A Quasi-Algebraic Multigrid Approach to Fracture Problems Based on Extended Finite Element Methods

B. Hiriyur[a], R.S. Tuminaro[b], H. Waisman[a], E.G. Boman[c], D. Keyes[d]

[a] *Dept. of Civil Engineering & Engineering Mechanics, Columbia University, New York, NY 10027*

[b] *Sandia National Laboratories, Livermore, CA 94551*

[c] *Sandia National Laboratories, Albuquerque, NM 87185*

[d] *Dept. of Applied Physics & Applied Mathematics, Columbia University, New York, NY 10027*

Abstract

The modeling of discontinuities arising from fracture of materials poses a number of significant computational challenges. The extended finite element method provides an attractive alternative to standard finite elements in that they do not require fine spatial resolution in the vicinity of discontinuities nor do they require repeated re-meshing to properly address propagation of cracks. They do, however, give rise to linear systems requiring special care within an iterative solver method. An algebraic multigrid method is proposed that is suitable for the linear systems associated with modeling fracture via extended finite elements. The new method follows naturally from an energy minimizing algebraic multigrid framework. The key idea is the modification of the prolongator sparsity pattern to prevent interpolation across cracks. This is accomplished by accessing the standard levelset functions used during the discretization process. Numerical experiments illustrate that the resulting method converges in a fashion that is relatively insensitive to mesh resolution and to the number of cracks or their location.

1. Introduction

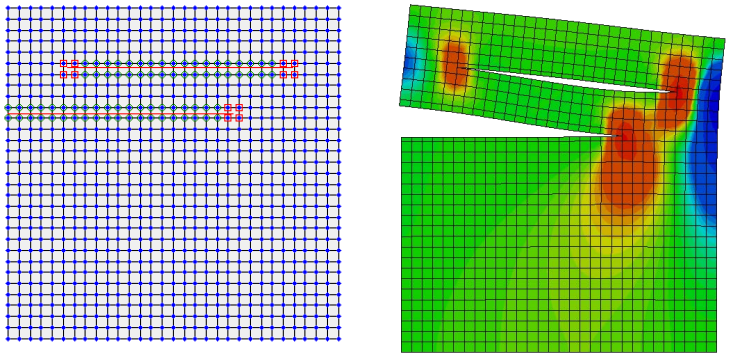
Numerical methods in mechanics often require modeling of discontinuities to obtain an accurate representation of the response. In solid mechanics, strong discontinuities in continuum fields are generally associated with fracture of structures. The classical finite element approach to modeling fracture involves meshes that conform to the internal geometric features causing the discontinuities. As the discontinuities evolve (e.g. propogation of cracks), the finite element meshes have to be regenerated to conform with the new internal boundaries. XFEM offers a computationally efficient alternative by working on a mesh that is independent of the internal boundaries. The discontinuities are modeled through special enrichment functions that have associated with them additional degrees of freedom and have a local support near the crack.

The linear system associated with an XFEM discretization includes unique features that pose challenges for a multigrid linear solver. One obvious complication is that the number of degrees-of-freedom at each node now depends on how many enrichment functions influence that node. This variability in the number of degrees-of-freedom per node is not addressed by most algebraic multigrid (AMG) algorithms or existing codes. While this is somewhat problematic, a greater difficulty is associated with the representation of discontinuities on coarser levels. Intuitively, the value of an XFEM discretization comes from its ability to address discontinuities. If an algebraic multigrid method does not preserve this ability on coarser levels (i.e. properly capture discontinuities), then it can be expected that the corresponding convergence rates will be poor. Standard algebraic multigrid methods are structured to accurately reproduce smooth modes such as constants or rigid body modes on coarse levels. Unfortunately, the presence of discontinuities introduces additional modes that must be accurately represented to maintain a rapid convergence rate. This, however, requires rethinking how prolongators are generated within an algebraic multigrid scheme. One final difficulty is that enrichment functions at crack tips are fairly non-standard. They are designed to capture singularities

which are essential to modeling the appropriate physics, but they do introduce a nontrivial degree of ill-conditioning into the linear system.

This paper proposes a new algebraic multigrid method suitable for the discrete linear systems associated with XFEM. It is motivated by an examination of the Schur complement formulated by eliminating degrees-of-freedom associated with enrichment functions, though the new method does not require formation of a Schur complement. A key idea which follows from a Schur complement perspective is that the prolongator sparsity pattern should be modified to prevent interpolation across cracks. This is accomplished within our algorithm by accessing the standard levelset functions used during the discretization process. This also allows the method to accurately represent discontinuities on coarse meshes, although the resulting method is no longer purely algebraic. Numerical experiments illustrate that the algorithm converges in a fashion that is relatively insensitive to the mesh resolution and to the number of cracks or their location.

1.1. XFEM



(a) (b)

Figure 1: 2D Plane-stress structure subjected to fixity of bottom edge and uniform traction on the top edge (a) XFEM Mesh (b) Computed σ_{yy} stress

The key idea of XFEM is to use a standard finite element mesh that is independent of internal boundaries. The discontinuities (along the crack interface) and singularities (near the crack tip) are instead captured through an "enriched" space of basis functions that have local support near a crack and satisfy a partition of unity. The use of an enriched space of basis functions alleviates the need for remeshing the domain in the case of propagating cracks. The discrete trial function for the displacement field in XFEM takes the following form:

$$u^h(\mathbf{x}) \quad = \quad \sum_{I=1}^{n} N_I(\mathbf{x}) u_I + \sum_{i=1}^{n_h} N_{I_i}(\mathbf{x}) H(\mathbf{x}) a_{I_i} + \sum_{i=1}^{n_f} N_{\hat{I}_i}(\mathbf{x}) \sum_{J=1}^{n_J} F_J(\mathbf{x}) b_{\hat{I}_i J} \tag{1.1}$$

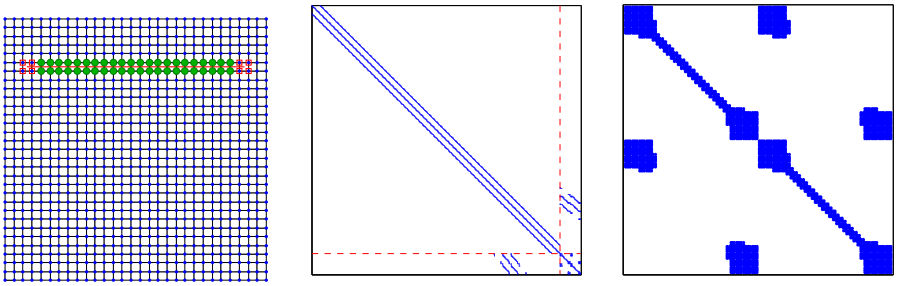
where n is the total number of nodes, n_h is the number of nodes which define at least one element bisected by a crack, n_f is the number of nodes which define elements associated with crack tips, n_J is the number of tip singularity functions, $N_I(\mathbf{x})$ are standard nodal basis functions, I_i gives the index of the i^{th} node associated with an element bisected by a crack, and $\hat{I}_i$ gives the index of the i^{th} node associated with elements containing tips. $H(\mathbf{x})$ and $F_J(\mathbf{x})$ are the enrichment functions for crack interface and the crack-tip respectively and are enveloped by the standard shape functions $N_I(\mathbf{x})$ to ensure a partition of unity. The strain displacement relations are obtained by applying the symmetric gradient operator on the shape function matrix that includes the enrichment functions for the elements through which the cracks pass: $\mathbf{B}^e_{enr} = \nabla_{sym} \mathbf{N}^e_{enr}$. The XFEM element stiffness matrix is obtained from the following:

$$\mathbf{A}_e = \int_{\Omega_e} \left(\mathbf{B}^e_{enr}\right)^T \mathbf{D} \mathbf{B}^e_{enr} d\Omega_e \tag{1.2}$$

Because of the presence of discontinuous fields, the numerical quadrature rule for the integration in Eq. (1.2) has to be performed carefully. The most common method is to subdivide the enriched elements along the lines of discontinuity and using higher order Gauss quadrature for these sub-elements. Once all the element stiffness matrices are obtained, they are assembled to form the XFEM global linear system as follows:

$$\begin{bmatrix} A_{rr} & A_{rx} \\ A_{xr} & A_{xx} \end{bmatrix} \begin{bmatrix} u_r \\ u_x \end{bmatrix} = \begin{bmatrix} \tilde{f}_r \\ \tilde{f}_x \end{bmatrix} \tag{1.3}$$

In Eq. (1.3), all the enriched degrees of freedom a_{I_i} and $b_{\hat{I}_i}$ are grouped together at the end in u_x and the corresponding stiffness coefficients form the block A_{xx}. The sparsity patterns of A and A_{xx} are shown in Fig. (2) for a domain with one interior crack. Further information about XFEM may be obtained in the following references: [1, 6]



(a) (b) (c)

Figure 2: XFEM Stiffness Matrix (a) Mesh (b) Sparsity pattern of A (c) Sparsity pattern of A_{xx}

1.2. Energy Minimization AMG

It is well known that multigrid is effective for solving many discrete partial differential equations, see e.g, [3, 8]. The key is to capture errors by utilizing multiple resolutions. High energy (or oscillatory) components are reduced through a simple smoothing procedure, while low energy (or smooth) components are tackled using an auxiliary lower resolution version. The scheme is then applied recursively on the next coarser level. In standard multigrid, this is accomplished by generating a hierarchy of meshes, $\mathcal{G}^{[k]}$, corresponding to differing resolutions where the superscript $^{[k]}$ indicates the grid level. Grid transfers are defined to move data between meshes and discretizations are constructed on all meshes by either re-utilizing the discretization procedure or Galerkin projection:

$$A^{[k+1]} = \mathcal{R}^{[k]} A^{[k]} P^{[k]} \tag{1.4}$$

where $P^{[k]}$ interpolates from $\mathcal{G}^{[k+1]}$ to $\mathcal{G}^{[k]}$ and $A^{[k]}$ is the discretization on $\mathcal{G}^{[k]}$. To complete the specification, relaxation $\mathcal{R}^{[k]}$ and the $P^{[k]}$ must be defined. The key to fast convergence is their complementary nature; errors not reduced by $\mathcal{R}^{[k]}$ must be well interpolated by $P^{[k]}$. In algebraic multigrid (AMG), the $\mathcal{G}^{[k]}$ are not supplied and instead a notion of mesh is developed from matrix data. This mesh is coarsened via graph algorithms, the $P^{[k]}$ are deduced from algebraic principles, and used in conjunction with (1.4) to recursively generate a hierarchy. We follow an *energy minimizing* philosophy to generate the $P^{[k]}$. The details of this particular philosophy are not critical for this paper. It is chosen here because nonstandard prolongator sparsity patterns are easily incorporated. A brief description is given where superscripts indicating level are omitted to simplify the presentation; more information can be found in [2, 4, 5, 7, 9, 10]. The basic principle is that interpolation is chosen to minimize the energy norm of prolongator basis functions subject to constraints. That is:

$$P = \text{argmin}_P \sum_j \|P_j\|_A \quad \text{and} \quad PB^C = B, \text{ with } P \in \mathcal{N}, \tag{1.5}$$

where P_j is the j^{th} column of P, B and B^C are column matrices of fine and coarse level near-null-space modes respectively, $P \in \mathcal{N}$ specifies that P's sparsity pattern conforms to a specified form, $\mathcal{N}$, and $\|v\|_A \equiv \sqrt{v^T A v}$ defines energy. B and B^C have m columns so the relation $PB^C = B$ defines a set of m vectors which P must exactly interpolate. In this paper, these vectors are rigid body modes (i.e., $m = 3$ in two dimensions). The main idea is that interpolation complements relaxation. The more poorly an error mode is reduced by fine level relaxation, the better it should be captured by interpolation so that it can be reduced on a coarse level. Standard relaxation is ineffective for modes corresponding to eigenvectors with small eigenvalues. This is most easily seen for Jacobi relaxation. For unattached domains (no Dirichlet boundary conditions) the rigid body modes correspond to eigenvectors with zero eigenvalue. Thus, the constraint $PB^C = B$ guarantees that these lowest energy modes are exactly interpolated. It is also important that other modes corresponding to small eigenvalues be well interpolated with interpolation accuracy of an eigenvector proportional to the reciprocal of its associated eigenvalue. Energy minimization in conjunction with the constraint $PB^C = B$ accomplishes this.

2. A New AMG

2.1. XFEM Schur Complements

The special degrees-of-freedom in XFEM pose many challenges for algebraic multigrid. One difficulty is the treatment of low energy modes. Since constants and more generally rigid body modes correspond to low energy modes, standard relaxation is not effective on these components and therefore they must be accurately represented on a coarse level. In the XFEM context, the cracks may introduce additional low-energy modes (e.g. cracks completely traversing the domain may lead to an unattached structure with independent rigid body modes). All of these low energy vectors must be well-represented in the range of interpolation, but unfortunately the new vectors are discontinuous at the crack interface. It is unlikely that a standard AMG prolongator will preserve this discontinuity when interpolating between coarse and fine meshes. Therefore these discontinuous modes are not well-addressed by either standard relaxation or the coarse level correction. This implies that coarse values on one side of a crack should not be used to interpolate to fine values on the other side of a crack when these values are far from a crack tip. Another AMG difficulty arises from having a variable number of degrees-of-freedom (dofs) at each fine level node. This variability thwarts capabilities within most multigrid codes for addressing PDE systems. In particular, PDE systems are often tackled by grouping dofs at each node. Most AMG codes, however, assume that the number of dofs per node is constant and so in this case blocking cannot be utilized. Blocking information can be ignored, but this is akin to treating a PDE system as a scalar PDE. Specifically, blocking is often used to guarantee exact interpolation of constant functions for each degree-of-freedom at coarse nodes and so this is lost if blocking is not done. Finally, in the specific case of energy minimization, interpolation of constants is accomplished by the rigid body mode constraints. However, it is not clear how B should be defined for degrees-of-freedom associated with enrichment functions.

In view of all these challenges, one possible remedy is to algebraically eliminate special degrees-of-freedom from the system. That is to instead consider the Schur complement:

$$Su_r = \tilde{f}_r - A_{rx}A_{xx}^{-1}\tilde{f}_x \qquad S = A_{rr} - A_{rx}A_{xx}^{-1}A_{xr}. \tag{2.1}$$

Removal of special degrees-of-freedom effectively renders the problem amenable to algebraic multigrid methods. Table 1 illustrates convergence data for fairly standard 2-level AMG algorithms applied directly to the full system as well as applied to the Schur complement. Two different AMG versions are shown for the full system. The variable block AMG maintains the relationship between dofs and nodes while scalar AMG does not. The Schur complement AMG uses constant block sizes of two. In each case, one Krylov-like iteration [7] is used to improve the initial prolongator and generate a final prolongator that approximates(1.5). For scalar AMG and variable block AMG, $B(B_C)$ are taken to be zero for enriched degrees-of-freedom. All AMG methods drop weak connections during prolongator construction using $\varepsilon = 10^{-2}$. This dropping does not have a big effect on scalar or variable block AMG, but this dropping effectively removes coupling across cracks within the Schur AMG method. In addition to dropping small entries, Schur complement nonzeros that are

not present in A_{rr} are removed when constructing $\hat{S}$. Ensuring that $\hat{S}$ does not have large stencils avoids large aggregates and sparsity patterns with too much overlap.

In all experiments, a direct solver is always used for coarse level relaxation while on finer levels, one symmetric block Gauss-Seidel defines the $\mathcal{R}^{[k]}$ where blocks are associated with grouping dofs at nodes for the Schur AMG and block AMG. The block size is one for scalar AMG. "-" indicates that preconditioned conjugate gradient did not achieve a residual reduction of 10^{-8} in 200 iterations. AMG W cycles are used for all the results presented here. It is clear from Table 1 that AMG applied directly to the full system is problematic while AMG applied to the Schur complement is ideal. The key reason for this is that the Schur complement method does not coarsen enrichment unknowns and that Schur complement prolongators do not interpolate across cracks. This means that the interpolation respects the XFEM discontinuities.

Mesh	Scalar AMG	Variable Block AMG	AMG on Schur Complement
30×30	180	89	10
60×60	-	103	11
90×90	-	114	11
120×120	-	126	12

Table 1: PCG iterations for different AMG approaches on a six crack problem.

The remainder of this paper focuses on an algebraic multigrid method which avoids the explicit computation of the Schur complement. This is particularly important for large parallel three dimensional calculations with many cracks. It should be noted that Schur complement multilevel methods have been considered in domain decomposition where unknowns within subdomain interiors are removed from the linear system. In this case, spectral equivalence with the original operator is often used to avoid an explicit Schur complement, though the Schur complement can be efficiently computed for the Galerkin projection within a domain decomposition scheme.

2.2. Implicit Schur Complements

While avoiding the explicit computation of S, we want an algorithm with similar convergence characteristics to that of applying AMG to the Schur complement. To this end, we begin with three Lemmas relating AMG applied to the Schur complement and AMG applied to the 2×2 block linear system.

Lemma 2.1. *Schur complement/projection commutativity. Consider* (1.3) *and its associated Schur complement given by* (2.1). *Let $\mathcal{P}$ be a prolongation operator used in a Schur complement multigrid algorithm and let*

$$P = \begin{bmatrix} \mathcal{P} & 0 \\ 0 & I \end{bmatrix} \tag{2.2}$$

be a prolongation operator used to project the full system. Then, the projected Schur complement,

$$S_H = \mathcal{P}^T S \mathcal{P},$$

and the Schur complement of the projected full system are equivalent.

Proof. Follows by comparison of the two coarse level Schur complements which in both cases is given by $\mathcal{P}^T A_{rr} \mathcal{P} - \mathcal{P}^T A_{rx} A_{xx}^{-1} A_{xr} \mathcal{P}$. □

Remark 2.2. The above two-level result generalizes to a multilevel setting using recursive arguments.

It is important to notice the computational differences between projecting the Schur complement versus taking the Schur complement of a projected system. In both cases, the following quantity is computed:

$$\mathcal{P}^T A_{rx} A_{xx}^{-1} A_{xr} \mathcal{P}.$$

The difference is in the order in which operations occur. Projection of the full system effectively performs the product $A_{xr}\mathcal{P}$. The result has significantly fewer columns and so application of A_{xx}^{-1} is less expensive than first forming $A_{xx}^{-1}A_{xr}$.

Lemma 2.3. *Assume the full linear system* (1.3) *is given along with* $S = A_{rr} - A_{rx}A_{xx}^{-1}A_{xr}$ *and a reduced right hand side* $f_r = \tilde{f}_r - A_{rx}A_{xx}^{-1}\tilde{f}_x$. *Consider two relaxation procedures corresponding to the reduced system and the full system:*

Reduced Relaxation

$$u_r \leftarrow u_r + M_{rr}^{-1} r_r$$

Full System Relaxation

$$\begin{aligned} (a) \quad \tilde{u}_x &\leftarrow A_{xx}^{-1}(\tilde{f}_x - A_{xr}\tilde{u}_r) \\ (b) \quad \begin{bmatrix} \tilde{u}_r \\ \tilde{u}_x \end{bmatrix} &\leftarrow \begin{bmatrix} \tilde{u}_r \\ \tilde{u}_x \end{bmatrix} + \begin{bmatrix} M_{rr} & 0 \\ A_{xr} & A_{xx} \end{bmatrix}^{-1} \begin{bmatrix} \tilde{r}_r \\ \tilde{r}_x \end{bmatrix} \end{aligned}$$

where r_r *denotes the residual associated with the Schur system while* $\tilde{r}_r$ *and* $\tilde{r}_x$ *give residual components for the full system.* M_{rr} *is non-singular and typically approximates* A_{rr}. *Further, assume that both procedures begin with identical initial guesses for the regular degrees-of-freedom (i.e.,* $u_r = \tilde{u}_r$ *on entry). Then, the final* u_r *and* $\tilde{u}_r$ *produced by Reduced Relaxation and by Full System Relaxation are equivalent and* $\tilde{r}_x = 0$ *after completion of both step* (a) *and* (b) *of Full System Relaxation.*

Proof. Verification of $\tilde{r}_x = 0$ after step (a) follows trivially from the definition of $\tilde{r}_x$ and the assigned value to $\tilde{u}_x$. Verification after step (b) is seen by transforming to a residual update. This is done by pre-multiplication of step (b) with the 2×2 block system and subtracting this from the right hand side. This results in:

$$\begin{bmatrix} \tilde{r}_r \\ \tilde{r}_x \end{bmatrix} \leftarrow \begin{bmatrix} \tilde{r}_r \\ \tilde{r}_x \end{bmatrix} - \begin{bmatrix} A_{rr} & A_{rx} \\ A_{xr} & A_{xx} \end{bmatrix} \begin{bmatrix} M_{rr} & 0 \\ A_{xr} & A_{xx} \end{bmatrix}^{-1} \begin{bmatrix} \tilde{r}_r \\ \tilde{r}_x \end{bmatrix}$$

where we note that the second matrix is block triangular so the diagonal blocks can simply be inverted. It follows that $\tilde{r}_x = 0$. Finally, substitution of $\tilde{u}_x = A_{xx}^{-1}(\tilde{f}_x - A_{xr}\tilde{u}_r)$ into $\tilde{r}_r = \tilde{f}_r - A_{rr}\tilde{u}_r - A_{rx}\tilde{u}_x$ reveals that if $\tilde{u}_r = u_r$ on entry, then $\tilde{r}_r = r_r$ after step (a) of Full System Relaxation. From this it follows that $u_r = \tilde{u}_r$ at termination. □

Lemma 2.4. *Assume the full linear system* (1.3) *is given along with a reduced right hand side* $f_r = \tilde{f}_r - A_{rx}A_{xx}^{-1}\tilde{f}_x$ *and* $S = A_{rr} - A_{rx}A_{xx}^{-1}A_{xr}$. *Consider two m-level multigrid V cycle procedures which are fully defined by relaxation and grid transfers; one applied to the Schur complement and the other applied to the full system.*

Reduced Multigrid

- $\mathcal{P}^{[k]}$ *prolongates from level* $k+1$ *to level* k
- *relaxation consists of* $\nu > 0$ *pre- and post-sweeps given by*

$$u_r^{[k]} \leftarrow u_r^{[k]} + (M_{rr}^{[k]})^{-1} r_r^{[k]}$$

Full System Multigrid

- *prolongation from level* $k+1$ *to level* k *is accomplished via*

$$\begin{bmatrix} \mathcal{P}^{[k]} & 0 \\ 0 & I \end{bmatrix}.$$

- *relaxation consists of $\nu > 0$ pre- and post-sweeps given by*

$$(a) \quad \tilde{u}_x^{[k]} \leftarrow (A_{xx}^{[k]})^{-1}(\tilde{f}_x^{[k]} - A_{xr}^{[k]}\tilde{u}_r^{[k]})$$

$$(b) \quad \begin{bmatrix} \tilde{u}_r^{[k]} \\ \tilde{u}_x^{[k]} \end{bmatrix} \leftarrow \begin{bmatrix} \tilde{u}_r^{[k]} \\ \tilde{u}_x^{[k]} \end{bmatrix} + \begin{bmatrix} M_{rr}^{[k]} & 0 \\ A_{xr}^{[k]} & A_{xx}^{[k]} \end{bmatrix}^{-1} \begin{bmatrix} \tilde{r}_r^{[k]} \\ \tilde{r}_x^{[k]} \end{bmatrix}$$

where the superscript $^{[k]}$ again denotes grid level. Then, under the additional assumption that both procedures begin with the same initial guess for regular degrees-of-freedom (i.e. $u_r^{[1]} = \tilde{u}_r^{[1]}$ on entry), the two multigrid procedures produce identical solutions for the regular degrees-of-freedom at termination.

The proof of the last lemma is not provided explicitly here. The Lemma obviously holds for $m = 1$ as Lemma 2.3 directly applies. For higher levels, the proof follows by induction. The last Lemma establishes that a multigrid cycle can be constructed for the 2×2 block system which is completely equivalent to multigrid applied directly to the explicit Schur complement. To do this without forming an explicit Schur complement, however, requires that $M_{rr}^{[k]}$ and $\mathcal{P}^{[k]}$ be defined without relying on $S^{[k]}$. Additionally, an efficient procedure for solving systems of the form $A_{xx}^{[k]}w = b$ is needed within relaxation. However, $M_{rr}^{[k]}$ and $\mathcal{P}^{[k]}$ employ a Schur complement matrix in their definition and so it is not practical to make an entirely equivalent multigrid cycle based on the 2×2 block system. Instead, an approximation is pursued.

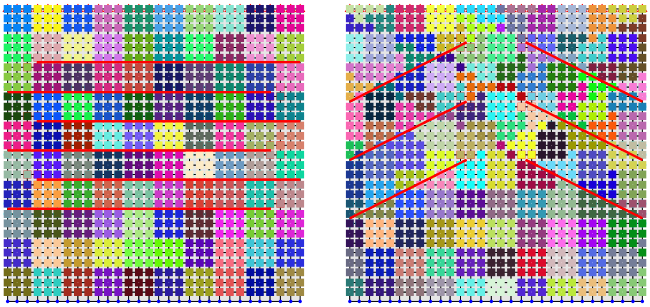
2.3. Algorithm Details

The new AMG algorithm uses a hybrid prolongator given by (2.2). $\mathcal{P}$ is defined by applying a standard algebraic multigrid algorithm to a modified form of $\hat{A}_{rr}$ which is intended to mimic the Schur complement by dropping entries associated with crack crossings. This is done with the help of the function `Drop` detailed in Alg. 1. Sample aggregates formed using this algorithm are shown in Fig. 3. As ϕ and ψ require coordinates, these are projected using a restriction operator based on averaging coordinate values at all nodes within an aggregate to define the associated coordinate location on the coarse level.

	Schur		Hybrid		
	$P(S, \hat{S})$	$P(A_{rr}, \hat{S})$	$P(A_{rr}, \hat{A}_{rr})$	$P(A_{rr}, \hat{A}_{rr})$	$P(A_{rr}, \hat{A}_{rr})$
Mesh	M_{rr} = GS on S		R_r =GS on S	R_r =GS on A_{rr}	R_r =GS on A_{rr}
			R_x =Direct Solve	R_x =Direct Solve	R_r =GS on A_{xx}
30×30	10	11	14	18	18
60×60	11	11	13	17	17
90×90	11	11	13	17	17
120×120	12	11	13	17	17

Table 2: Schur Complement Approximations

Table 2 assesses the individual affects of these different approximations on the convergence rate of the overall method. In particular, CG with AMG preconditioning is applied to a problem with six cracks. $P(C, G)$ indicates that the prolongator was generated via energy minimization using the matrix C in its definition of energy and using G to generate the sparsity pattern. In the table, $\hat{A}_{rr}$ refers to the modified form with crack crossings removed. The second and third columns give preconditioned AMG iterations applied to the true Schur complement with relaxation as defined in Lemma 2.3. The remaining columns refer to the hybrid prolongator with different forms of Hybrid Smoother relaxation. Specifically, the fourth column is associated with the Hybrid Smoother in Lemma 2.3 while the rightmost columns use either symmetric Gauss-Seidel (indicated by GS) or a direct solver to define relaxation components. AMG W cycles are used for all the results presented here. The reader should note that the third and fourth columns are nearly identical. When $\hat{A}_{rr}$ is used instead of $\hat{S}$ to generate sparsity patterns for the Schur method, this does in fact give identical convergence to the hybrid method in column four and these two variants correspond to the W cycle version



(a) (b)

Figure 3: Sample aggregates. Each grid point is colored corresponding to the aggregate that contains it. The red lines represent cracks.

of Lemma 2.4. Overall, it is clear that there is a very modest increase in the number of iterations reading from left to right corresponding to different approximations, and that the rightmost column still appears to be nearly mesh independent.

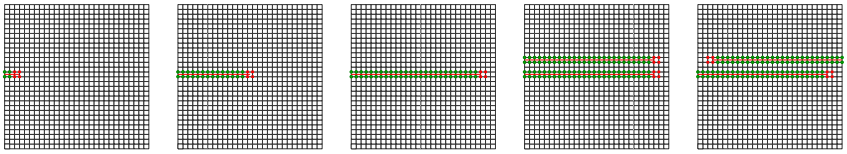
Algorithm 1 $\hat{A}_{rr} \leftarrow Drop(A_{rr}, BlkSize, coords, tol, \phi, \psi)$

Require: matrix A_{rr} with constant blocksize of $BlkSize$.
Require: coordinate locations $coords$ of nodes
Require: drop tolerance tol for removing small entries
Require: levelset functions ϕ, ψ representing cracks
  $\hat{A}_{rr} \leftarrow StandardDrop(A_{rr}, tol)$
  for $i = 1$ to $dimension(A_{rr})$ **do**
    $C_i \leftarrow \{j \mid [\hat{A}_{rr}]_{ij} \neq 0\}$
    $(x_1, y_1) \leftarrow (coords(i/BlkSize, 1), coords(i/BlkSize, 2))$
    for $j \in C_i$ **do**
      $(x_2, y_2) \leftarrow (coords(j/BlkSize, 1), coords(j/BlkSize, 2))$
      for each crack c **do**
        if $\psi^c(x_1, y_1)\psi^c(x_2, y_2) < 0$ AND $\phi^c(x_1, y_1) \leq 0$ **then**
          $[\hat{A}_{rr}]_{ij} \leftarrow 0$
          $[\hat{A}_{rr}]_{ji} \leftarrow 0$
        end if
      end for
    end for
  end for

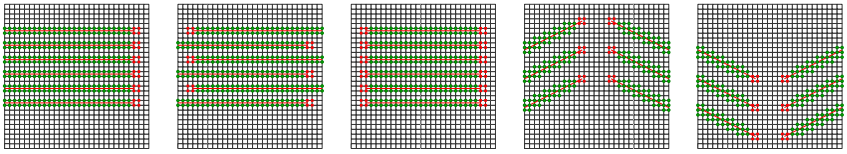
3. Numerical Results

Some numerical results in terms of PCG iterations (to achieve a residual reduction of 10^{-8}) using three different AMG preconditioners are presented for the select crack configurations in Fig. 4. Four different mesh densities are employed and the crack configurations vary in terms of the number, type and orientation of cracks. In Table 3, VBlk AMG corresponds to relatively standard algebraic multigrid where all dofs at a common geometric position are blocked together. These blocks are used when coarsening (or aggregating) as well as during the block symmetric Gauss-Seidel relaxation. The Quasi AMG method is the new method advocated in this paper. It uses the hybrid prolongator (2.2) in conjunction with the specialized dropping procedure given in Algorithm 1. The HybridStandard AMG method also uses the hybrid prolongator of (2.2). However, the key difference is that instead of the specialized dropping procedure which removes crack crossing, standard dropping (with specified tolerance $\varepsilon = 8 \cdot 10^{-2}$) is used. Therefore the first two methods are fully algebraic and only Quasi-AMG method uses the geometric levelset information. The previously discussed

hybrid relaxation scheme is not employed in any of these experiments. This was primarily a theoretical tool that helped establish the equivalence with the Schur complement AMG method. Instead, point symmetric Gauss-Seidel relaxation is employed within the new Quasi AMG method and within the HybridStandard AMG method. Only the VBlk AMG uses the more expensive block symmetric Gauss-Seidel relaxation. For VBlk AMG, $B(B_C)$ are taken to be zero for enriched degrees-of-freedom. A direct solver is always used for coarsest level relaxation. "-" indicates that preconditioned conjugate gradient did not converge in 200 iterations. It is clear from the results that only the Quasi AMG method gives acceptable performance and that the number of iterations required for convergence does not grow.



(a) Case 1a (b) Case 1b (c) Case 1c (d) Case 2a (e) Case 2b



(f) Case 3a (g) Case 3b (h) Case 4 (i) Case 5a (j) Case 5b

Figure 4: Test crack configurations (1a,b,c) Single propagating crack (2a,b) Two edge cracks (3a,b) Six edge cracks (4) Six interior cracks (5a,b) Inclined cracks

Case	VBlk AMG	Hybrid Standard AMG	Quasi AMG	Mesh	Case	VBlk AMG	Hybrid Standard AMG	Quasi AMG
1a	28	13	11	30^2	3a	154	-	16
	29	15	10	60^2		127	-	14
	37	17	12	90^2		-	-	25
	37	19	12	120^2		-	-	21
1b	24	22	11	30^2	3b	-	-	18
	24	29	12	60^2		-	-	21
	36	35	14	90^2		-	-	28
	35	41	13	120^2		-	-	22
1c	31	31	13	30^2	4	116	107	15
	32	43	14	60^2		102	154	21
	47	53	16	90^2		142	190	23
	45	61	15	120^2		151	-	22
2a	64	57	15	30^2	5a	80	76	12
	52	80	14	60^2		91	107	13
	87	98	20	90^2		124	131	15
	92	113	18	120^2		140	151	15
2b	73	59	16	30^2	5b	89	81	16
	72	81	17	60^2		103	116	15
	97	104	21	90^2		134	143	17
	95	122	19	120^2		151	165	16

Table 3: Preconditioned CG iterations

4. Conclusions

A new quasi-algebraic multigrid preconditioner has been proposed to tackle the linear systems arising from XFEM discretizations which pose difficulties for standard AMG. The new method uses levelset information normally available in the discretization process to avoid crack crossings in the prolongator stencils. Additionally, coarsening is only applied to degrees-of-freedom associated with standard nodal basis functions. The resulting preconditioner performs in a way that one expects for algebraic multigrid. In particular, a small number of iterations is required for convergence and the number of iterations does not grow as the mesh is refined. Numerical experiments demonstrate that our quasi-algebraic method is superior to more standard AMG variations not adapted to XFEM.

The lack of coarsening for enrichment degrees-of-freedom is not problematic when a modest percentage of the domain contains cracks. An extension based on coarsening Heaviside degrees-of-freedom will be considered in a future paper for highly cracked materials.

References

[1] T. Belytschko, R. Gracie, and G. Ventura. A review of extended/generalized finite element models for material modeling. *Modeling and Simulation in Materials Science and Engineering*, 17:1–24, 2009.

[2] Achi Brandt. General highly accurate algebraic coarsening. *Electronic Trans. Num. Anal*, 10:1–20, 2000.

[3] William L. Briggs, Van Emden Henson, and Steve McCormick. *A Multigrid Tutorial, Second Edition*. SIAM, Philadelphia, 2000.

[4] Tzanio V. Kolev and Panayot S. Vassilevski. AMG by element agglomeration and constrained energy minimization interpolation. *Numer. Linear Algebra Appl.*, 13(9):771–788, 2006. ISSN 1070-5325. doi: 10.1002/nla.494. URL `http://dx.doi.org/10.1002/nla.494`.

[5] J. Mandel, M. Brezina, and P. Vaněk. Energy optimization of algebraic multigrid bases. *Computing*, 62: 205–228, 1999.

[6] N. Moes, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46:131–150, 1999.

[7] L. N. Olson, J. Schroder, and R. S. Tuminaro. A general interpolation strategy for algebraic multigrid using energy-minimization. *submitted to J. Comp. Phys.*, 2010.

[8] U. Trottenberg, C. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, London, UK, 2001.

[9] W. L. Wan, Tony F. Chan, and Barry Smith. An energy-minimizing interpolation for robust multigrid methods. *SIAM J. Sci. Comput.*, 21(4):1632–1649, 2000. ISSN 1064-8275. doi: http://dx.doi.org/10.1137/S1064827598334277.

[10] Jinchao Xu and Ludmil Zikatanov. On an energy minimizing basis for algebraic multigrid methods. *Computing and Visualization in Science*, 7:121–127(7), October 2004. doi: doi:10.1007/s00791-004-0147-y. URL `http://www.ingentaconnect.com/content/klu/791/2004/00000007/F0020003/art00003`.