{CG}$ we use the fact that $\mathbf{r}^k = \mathbf{A}(\mathbf{u}^* - \mathbf{u}^k)$ and expand $\mathbf{r}^k$ in the eigenfunctions of A to obtain:

$$\|\mathbf{r}^{k+1}\|_2 \leq \widehat{CG}_{\max} \|\mathbf{r}^k\|_2 \tag{27}$$

Comparing the observed ρ_2 values with the piecewise linear coarse grid correction factor $\widehat{CG}$ plotted in Figure 3 we see that the observed convergence factors are consistently less than the theoretical upper bound as given by (27). The closeness of the observed convergence factors to the theoretical ones will depend on the mix of frequency components in the residual.

In Figure 3 the PWQ CG correction operator (shown by a dotted line) can be seen to exhibit the same behavior as the PWL operator except that for the PWQ case the monotonicity property persists for the larger value of $q = 4$.

3.2.4 *Efficiency of MG and enhancements: lumping*

Operation count and memory requirements of MG: In practice the soluton is not actualy determined directly on the coarse grid but rather a sequence of coarser grid corrections are used to obtain the solution on the coarse grid. A number of such cycles of solution between various grids can be used, including: V-cycles, W-cycles, and the FMG full multigrid algorithm (e.g. see Reference 6). The performance of these three MG strategies when applied to the solution of the discretized crack problem (2) will be discussed in Section 4.

The computational complexity associated with the MG solution of (2) can be determined by a procedure similar to that used to determine the complexity of MG solution of finite difference equations.[5,6] We assume that the finest grid has 2^M elements. A single Gauss–Seidel iteration will involve 2^{2M} (or $O(N^2)$) operations since the BE matrices are full. If we perform a V-cycle we would have to perform:

$$\begin{aligned} \mathrm{NOP}_{\mathrm{MG}} &= q\mathfrak{Q}^{2M}\left[1 + \left(\frac{1}{2^2}\right) + \left(\frac{1}{2^2}\right)^2 + \cdots + \left(\frac{1}{2^2}\right)^M\right] \\ &= \frac{4}{3} q\mathfrak{Q}^{2M}\left[1 - \left(\frac{1}{2^2}\right)^{M+1}\right] \\ &< \frac{4}{3} q\mathfrak{Q}^{2M} \end{aligned} \tag{28}$$

Thus we see that there are only marginally more operations required for one MG V-cycle than required for a GS iteration since it is typical that $q \leq 3$. This calculation does not include the overhead involved in injection and interpolation which will be $O(2^M)$ operations.

A similar calculation shows that MG solution only increases the amount of memory required to store all the influence coefficients by a factor of 4/3. This is not particularly important in the special case of the simple model problem (1a) (because of the simple Toeplitz structure of the

discretized equations (2)), but will be significant for problems involving large numbers of interacting 2D cracks that are not collinear.

In (28) we saw that GS and MG iteration both require $O(N^2)$ operations for a single MG step. The crucial property that makes MG so efficient is that the highly desirable error damping characteristic of MG does not depend on N. Thus the number of operations for a MG scheme to achieve a desired accuracy is still $O(N^2)$, whereas the number of operations for GS iteration to achieve the prescribed accuracy is $O(N^3)$. The $O(N^2)$ complexity in each MG step originates from the number of operations required to perform the matrix multiplications in the iterative solution of (2).

Lumping combined with MG: In this section we consider a combination of lumping with MG to achieve a faster iterative algorithm. The essential idea of lumping can be understood in terms of an asymptotic approximation of the integral in (1) over a subinterval $(-a, a) \subset (-b, b)$. For simplicity we have chosen the subinterval to be symmetric but the analysis applies equally well to any subinterval of $(-b, b)$. Let x be a remote receiving point i.e. $|x| \gg a$ then we have the following approximation:

$$\int_{-a}^{a} \frac{U(\xi)}{(x-\xi)^2}\,d\xi = \frac{1}{x^2}\int_{-a}^{a} U(\xi) \sum_{n=0}^{\infty} (n+1)\left(\frac{\xi}{x}\right)^n d\xi \approx \frac{\int_{-a}^{a} U(\xi)\,d\xi}{x^2} + O\left(\frac{1}{x^3}\right)$$

Thus for remote receiving points the integral over the interval $(-a, a)$ can be approximated by the product of the far-field behavior of the kernel $1/x^2$ and the average value of U over the interval. We notice that higher order corrections to this approximation involve the first and higher moments of U on $(-a, a)$. The idea of lumping is to define some cut-off window of width L about any receiving element within which influences are calculated exactly (to the level of discretization) and outside of which the influences are calculated by averaging the solution U and using the far-field kernel. We notice that remoteness is defined relative to a, the width of the averaging region, so that farther from the receiving point the solution can be averaged over larger and larger subregions while maintaining the same level of approximation. This enables one to set up a hierarchy of lumps in which the appropriate lump size is determined by the distance from the receiving point. Since MG has a similar hierarchy of elements over which the solution is averaged, lumping can be built into the MG data structure relatively easily.

We have implemented such a hierarchical lumping scheme within the MG solution algorithm. The scheme approximates influences on a receiving element by averaged influences from coarser grids for all elements further than a prescribed distance L from the receiving element. Within a distance L the influences are calculated directly. An element on level k is lumped in with its neighbor (as was done in Figure 2 when transferring data to a coarser grid) for the purpose of receiving and sending. Averaged influences on the coarser grid $k+1$ are received from lumped elements beyond a distance L on grid $k+1$ but still within a distance L in co-ordinates measured on grid $k+2$. This process is repeated recursively until the whole interval $[-b, b]$ has been spanned. Since there is an overhead in setting up the lumping process it is not efficient if the mesh on which the iteration takes place is already coarse. Thus we choose to perform lumping only on the finest meshes—those on which it will have the greatest effect. Since the lumps are defined to be compatible with the coarser meshes of the MG part of the algorithm, it is possible to use a common data structure and the same influence coefficients.

We now analyze the complexity of the combined MG-lumping MGL algorithm. First we consider a MG mesh (which we shall refer to as the level 0 lump) comprising 2^M elements and assume that there are k coarser levels of lumping associated with this mesh. It is convenient to

consider one of the 2^{M-k} receiving lumps on the kth level (i.e. the coarsest level of lumping). By means of the lumping algorithm described above the approximate influences on all the level 0 elements within the coarsest receiving lump can be evaluated in $2^{M-k} + (2^{k+1} - 3)(2L - 1)$ operations. Since there are 2^{M-k} coarse receiving lumps, a single lumped relaxation requires a total of

$$\mathcal{L}(M, k, L) = 2^{2(M-k)} + (2^{M+1} - 3 \times 2^{M-k})(2L - 1) = O(2^{2(M-k)}, 2^M) \tag{29}$$

operations. Here $O(a, b)$ is used to denote the larger of $O(a)$ and $O(b)$, and we have neglected the $O(2^M)$ operations required to inject the solution from the level zero lump to the coarser lumps. Making use of the function $\mathcal{L}(M, k, L)$ defined in (29), the number of MGL operations can be written in the form:

$$\text{NOP}_{\text{MGL}} = \mathcal{L}(M, k_0, L) + \mathcal{L}(M - 1, k_1, L) + \cdots + \mathcal{L}(0, k_M, L) \tag{30}$$

The number of MGL operations depends on the level k_l to which each multigrid mesh is lumped. One strategy is to lump each MG mesh until all the sending lumps on the kth level fall within a distance L of each receiving lump. In this case $2^{M-l-k_l} = L$ and (30) reduces to

$$\text{NOP}_{\text{MGL}} \leq ML^2 + 2(2^{M+1} - 3L)(2L - 1) = O(2^M) \tag{31a}$$

Thus in theory MGL can reduce the complexity of the iteration scheme to $O(N)$ operations. From a practical point of view, it is difficult to preserve the GS nature of relaxations within the MGL iteration scheme if the lump level k is chosen to be as high as that in (31a). In particular, if $L < 2^k$ then level 0 elements within the same level k receiving lump will be lumped on level 1 or 2 to other level 0 elements. To preserve the GS structure of the algorithm, it will be necessary to inject new level 0 values of the solution to level 1 and 2 as soon as they are known. This complication can be avoided by placing the restriction $k = \min\{\log_2 L, M - \log_2 L\}$ and the operation count becomes:

$$\text{NOP}_{\text{MGL}} = O[2^{2(M-k)}, 2^{M+k}] \tag{31b}$$

3.3. Modelling nonlinear material properties in the MG algorithm

We adapt the full approximation scheme (FAS)[5] to solve the nonlinear equation (1b) introduced in Section 2. We have chosen to use nonlinear GS iteration to perform the smoothing relaxations within the FAS algorithm. Consider the GS iteration scheme when applied to solving $\mathbf{A}_h\mathbf{u}_h + \mathbf{p} = \mathbf{F}(\mathbf{u}_h)$, where $\mathbf{p} = [p, p, \ldots, p]^T$ and $\mathbf{F}(\mathbf{u}_h) = [F(u_{h,-N}), F(u_{h,-N+1}), \ldots, F(u_{h,N})]^T$. This algorithm can be expressed pointwise in the form:

$$\underbrace{\sum_{m \neq n} A_{h,mn} u_{h,n} + p}_{\sigma^{\text{ext}}} + A_{h,mm} u_{h,m} = F(u_{h,m}) \tag{32}$$

Here $\sum_{m \neq n} A_{h,mn} u_{h,n}$ represents the stresses on the mth element from all the other elements in the crack (i.e. the combined influences $\mathbf{A}^+ + \mathbf{A}^-$). Adding these external influences on the mth element to the ambient stress we obtain the external stress denoted by σ^{ext}. The nonlinear GS iteration process is represented graphically in Figure 4 on the plot for the grid X_h. The diagonal line joining σ_{ext} with the curve $\sigma = F(u)$ has a slope of $A_{h,mm}$. The value of $u_{h,m}$ is obtained by solving the scalar equation $\sigma_{\text{ext}} + A_{h,mm} u_{h,m} = F(u_{h,m})$.

The FAS algorithm we use to solve $\mathbf{A}_h\mathbf{u}_h + \mathbf{p} = \mathbf{F}(\mathbf{u}_h)$ is as follows:

1°. Smooth high frequency errors by performing q_1 nonlinear GS iterations on the finest mesh X_h (see (32))

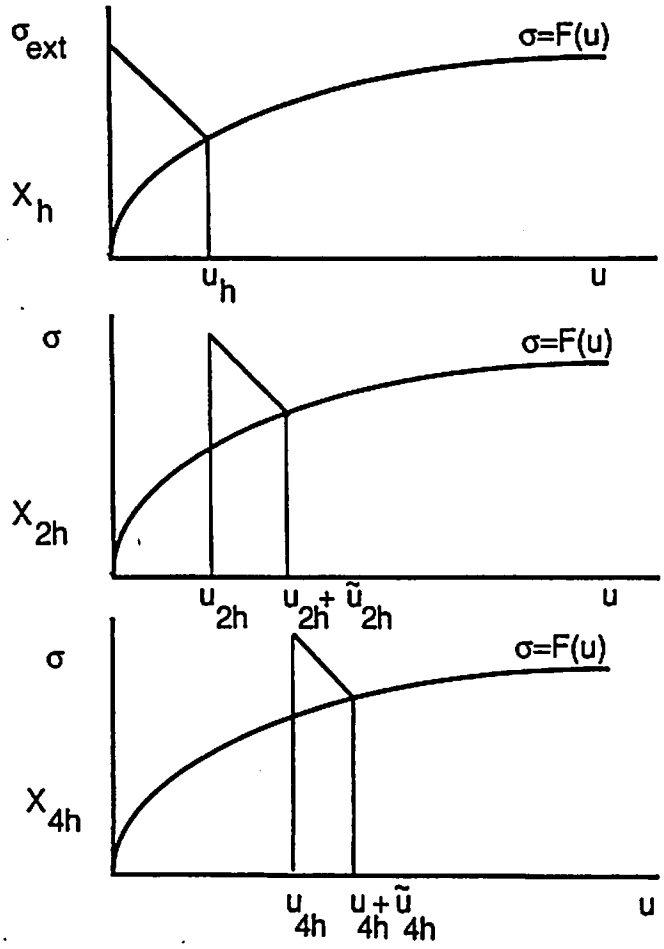

Figure 4. The nonlinear GS iteration process is represented graphically on the plot for the grid X_h. The diagonal line joining σ_{ext} with the curve $\sigma = F(u)$ has a slope of $A_{h,mm}$. The value of $u_{h,m}$ is obtained by solving the scalar equation $\sigma_{ext} + A_{h,mm}u_{h,m} = F(u_{h,m})$. The nonlinear MG solution steps 3^1 and 3^2 (see text) are represented schematically on a stress–displacement diagram for the grids X_{2h} and X_{4h} respectively. A pseudo external stress is constructed from the residual stress on that level and the value of $u_{2h} + \tilde{u}_{2h}$ is calculated. Notice that the total approximate solution $u_{2h} + \tilde{u}_{2h}$ should be injected to the next level to give u_{4h} so that the gradient of the function F(u) in the vicinity of the solution point $u_{4h} + \tilde{u}_{4h}$ on the grid X_{4h} is used

$2^{0,1}$. Inject the residual $\mathbf{r}_h = \mathbf{F}(\mathbf{u}_h) - \mathbf{p} - \mathbf{A}_h\mathbf{u}_h$ and $\mathbf{u}_h$ to the coarser mesh X_{2h}: $\mathbf{r}_{2h} = \mathbf{I}_h^{2h}\mathbf{r}_h$ and $\mathbf{u}_{2h} = \mathbf{I}_h^{2h}\mathbf{u}_h$

3^1. Set $\tilde{\mathbf{u}}_{2h} = 0$ and perform q_1 nonlinear GS iterationson the equation:

$$\mathbf{A}_{2h}\tilde{\mathbf{u}}_{2h} = \mathbf{F}(\mathbf{u}_{2h} + \tilde{\mathbf{u}}_{2h}) - \mathbf{F}(\mathbf{u}_{2h}) + \mathbf{r}_{2h}$$

$2^{1,2}$. Inject the residual $\tilde{\mathbf{r}}_{2h} = \mathbf{F}(\mathbf{u}_{2h} + \tilde{\mathbf{u}}_{2h}) - \mathbf{F}(\mathbf{u}_{2h}) + \mathbf{r}_{2h} - \mathbf{A}_{2h}\tilde{\mathbf{u}}_{2h}$ and the *total* approximate solution $\mathbf{u}_{2h} + \tilde{\mathbf{u}}_{2h}$ to the mesh X_{4h}: $\mathbf{r}_{4h} = \mathbf{I}_{2h}^{4h}\tilde{\mathbf{r}}_{2h}$ and $\mathbf{u}_{4h} = \mathbf{I}_{2h}^{4h}(\mathbf{u}_{2h} + \tilde{\mathbf{u}}_{2h})$

3^2. Set $\tilde{\mathbf{u}}_{4h} = 0$ and perform q_1 nonlinear GS iterations on the equation:

$$\mathbf{A}_{4h}\tilde{\mathbf{u}}_{4h} = \mathbf{F}(\mathbf{u}_{4h} + \tilde{\mathbf{u}}_{4h}) - \mathbf{F}(\mathbf{u}_{4h}) + \mathbf{r}_{4h}$$

. . . . . .

$4^{3,2}$. Interpolate the correction back to the finer grid X_{4h} and update the approximate solution: $\tilde{u}_{4h} \leftarrow \tilde{u}_{4h} + I_{8h}^{4h}\tilde{u}_{8h}$

5^2. Perform q_2 nonlinear GS iterations on the equation:

$$A_{4h}\tilde{u}_{4h} = F(u_{4h} + \tilde{u}_{4h}) - F(u_{4h}) + r_{4h}$$

$4^{2,1}$. Interpolate the correction back to the finer grid X_{2h} and update the approximate solution: $\tilde{u}_{2h} \leftarrow \tilde{u}_{2h} + I_{4h}^{2h}\tilde{u}_{4h}$

5^1. Perform q_2 nonlinear GS iterations on the equation:

$$A_{2h}\tilde{u}_{2h} = F(u_{2h} + \tilde{u}_{2h}) - F(u_{2h}) + r_{2h}$$

$4^{1,0}$. Interpolate the correction back to the finer grid X_h and update the approximate solution: $u_h \leftarrow u_h + I_{2h}^{h}\tilde{u}_{2h}$

5^0. Smooth high frequency errors by relaxing q_2 times on the finest mesh X_h as in 1.

Here the superscripts refer to the mesh level on which the particular step is being performed. If a superscript contains two numbers then information is being transferred from the level of the first number to the level of the second number.

In Figure 4 the solution steps 1^0, 3^1 and 3^2 are represented schematically on a stress–displacement diagram. On the grids X_{2h} and X_{4h} an iteration process similar to that in (32) is performed. A pseudo-external stress is constructed from the residual stress on that level and the value of $u_{2h} + \tilde{u}_{2h}$ is calculated. It is important to note in step $2^{1,2}$ that the total approximate solution $u_{2h} + \tilde{u}_{2h}$ should be injected to the next level (and similarly with the higher level injections) in order that the gradient of the function F(u) in the vicinity of the solution point $u_{4h} + \tilde{u}_{4h}$ on the next grid is used. If, for example, u_h were transferred to the coarser grids then the gradient of F(u) used in the calculation of the corrections $\tilde{u}_{2h}, \tilde{u}_{4h}, \ldots$ would be that at the beginning of the iteration step. This would lead to an algorithm that is akin to the modified Newton method in which the stiffness matrix is set at the beginning of a sequence of iterations and updated only periodically. In the algorithm described above the local gradient is updated throughout the iteration process in order to accelerate convergence.

4. NUMERICAL RESULTS

4.1. Linear model problem

We used a number of MG algorithms with a variety of algorithm parameters to solve the linear model problem (2) and measured the required CPU times. The GS technique was used to solve the same problem so that the CPU times could be compared with those for the MG algorithms. All the CPU times quoted here were performed on a Macintosh II computer. The parameters chosen for the model problem were as follows: $2b = 10$ m, $E = 70000{\cdot}0$ MPa, $\nu = 0{\cdot}2$, $p(x) = 30{\cdot}0$ MPa and N, in what follows, will be used to denote the total number of elements in the discretization.

4.1.1. MG and FMG algorithms. In Table 1 we list the CPU times (in seconds) required to solve the BE equations until the maximum residual stress on the elements is less than $\varepsilon = 10^{-7}$. This small tolerance was chosen to allow significant CPU times to be measured for small meshes. All the MG algorithms used to produce the data for this table exploit PWL interpolation.

Direct GS solution required 182 iterations for $N = 32$, 360 iterations for $N = 64$, and 720 iterations for $N = 128$. Therefore, as was predicted in (11) the number of GS iterations required to

achieve the final solution grows in proportion to the number of mesh points N. Since the convergence rate for the MG algorithms does not depend on the mesh size, the number of MG cycles does not change with the mesh size. Therefore, the GS scheme requires $O(N^3)$ operations to solve the equations while all the MG algorithms require $O(N^2)$ operations. This fact is illustrated clearly in Figure 5.

Table I

	Gauss seidel			M.G. V. Cycles			M.G. W. Cycles		F.M.G.		M.G. V + Lumping	
N	Time	G.S. + Lump L = 8		q = 1	q = 2	q = 3	q = 2	q = 3	V, q = 3	W, q = 3	L = 16, q = 2	
32	8	time	error	2	2	2	4	3	1	3	time	error
64	64	–	–	10	8	8	12	10	7	10	–	–
128	504	197	7%	39	32	31	47	40	29	36	19	1·3%
256	3978	840	13%	142	126	140	180	154	110	137	53	3·3%
512	(31824)	3488	21·6%	566	495	558	706	606	438	637	121	6·3%

*The bracketed run times were not actually measured but were extrapolated from the results on smaller meshes.

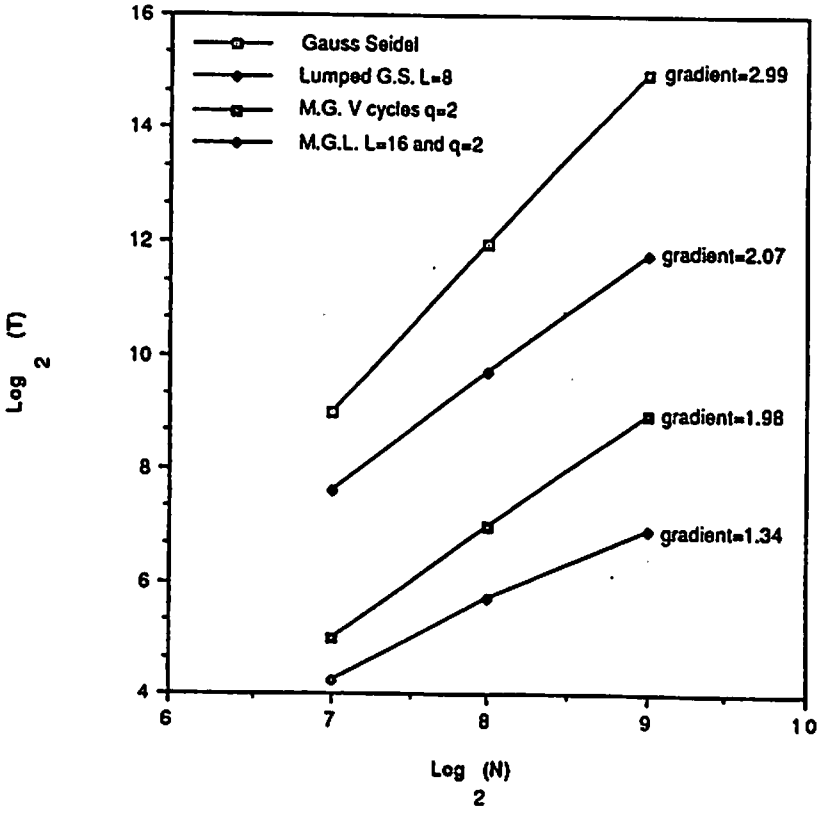

Figure 5. The log of the CPU time T is plotted against the log of the number of elements N (the data is taken form Table 1). It can be seen that the GS scheme requires $O(N^3)$ operations, MG V-cycles require $O(N^2)$ operations, while the MGL algorithms require $O(N^{1+\delta})$, $0 < \delta < 1$ operations. The operation count $O(N)$ is not quite achieved because the restriction $k = \min\{\log_2 L, M - \log_2 L\}$ has been used for ease of implementation, which is consistent with the $O(N^{1+\delta})$ operation count that is observed

For the problem considered and the tolerance ε specified the MG algorithms typically required: 15 cycles for $q = 1$, 9 cycles for $q = 2$, 8 cycles for $q = 3$, and 7 cylces for $q = 4$, to achieve the desired solution. By comparison, the two grid CG correction scheme with PWL interpolation (see Figure 3) requires: 13 cycles for $q = 1$, 6 cycles for $q = 2$, 4 cycles for $q = 3$, and 3 cycles for $q = 4$. We see that in each case, slightly more MG cycles are required than two grid CG correction cycles, because the former does not solve the coarse grid problem exactly in each cycle. However, since each MG cycle only calculates an accurate solution on the coarsest mesh, each MG cycle requires much less time than a CG cycle which calculates an accurate solution on the mesh X_{2h}.

From Table I it can be seen that the MG V-schemes perform better than the corresponding MG W-schemes having the same q values. This pattern is also found for the FMG. V- and W-schemes. For the FMG V-algorithm too little high frequency smoothing, $q = 1$, results in an inefficient algorithm. More high frequency smoothing steps, $q = 2, 3$, result in more efficient algorithms and any more high frequency smoothing steps, $q > 3$, prove to be redundant resulting in a degradation in the efficiency of the algorithm. The most efficient standard MG procedure (i.e. without lumping) considered in Table I is the FMG V-algorithm with $q = 3$. However, it is less than 15 per cent more efficient than the MG V-cycle with $q = 2$. This slight gain in efficiency may not warrant the additional programming effort to implement the FMG algorithm.

The results in Table I indicate that the MG algorithm is an algorithm that provides gains in efficiency that are already noticeable for small meshes (as much as 8 times when N is as small as 32) and that improve as the number of degrees of freedom increase (up to 72 times when $N = 512$). Owing to the weak degradation of GS iteration with increasing N, in Section 3.1 we raised the issue of whether the MG algorithm provides an improvement over GS iteration for the solution of the discretized crack equations (2). We see that the degradation of the convergence of GS iteration does become severe for $N \geq 512$—a size of problem that can be expected if large number of interacting cracks are to be considered.

4.1.2. MGL—MG combined with lumping. In Table I we also list the CPU times (in seconds) required to solve the BE equations (2) by using a GS based lumping algorithm (without 'multigridding') and a combined MG-lumping algorithm. Comparing the lumped GS algorithm to the standard MG algorithms (without lumping) we see that the MG algorithm consistently outperforms the lumped GS algorithm. Although both these acceleration schemes reduce the operation count to $O(N^2)$ operations (see Figure 5) it is the constant that multiplies the asymptotic operation count that determines the practical operating range of the two methods. This constant is roughly six times smaller for the MG algorithm (without lumping) than it is for the GS algorithm with $L = 8$. A drawback of lumping is that, in order to perform the matrix multiplication Ax more efficiently, remote influences are approximated. As an *a postiori* measure of the lumping error we calculated the solution iteratively using lumping and then calculated the maximum residual stress on the elements exactly (i.e. without lumping) These are the errors (expressed as a percentage of the imposed stress $p = 30{\cdot}0$ MPa) listed with the lumped run times in Table I. The lump error increases as the width L of the lump zone decreases. As is to be expected, the lump error also increases as N increases since the physical extent of the direct access zone decreases.

The MGL scheme achieves lower CPU times than any of the MG schemes, which are in turn faster than the lumped GS scheme. The lumped GS scheme and the MGL scheme for the same value of L have precisely the same lump error. When using lumping a trade-off has to be made between the savings in CPU time and an acceptable lump error ~~becomes a must~~. The asymptotic operation count of the MGL scheme is $O(N^{1+\delta})$, $0 < \delta < 1$ as can be seen from Figure 5. Since the restriction $k = \min\{\log_2 L, M - \log_2 L\}$ has been used for ease of implementation, ~~in determining these~~

[illegible], the appropriate theoretical operation count is $O(N^{2[1-(k/M)]}, N^{[1+(k/M)]})$, which is consistent with the order of convergence observed in Figure 5.

4.2. Nonlinear model problem

We consider as a model problem the solution of (1b). The particular situation we consider represents a crack-like excavation in a soft seam for which we propose the following nonlinear stress–strain relation:

$$F(U) = Y(1 - e^{-\alpha U/w})$$

Here Y is the ultimate yield strength of the seam material and w is the width of the seam. The parameter α is chosen so that for small strains

$$\lim_{U \to 0} F'(U) = \frac{\alpha Y}{w} = \frac{E}{w}$$

The following parameter values are used in the model: $E = 70000$ MPa, $\nu = 0{\cdot}2$, $p = 60$ MPa and 120 MPa, $w = 1$ m and $Y = 200$ MPa which implies that $\alpha = 350$. As the domain of the test problem we choose $[-b, b] = [-30\text{ m}, 30\text{ m}]$ and consider the stress free crack to be located in the interval $(-10\text{ m}, 10\text{ m})$ while the remainder of the domain of the problem is assumed to comprise soft seam material. This situation is summarized by defining the right hand side of (1b) that applies to the model problem:

$$F(x, U) = \begin{cases} Y(1 - e^{-\alpha U/w}) & \text{for } x \in (-30\text{ m}, -10\text{ m}) \\ 0 & \text{for } x \in (-10\text{ m}, 10\text{ m}) \\ Y(1 - e^{-\alpha U/w}) & \text{for } x \in (10\text{ m}, 30\text{ m}) \end{cases} \tag{33}$$

The nonlinear MG algorithm described in Section 3.3 is used to solve the above model problem. In order to cater for piecewise changes in the RHS (33) of (1b) within the MG algorithm the domain is divided into segments. Each segment corresponds to a collection of elements that lie in a connected line segment and which have a common RHS and therefore the same constitutive law. Multigridding is performed on each segment separately i.e. coarser grids for one segment are restricted to that segment and are not combined to form coarser grids in common with adjacent segments. This approach avoids the difficulty of having to define 'averaged' constitutive laws which would be necessary if multigridding were performed on the whole domain or across adjacent segment boundaries. Coarser grids from different segments of the same hierarchical level (i.e. the same number of injection steps up from the finest grid) are assumed to communicate with one another. In this paper we assume, for ease of implementation, that all the segments have the same number of elements. This restriction could be removed by communicating between coarser grids for different segments on the same hierarchical level until the grid in one of the segments has reached its coarsest level. Further coarsening on the remaining segments can be performed by introducing pseudo coarser grids for those segments that have reached their highest level of coarsening. A pseudo coarser grid for a given segment will have the same number of unknowns as the coarsest grids for that segment but will have different variable sets with which to continue the coarsening process on the remaining segments. The disadvantage of the segmented MG approach is that it will not be able to achieve the same level of coarseness (and therefore reduction in computational effort) as a global MG strategy defined on the whole grid. The segmented MG approach considered here provides a simple way to model large numbers of noncollinear interacting cracks in 2D or regions having different nonlinear material properties. Since segmented MG will not achieve the same efficiency as the MG algorithm with globally defined grids

discussed in Section 4.1, it is important to determine whether the segmented MG approach provides a practical level of reduction in computational effort.

In order to solve the nonlinear algebraic equations (32) at each mesh-point we use Newton's method. If a simpler form for $F(U)$ is chosen [e.g. $F(U) = a_0 + a_1(U/w) + a_2(U/w)^2$] then it will not necessary to perform Newton iteration because the formula for the roots of a quadratic equation could be used to give the solution exactly at every point. The precision with which to solve the pointwise nonlinear equations (32) at each step of the global iteration process is unclear. However, numerical experiments indicate that decreasing the tolerance to which Newton iterations at each mesh-point are performed does not have a significant effect on the run times.

In Table II the CPU times for the segmented MG solution of the nonlinear test problem are compared with the CPU times required by nonlinear GS iteration. Even in this nonlinear environment the nonlinear MG algorithm exhibits the same $O(N^2)$ operation count characteristic of its linear counterpart seen in Section 4.1. Owing to the segmented nature of the algorithm we do not expect the savings in computational effort to be as large as the globally defined MG algorithms considered in Section 4.1. However, the segmented MG algorithm still provides reduction of computational effort for all the meshes considered, which is up to a factor of 12 for $N = 384$ with $p = 60$ MPa and up to a factor of 15 for $N = 384$ and $p = 120$ MPa. For the larger value of the virgin stress $p = 120$ MPa the CPU times increase due to the increased level of nonlinear strain in the seam.

5. SUMMARY OF RESULTS AND CONCLUSIONS

1. (Performance of linear MG on the crack problem): A Fourier analysis of the convergence characteristics of MG applied to the crack problem was performed. A number of interpolation schemes were considered, PWC, PWL, and PWQ. If the number of smoothing GS relaxations $q = 1$ then the MG algorithm associated with all three interpolation schemes are found to inherit the high frequency damping characteristic of the GS relaxation component. In this case there was essentially no difference between the MG convergence factor for these three interpolation schemes. However, if q is increased the level of high frequency errors drops below the level of the interpolation error and the differences between the different types of interpolation schemes become apparent. Theoretical indications from the Fourier analysis are that MG provides an algorithm for which the number of MG cycles does not depend on the number of degrees freedom in the model. This implies that MG iteration can solve the crack problems in $O(N^2)$ operations, which is the same order of operations as that required to perform a single convolution sum directly. Numerical results indicate that MG can substantially reduce the amount of computational effort for discretizations with as few as $N = 32$ elements. Thus even though GS iteration

Table II

	Non Linear G.S.		Non Linear M.G. q = 3	
N	p = 60 MPa	p = 120 MPa	p = 60 MPa	p = 120 MPa
3 × 16 = 48	12	18	5	8
3 × 32 = 96	93	136	24	32
3 × 64 = 192	712	1040	98	131
3 × 128 = 384	5553	8119	464	548
order	N^3	N^3	N^2	N^2

does degrade relatively slowly as N is increased, MG still provides a means of substantially reducing the solution times.

2. (The combination of lumping with MG): The full matrix multiplications required to perform the fine grid relaxations in the MG algorithm are responsible for the $O(N^2)$ operations that characterize the MG algorithm. We therefore considered the possibility of reducing the operation count even further by combining MG with lumping. We demonstrated theoretically that if lumping is performed to the coarsest grid then this combined approach should reduce the operation count to $O(N)$ operations. Numerical results for an algorithm that does not even lump right to the lowest level indicate that an operation count as low as $O(N^{1.3})$ can be achieved. One of the disadvantages of lumping in general is the error introduced by the approximation of the remote influences. The numerical studies performed also enabled GS with lumping to be compared to MG without lumping—both $O(N^2)$ methods. Numerical results indicate that MG without lumping is more efficient. MG has the additional advantage that no approximation has to be made whereas lumping introduces errors that restrict the size of lump zone L and therefore the efficiency of lumping.

3. (Modelling nonlinear material properties in MG): The FAS multigrid approach was adapted to model nonlinear material behavior. In order to cater for distinct segments of different nonlinear material a segmented MG algorithm was developed. Each MG segment corresponds to a collection of elements that lie in a connected line segment and which have a common RHS. The segmented MG approach will provide a simple way to model large number of non-collinear interacting cracks or regions having different nonlinear material properties in two or three dimensions. A segmented MG algorithm will not achieve the same efficiency as a MG algorithm with globally defined grids. However, the numerical results indicate that the $O(N^2)$ characteristic of the linear MG algorithm is also exhibited by the segmented MG algorithm when it is applied to a nonlinear problem. The segmented MG algorithm achieves a reduction of computational effort for N as low as 48. This implies that the size of problem for which segmented MG can achieve savings is well within the operating range of typical problems that occur in practice. As N is increased the segmented MG algorithm improves in efficiency relative to nonlinear GS iteration—decreasing the CPU times by a factor of 15 for $N = 384$. In addition, as the extent of the nonlinear region increases so too does the performance of the segmented MG algorithm relative to the nonlinear GS technique.

4. (Implications of MG for analyzing large engineering problems): In a two dimensional model of a single excavation interacting with a fault approximately 100 elements are required. On a 1 MIP machine the solution takes approximately 1 hour of CPU time. In order to analyze practical problems in which the dominant geological discontinuities and mining induced fractures are included, about 1000 elements are required. If standard solution methods (taking $O(N^3)$ operations) are used, the solution time for this problem will be 1000 hours of CPU time, which will make the solution of such problems infeasible given the computing capacity available to field-design engineers. Thus the MG algorithm would be an indispensable technique for providing the field-design engineer with the necessary analytical tools to model excavations in the vicinity of large number of geological dicontinuities. The importance of the results in this paper is that they povide field-design engineers a theoretical basis for making appropriate design decisions when developing general purpose MG codes. For example, the MG convergence analysis demonstrates that there is little advantage in implementing more than linear interpolation operators. We have also demonstrated that it is possible to combine MG with lumping to achieve an $O(N)$ solution algorithm. We have established a technique to implement nonlinear constitutive laws within the framework of MG and demonstrated the effectiveness of segmented MG. All these techniques can be used in developing general purpose solvers for fractured rock masses.

ACKNOWLEDGEMENTS

The author wishes to acknowledge the support of the Chamber of Mines of South Africa and the National Science and Engineering Research Council of Canada.

REFERENCES

1. P. K. Banerjee and R. Butterfield, *Boundary Element Methods in Engineering Science*, McGraw-Hill, Maidenhead, 1981.
2. S. L. Crouch, 'Analysis of stresses and displacements around underground excavations: an application of the displacement discontinuity method', *Geomechanics report* to the National Science Foundation, University of Minnesota, Minneapolis, 1976.
3. A. P. Peirce, 'The applicability of the nonlinear boundary element method in the modelling of mining excavations', *M.Sc. Thesis*, University of the Witwatersrand, South Africa, 1983.
4. A. Brandt, 'Multi-level adaptive solutions to boundary value problems', *Math. Comp.*, 31, p. 333, (1977).
5. K. Stuben and U.Trottenberg, 'Multigrid methods: Fundamental algorithms, model problem analysis and applications', in *Multigrid Methods, Proc. Conference* held at Koln-Porz, Nov. 1981, W. Hackbusch and U. Trottenberg, *Lecture Notes in Mathematics*, 960, Springer, Berlin, pp. 1–176, 1982.
6. W. L. Briggs, 'A multigrid tutorial, *SIAM*, Philadelphia', 1987.
7. S. F. McCormick ed, 'Multigrid methods', *SIAM*, Philadelphia, 1987.
8. R. P. Plewman, F. H. Deist and W. D. Ortlepp, 'The development and application of a digital computer method for the solution of strata control problems', *J. S. Afr. Inst. Min. Metall.*, 70, p. 214, 1969.
9. A Stewart, 'An application of the fast Fourier transform in numerical elasticity', *M.Sc. Thesis*, University of the Witwatersrand, South Africa, 1979.
10. H. Schippers, 'Application of multigrid methods for integral equations to two problems from fluid dynamics', *J. Comput. Phys.*, 48, p. 141 (1982).
11. B. Oksam and J. M. J. Fray, 'General relaxation schemes in multigrid algorithms for higher-order singularity methods', *J. Comput. Phys.*, 48, p. 423 (1982).
12. M. D. G. Salamon, 'Elastic analysis of displacements and stresses induced by mining of seam or reef deposits. Parts I–IV', *J. S. Afr. Inst. Min. Metall.*, 63–65, (1964).
13. D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd edn, Addison-Wesley, Reading, 1984.
14. I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series and Products*, Academic Press, Orlando, 1980.
15. M. J. Lighthill, *An Introduction to Fourier Analysis and Generalized Functions*, Cambridge University Press, 1980.