Developments in The Extended Finite Element Method & Algebraic
Multigrid for Solid Mechanics Problems Involving Discontinuities

Badri Krishna Jainath Hiriyur

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY
2012
Abstract

Developments in The Extended Finite Element Method & Algebraic Multigrid for Solid Mechanics Problems Involving Discontinuities

Badri Krishna Jainath Hiriyur

In this dissertation, some contributions related to computational modeling and solution of solid mechanics problems involving discontinuities are discussed. The main tool employed for discrete modeling of discontinuities is the extended finite element method and the primary solution method discussed is the algebraic multigrid. The extended finite element method has been shown to be effective for both weak and strong discontinuities. With respect to weak discontinuities, a new approach that couples the extended finite element method with Monte Carlo simulations with the goal of quantifying uncertainty in homogenization of material properties of random microstructures is presented. For accelerated solution of linear systems arising from problems involving cracks, several new methods involving the algebraic multigrid are presented. In the first approach, the Schur complement of the linear system arising from XFEM is used to develop a Hybrid-AMG method such that crack-conforming aggregates are formed. Another alternative approach involves transforming the original linear system into a modified system that is amenable for a direct application of algebraic multigrid. It is shown that if only Heaviside-enrichments are present, a simple transformation based on the phantom-node approach is available, which decouples the linear system along the discontinuities such that crack conforming aggregates are automatically generated via smoother aggregation algebraic multigrid. Various numerical examples are presented to verify the accuracy of the resulting solutions and the convergence properties of the proposed algorithms. The parallel scalability performance of the implementation are also discussed.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contents</td>
<td>i</td>
</tr>
<tr>
<td>List of Tables</td>
<td>v</td>
</tr>
<tr>
<td>List of Figures</td>
<td>vii</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>xi</td>
</tr>
</tbody>
</table>

## I Introduction
1. **Scope-Outline**
   1.1 Introduction 2
   1.2 Outline 2

## II Weak Discontinuities
2. **State-of-the-art**
   2.1 Motivation & Literature Survey 6

## 3 XFEM-Inclusions
   3.1 XFEM for Weak Discontinuities
      3.1.1 Fundamental equations 11
      3.1.2 Enrichment function 15
      3.1.3 Convergence study of XFEM solution for single inclusion 16
      3.1.4 Convergence study of XFEM solution for multiple inclusions 21

## 4 XFEM-MC
   4.1 Monte Carlo approach 25
4.1.1 Random microstructure generation ............................................. 26
4.2 Homogenization ................................................................. 27
  4.2.1 Application and results ...................................................... 27
  4.2.2 Discussion ................................................................. 40
4.3 Concluding remarks ............................................................. 40

III Strong Discontinuities ......................................................... 42

5 Introduction ................................................................. 43
  5.1 Motivation & Literature Survey .............................................. 43
  5.2 XFEM for Strong Discontinuities ............................................ 46
    5.2.1 Governing Equations .................................................. 46
    5.2.2 Weak Form ............................................................ 47
    5.2.3 Levelset Functions .................................................... 49
    5.2.4 XFEM Linear System .................................................. 51
  5.3 Algebraic Multigrid ........................................................ 53
  5.4 AMG for XFEM .............................................................. 58

6 XAMG-I ................................................................. 60
  6.1 Schur complement of XFEM matrix ....................................... 60
  6.2 A new AMG method ...................................................... 64
  6.3 Algorithm Details .......................................................... 68
  6.4 Numerical Results ........................................................ 72
    6.4.1 Application in 3D .................................................... 76
  6.5 Extensions ................................................................. 78

7 XAMG-II ................................................................. 79
  7.1 Introduction ............................................................... 79
  7.2 Phantom node representation ............................................. 80
    7.2.1 Choice of enrichment form ....................................... 82
C XICE

C.1 FEAP User Element for 3D XFEM .............................................. 127
  C.1.1 Input format ............................................................ 128
  C.1.2 Stiffness generation for jump-enriched element ............... 128
### List of Tables

<table>
<thead>
<tr>
<th>Number</th>
<th>Table Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Problem size comparison</td>
<td>23</td>
</tr>
<tr>
<td>4.1</td>
<td>Stiff circular inclusions: Statistics of $E_{eff}$</td>
<td>32</td>
</tr>
<tr>
<td>4.2</td>
<td>Stiff elliptical inclusions: Statistics of $E_{eff}$</td>
<td>33</td>
</tr>
<tr>
<td>4.3</td>
<td>Soft elliptical inclusions: Statistics of $E_{eff}$</td>
<td>33</td>
</tr>
<tr>
<td>6.1</td>
<td>PCG iterations for different AMG approaches on a six crack problem (case 5b</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>in Figure 6.3i)</td>
<td></td>
</tr>
<tr>
<td>6.2</td>
<td>Assessment of Schur complement approximations for case 5a in Figure 6.3i</td>
<td>70</td>
</tr>
<tr>
<td>6.3</td>
<td>Preconditioned CG iterations for two-dimensional problems</td>
<td>75</td>
</tr>
<tr>
<td>6.4</td>
<td>AMG operator complexities for Quasi AMG on Case 5b</td>
<td>76</td>
</tr>
<tr>
<td>6.5</td>
<td>Preconditioned CG iterations for a three-dimensional problem</td>
<td>77</td>
</tr>
<tr>
<td>6.6</td>
<td>AMG operator complexities for a three-dimensional problem</td>
<td>77</td>
</tr>
<tr>
<td>7.1</td>
<td>Comparison of different solvers: Direct, CG (unpreconditioned, “black-box”</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>AMG preconditioned and “XAMG” preconditioned). The number of pre- and post-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sweeps of symmetric Gauss-Seidel smoother is also varied (S1 and S5) for the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>preconditioner. “nits” is the number of CG iterations and cpu time is measured in seconds</td>
<td></td>
</tr>
<tr>
<td>7.2</td>
<td>Solution of a fracture problem involving plasticity using CG-brute</td>
<td>91</td>
</tr>
<tr>
<td>7.3</td>
<td>Solution of a fracture problem involving plasticity using “black-box” AMG-S</td>
<td>91</td>
</tr>
<tr>
<td>7.4</td>
<td>Solution of a fracture problem involving plasticity using XAMG-S1</td>
<td>91</td>
</tr>
<tr>
<td>A.1</td>
<td>Summary of the convergence results for the problem considered in Fig. A.4.</td>
<td>115</td>
</tr>
</tbody>
</table>
B.1 WAM-GA scheme
## List of Figures

2.1 (a,b) Scanning Electron Microscope micrographs of ceramic matrix composite from Fair et. al. [21] (c) SEM micrograph of carbon nanotubes from Zhang et. al. [85] ... 7

2.2 Multiscale information-passing approach: (a) composites scale (b) fiber scale (c) coating scale (d) nanotube scale .......................... 8

3.1 Multiple inclusions within element boundary ............................... 12

3.2 Multiple inclusions (a) Enriched element (b) Enriched DOF ................. 13

3.3 The two coordinate systems ................................................. 16

3.4 Enrichment function for an elliptical inclusion within an element: (a) \( \varphi(x) \), (b) Contour levels of \( \varphi(x) \), (c) \( \frac{\partial \varphi}{\partial x} \), (d) \( \frac{\partial \varphi}{\partial y} \) .................. 17

3.5 Schematic of load and boundary conditions .................................. 18

3.6 Displacement solution comparison (a) XFEM mesh (b) XFEM displacements (c) FEM mesh (d) FEM displacements .......................... 19

3.7 Convergence of XFEM results to FEM results .............................. 20

3.8 Effect of elastic moduli ratio on accuracy of XFEM solution .................. 22

3.9 Mesh discretization: (a) ABAQUS FEM model (b) MATLAB XFEM model ...... 23

3.10 Convergence of XFEM results to benchmark FEM results for the case of multiple inclusions: (a) Convergence of \( x \) displacements along right edge (b) Relative norm of the difference in edge displacements versus XFEM element size .............. 24

4.1 Sample realizations of generated random microstructures ................. 28
4.2 Boundary Conditions (arrows here indicate enforced displacements) 31

4.3 Stiff circular inclusions: (a) $E_{\text{eff}}$ versus $\lambda$, (b) Histograms of $E_{\text{eff}}$, (c) $E_{\text{eff}}$ versus $n_p$ 34

4.4 Stiff circular inclusions: Convergence of Mean and COV of $E_{\text{eff}}$ 35

4.5 Stiff elliptical inclusions: (a) $E_{\text{eff}}$ versus $\lambda$, (b) Histograms of $E_{\text{eff}}$, (c) $E_{\text{eff}}$ versus $n_p$ 36

4.6 Stiff elliptical inclusions: Convergence of Mean and COV of $E_{\text{eff}}$ 37

4.7 Soft elliptical inclusions: (a) $E_{\text{eff}}$ versus $\lambda$, (b) Histograms of $E_{\text{eff}}$, (c) $E_{\text{eff}}$ versus $n_p$ 38

4.8 Soft elliptical inclusions: Convergence of Mean and COV of $E_{\text{eff}}$ 39

5.1 2D Fracture domain 47

5.2 (a) Thick lines (red) depict cracks. Circles (green) and squares (red) give nodes enriched by $H(x)$ and $F_J(x)$, respectively. (b) Computed stress $\sigma_{yy}$ 49

5.3 Levelset functions for a single crack effectively subdividing $\Omega$ into 4 regions. 50

5.4 Levelsets in three dimensions (a) Mesh with crack (b) $\Phi$, contours (c) $\Psi$, contours 51

5.5 (a) XFEM mesh for a single crack (b) Sparsity pattern of $A$ (c) Sparsity pattern of $A_{xx}$ 52

5.6 Multigrid V-cycle 54

5.7 Multigrid V cycle consisting of $\ell$ levels to solve $A^{[0]}u^{[0]} = b^{[0]}$, where level 0 is the finest level. 54

5.8 Basis functions and sine wave representation. 56

5.9 Convergence of CG with and without preconditioning 59

6.1 Graphic illustration of Schur complement stencil just left of a crack. 62

6.2 Sparsity Pattern Modification: Blue circles indicate sparsity pattern of a single prolongator column and black squares are removed pattern entries due to a crack. 69

6.3 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 73

6.4 Sample aggregates. Each grid point is colored corresponding to the aggregate that contains it. The red lines represent cracks. 74
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.1</td>
<td>Convergence study of crack modeled with elliptical enrichment</td>
<td>119</td>
</tr>
<tr>
<td>B.2</td>
<td>The XFEM-GA algorithm flowchart</td>
<td>121</td>
</tr>
<tr>
<td>B.3</td>
<td>Mesh generation, loading configuration, sensor placement and assumed linear crack locations</td>
<td>125</td>
</tr>
<tr>
<td>B.4</td>
<td>Estimated elliptical approximations for the assumed linear cracks of size $l_1 = 1\sqrt{2}$</td>
<td>125</td>
</tr>
</tbody>
</table>
Acknowledgements

I am extremely thankful to my wife, Malini Enjeti, without whose support this work would not have been possible. Since the day that I first mentioned to her about my plans to quit employment and pursue this endeavour, she has been a constant source of support and encouragement and has unwaveringly stood by my side throughout.

I am thankful to my parents Jainath and Prabha Hiriyur, who have provided me with all the love and taught me so much that has enabled me to reach this stage today. I would like to thank all my family including - Mrs. Sita Pullella, Dr. Seetharam & Mrs. Mahalakshmi Enjeti, Deepti & Gopala Suresha, Anupama & Raghu Gummaraju for their support.

I have been very fortunate in having Prof. Haim Waisman as my advisor here at Columbia. He has been an excellent mentor through this academic journey and has provided all the guidance and encouragement whenever I needed it, while allowing the freedom to pursue my own ideas. His unbounded enthusiasm and drive to pursue more achievements is perfectly matched by his calm patience through the slow phases of my research. I am extremely grateful to my co-advisor Prof. George Deodatis. He is one of the best teachers I have ever come across and I have learnt a lot from him, both in the classroom and from working closely with him on research. I have also been very fortunate to have had the opportunity to work for two summers with Ray Tuminaro at Sandia Labs. Ray has been instrumental in many of the successes achieved during the course of my research and I am very thankful to him for his guidance and mentoring. I thank Prof. David Keyes and Prof. Jacob Fish for agreeing to serve on my dissertation committee and providing many useful comments that have helped improve the quality of this dissertation. I am thankful to all the professors in the department of CEEM here at Columbia. Thanks to Axel Gerstenberger and others at Sandia with whom I have had the pleasure of working on exciting research projects. Thanks to all my friends here at CEEM and beyond - Arturo, Arundhati & Ravindra, Brett, Eleni, Patrick, Kirubel, Luciana, Mahua & Uttam, Matt, Pablo, Suparno, Tavishi & Mahesh, Tulika & Souvik, Xia and others, who have made this PhD journey memorable.
Part I

Introduction
In this introductory chapter, the scope of the author’s research is presented along with a brief outline of this dissertation.

1.1 Introduction

In this work, some contributions made by the author in the field of computational simulation of solid mechanics problems involving discontinuities are presented. Both modeling and solution of discontinuities are considered. To model the discontinuities, the main tool employed is the extended finite element method (XFEM). Both weak discontinuities (material homogeneities, inclusions etc.) and strong discontinuities (voids, cracks etc.) are considered.

Algebraic multigrid (AMG) is a powerful tool to accelerate iterative solvers to find the solution of large sparse linear systems. However, it is shown that a “black-box” application of the AMG is not suitable for linear systems arising from fracture mechanics problems modeled with XFEM. In this work, some new methods are proposed to improve the performance of AMG for these fracture problems modeled with XFEM. Here is a brief outline of this dissertation.

1.2 Outline

This dissertation is organized into four parts. Part I contains this introduction and outline.

Part II is devoted to the computational modeling of weak discontinuities - primarily material inhomogeneities - and contains three chapters. In Chapter 2, a brief survey of the state-of-the-art
in the field of computational modeling of heterogeneous microstructures is presented. In Chapter 3, the modeling of multiphase materials using an extended finite element method is presented. A new enrichment strategy that is suitable for multiple elliptical inclusions is also introduced. Finally in Chapter 4, an extended finite element method (XFEM) coupled with a Monte Carlo approach is proposed to quantify the uncertainty in the homogenized effective elastic properties of multiphase materials. The methodology allows for an arbitrary number, aspect ratio, location and orientation of elliptic inclusions within a matrix, without the need for fine meshes in the vicinity of tightly packed inclusions and especially without the need to remesh for every different generated realization of the microstructure. Moreover, the number of degrees of freedom in the enriched elements is dynamically reallocated for each Monte Carlo sample run based on the given volume fraction. The main advantage of the proposed XFEM-based methodology is a major reduction in the computational effort in extensive Monte Carlo simulations compared to the standard FEM approach. Monte Carlo and XFEM appear to work extremely efficiently together. The Monte Carlo approach allows for the modeling of the size, aspect ratios, orientations, and spatial distribution of the elliptical inclusions as random variables with any prescribed probability distributions. Numerical results are presented and the uncertainty of the homogenized elastic properties is discussed.

Part III is devoted to the computational modeling of strong discontinuities - primarily cracks - and contains four chapters. In Chapter 5, a brief literature survey describing the state-of-the-art in solution methods using Algebraic Multigrid (AMG) is presented. An introduction to modeling cracks using the extended finite element method is presented along with an introduction to algebraic multigrid. It is shown that the linear systems arising from the discretization of fracture mechanics problems using the XFEM are not suitable for a “black-box” application of AMG. In Chapter 6, a new 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. In Chapter 7, an alternative AMG approach is used to accelerate the iterative solvers for linear systems arising from XFEM fracture problems. The methods described in Chapter 6 require several modifications to the AMG to improve convergence. Here the XFEM system itself is transformed into a different representation that makes it more suitable for a direct application of AMG. It is shown that a transformation exists to represent the XFEM system in an equivalent phantom node representation that decouples the graph of the global stiffness matrix across strong discontinuities. Some parallel implementation aspects and numerical examples are also discussed.

Finally Part IV contains the summary and concluding remarks. A list of the author’s primary contributions in the field of modeling and solution of solid mechanics problems involving discontinuities is presented. Following this part is the bibliography and the appendices. In Appendix A, a collaborative work with Luc Berger-Vergiat involving a domain decomposition method for improving the AMG method for XFEM linear systems is described. In Appendix B, another collaborative effort with Eleni Chatzis is described and it involves the use of XFEM coupled with genetic algorithms for flaw detection in structures. Finally a brief description of the XFEM and AMG programs developed is provided in Appendix C.
Part II

Weak Discontinuities
Chapter 2

Motivation & Literature Survey

In this chapter, the motivation to study the computational modeling of weak discontinuities is presented. Further a brief survey of the state-of-the-art in this field is also presented. The inputs from co-authors of the paper (Hiriyur et al. [31]) from which the main chapters in this section are reproduced are gratefully acknowledged.

2.1 Motivation & Literature Survey

Composite materials are commonly used in engineering practice as they can be designed to provide a desired mechanical behavior, while satisfying other requirements such as weight density, thermal conductivity, durability etc. Fiber-reinforced materials, metal composites, concrete and ceramics are some of the widely used materials of composite nature. Several examples of complex multiphase materials including fiber-reinforced ceramic metal composites and carbon nanotubes are shown in Fig. 2.1. The mechanical behavior of composites is governed by the mechanical properties of their individual components, their volume fractions and other parameters defining their spatial and size distribution. In many cases, only the macroscopic mechanical behavior is of interest, but understanding the microstructure is also extremely important. However, a complete deterministic analysis of the medium taking the microstructure into account would involve too much computational effort and may not be feasible. It is therefore necessary to approximate the complex microstructures with equivalent effective homogeneous material properties.

To obtain the effective homogeneous properties of the composite materials, various methods have been proposed and used - both analytical [20, 18, 67, 39] and numerical [23, 36, 64, 52, 3, 83].
Monte Carlo based stochastic homogenization involves the computational analysis of a large number of randomly generated realizations of the composite medium. The results from these analyses are then used to derive the effective properties of an equivalent homogeneous medium and quantify their inherent uncertainties. The heterogeneity of composites can have significant variation across different spatial scales. Consequently, various multi-scale approaches to analyze such materials have been proposed, along with damage models for particle decohesion [42, 24, 53, 38, 35]. Figure 2.2 shows one such information passing approach applied to the modeling of nanocomposites within the coatings of traditional fiber laminated composites. Information passing or sequential multiscale methods involves the computation of quantities at the finer scale and then injecting them on to the coarser scale [22, 66]. Classical finite element methods are commonly used to analyze complex microstructures. In this case, the mesh conforms to the internal material interface boundaries that cause the strong or weak discontinuities in the displacement solution field. While fast meshing algorithms are available to discretize a domain with such internal features, this step still involves a significant computational effort. This is especially true when a large number of simulations has to be performed along the lines of a Monte Carlo approach reflecting the various uncertainties involved. Alternatively pixel and voxel based methods have also been used to analyze complex microstructures by constructing finite elements directly from microstructure images [57]. By incorporating brute-force techniques, these methods may yield accurate represen-
tations of complex microstructure features. However, these approaches are also computationally intensive and additionally, the effort involved is proportional to the pixel/voxel resolution, which should be sufficiently high to capture complexities in the microstructure that it models. Moreover, the generation of voxel-based elements from sets of 3-dimensional images is not a trivial task and involves significant complexities.

Development of XFEM [4, 46] has provided a way of dispensing with the need for remeshing each random realization of the material microstructure in a Monte Carlo formulation. XFEM uses nodal enrichment functions within the framework of the partition of unity method to augment the finite element approximations over a structured mesh [44, 60, 9]. These enrichment functions act as additional bases to model strong or weak discontinuities that are known to occur along the interface boundaries. References [5, 47] provide exhaustive reviews of the developments in extended finite element methods to date. While much of the focus of the research community in this field has been on using this method for numerical modeling of fracture and crack propagation [32, 7], there have also been many papers on the application of XFEM to model weak discontinuities in solid mechanics problems. For example, the use of level-sets within the framework of XFEM for modeling holes and inclusions has been proposed by Sukumar et al. [63] and GFEM enrichment functions for discontinuous gradient fields have been studied by Aragon et al. [1]. Alternate methods similar to XFEM have been proposed to model strong and weak dis-
continuities but using polynomial approximations similar to classical FEM and not modeling the jump in displacement or strain as explicit variables [27]. More recently the use of XFEM to model inclusion problems in viscoelastic materials has been explored [84] and extended stochastic finite element methods for solving stochastic PDEs have been proposed [50].

In previous literature on the use of XFEM for modeling multiphase media, the discrete approximations of the displacement fields were enriched by a single weakly discontinuous function. This results in the need for high mesh densities in regions where the inclusions are tightly packed. In the author’s work, the idea is to extend the framework of XFEM to incorporate the enrichments of multiple weakly discontinuous functions over a single element domain. The scalar coefficients corresponding to these multiple enrichment functions then act as unknown virtual degrees of freedom added on to the global system of equations. The number of these additional degrees of freedom associated with a particular node is also a variable that depends on the number of inclusions enriching the elements containing the node. A similar approach of using multiple level sets to prevent numerical artefacts arising from nearby inclusions in XFEM has been proposed by Tran et al. [68]. To quantify the uncertainty in the homogenized effective properties of these multiphase materials, the model is coupled with a Monte Carlo simulation approach, where a large number of random realizations of the microstructure are analyzed. Therefore the probability distributions of the size, aspect ratio, orientation and spatial distribution of the elliptical inclusions considered in this work can be readily incorporated into the analysis of the output uncertainties. The following chapters in this section are organized as follows: In section 3.1, the XFEM equations to model elliptical inclusions are developed and the framework extended to include multiple inclusions within an element domain. A comparison of the XFEM implementation with results obtained from a benchmark FEM solution is also presented. In section 4.1, a description is provided for the Monte Carlo simulation approach that is used to obtain the probability distribution of the output effective properties while modeling the randomness in the input microstructure. Finally the approach used for finding the effective elastic properties of the homogenized microstructure is described in section 4.2. The proposed approach is applied in three cases: stiff circular inclusions, stiff elliptical
inclusions and finally soft elliptical inclusions.
Chapter 3

Modeling Inclusions via XFEM

In this chapter, an introduction to the extended finite element method in the context of modeling weak discontinuities such as inclusions and material inhomogeneities is presented. A simple enrichment formulation suitable for elliptical inclusions is presented. The traditional XFEM approach to modeling inclusions is extended to accommodate multiple phase-separations within a single element domain.

3.1 XFEM for Weak Discontinuities

3.1.1 Fundamental equations

The discrete approximation for the displacement field within an element as used in classical FEM is given in Eq. (3.1):

$$ \mathbf{u}^h(x) = \sum_{j=1}^{nen} N_j(x) u_j $$

where \( nen \) represents the number of element nodes and the nodal shape functions satisfy the partition of unity condition described as \( \sum_{\forall l} N_l(x) = 1, \quad \forall x \in \Omega^e \).

An element that contains inclusions with discontinuous material properties within its boundaries will have a weakly discontinuous displacement field along the interface boundaries. Clearly, the nodal shape functions \( N_j \), which form a set of smooth and continuous functions are by themselves inadequate to model this weakly discontinuous displacement field. In XFEM, the basis functions are “enriched” through a function \( \varphi(x) \) that satisfies the local character of the displacement field that the discrete approximation aims to model. A description of the enrichment func-
tion $\varphi(x)$ that has been developed for an arbitrarily oriented elliptical inclusion is provided in section 3.1.2. To satisfy partition of unity, the enrichment function is enveloped by the original shape functions $N_j$ and corresponding additional scalar nodal coefficients $a_j$ introduced in the equation. This leads to the XFEM discrete approximation for an element containing a single material inclusion:

$$u^h(x) = \sum_{j=1}^{nen} N_j [u_j + \varphi(x)a_j]$$

(3.2)

When multiple inclusions are closely packed together as shown in Fig. 3.1, the mesh density would have to be quite large so that the enrichment functions corresponding to the different inclusions do not interfere. To overcome this limitation, the formulation is further extended to include multiple enrichment functions as described in Eq. (3.3). Here, we have enrichment functions $\varphi_i$ corresponding to each inclusion $i$ within the element domain that add to the set of basis functions in modeling the displacement field. The additional scalar nodal coefficients $a_{jk}$ corresponding to these functions are unknowns to be found, and act as virtual degrees of freedom in the global

![Figure 3.1: Multiple inclusions within element boundary](image-url)
system of equations:

\[
\mathbf{u}^h(\mathbf{x}) = \sum_{j=1}^{nen} N_j(\mathbf{x}) u_j + \sum_{j=1}^{nen} N_j(\mathbf{x}) \left[ \begin{array}{ccc}
\phi_{j1}(\mathbf{x}) a_{j1} + \phi_{j2}(\mathbf{x}) a_{j2} + \cdots + \phi_{jn_0}(\mathbf{x}) a_{jn_0}
\end{array} \right]
\]

\[
= \sum_{j=1}^{nen} N_j(\mathbf{x}) \left[ u_j + \sum_{k=1}^{n_0} \phi_{jk}(\mathbf{x}) a_{jk} \right]
\]

In equation (3.3), \( n_0 \) represents the number of inclusions in the element domain. An illustration of the multiple inclusion model and a representation of the additional scalar coefficients \( a_{jk} \) as the virtual degrees of freedom is shown in Fig. 3.2. It is interesting to note that the total number of additional degrees of freedom at a particular node is dependent on the number of inclusions in the elements containing the node and is therefore not a fixed number. In the simulations that follow, it is dynamically reallocated based on the Monte Carlo scheme described in section 4.1. In Fig. 3.2(b), the arrows at the nodes and their colors represent the added virtual degrees of freedom corresponding to the inclusions of the same colors.

To ensure that the approximation results for the classical degrees of freedom model the displacement at the nodes exactly, a suitable shifting operation is performed:

\[
\mathbf{u}^h(\mathbf{x}) = \sum_{j=1}^{nen} N_j(\mathbf{x}) \left[ u_j + \sum_{k=1}^{n_0} \left\{ \phi_k(\mathbf{x}) - \phi_k(\mathbf{x}_j) \right\} a_{jk} \right]
\]

(3.4)
The strain field is related to the displacement field through the gradients of the basis functions $\mathbf{B}(x)$ as shown in Eq. (3.5):

$$\varepsilon(x) = \mathbf{B}(x) \cdot \mathbf{u}$$

(3.5)

In the case of XFEM representation of the displacement field, the gradients of the enrichment functions also influence the strain field. Thus the expression for the strain $\varepsilon$ at any point $x$ along the direction $x_i$ is written as in Eq. (3.6):

$$\varepsilon_i(x) = \sum_{j=1}^{n_{en}} B_{ij}(x) \left[ u_j + \sum_{k=1}^{n_0} \left\{ \varphi(x) - \varphi(x_j) \right\} a_{jk} \right] + \sum_{j=1}^{n_{en}} N_j(x) \left[ u_j + \sum_{k=1}^{n_0} \frac{\partial \varphi_k(x)}{\partial x_i} a_{jk} \right]$$

(3.6)

$$\implies \mathbf{B} = \begin{bmatrix} \mathbf{B}^{FEM} & \mathbf{B}^{ENR}_1 & \mathbf{B}^{ENR}_2 & \cdots & \mathbf{B}^{ENR}_{n_0} \end{bmatrix}$$

(3.7)

Here, the $\mathbf{B}$ matrix in the XFEM approximation can be seen as consisting of two parts - the classical FEM part which corresponds to the nodal displacement degrees of freedom $u_j$ and the augmented virtual part which corresponds to the enriched degrees of freedom $a_{jk}$. The size of the augmented virtual part is dependent on the number of inclusions enriching the corresponding element nodes. In the stochastic homogenization approach, this number is a random variable and depends on the parameters describing the size and spatial distribution of inclusions. Using this $\mathbf{B}$ matrix and the constitutive equations, the stiffness terms can be computed from numerical integration as shown in equations (3.8) and (3.9). In these equations, $ngp$ refers to the number of quadrature points and $w_i$ are the corresponding weights. The numerical quadrature may be performed by either partitioning the element domain into sub-triangles/sub-quads and using Gauss points with appropriate weights or by using a large number of equidistant trapezoidal integration points along each spatial dimension. Recently, an alternative approach to integrating strong and weak discontinuities in XFEM without using integration subcells has been proposed [49]. The trapezoidal integration method with 64 quadrature points in an equispaced $8 \times 8$ grid is used in the current
work:

\begin{equation}
K_{ij}^e = \int_{\Omega^e} B_i DB_j d\Omega^e \tag{3.8}
\end{equation}

\begin{equation}
= \sum_{k=1}^{ngp} w_i B_i(x_k) D(x_k) B_j(x_k) \tag{3.9}
\end{equation}

where \( D \) represents the material constitutive matrix.

### 3.1.2 Enrichment function

A planar enrichment function and its gradients are developed for an elliptical inclusion arbitrarily oriented with respect to the coordinate axes. The equation of an ellipse with major radius \( a \) and minor radius \( b \) is given by the zero valued locus of the following function:

\begin{equation}
f(\bar{x}) = \sqrt{\frac{x^2}{a^2} + \frac{y^2}{b^2}} - 1 \tag{3.10}
\end{equation}

Here \( \bar{x} = \{ \bar{x}, \bar{y} \} \) represents the transformed coordinates with origin at the ellipse center \((x_c, y_c)\) and with orientation \( \theta \) along the ellipse radii. The transformation with respect to the global coordinate system (see Fig. 3.3) can be written as follows:

\begin{equation}
\bar{x} = (x - x_c) \cos \theta + (y - y_c) \sin \theta \\
\bar{y} = -(x - x_c) \sin \theta + (y - y_c) \cos \theta \tag{3.11}
\end{equation}

The enrichment function which requires a weak discontinuity along the ellipse boundary is obtained by the absolute value of the function in Eq. (3.10):

\begin{equation}
\varphi(x) = |f(x)| \tag{3.12}
\end{equation}

The gradients of the enrichment function are also obtained as follows:

\begin{equation}
\frac{\partial \varphi}{\partial x} = \text{sign}(f) \left( \frac{\partial f}{\partial x} \frac{\partial x}{\partial x} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial x} \right) \\
\frac{\partial \varphi}{\partial y} = \text{sign}(f) \left( \frac{\partial f}{\partial x} \frac{\partial x}{\partial y} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial y} \right) \tag{3.13}
\end{equation}
Where the gradient functions with respect to the transformed coordinates are given as follows:

\[
\begin{align*}
\frac{\partial f}{\partial \bar{x}} &= \frac{\bar{x}}{a^2} \left( \sqrt{\frac{\bar{x}^2}{a^2} + \frac{\bar{y}^2}{b^2}} \right)^{-1} \\
\frac{\partial f}{\partial \bar{y}} &= \frac{\bar{y}}{b^2} \left( \sqrt{\frac{\bar{x}^2}{a^2} + \frac{\bar{y}^2}{b^2}} \right)^{-1}
\end{align*}
\]  

(3.14)

The final form of the gradients of the enrichment function are obtained by plugging (3.14) in (3.13):

\[
\begin{align*}
\frac{\partial \varphi}{\partial \bar{x}} &= \text{sign}(f) \left( \sqrt{\frac{\bar{x}^2}{a^2} + \frac{\bar{y}^2}{b^2}} \right)^{-1} \left( \frac{\bar{x} \cos(\theta)}{a^2} - \frac{\bar{y} \sin(\theta)}{b^2} \right) \\
\frac{\partial \varphi}{\partial \bar{y}} &= \text{sign}(f) \left( \sqrt{\frac{\bar{x}^2}{a^2} + \frac{\bar{y}^2}{b^2}} \right)^{-1} \left( \frac{\bar{x} \sin(\theta)}{a^2} + \frac{\bar{y} \cos(\theta)}{b^2} \right)
\end{align*}
\]  

(3.15)

A graphical description of the enrichment function and its gradients is provided in Fig. 3.4.

### 3.1.3 Convergence study of XFEM solution for single inclusion

A convergence study is performed to compare the XFEM solution developed using the enrichment function described in section 3.1.2 against a standard FEM solution. In the latter case, an FEM mesh conforming to the interface boundaries is formed using quadrilateral bilinear elements. These quadrilateral meshes are obtained using a python script within ABAQUS. A unit cell of size 1 \times 1 is subject to the boundary and loading configuration shown in Fig. 3.5. The magnitude of the edge traction along the right edge is 1E3. The matrix and the inclusion are modeled using linear
Figure 3.4: Enrichment function for an elliptical inclusion within an element: (a) $\varphi(x)$, (b) Contour levels of $\varphi(x)$, (c) $\frac{\partial \varphi}{\partial x}$, (d) $\frac{\partial \varphi}{\partial y}$
isotropic materials with the following elastic properties:

\[ \text{Matrix} : \quad E_m = 70 \cdot 10^3 \text{ MPa} \]

\[ \text{Inclusions} : \quad E_p = 410 \cdot 10^3 \text{ MPa} \]

(3.16)

Poisson’s ratio for both the matrix and the inclusion is set equal to 0.3. To obtain the stiffness quantities in XFEM, the numerical integration in Eq. (3.8) is performed in the element domain using \(8 \times 8\) equispaced quadrature points. The following variable parameters are considered in this study: the inclusion aspect ratio (0.2, 0.4 0.6, 0.8 and 1) and the orientation angle \((0, 45^\circ, 90^\circ)\) with respect to the loading direction. Figure 3.6 shows the meshes and the displacement solutions obtained using XFEM and FEM for the case of the inclusion ellipse aspect ratio being equal to 0.4 and the orientation angle of the major axis being equal to \(45^\circ\) with respect to the global \(x\) axis. Agreement between FEM and XFEM solutions is excellent. Similar quality of agreement was observed for all the cases considered involving the aforementioned values for the aspect ratio and the orientation angle.

Figure 3.5: Schematic of load and boundary conditions

The differences in the solutions obtained from XFEM and FEM analyses are plotted against the logarithm of the XFEM element size in Fig. 3.7 for five different aspect ratios \(r\). The plots in the left column show the relative norm of the difference in displacement solutions and the plots
in the right column show the difference in strain energies computed by XFEM and FEM. The results indicate that when the orientation of the elliptical inclusion is parallel or perpendicular to the loading direction ($\theta = 0$ or 90 degrees), the convergence rates for all the ellipse aspect ratios considered are very close to each other. When the orientation of the elliptical inclusion is 45 degrees, the difference in the convergence rates for the different aspect ratios vary slightly; the convergence being slowest when the aspect ratio is 0.2.

Since the enrichment function described in section 3.1.2 is designed for approximation of
Figure 3.7: Convergence of XFEM results to FEM results
weakly discontinuous $C^0$ fields, it is ideally suited when the difference in elastic moduli between the matrix and inclusions is relatively small. Some studies have shown that an XFEM formulation using an enrichment function that is based on a discontinuous deformation map is better suited when there is a large modulus mismatch [62]. To study the effect of the ratio of the elastic moduli of the two phases on the convergence properties of the XFEM enrichment function described in section 3.1.2 towards a reference benchmark solution, the following test is performed. A unit cell of size $1 \times 1$ with a single elliptical inclusion (of aspect ratio $r = 0.5$ and orientation angle $\theta = 45^\circ$) is subjected to the same boundary conditions and loading as described earlier in this section. In this study, only the ratio of the elastic moduli of the two phases is varied ($E_m$ is fixed at $70 \cdot 10^3$ MPa and the ratio $E_p/E_m$ is varied from 2 to 1,000 for stiff inclusions and from 0.001 to 0.5 for soft inclusions) keeping all other parameters unchanged. XFEM analyses and reference FEM analyses are performed for each ratio of the moduli and the two solutions are compared. In Fig. 3.8, the relative differences (in terms of strain energy and displacements) of the two solutions are plotted against the corresponding ratio of the elastic moduli. It is observed that the XFEM solution is increasingly diverging from the reference FEM solution as the moduli ratio increases. The divergence appears to be larger for the case of stiff inclusions. However, even for the largest ratio considered (1,000), the difference in the two solutions is only about 1.4%. For the moduli ratios considered in the convergence studies (refer to sections 3.1.3 and 3.1.4) and the Monte Carlo simulations (refer to section 4.2.1), the differences between XFEM and the reference FEM solution are at most 0.3%.

### 3.1.4 Convergence study of XFEM solution for multiple inclusions

Implementation of the proposed XFEM approach for multiple inclusions is performed in MATLAB. For the benchmark verification study between XFEM and FEM, a specific realization involving 15 elliptical inclusions of volume fraction 0.3, uniformly scattered in a 2D unit cell is studied. The test configuration (size of the unit cell, material properties of the two phases, boundary conditions and loads) is identical to the description provided in the preceding section. Figure 3.9 shows the mesh discretization used for the reference FEM study and a representative XFEM discretiza-
tion (among the many considered for this convergence study). The benchmark FEM analysis was performed using a quadrilateral mesh containing 12,842 nodes and 12,641 elements (shown in Fig. 3.9a). The XFEM analyses were performed using structured square meshes of sizes ranging from $9 \times 9$ elements to $94 \times 94$ elements. Table 3.1 shows the comparison of the problem sizes (nodes, elements and degrees of freedom) for the different discretizations considered. The XFEM analyses involved the solution of smaller systems of equations compared to the corresponding reference FEM mesh. However, it must be noted that in XFEM case, the establishment of the element stiffness matrices involves a larger effort compared to FEM since more quadrature points are required to accurately integrate over a weakly discontinuous field. This drawback however is offset by the benefits of having far simpler structured mesh generation and solving a smaller global system of equations. As the mesh is refined, the number of global system of equations (unknown degrees of freedom) in XFEM is approaching the size of the benchmark FEM system. However, a majority of these degrees of freedom correspond to nodal displacements (i.e., the regular FEM
degrees of freedom and not the enrichments), which in sophisticated implementations can be retained for multiple simulations where the underlying mesh need not change (e.g. Monte Carlo simulations described in following sections). The convergence of the XFEM nodal displacement solution at the right edge (refer to Fig. 3.5) to the corresponding reference solution from FEM is shown in Fig. 3.10.

![Mesh discretization: (a) ABAQUS FEM model (b) MATLAB XFEM model](image)

Figure 3.9: Mesh discretization: (a) ABAQUS FEM model (b) MATLAB XFEM model

<table>
<thead>
<tr>
<th></th>
<th>FEM</th>
<th>9 × 9</th>
<th>20 × 20</th>
<th>44 × 44</th>
<th>94 × 94</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>12,842</td>
<td>100</td>
<td>441</td>
<td>2,025</td>
<td>9,025</td>
</tr>
<tr>
<td>Number of elements</td>
<td>12,641</td>
<td>81</td>
<td>400</td>
<td>1,936</td>
<td>8,836</td>
</tr>
<tr>
<td>Number of equations</td>
<td>25,684</td>
<td>486</td>
<td>1,588</td>
<td>5,704</td>
<td>21,644</td>
</tr>
</tbody>
</table>

Table 3.1: Problem size comparison
Figure 3.10: Convergence of XFEM results to benchmark FEM results for the case of multiple inclusions: (a) Convergence of $x$ displacements along right edge (b) Relative norm of the difference in edge displacements versus XFEM element size
Chapter 4

Homogenization of Random Microstructures

In this chapter, the extended finite element method is coupled with the Monte-Carlo approach to be used in uncertainty quantification in homogenization of random heterogeneous microstructures.

4.1 Monte Carlo approach

The first step of the proposed Monte Carlo based stochastic homogenization involves the generation of a large number of random realizations of the microstructure geometry based on a given volume fraction of the inclusions and other parameters representing the uncertainties in their number, aspect ratios, spatial distribution and orientation. There are several methods available for this purpose, an example being the random growth algorithm [40]. The approach used in this work (with certain similarities to the ballistic deposition algorithm [79]) is described in section 4.1.1. Once the random microstructures are obtained, deterministic elastic analyses are performed for a unit cell containing these generated microstructures, when subjected to a predetermined set of loads and boundary conditions. Finally, the effective homogeneous properties corresponding to these random microstructures are obtained by finding the best-fit material properties for an equivalent homogeneous unit cell subjected to the same loads and boundary conditions. When XFEM is used, the equivalent homogeneous unit cell is obtained from the same mesh with the internal boundaries and enrichments removed. Through the residual minimization routine, uniform material properties are assigned at all the quadrature points. The discrete problem for the
homogeneous medium then reduces to a classical FEM one and the solution is obtained at the
same nodal points as the heterogeneous case.

A strain energy approach is used in this work to determine the homogenized effective prop-
erties. For the case considered here involving a linear isotropic material under plane stress con-
ditions, there are two independent effective properties to be found: the elastic modulus $E_{\text{eff}}$ and
Poisson’s ratio $\nu_{\text{eff}}$. To obtain the best-fit estimates for these two quantities, an optimization algo-
riothm is used with a residual quantity $R$ as the objective function to be minimized and the material
properties as the control variables. As shown in Eq. (4.1), the residual quantity $R$ is defined in
terms of the difference in internal strain energy stored in the heterogeneous system (computed us-
ing XFEM including enrichments) and the corresponding homogenized systems (computed using
classical FEM without enrichments):

\[
R(U(E, \nu, \cdots)) \equiv \langle \int_{\Omega} \sigma : \varepsilon d\Omega \rangle_{\text{XFEM}} - \langle \int_{\Omega} \sigma : \varepsilon d\Omega \rangle_{\text{FEM}}
\] (4.1)

The most important benefit of XFEM is realized when performing the elastic analyses of the
multiple randomly heterogeneous microstructures on a fixed structured mesh. Only the enrich-
ments and the corresponding additional degrees of freedom change in each Monte Carlo simu-
lation, thereby avoiding the need to remesh every generated random microstructure. An added
benefit of using XFEM is that it allows for easy comparison of individual nodal and elemental
solution quantities across different random realizations of the microstructure and equivalent ho-
mogeous systems as the mesh remains the same. A similar type of optimization using XFEM
has been proposed for adapting the enrichment function to the solution [76] and for detection of
flaws in structures [54, 77].

4.1.1 Random microstructure generation

The Monte Carlo procedure to generate a large number of sample realizations of a unit cell of the
microstructure is described in Algorithm 4.1.1. The inclusions are described through the volume
fraction $\lambda$, the total number of inclusions $n_p$ within the unit cell and independent probability
distributions for the following: inclusion relative major radius $f_{\hat{a}}$, aspect ratio $f_r$, location of the ellipse center $f_c$ and ellipse orientation $f_\theta$. The first step is to generate the elliptical inclusions. To this end, values representing the relative inclusion major radius $\hat{a}_i$ and the inclusion aspect ratio $r_i$ are generated according to prescribed PDF’s (assumed independent). The subscript $i$ ranges from $1$ to $n_p$. The minor radii are calculated as $\hat{b}_i = r_i \hat{a}_i$. The inclusion sizes obtained are relative to the prescribed distributions and need to be scaled to represent the prescribed volume fraction $\lambda$. The cumulative area of all the randomly generated inclusions is then computed and an appropriate scaling factor is determined and applied to the major and minor radii to obtain the final inclusion sizes reflecting the prescribed volume fraction. It should be mentioned that the scaled inclusion major radii $a_i$ do not follow strictly the prescribed PDF $f_{\hat{a}}$. Its shape is preserved but not its mean value and variance.

The next step is to spatially distribute the generated inclusions within the base matrix representing the unit cell, starting with the largest inclusion and following with the remaining inclusions in decreasing order of size. Values representing the coordinates of the ellipses’ centers and their orientations are generated according to the prescribed probability distributions ($f_c$ and $f_\theta$ respectively). If an inclusion centered at such generated coordinates is found to overlap with any other previously generated and spatially placed inclusion(s), new center coordinates and orientation are generated until no overlap is observed. The procedure continues then with the next smaller inclusion. By spatially distributing the inclusions in decreasing order of size, the probability of overlap of a particular inclusion with others that have been previously placed within the unit cell is reduced. A set of sample realizations obtained using Algorithm 4.1.1 for different volume fractions $\lambda$ and number of inclusions $n_p$ is shown in Fig. 4.1.

### 4.2 Homogenization

#### 4.2.1 Application and results

The proposed framework of using XFEM coupled with Monte Carlo simulations is used to obtain the probability distribution of the effective elastic modulus for a plane-stress medium containing
Figure 4.1: Sample realizations of generated random microstructures
Algorithm 4.1.1 Random Microstructure Generation

- **INPUT**
  - $X_1, X_2$: Size of the unit cell
  - $\lambda$: Inclusion volume fraction
  - $n_p$: Total number of inclusions in unit cell
  - $f_{\hat{a}}, f_r, f_c, f_\theta$: Independant probability distributions for the relative major radius, aspect ratio, center coordinates and orientation angle

- **GENERATE/SCALE/SORT INCLUSIONS**
  - Generate $n_p$ random numbers to represent relative major radius $\hat{a}_i$ following the prescribed probability distribution $f_{\hat{a}}$
  - Generate $n_p$ random numbers to represent the aspect ratio $r_i$ following the prescribed probability distribution $f_r$
  - Obtain the relative minor radii $\hat{b}_i = r_i \hat{a}_i$ and cumulative area $\hat{A}_{incl} = \pi \sum_{k=1}^{n_p} \hat{a}_i \hat{b}_i$
  - Scale radii: $a_i = \hat{a}_i \sqrt{\lambda \cdot \frac{X_1 X_2}{\hat{A}_{incl}}}$ and $b_i = \hat{b}_i \sqrt{\lambda \cdot \frac{X_1 X_2}{\hat{A}_{incl}}}$
  - Sort inclusions in decreasing order of size

- **SPATIALLY DISTRIBUTE ON UNIT CELL**
  - Loop over inclusions $k = 1$ to $n_p$
    - Generate random numbers $x_k, y_k$ (uniform in $[0, X_1]$ and $[0, X_2]$ respectively) and $\theta_k$ (uniform in $[0, 2\pi]$) to represent inclusion ellipse center and orientation
    - Check overlap with previously positioned inclusions $1$ to $k$
      - If TRUE, repeat step for inclusion $k$ with new random values for coordinates $x_k, y_k$ and orientation $\theta_k$
      - if FALSE, proceed to next smaller inclusion

Elliptical inclusions. A linear isotropic material model is used for the matrix and the inclusion phases and for simplicity, the homogenization is assumed to preserve this property. Therefore only two independent effective material properties are computed: effective elastic modulus $E_{eff}$ and effective Poisson’s ratio $\nu_{eff}$. It should be noted that a more general orthotropic model would be more accurate for the resulting homogenized medium. However using such a model involves a significantly higher computational cost and therefore it is not used in the current study as it
is not contributing towards the main objectives of this work. Here, the effective Poisson’s ratio is fixed \textit{a priori} to the common Poisson’s ratio for the two phases \((\nu_m = \nu_p = 0.3 \rightarrow \nu_{\text{eff}})\) and the residual \(R_U\) in Eq. (4.1) is minimized with respect to the effective elastic modulus \(E_{\text{eff}}\) alone. In the examples provided here, the size of the domain is smaller than the typical representative volume element and therefore a strict comparison to the homogenized effective properties would not be accurate. However for comparison and perspective, the following three models for effective elastic properties are provided:

\textbf{a) Voigt Model:} This model [74] is based on the assumption of uniform strain throughout the medium. The effective elastic modulus is given by the volume fraction weighted average (arithmetic mean) of the elastic moduli of the individual phases:

\[
E_V = (1 - \lambda)E_m + \lambda E_p
\]  

\(4.2\)

\textbf{b) Reuss Model:} This model [56] is based on the assumption of uniform stress throughout the medium. The effective elastic modulus is given by the reciprocal volume fraction weighted average (harmonic mean) of the elastic moduli of the individual phases:

\[
\frac{1}{E_R} = \frac{(1 - \lambda)}{E_m} + \frac{\lambda}{E_p}
\]  

\(4.3\)

Hill [29] shows that when there is a sufficiently large number of inclusions and when the medium is macroscopically homogenous, the effective moduli from the Voigt and Reuss models form strict upper and lower bounds to the actual effective modulus respectively:

\[
E_R \leq E_{\text{eff}} \leq E_V
\]  

\(4.4\)

\textbf{c) Mori-Tanaka Model:} This model [48] is based on the assumption that the actual stress in the medium is the average stress plus a fluctuating stress (influenced by the inclusions), the average of which vanishes in the medium. The effective bulk modulus \(\bar{K}\) and the effective shear modulus
\( \bar{G} \) are obtained as follows:

\[
\bar{K} = K_m + \frac{\lambda K_m (K_p - K_m)}{K_m + \beta_2 (1 - \lambda)(K_p - K_m)}
\]

\[
\bar{G} = G_m + \frac{\lambda G_m (G_p - G_m)}{G_m + \beta_1 (1 - \lambda)(G_p - G_m)}
\]

where \( \beta_1 = \frac{2(4 - 5\nu_m)}{15(1 - \nu_m)} \) and \( \beta_2 = 3 - 5\beta_1 \). The effective elastic modulus is then given as follows:

\[
E_{MT} = \frac{9 \bar{K} \bar{G}}{3 \bar{K} + \bar{G}} \tag{4.5}
\]

In the Monte Carlo simulations that follow, three different cases are considered: (a) stiff circular inclusions, (b) stiff elliptical inclusions and (c) soft elliptical inclusions. A total of 2000 Monte Carlo simulations are performed for each volume fraction \( \lambda \) considered ranging from 0.1 to 0.5. The number of inclusions \( n_p \) varies from 1 to 30. A unit cell of size \( 1 \times 1 \) subjected to the boundary conditions shown in Fig. 4.2 is used for this study. The enforced uniform displacements (compressive) on the top and bottom edges are 0.3 times the uniform displacements enforced on the left and right edges (tensile).

Figure 4.2: Boundary Conditions (arrows here indicate enforced displacements)
4.2.1.1 Case 1 - Stiff circular inclusions:

The elastic modulus of the inclusions $E_p$ is related to that of the matrix $E_m$ as follows: $E_p / E_m = 5.8$ (the values of $E_p$ and $E_m$ mentioned in section 3.1.3 are used). The relative radii of the circular inclusions follow a uniform distribution in [0.5, 1]. Figure 4.3(a) displays the results of the Monte Carlo simulations regarding the variation of the effective modulus in terms of the volume fraction $\lambda$. Figure 4.3(b) displays the corresponding histograms of $E_{eff}$ for the five values of $\lambda$ considered. And the variation of $E_{eff}$ with respect to the number of inclusions $n_p$ is provided in Fig. 4.3(c). The resulting mean, standard deviation and coefficient of variation of $E_{eff}$ for each volume fraction $\lambda$ considered are presented in Table 4.1. Figure 4.4 shows the convergence of the mean and COV of $E_{eff}$ with increasing number of Monte Carlo realizations.

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>Mean</th>
<th>Std. Devn.</th>
<th>COV</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>7.85E+04</td>
<td>5.62E+02</td>
<td>0.0072</td>
</tr>
<tr>
<td>0.2</td>
<td>8.99E+04</td>
<td>1.13E+03</td>
<td>0.0126</td>
</tr>
<tr>
<td>0.3</td>
<td>1.04E+05</td>
<td>2.01E+03</td>
<td>0.0194</td>
</tr>
<tr>
<td>0.4</td>
<td>1.22E+05</td>
<td>2.64E+03</td>
<td>0.0217</td>
</tr>
<tr>
<td>0.5</td>
<td>1.44E+05</td>
<td>3.08E+03</td>
<td>0.0214</td>
</tr>
</tbody>
</table>

4.2.1.2 Case 2 - Stiff elliptical inclusions:

As in the previous case, the elastic modulus of the inclusions $E_p$ is related to that of the matrix $E_m$ as follows: $E_p / E_m = 5.8$ (the values of $E_p$ and $E_m$ mentioned in section 3.1.3 are used). The relative major radii $\hat{a}_i$ of the inclusions follow a uniform distribution in [0.5, 1] and the ellipses’ aspect ratios $r_i$ follow a uniform distribution also in [0.5, 1]. Figure 4.5(a) displays the results of the Monte Carlo simulations regarding the variation of the effective modulus in terms of the volume fraction $\lambda$. Figure 4.5(b) displays the corresponding histograms of $E_{eff}$ for the five values of $\lambda$ considered. And the variation of $E_{eff}$ with respect to the number of inclusions $n_p$ is provided in Fig. 4.5(c). The resulting mean, standard deviation and coefficient of variation of $E_{eff}$ for each volume fraction $\lambda$ considered are presented in Table 4.2. Figure 4.6 shows the convergence of the
mean and COV of $E_{\text{eff}}$ with increasing number of Monte Carlo realizations.

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>Mean</th>
<th>Std. Devn.</th>
<th>COV</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>7.84E+04</td>
<td>6.10E+02</td>
<td>0.0078</td>
</tr>
<tr>
<td>0.2</td>
<td>9.00E+04</td>
<td>1.21E+03</td>
<td>0.0135</td>
</tr>
<tr>
<td>0.3</td>
<td>1.04E+05</td>
<td>2.15E+03</td>
<td>0.0206</td>
</tr>
<tr>
<td>0.4</td>
<td>1.22E+05</td>
<td>3.35E+03</td>
<td>0.0275</td>
</tr>
<tr>
<td>0.5</td>
<td>1.44E+05</td>
<td>3.97E+03</td>
<td>0.0275</td>
</tr>
</tbody>
</table>

### 4.2.1.3 Case 3 - Soft elliptical inclusions:

In the third case, the elastic modulus of the matrix is higher than that of the inclusions as: $E_m/E_p = 5.8$ ($E_p = 70 \cdot 10^3$ MPa and $E_m = 410 \cdot 10^3$ MPa). The relative major radii $\hat{a}_i$ of the inclusions follow a uniform distribution in $[0.5, 1]$ and the ellipses’ aspect ratios $r_i$ follow a uniform distribution also in $[0.5, 1]$. Figure 4.7(a) displays the results of the Monte Carlo simulations regarding the variation of the effective modulus in terms of the volume fraction $\lambda$. Figure 4.7(b) displays the corresponding histograms of $E_{\text{eff}}$ for the five values of $\lambda$ considered. And the variation of $E_{\text{eff}}$ with respect to the number of inclusions $n_p$ is provided in Fig. 4.7(c). The resulting mean, standard deviation and coefficient of variation of $E_{\text{eff}}$ for each volume fraction $\lambda$ considered are presented in Table 4.3. Figure 4.8 shows the convergence of the mean and COV of $E_{\text{eff}}$ with increasing number of Monte Carlo realizations.

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>Mean</th>
<th>Std. Devn.</th>
<th>COV</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>3.37E+05</td>
<td>3.32E+03</td>
<td>0.0099</td>
</tr>
<tr>
<td>0.2</td>
<td>2.82E+05</td>
<td>5.39E+03</td>
<td>0.0191</td>
</tr>
<tr>
<td>0.3</td>
<td>2.37E+05</td>
<td>6.74E+03</td>
<td>0.0284</td>
</tr>
<tr>
<td>0.4</td>
<td>2.00E+05</td>
<td>6.71E+03</td>
<td>0.0336</td>
</tr>
<tr>
<td>0.5</td>
<td>1.69E+05</td>
<td>5.77E+03</td>
<td>0.0342</td>
</tr>
</tbody>
</table>
Figure 4.3: Stiff circular inclusions: (a) $E_{\text{eff}}$ versus $\lambda$, (b) Histograms of $E_{\text{eff}}$, (c) $E_{\text{eff}}$ versus $n_p$. 
CHAPTER 4. XFEM-MC

Figure 4.4: Stiff circular inclusions: Convergence of Mean and COV of $E_{\text{eff}}$.
Figure 4.5: Stiff elliptical inclusions: (a) $E_{\text{eff}}$ versus $\lambda$, (b) Histograms of $E_{\text{eff}}$, (c) $E_{\text{eff}}$ versus $n_p$
Figure 4.6: Stiff elliptical inclusions: Convergence of Mean and COV of $E_{\text{eff}}$.
Figure 4.7: Soft elliptical inclusions: (a) $E_{\text{eff}}$ versus $\lambda$, (b) Histograms of $E_{\text{eff}}$, (c) $E_{\text{eff}}$ versus $n_p$
Figure 4.8: Soft elliptical inclusions: Convergence of Mean and COV of $E_{\text{eff}}$. 
4.2.2 Discussion

In all three cases examined, the uncertainty in the effective elastic modulus $E_{\text{eff}}$ (indicated by the spreading of the histograms and the COV values) is increasing as the volume fraction $\lambda$ increases. For all three cases too, the general trend is that the coefficient of variation of $E_{\text{eff}}$ decreases as the number of inclusions increases. Finally, observing the results in cases 1 and 2 where the only difference is the shape of the inclusions (circular vs. elliptical) and taking into account that there are only minor differences between the results of these two cases, it appears that the uncertainty in $E_{\text{eff}}$ is not greatly affected by the ellipse aspect ratio (at least for the values considered in this study).

The computed effective moduli respect the Voigt and Reuss elastic bounds given in Eq. (4.4) and are reasonably close to the effective elastic modulus predicted by the Mori-Tanaka model given in Eq. (4.5).

4.3 Concluding remarks

This work demonstrated the application of extended finite element methods for modeling systems with known weak discontinuities in the solution field in combination with a Monte Carlo simulation approach to quantify the uncertainty of homogenized elastic properties for a random two-phase composite in 2D. XFEM methods offer a computationally superior alternative to classical FEM methods for such problems, especially when a large number of Monte Carlo simulations is necessary. The numerical examples considered in this work indicated that the effective homogeneous properties exhibit increasingly higher uncertainty as the volume fraction increases, but this uncertainty is largely insensitive to other parameters.

The main objective of this work was to demonstrate the excellent synergy of XFEM and Monte Carlo simulation compared to standard FEM combined with MC simulation. Though the problems considered in this work were limited to two dimensions, it is expected that for three dimensional problems (where meshing of discontinuous domains is more complicated and larger sizes
of resulting linear systems involve significantly higher computational expense) the synergy provided by coupling XFEM and Monte Carlo simulations is further advantageous. Further, sophisticated implementations of the XFEM-MC approach may be pursued wherein the regular degrees of freedom can be retained across all the realizations and only the enriched degrees of freedom (and its couplings with regular degrees of freedom) are computed for each individual simulation, thereby resulting in additional savings of computational effort. Although the uncertainties involved in the problem considered here were relatively simple and the problem was a linear one, the main purpose was to demonstrate the capabilities of the overall methodology. Extensions to more complex uncertainties modeled by random fields and to nonlinear problems where a Monte Carlo simulation approach is the only option will be explored in the future.
Part III

Strong Discontinuities
Chapter 5

Introduction

In this chapter, the motivation to study the computational modeling of strong discontinuities is presented along with a brief survey of the state-of-the-art in the modeling and solution methods involving multigrid methods applied to XFEM linear systems. This chapter also features an introduction to extended finite element method in the context of modeling strong discontinuities such as cracks and an introduction to the algebraic multigrid.

5.1 Motivation & Literature Survey

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. Any material will fracture depending on the loading circumstances and environmental conditions. Typically fracture is classified as brittle or ductile. This work is mainly concerned with brittle fracture. Some examples of brittle fracture include delamination of composite structures due to fatigue loadings and the cracking of ice sheets in Greenland due to global warming.

Unfortunately, the presence of discontinuities poses a number of numerical challenges associated with discretization and solvers. Standard finite element methods are severely limited to a small number of discontinuities or simplified problems as very fine meshes are required in the vicinity of these discontinuities. Moreover, if these discontinuities propagate due to quasi-static or fatigue loadings, the domain must be re-meshed at every step which makes modeling by standard
finite element methods highly challenging. The extended finite element method (XFEM) offers an alternative [44, 4, 46, 6, 37, 47]. The key idea of XFEM is to use a standard finite element mesh that is independent of internal boundaries. The discontinuities (along the interface) and singularities (near the tip) are instead captured through an “enriched” space of basis functions that have local support near a crack and satisfy a partition of unity. These additional enrichment functions model the brittle fracture that certain materials exhibit. The use of an enriched space of basis functions alleviates the need for remeshing the domain in the case of propagating cracks.

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 approximate 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.

XFEM and geometric multigrid are considered in [55]. The importance of treating different types of enrichment functions separately within interpolation as well as approximating fine level basis functions within interpolation is recognized. Additionally, a consistent levelset function across all multigrid levels is developed. These themes also appear in the method proposed here.
Domain decomposition is used to develop preconditioners suitable for extended finite elements in [45]. In [8], the domain is decomposed into “cracked” and “healthy” subdomains with AMG acting only on the latter, while a Schwarz multiplicative method is used to develop the full solution. In [17] a multilevel BPX preconditioner is considered for the generalized finite element method (GFEM) and is appropriate for simplicial grids. Here, an auxiliary matrix based approach is pursued where one supplies an additional matrix which is used to guide the preconditioner. In [82] point smoothers are analyzed for a model GFEM problem and it is shown how standard smoothers do not in fact damp all high frequencies. This leads the authors to consider line Gauss-Seidel schemes within an algebraic multigrid context which are shown to perform suitably on a model problem.

In this section, the author presents two new methods based on algebraic multigrid suitable for the discrete linear systems associated with XFEM. The first method described in Chapter 6 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 the proposed 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.

The second new method described in Chapter 7 is derived from a phantom-node representation of XFEM. It is shown that the linear systems arising from this representation are directly suitable for application of AMG. The transformation between a typical XFEM system and the corresponding phantom node representation is presented. Some numerical results are shown to illustrate the effectiveness of the proposed algorithms. In this work, three dimensional cracks, nonlinear materials and parallel implementation issues are also considered.
5.2 XFEM for Strong Discontinuities

5.2.1 Governing Equations

The governing equations are now presented for an elasto-static physical system. Consider a two-dimensional bounded domain $\Omega$ with internal crack surfaces $\Gamma_c$. Body forces are represented by $b$, prescribed displacements $\bar{u}$ are applied along $\Gamma_u$ and prescribed tractions $\bar{t}$ along $\Gamma_t$. The linear elastic equations are

$$
\nabla^T_s \sigma + b = 0 \quad \text{in } \Omega \\
\bar{u} = \bar{u} \quad \text{on } \Gamma_u \\
\bar{\sigma} \cdot n = \bar{t} \quad \text{on } \Gamma_t \\
\sigma \cdot n = 0 \quad \text{on } \Gamma_{c+}, \Gamma_{c-}
$$

(5.1)

where $n$ is a unit normal vector on $\Gamma$ and $\nabla^T_s$ is the symmetric gradient operator

$$
\nabla^T_s = \begin{bmatrix}
\frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial y} \\
0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x}
\end{bmatrix}.
$$

(5.2)

Indeed the last condition in (5.1) corresponds to a linear elastic material where all cracks have opened. In a more general case, residual stresses, near-tip plasticity or simply complex loading conditions may cause cracks to close during the growth process [58]. In such cases, the following no-penetration condition must also be applied:

$$
\llbracket u \rrbracket \cdot n \geq 0 \quad \text{on } \Gamma_c
$$

(5.3)

where $\llbracket u \rrbracket$ refers to the jump in displacement at crack-interface. This condition is checked a posteriori similar to contact problems [80] thus leading to nonlinear boundary contact treatment. Nonetheless, in the problems considered in this work, the loading conditions are chosen such that cracks are only opening and thus condition (5.3) can be neglected in the solution process.

For the linear elastic case considered here, the stress $\sigma$ is related to the strain $\epsilon$ through a standard Hookean elasticity matrix $D$, where the strain $\epsilon$ for small displacements is obtained by
applying the gradient operator on displacements $\mathbf{u}$:

$$
\sigma = \mathbf{D} \cdot \mathbf{e} \quad \text{and} \quad \mathbf{e} = \nabla_s \cdot \mathbf{u}.
$$

### 5.2.2 Weak Form

The weak form of (5.1) corresponds to finding a $\mathbf{u} \in \mathcal{U}$ such that

$$
\int_{\Omega} (\nabla_s \mathbf{w})^T \mathbf{D} (\nabla_s \mathbf{u}) \, d\Omega = \int_{\Gamma_t} \mathbf{w}^T \mathbf{t} \, d\Gamma_t + \int_{\Omega} \mathbf{w}^T \mathbf{b} \, d\Omega \quad \forall \mathbf{w} \in \mathcal{W} \quad (5.4)
$$

where $\mathcal{U} = \{ \mathbf{u} \mid \mathbf{u} \in H^1, \mathbf{u} = \bar{\mathbf{u}} \text{ on } \Gamma_u \}$, $\mathcal{W} = \{ \mathbf{w} \mid \mathbf{w} \in H^1, \mathbf{w} = 0 \text{ on } \Gamma_t, \Gamma_u \}$, and $H^1$ is the Sobolev Space of $L_2$ functions with square integrable derivatives. The crack satisfies the above weak form since the interfaces are traction free.

The discretization of the weak form leads to:

$$
\int_{\Omega^h} (\nabla_s \mathbf{w}^h)^T \mathbf{D} (\nabla_s \mathbf{u}^h) \, d\Omega^h = \int_{\Gamma_{t}^h} \mathbf{w}^h \mathbf{t} \, d\Gamma_{t}^h + \int_{\Omega^h} \mathbf{w}^h \mathbf{b} \, d\Omega^h \quad (5.5)
$$

where superscript $h$ denotes discrete quantities. The XFEM discrete trial function $\mathbf{u}^h \in \mathcal{U}^h$ for fracture is then

$$
\mathbf{u}^h(\mathbf{x}) = \sum_{i=1}^{n} N_i(\mathbf{x}) \mathbf{u}_i + \sum_{i=1}^{n_h} N_{i_i}(\mathbf{x}) (H(\mathbf{x}) - H(\mathbf{x}_{i_i})) \mathbf{a}_{i_i} + \sum_{i=1}^{n_f} \sum_{j=1}^{n_f} N_{i_f}(\mathbf{x}) \left( F_{j_i}(\mathbf{x}) - F_{j_f}(\mathbf{x}_{i_f}) \right) \mathbf{b}_{i_f} \quad (5.6)
$$
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, \( n_f \) is the number of tip singularity functions, \( N_i(x) \) 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. Discontinuities along the crack interface are modeled through Heaviside step functions

\[
H(x) = \begin{cases} 
1 & \text{above } \Gamma_{c+} \\
-1 & \text{below } \Gamma_{c-} 
\end{cases}
\] (5.7)

where \( \Gamma_{c-} \) and \( \Gamma_{c+} \) refer to the two surfaces on either side of the crack interface \( \Gamma_c \).

The \( F_j(x) \) are given in local polar coordinates \((r, \theta)\) as

\[
F_j(r, \theta) = \begin{pmatrix} 
\sqrt{r} \sin \left( \frac{\theta}{2} \right), \\
\sqrt{r} \cos \left( \frac{\theta}{2} \right), \\
\sqrt{r} \sin \left( \frac{\theta}{2} \right) \sin(\theta), \\
\sqrt{r} \cos \left( \frac{\theta}{2} \right) \sin(\theta)
\end{pmatrix}
\] (5.8)

Enveloping \( H(x) \) and \( F_j(x) \) with standard basis functions provides local support near the crack and ensures that the new set of basis functions define a partition of unity.

Figure 5.2 graphically illustrates the idea. Additional material can be found in [4, 6, 44, 46, 47]. In the current work, only small-deformation, linear-elastic fracture mechanics is considered. If plasticity were to be introduced, the singularity at the crack tip may disappear (perfect plasticity), may become of a different power (hardening plasticity) or may be completely unknown as analytical solutions are not generally available (thermo-plasticity, visco-plasticity, shear banding, etc.). In such cases, the use of tip-enrichment functions (5.8) is not valid anymore and typically only the Heaviside function (5.7) is employed in the discretization. Thus, on one hand, the material law becomes more complicated and requires nonlinear solution techniques (e.g. global Newton iterations with return mapping algorithms etc.), but on the other hand, the XFEM model becomes simpler. Furthermore, since the focus of this work is on AMG preconditioning of the XFEM linear systems, crack propagation models and the enforcement of non-penetrating condition of cracks
are not considered in this work.

Figure 5.2: (a) Thick lines (red) depict cracks. Circles (green) and squares (red) give nodes enriched by $H(x)$ and $F_j(x)$, respectively. (b) Computed stress $\sigma_{yy}$.

5.2.3 Levelset Functions

Levelset functions are now described as they play a role in the proposed multigrid algorithm. The XFEM enrichment functions are developed by utilizing a normal levelset function $\psi(x)$ and a tangential levelset function $\phi(x)$ for each crack. These functions represent a local coordinate system oriented along the crack with the origin at the crack-tip as shown in Figure 5.3. The signs of the two functions effectively define four subregions of the domain. Levelset functions are used to identify whether a crack completely cuts across an element (signifying that Heaviside enrichment functions should be added) or whether the tip is located within an element (signifying that tip enrichments should be added). This is done by simply evaluating $\psi(x)$ and $\phi(x)$ at nodes defining each element. $H(x)$ is then determined by the sign of $\psi(x)$ and the polar-coordinates $(r, \theta)$ for specifying tip-enrichment are

$$ r(x) = \sqrt{\psi^2(x) + \phi^2(x)} $$
$$ \theta(x) = \tan^{-1}\left(\frac{\psi(x)}{\phi(x)}\right). $$
This concept is generalized for multiple cracks using levelset functions defined for each crack interface and each tip.

![Levelset functions for a single crack effectively subdividing Ω into 4 regions.](image)

Figure 5.3: Levelset functions for a single crack effectively subdividing Ω into 4 regions.

In the three-dimensional implementation of XFEM, only planar cracks bounded by general polygonal edges are considered. The crack planes are represented in terms of the equation of a plane (5.9). This facilitates the easy evaluation of the levelset functions with respect to the crack, e.g., the left-hand-side of (5.9) is itself the normal levelset $\Phi_n(x)$ or the perpendicular distance from the point $x$ to the crack plane. The tangent levelset at a point $\Phi_t(x)$ denotes the perpendicular distance to the closest edge of the crack along the crack plane. This is evaluated by first projecting the point to the crack plane and then finding the perpendicular distance to the crack edges represented by their corresponding line equations. The use of levelsets simplifies the identification of nodes to be enriched, the evaluation of the enrichment functions, and it also facilitates the modeling of propagating cracks. A graphical representation of the levelset functions in three dimensions is shown in Fig. 5.4.

$$Ax + By + Cz + D = 0$$  \hspace{1cm} (5.9)

From a multigrid perspective, it is important to notice that the evaluation of levelset functions requires coordinates. This type of information is not normally available within an algebraic multigrid solver, but will be used in the proposed quasi-algebraic approach. In particular, fine level coordinates and the levelset functions must be supplied along with the fine level discretiza-
5.2.4 XFEM Linear System

The discrete solution in (5.6) within an element $e$ may be written in terms of an augmented shape-function matrix $\mathbf{N}_{enr}$ and the nodal degrees of freedom $\mathbf{u}^{enr}_{enr}$ (which includes $u^e_I$, $a^e_I$ and $b^e_{Ij}$):

$$\mathbf{u}^h(x^e) = \mathbf{N}^e_{enr} \mathbf{u}^{enr}_{enr}$$  \hfill (5.10)

A gradient operator is applied to the augmented shape-function matrix to obtain

$$\mathbf{B}^e_{enr} = \nabla \mathbf{N}^e_{enr}$$  \hfill (5.11)

The XFEM stiffness matrix for the element $e$ is then written as

$$\mathbf{A}^e = \int_{\Omega^e} (\mathbf{B}^e_{enr})^T \mathbf{D} \mathbf{B}^e_{enr} d\Omega^e.$$  \hfill (5.12)

It should be noted that numerical quadrature must take into account the presence of discontinuous fields in $\mathbf{B}^{enr}$.

The global stiffness matrix $\mathbf{A}$ is obtained by the assembly of the element stiffness matrices. Grouping all the standard and the enriched degrees of freedom results in a linear system of the
following form:

\[
\begin{bmatrix}
A_{rr} & A_{rx} \\
A_{xr} & A_{xx}
\end{bmatrix}
\begin{bmatrix}
u_r \\
u_x
\end{bmatrix} =
\begin{bmatrix}
\tilde{f}_r \\
\tilde{f}_x
\end{bmatrix}
\] (5.13)

where \(u_r\) are associated with regular nodal degrees-of-freedom and \(u_x\) are associated with special degrees-of-freedom. The submatrix \(A_{rr}\) is a standard nodal finite element discretization without any cracks. \(A_{xx}\) contains couplings between special degrees-of-freedom. This includes contributions from Heaviside and tip functions. Cracks within two dimensional domains can essentially be mapped to a single dimension. Figure 5.5 illustrates an XFEM mesh and the corresponding sparsity pattern of \(A_{xx}\). The dense portions are associated with the tip enrichments.

Figure 5.5: (a) XFEM mesh for a single crack (b) Sparsity pattern of \(A\) (c) Sparsity pattern of \(A_{xx}\)
5.3 Algebraic Multigrid

The primary solution method used in this section is the algebraic multigrid method. It is well known that multigrid is effective for solving many discrete partial differential equations, see e.g. [14, 26, 69]. 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, \( G^k \), 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:

\[
A^{k+1} = (P^k)^T A^k P^k
\]  

(5.14)

where \( P^k \) interpolates from \( G^{k+1} \) to \( G^k \) and \( A^k \) is the discretization on \( G^k \). A sample multilevel iteration is given in Figure 5.7 to solve

\[
A^0 u^0 = b^0.
\]  

(5.15)

To complete the specification, relaxation \( R^k \) and the \( P^k \) must be defined. The key to fast convergence is their complementary nature; errors not reduced by \( R^k \) must be well interpolated by \( P^k \).

Algebraic multigrid differs in that the \( G^k \)'s 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 (5.14) to recursively generate a hierarchy. In Chapter 6, an energy minimizing philosophy is followed to generate the \( P^k \) and in Chapter 7, a smoothed aggregation framework is adopted. The details of these particular philosophies are not critical for this work. Energy minimization is chosen in the former case because nonstandard prolongator sparsity patterns are easily incorporated. A brief description is given where super-
Figure 5.6: Multigrid V-cycle

// Solve $A^k u^k = b^k$
procedure multilevel($b^k$, $u^k$, $k$)
  $u^k = R^k(A^k, b^k, u^k)$;
  if ($k \neq \ell$)
    $r^k = b^k - A^k u^k$;
    $u^{k+1} = 0$;
    $u^{k+1} = \text{multilevel}((P^{[k]}(r^k), u^{[k+1]}, k+1))$;
  $u^{[k]} = u^k + P^{[k]} u^{[k+1]}$;
  $u^{[k]} = R^{[k]}(A^{[k]}, b^{[k]}, u^{[k]})$;

Figure 5.7: Multigrid V cycle consisting of $\ell$ levels to solve $A^{[0]} u^{[0]} = b^{[0]}$, where level 0 is the finest level.
scripts indicating level are omitted to simplify the presentation; more information can be found in [78, 13, 41, 51, 12, 11, 75, 81, 34, 73]. The basic principle is that interpolation is chosen to minimize the energy norm of prolongator basis functions subject to constraints. That is,

$$P = \arg\min_P \sum_j \|P_j\|_A \quad \text{and} \quad PB^C = B, \text{ with } P \in \mathcal{N}, \quad (5.16)$$

where $P_j$ is the $j^{th}$ column of $P$, $B$ ($B^C$) is a column matrix of fine (coarse) level near null-space modes, $P \in \mathcal{N}$ 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 work, these vectors are rigid body modes (i.e., $m = 3$ in two dimensions and $m = 6$ in three 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 [10, 43]. Energy minimization in conjunction with the constraint $PB^C = B$ accomplishes this. Figure 5.8 provides an illustration. The figure shows two interpolants to a sine wave, the eigenvector with the smallest nonzero eigenvalue of a one dimensional periodic Poisson operator. While a constant is contained within the span of both sets of basis functions, the lower energy basis functions more accurately capture the sine wave.

Sparsity patterns play a key role in this work. Recall that $P_j$ is the $j^{th}$ coarse level basis function. Thus, $\mathcal{N}$ determines the nonzero support associated with coarse level basis functions. The number of nonzeros in the coarse level discretization matrix (defined via (5.14)) grows as overlap between basis functions increases. However, less overlap gives rise to higher energy basis functions due to
the presence of more constraints in (5.16). Thus, \( N \) must balance these considerations. There are many possible choices, however, a smoothed aggregation philosophy is now discussed \[72, 71\].

The first step is to define a strong-coupling matrix via

\[
\hat{A}_{ij} = \begin{cases} 
1 & \text{if } |A_{ij}| > \varepsilon \sqrt{A_{ii}A_{jj}} \\
0 & \text{otherwise}
\end{cases} \tag{5.17}
\]

where \( \varepsilon \) is a user-specified drop tolerance to exclude weak connections and \( \hat{A} \) has the same dimensions as \( A \). Then, the degree-of-freedom matrix is transformed to a nodal matrix. In particular, let \( d(i) \) be a map which provides the degrees-of-freedom corresponding to the \( i^{th} \) mesh node and define \( \tilde{A} : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n} \) such that

\[
\tilde{A}_{ij} = \max(\hat{A}_{d(i),d(j)}) \tag{5.18}
\]

where \( n \) is the number of nodes. That is, \( \tilde{A}_{ij} \) is nonzero only if there is a strong coupling in the submatrix associated with degrees-of-freedom at nodes \( i \) and \( j \). A matrix graph is then constructed which occupies the role of \( G \). Specifically,

\[
G = \{ V, E \} \tag{5.19}
\]

with vertices

\[
V = \{ 1, 2, \ldots n \} \tag{5.20}
\]
and undirected edges
\[ E = \{(i, j) : i, j \in V, \ j \leq i, \ \tilde{A}_{i,j} \neq 0\}. \] (5.21)

In the current notation, \((i, j)\) and \((j, i)\) refer to the same undirected edge. \(G\) is automatically coarsened by aggregating neighboring vertices together. Each aggregate defines a vertex on the next coarser mesh. Formally, the \(q\)th aggregate corresponds to a set \(A(q)\) such that
\[ A(q) \cap A(p) = \emptyset \ q \neq p \quad \text{and} \quad V = \bigcup_{j=1}^{\hat{n}} A(j) \] (5.22)
where \(\hat{n}\) is the total number of aggregates and \(\emptyset\) is the empty set. For details on aggregation, readers may refer to [72, 70]. The goal is to create ideal aggregates which consist of a single central vertex and all of its immediate neighbors. While it is not usually possible to coarsen completely with ideal aggregates, a large fraction of the computed aggregates are typically ideal.

An initial prolongator guess (or tentative prolongator) to the energy minimization process is now given by
\[
P^{\text{tent}} = \begin{bmatrix} Q_1 & 0 & \ldots \\ 0 & Q_2 & 0 & \ldots \\ \vdots \\ \ldots & 0 & Q_{\hat{n}} \end{bmatrix}, \quad B^C = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_{\hat{n}} \end{bmatrix},
\]
(5.23)
where first \(B_{(i)}\) is defined as a submatrix of \(B\) obtained by taking only rows associated with degrees-of-freedom within nodes assigned to \(A(i)\) and then \(B_{(i)} = Q_{(i)}R_{(i)}\) is constructed via a QR factorization. It follows that \(B = P^{\text{tent}}B^C\). Finally, the sparsity pattern \(\mathcal{N}\) is given by the sparsity pattern of \(\hat{A}P^{\text{tent}}\). Notice that \(P^{\text{tent}}\) contains no overlap between basis functions associated with different aggregates. The sparsity pattern choice extends each basis function’s support by one node in each strong-coupling direction. When used with diameter three aggregates (as in the ideal aggregate case), this generally provides sufficiently low energy basis functions without creating many nonzeros in the Galerkin projection. Figure 5.8 illustrated this for a scalar one dimensional Poisson problem. Each ideal aggregate includes three mesh points giving rise to constant basis functions (corresponding to columns of \(P^{\text{tent}}\)) with three nonzeros while the energy minimizing
basis functions have five nonzeros.

In practice, exact minimization is not needed and with initial guess given by (5.23) only a couple of Krylov-like iterations are sufficient to approximate (5.16). Constraint satisfaction is also straight-forward requiring solution of $n$ linear systems of size $m \times m$. The following reference [51] is provided for additional aspects of energy minimization.

### 5.4 AMG for XFEM

In Fig. 5.9, the convergence of the solution obtained from a “black-box” application of AMG on both an XFEM system and a regular FEM system is shown. The norm of the residual is plotted against the number of conjugate gradient iterations with and without AMG preconditioning. It may be observed from the figure that while the improvement in convergence is observed, AMG is not as effective for XFEM as it is for FEM. To understand this, we must consider the sparsity pattern algorithms described in the previous section in the context of XFEM. As mentioned previously, the grid transfer operators in AMG are obtained via the graph of the matrix constructed from the couplings across various degrees of freedom in the linear system. For a stiffness matrix generated via XFEM, strong couplings do exist across degrees of freedom corresponding to nodes of an enriched element, even on opposite sides of a crack. This indicates that aggregates can span across the cracks, leading to an inaccurate representation of the discontinuities on coarse levels. Additionally, the variability of the number of degrees of freedom in XFEM enriched nodes, pose problems for the AMG preconditioner. Further elaborations on the mismatch of XFEM linear systems for AMG are provided in Chapters 6 and 7 in addition to proposed methods to alleviate the difficulties.
Figure 5.9: Convergence of CG with and without preconditioning
Chapter 6

AMG for XFEM - I

In this chapter, a new quasi-algebraic multigrid method that is suitable for linear systems obtained from discretizing fracture mechanics problems using the extended finite element method is presented. This new method derives from the application of AMG on the Schur Complement of the XFEM matrix. Primarily two-dimensional, static and linear-elastic fracture mechanics implemented on the MATLAB platform is considered. The inputs from co-authors of the paper (Hiriyur et. al. [30]) from which this chapter is reproduced are gratefully acknowledged.

6.1 Schur complement of XFEM matrix

We now consider application of the algebraic multigrid idea to the $2 \times 2$ block system which is repeated here for reference:

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

Unfortunately, the special degrees-of-freedom pose potential problems for algebraic multigrid. One difficulty is the treatment of low energy modes. Recall that constants and more generally rigid body modes correspond to low energy modes. Standard relaxation is not effective on these components and so they must be accurately represented on a coarse level. In the XFEM context, it is natural to ask whether cracks introduce additional low energy modes which must be treated with care. Suppose, for example, that a crack completely traverses an unattached structure (i.e. without Dirichlet boundary conditions) thereby dividing it into two floating subdomains. This
division doubles the number of rigid body modes reflecting the independent movement of the individual subdomains. This larger space can be represented by defining a set of vectors corresponding to standard rigid body modes and augmenting these with the same set of vectors except that entries associated with one of the floating subdomains are redefined to be zero. 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. Thus, these discontinuous modes are not well-addressed by either standard relaxation or the coarse level correction. Of course, cracks do not always completely traverse the domain. While prolongation for partial crossings is somewhat less clear, it seems natural that interpolation should preserve local discontinuities of rigid body modes across crack boundaries. 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 defeats capabilities within most multigrid codes for addressing PDE systems. In particular, PDE systems are often tackled by grouping dofs at each node (as described for creating the nodal matrix $\tilde{A}$ in Section 5.3). 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. 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.

One possible remedy is to algebraically eliminate special degrees-of-freedom from the system.
That is to instead consider the Schur complement

\[ S_{ur} = \tilde{f}_r - A_{rx} A_{xx}^{-1} \tilde{f}_x \quad \text{with} \quad S = A_{rr} - A_{rx} A_{xx}^{-1} A_{xr}. \quad (6.2) \]

Figure 6.1 illustrates a single crack and the stencil at an unknown near the left side of the crack.

![Figure 6.1](image)

**Figure 6.1:** Graphic illustration of Schur complement stencil just left of a crack.

Specifically, a matrix row is displayed of the Schur complement which has undergone symmetric diagonal scaling. The figure plots absolute value of entries associated with the first degrees-of-freedom at corresponding nodal locations. The largest entry corresponds to the matrix diagonal. Entries to the left correspond to couplings away from the crack while those to the right are couplings across the crack. The figure illustrates that this row contains nonzeros along the crack interface but that these nonzeros are small far from the diagonal entry and almost negligible across the crack. This reflects the nature of the problem in that adjacent dofs on opposite sides of a crack are actually quite distant in terms of the underlying physics.

Removal of special degrees-of-freedom effectively renders the problem amenable to algebraic multigrid methods. Table 6.1 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 [51] is used to improve the initial prolongator and generate a final prolongator that approximates (5.16). 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 \( R^{(k)} \) 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 V cycles are used for all the results presented here. It is clear from Table 6.1 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.

Table 6.1: PCG iterations for different AMG approaches on a six crack problem (case 5b in Figure 6.3i).

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Scalar AMG</th>
<th>Variable Block AMG</th>
<th>AMG on Schur Complement</th>
</tr>
</thead>
<tbody>
<tr>
<td>30×30</td>
<td>180</td>
<td>89</td>
<td>10</td>
</tr>
<tr>
<td>60×60</td>
<td>-</td>
<td>103</td>
<td>11</td>
</tr>
<tr>
<td>90×90</td>
<td>-</td>
<td>114</td>
<td>11</td>
</tr>
<tr>
<td>120×120</td>
<td>-</td>
<td>126</td>
<td>12</td>
</tr>
</tbody>
</table>

The remainder of this work focuses on an algebraic multigrid method which avoids the ex-
licit 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 in-
teriors 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 [15].

\section{A new AMG method}

While avoiding the explicit computation of $S$, an algorithm with similar convergence character-
istics to that of applying AMG to the Schur complement is sought. To this end, three Lemmas
relating AMG applied to the Schur complement and AMG applied to the $2 \times 2$ block linear system
are presented. Equivalences between Schur complements and full systems have been explored in
a non-multilevel context in [33].

\textbf{Lemma 6.2.1. Schur complement/projection commutativity.} Consider (6.1) and its associated Schur com-
plement given by (6.2). Let $P$ be a prolongation operator used in a Schur complement multigrid algorithm
and let

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

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

$$
S_H = P^T S P,
$$

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

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

\textbf{Remark 6.2.2.} 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 comple-
ment versus taking the Schur complement of a projected system. In both cases, the following
quantity is computed

$$
P^T A_{rx} A_{xx}^{-1} A_{xr} P.
$$
The difference is in the order in which operations occur. Projection of the full system effectively performs the product $A_{xr}P$. 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 6.2.3.** Assume the full linear system (6.1) is given along with $S = A_{rr} - A_{rx}A_{xx}^{-1}A_{xr}$ and a reduced right hand side $f_r = f_r - A_{rx}A_{xx}^{-1}f_x$. Consider two relaxation procedures corresponding to the full system and the reduced system:

### Reduced Relaxation

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

### Full System Relaxation

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

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

where $r_r$ denotes the residual associated with the Schur system while $\tilde{r}_x$ 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 $u_r$ 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 $2 \times 2$ block system and subtracting this from the right hand side. This results in

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

where it may be noted 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 $\tilde{r}_r = \tilde{f}_r - A_{rx}\tilde{u}_r - A_{rx}\tilde{u}_x$ 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.

Notice that repeated Full System Relaxation sweeps require that step (a) only be performed on the first sweep as it is only before the first step of the first sweep that $\tilde{r}_x$ might be nonzero. Further notice that the initial value of $\tilde{u}_x$ is not used within Full System Relaxation.

**Lemma 6.2.4.** Assume the full linear system (6.1) 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 as described in Figure 5.7 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
• $P^k$ prolongates from level $k+1$ to level $k$

• relaxation consists of $v > 0$ pre- and post-sweeps given by

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

**Full System Multigrid**

• prolongation from level $k+1$ to level $k$ is accomplished via

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

• relaxation consists of $v > 0$ pre- and post-sweeps given by

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

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

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.

**Proof.** The Lemma obviously holds for $m = 1$ as Lemma 6.2.3 directly applies when relaxation is used on the coarsest level and a direct solver is simply a special case of Lemma 6.2.3 associated with $M_{rr}^{[m]} = S^{[m]}$. The proof is completed by induction. Assume the Lemma is true for $m - 1$ level versions of the multigrid procedures, and show that the Lemma holds for $m$ level versions of the multgrid procedures.

For the finest grid (i.e. $k = 1$) pre-relaxation of the $m$ level AMG versions, Lemma 6.2.3 applies so that after pre-relaxation $u_{r}^{[1]} = \tilde{u}_{r}^{[1]}$ and $r_{x}^{[1]} = 0$. A recursive invocation of the multilevel procedures follows the finest grid relaxation and this defines the second level of the $m$ level methods. This recursive invocation, however, also coincides with the finest grid of a $m - 1$ level multigrid method. Thus, the solutions coincide after this recursive invocation if we use the assumption that the Lemma is true for $m - 1$ versions of the multigrid procedures. To apply the $m - 1$ version of the Lemma, however, the Lemma’s assumptions (initial guesses, right hand sides, and linear systems) hold for this recursive invocation must first be validated. That is,

$$\begin{align*}
\tilde{u}_r^{[2]} &= \tilde{u}_r^{[2]} & \text{for the initial guesses} \\
S_{r}^{[2]} &= A_{rr}^{[2]} - A_{r}^{[2]}(A_{xx}^{[2]})^{-1}A_{r}^{[2]} \\
f_{r}^{[2]} &= f_{r}^{[2]} - A_{r}^{[2]}(A_{xx}^{[2]})^{-1}f_{x}^{[2]}
\end{align*}$$

(6.4)

The first statement is trivially true as coarse level initial guesses ($u_{r}^{[2]}$ and $\tilde{u}_{r}^{[2]}$) are taken as zero within a V cycle procedure. The second statement follows directly from Lemma 6.2.1. After pre-relaxation, the projected residual of the reduced system is the right hand side for the next level
and so
\[ f_{rr}^{[2]} = (P^{[1]})^T (f_{rr}^{[1]} - S^{[1]} u_{rr}^{[1]}), \]
\[ = (P^{[1]})^T (f_{rr}^{[1]} - A_{rr}^{[1]} (A_{xx}^{[1]})^{-1} f_x^{[1]} - S^{[1]} u_{rr}^{[1]}), \]
\[ = (P^{[1]})^T (f_{rr}^{[1]} - A_{rr}^{[1]} (A_{xx}^{[1]})^{-1} f_x^{[1]} - A_{rr}^{[1]} u_{rr}^{[1]} + A_{rr}^{[1]} (A_{xx}^{[1]})^{-1} A_{rr}^{[1]} u_{rr}^{[1]}). \]

The first component of the projected residual for the full system is
\[ f_{rr}^{[2]} = (P^{[1]})^T (f_{rr}^{[1]} - A_{rr}^{[1]} u_{rr}^{[1]} - A_{rr}^{[1]} u_{rr}^{[1]}), \]
\[ = (P^{[1]})^T (f_{rr}^{[1]} - A_{rr}^{[1]} u_{rr}^{[1]} - A_{rr}^{[1]} (A_{xx}^{[1]})^{-1} (f_x^{[1]} - A_{rr}^{[1]} u_{rr}^{[1]}))%, \]
\[
\text{where } r_x^{[1]} = 0 \text{ after pre-relaxation is used to eliminate } u_{rr}^{[1]} \text{ above. As } f_x^{[2]} = f_{rr}^{[1]} = 0, \text{ it follows that (6.4) holds and so the interpolated corrections coincide (} P_{rr}^{[1]} u_{rr}^{[2]} = P_{rr}^{[1]} u_{rr}^{[2]} \text{) after recursive invocation as the } m - 1 \text{ level version of the Lemma is assumed. The proof is completed by recognizing that the assumptions of Lemma 6.2.3 are satisfied for post-relaxation and so that } V \text{ cycle terminates with identical solutions for the regular degrees-of-freedom.}
\]

The last Lemma establishes that a multigrid cycle can be constructed for the $2 \times 2$ 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 $P^{[k]}$ be defined without relying on $S^{[k]}$. Additionally, an efficient procedure for solving systems of the form $A_{rr}^{[k]} w = b$ is needed within relaxation. Unfortunately, $M_{rr}^{[k]}$ and $P^{[k]}$ would normally employ a Schur complement matrix in their definition and so it is not practical to make an entirely equivalent multigrid cycle based on the $2 \times 2$ block system. Instead, an approximation is pursued.

Prolongator construction is based on energy minimization which in turn utilizes $S$ in the definition of energy and in the algorithm for building a prolongator sparsity pattern via $\hat{S}_{\text{tent}}$ where again superscripts indicating level have been dropped. Recall that in the experiments associated with Table 6.1, $\hat{S}$ is defined by not only removing small entries from $S$, but also by removing entries that do not appear in $A_{rr}$. This second removal avoids large stencils which give rise to large aggregates and dense sparsity patterns. First, consider how $\hat{S}$ and $A_{rr}$ differ. In particular, the sparsity pattern of $\hat{S}$ is identical to $\hat{A}_{rr}$ for entries associated with mesh locations far from any crack. This is due to the fact that $A_{rs}$ and $A_{xr}$ contain only nonzeros for dofs adjacent to cracks. For each crack, $S$ contains a dense block for all dofs along the crack. However, all entries not ap-
pearing in $A_{rr}$ were removed when generating Table 6.1. This means that the difference between $\hat{S}$ and $\hat{A}_{rr}$ is the coupling across cracks. As Figure 6.1 illustrated, this coupling is small in the Schur complement matrix and so it is eliminated when small entries are dropped. To mimic this effect without forming a Schur complement, the definition of $\hat{A}_{rr}$ is modified so that crack couplings are removed. This modification affects the aggregation (which gives rise to $P^{tent}$) and the resulting sparsity pattern given by $\hat{A}_{rr} P^{tent}$ (see Figure 6.2).

The only remaining occurrence of the explicit Schur complement during prolongator construction is the use of $S$ in the definition of energy. This is also replaced with $A_{rr}$. As stated earlier, $A_{rr}$ and $S$ are identical far from cracks. They are most dissimilar for nonzeros corresponding to crack crossings. However, these nonzeros do not play a significant role in the energy minimization process as crack crossings are not permitted in prolongator basis functions (due to truncation of the prolongator sparsity pattern). Numerical justification is given shortly to access the effects of these approximations.

To complete the multigrid cycle, relaxation must be defined without forming Schur complements. Assume, for now, that a factorization of $A_{xx}$ is available and so the only remaining issue is generating the $M_{rr}$. If a Jacobi smoother is used, then $M_{rr}$ is the diagonal of the Schur complement. Once again, far from cracks $A_{rr}$ and $S$ correspond. It is only for degrees-of-freedom adjacent to crack that they differ. In principle, one could use a technique such as probing [59] to approximate this diagonal. Probing basically combines coloring techniques with matrix-vector products to approximate a matrix. As will be shown, however, simply using the diagonal of $A_{rr}$ to define $M_{rr}$ (e.g., taking the diagonal for Jacobi smoothing or the lower triangular part for Gauss-Seidel) does not seriously impact convergence.

### 6.3 Algorithm Details

The new AMG algorithm uses a hybrid prolongator given by (6.3). $P$ 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 \texttt{Drop}. A simplistic version is detailed in Algorithm 6.3.1. As \( \phi \) and \( \psi \) 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.

Algorithm 6.3.2 summarizes prolongation construction.

\begin{algorithm}[h]
\caption{Algorithm 6.3.1 $\hat{A}_{rr} \leftarrow \text{Drop}(A_{rr}, \text{BlkSize}, \text{coords}, \text{tol}, \phi, \psi)$}
\begin{algorithmic}
\Require matrix $A_{rr}$ with constant blocksize of \text{BlkSize}.
\Require coordinate locations \text{coords} of nodes
\Require drop tolerance \text{tol} for removing small entries
\Require levelset functions $\phi$, $\psi$ representing cracks
\State $\hat{A}_{rr} \leftarrow \text{StandardDrop}(A_{rr}, \text{tol})$
\For {$i = 1$ to dimension($A_{rr}$)}
\State $C_i \leftarrow \{j \mid [\hat{A}_{rr}]_{ij} \neq 0\}$
\ForAll {$j \in C_i$}
\State $(x_1, y_1) \leftarrow (\text{coords}(i/\text{BlkSize}, 1), \text{coords}(i/\text{BlkSize}, 2))$
\ForAll {each crack $c$}
\If{$\psi^c(x_1, y_1)\psi^c(x_2, y_2) < 0$ AND $\phi^c(x_1, y_1) \leq 0$}
\State $[\hat{A}_{rr}]_{ij} \leftarrow 0$
\State $[\hat{A}_{rr}]_{ji} \leftarrow 0$
\EndIf
\EndFor
\EndFor
\EndFor
\end{algorithmic}
\end{algorithm}

Algorithm 6.3.3 depicts the relaxation which avoids explicit construction of the Schur comple-
ment and uses $A_{rr}$ to define $M_{rr}$. Table 6.2 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 (case 5a depicted in Figure 6.3i). $P(C, G)$ 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 6.2.3. 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 6.2.3 while the rightmost columns use either symmetric Gauss-Seidel (indicated by GS) or a direct solver to define relaxation components. AMG V cycles are used for all the results presented here. The reader should note that the third and fourth columns are 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 Lemma 6.2.4. 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.

Table 6.2: Assessment of Schur complement approximations for case 5a in Figure 6.3i

<table>
<thead>
<tr>
<th>mesh size</th>
<th>Schur</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>$30^2$</td>
<td>$P(S, \hat{S})$</td>
<td>$P(A_{rr}, \hat{S})$</td>
</tr>
<tr>
<td>$60^2$</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>$90^2$</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>$120^2$</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>
Algorithm 6.3.2 $P \leftarrow \text{ConstructP}(B, A_{\text{rr}}, \text{BlkSize}, \text{coords}, \text{tol}, \phi, \psi, s)$

Require: column matrix $B$ of rigid body modes
Require: matrix $A_{\text{rr}}$ with constant blocksize of $\text{BlkSize}$
Require: coordinate locations $\text{coords}$ of nodes
Require: drop tolerance $\text{tol}$ for removing small entries
Require: levelset functions $\phi, \psi$ representing cracks
Require: number of enrichment degrees-of-freedom $s$

$\hat{A}_{\text{rr}} \leftarrow \text{Drop}(A_{\text{rr}}, \text{BlkSize}, \text{coords}, \text{tol}, \phi, \psi)$ via Algorithm 6.3.1

$\bar{A}_{\text{rr}} \leftarrow \text{MakeNodalMatrix}(\hat{A}_{\text{rr}}, \text{BlkSize})$

$A_{(1)} \ldots A_{(n)} \leftarrow \text{StandardAggregation}(\bar{A}_{\text{rr}})$

$p_{\text{tent}}, B^C \leftarrow \text{StandardInitialProlongator}(B, A_{(1)} \ldots A_{(n)}, \text{BlkSize})$

$\mathcal{N} \leftarrow \text{Pattern}(\hat{A}_{\text{rr}}, p_{\text{tent}})$

$\mathcal{P} \leftarrow \text{StandardEnergyMinimization}(A_{\text{rr}}, p_{\text{tent}}, B, B^C, \mathcal{N})$

$P \leftarrow \begin{bmatrix} \mathcal{P} & 0 \\ 0 & I_s \end{bmatrix}$ where $I_s$ is an $s \times s$ identity matrix

$\text{coords} \leftarrow \text{AverageCoords(\text{coords}, A_{(1)} \ldots A_{(n)})}$

Algorithm 6.3.3 $u \leftarrow \text{HybridSmoothers}(A, u, f, \mathcal{R}_r, \mathcal{R}_x, s)$

Require: matrix $A$ with $s$ degrees-of-freedom corresponding to enrichments
Require: initial guess $u$
Require: right hand side $f$

Require: SubSmoothers: $\mathcal{R}_r, \mathcal{R}_x$

Apply one sweep of $\mathcal{R}_x$ to smooth on: $A_{xx}u_x = f_x - A_{xr}u_r$
Apply one sweep of $\mathcal{R}_r$ to smooth on: $A_{rr}u_r = f_r - A_{rx}u_x$
Apply one sweep of $\mathcal{R}_x$ to smooth on: $A_{xx}u_x = f_x - A_{xr}u_r$
6.4 Numerical Results

Crack configurations are shown in Figure 6.3 corresponding to different numbers of cracks, cracks of different length, different crack orientations, and completely interior cracks. The aggregation of a few sample cases is displayed in Figure 6.4. Aggregates for horizontal cracks tend to have somewhat regular rectangular shapes (though some aggregates are larger than others). Aggregates for inclined cracks tend not to follow any particular pattern (though no aggregates cross cracks due to the levelset dropping).

The total number of preconditioned conjugate gradient iterations are presented in Table 6.3 in order to achieve a residual reduction of $10^{-8}$. In all cases, the initial guess is zero and the right hand side corresponds to fixed Dirichlet boundary condition on the bottom boundary followed by Neumann conditions on all other boundaries. The side edges are traction free while uniform normal traction is applied on the top edge. 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 work. It uses the hybrid prolongator described in Algorithm 6.3.2 in conjunction with the specialized dropping procedure given in Algorithm 6.3.1. A constant block size of two (at the finest level) is used to construct $P$. 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 to be discussed. Only the VBlk AMG uses the more expensive block symmetric Gauss-Seidel relaxation. The HybridStandard AMG method also uses Algorithm 6.3.2 to build a hybrid prolongator. The key difference, however, is that specialized dropping which removes crack crossing is not used when constructing $\hat{A}_{rr}$. Instead, standard dropping given by (5.17) is used for HybridStandard AMG as well as for VBlk AMG. This makes both VBlk AMG and HybridStandard AMG truly algebraic as levelset...
Figure 6.3: 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
functions and coordinates are not needed. For standard dropping in all methods, \( \epsilon \) is \( 8 \cdot 10^{-2} \).

Similar to Quasi AMG, a block size of two is used during coarsening. AMG V cycles are used for all the results presented here. In all cases, one Krylov-like iteration is used to improve the initial prolongator and generate a final prolongator that approximates (5.16). 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. It is emphasized that the performance of standard AMG deteriorates as the number of cracks increase while the Quasi AMG method is not affected as much. Therefore the computational efficiencies introduced by the proposed method are further advantageous when multiple cracks are present.

Table 6.4 concludes with AMG operator complexities for the Quasi AMG method associated with Case 5b. Operator complexity is normally defined as

\[
\text{complexity} = \frac{\sum_{k=1}^{\ell} \text{nnz}(A^k)}{\text{nnz}(A^1)}
\]

where \( \ell \) is the total number of multigrid levels and \( \text{nnz}(C) \) is the number of nonzero entries in the matrix \( C \). Thus, the operator complexity gives a measure of how much additional memory is required to store the coarse level discretization operators as well as the computational work.
Table 6.3: Preconditioned CG iterations for two-dimensional problems

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Case</th>
<th># of dofs</th>
<th>VBlk AMG</th>
<th>Hybrid AMG</th>
<th>Quasi AMG</th>
<th>Case</th>
<th># of dofs</th>
<th>VBlk AMG</th>
<th>Hybrid AMG</th>
<th>Quasi AMG</th>
</tr>
</thead>
<tbody>
<tr>
<td>30^2</td>
<td>1a</td>
<td>1900</td>
<td>28</td>
<td>13</td>
<td>11</td>
<td>3a</td>
<td>2676</td>
<td>154</td>
<td>-</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7372</td>
<td>29</td>
<td>15</td>
<td>10</td>
<td></td>
<td>8784</td>
<td>127</td>
<td>-</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16444</td>
<td>41</td>
<td>18</td>
<td>14</td>
<td></td>
<td>18492</td>
<td>-</td>
<td>-</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>29116</td>
<td>41</td>
<td>20</td>
<td>14</td>
<td></td>
<td>31800</td>
<td>-</td>
<td>-</td>
<td>28</td>
</tr>
<tr>
<td>60^2</td>
<td>1b</td>
<td>1948</td>
<td>24</td>
<td>22</td>
<td>11</td>
<td>3b</td>
<td>2772</td>
<td>-</td>
<td>-</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7468</td>
<td>24</td>
<td>29</td>
<td>12</td>
<td></td>
<td>8808</td>
<td>-</td>
<td>-</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16588</td>
<td>40</td>
<td>37</td>
<td>18</td>
<td></td>
<td>18444</td>
<td>-</td>
<td>-</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td></td>
<td>29308</td>
<td>38</td>
<td>42</td>
<td>17</td>
<td></td>
<td>31680</td>
<td>-</td>
<td>-</td>
<td>26</td>
</tr>
<tr>
<td>90^2</td>
<td>1c</td>
<td>1996</td>
<td>31</td>
<td>31</td>
<td>13</td>
<td>4</td>
<td>2436</td>
<td>116</td>
<td>107</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7564</td>
<td>32</td>
<td>43</td>
<td>14</td>
<td></td>
<td>8328</td>
<td>102</td>
<td>154</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16732</td>
<td>54</td>
<td>55</td>
<td>21</td>
<td></td>
<td>17820</td>
<td>167</td>
<td>199</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td></td>
<td>29500</td>
<td>52</td>
<td>62</td>
<td>20</td>
<td></td>
<td>30912</td>
<td>173</td>
<td>-</td>
<td>25</td>
</tr>
<tr>
<td>120^2</td>
<td>2a</td>
<td>2132</td>
<td>64</td>
<td>57</td>
<td>15</td>
<td>5a</td>
<td>2436</td>
<td>80</td>
<td>76</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7808</td>
<td>52</td>
<td>80</td>
<td>14</td>
<td></td>
<td>8328</td>
<td>91</td>
<td>107</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>17084</td>
<td>100</td>
<td>102</td>
<td>26</td>
<td></td>
<td>17820</td>
<td>135</td>
<td>134</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>29960</td>
<td>106</td>
<td>117</td>
<td>23</td>
<td></td>
<td>30912</td>
<td>153</td>
<td>155</td>
<td>19</td>
</tr>
<tr>
<td>30^2</td>
<td>2b</td>
<td>2676</td>
<td>73</td>
<td>59</td>
<td>16</td>
<td>5b</td>
<td>2436</td>
<td>89</td>
<td>81</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8784</td>
<td>72</td>
<td>81</td>
<td>17</td>
<td></td>
<td>8328</td>
<td>103</td>
<td>116</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>18492</td>
<td>109</td>
<td>108</td>
<td>27</td>
<td></td>
<td>17820</td>
<td>149</td>
<td>147</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>31800</td>
<td>108</td>
<td>123</td>
<td>24</td>
<td></td>
<td>30912</td>
<td>165</td>
<td>170</td>
<td>19</td>
</tr>
</tbody>
</table>
associated with operating on a hierarchy as opposed to a single level method. In particular, the work per V cycle is proportional to the operator complexity. In Table 6.4 the column labeled “full complexity” corresponds to this. The “sub-block complexity” column redefines $\text{nnz}(A^{[k]})$ so that nonzeros in $A^{[k]}_{xx}$ are excluded. As the $A_{xx}$ block is identical on all levels, it need not be replicated each time within the multigrid hierarchy. Thus, sub-block complexity more accurately reflects the required storage for a more sophisticated implementation. In general, complexities less than two are considered acceptable which is the case for the presented results. Obviously, this depends on the number of cracks present in the problem. It should be noted that these operator complexities are about 25% higher than standard smoothed aggregation operator complexities for model two dimensional elasticity problems. Thus, we would expect that one iteration of the Quasi AMG method would roughly be about 25% more costly than a standard smoothed aggregation iteration.

Table 6.4: AMG operator complexities for Quasi AMG on Case 5b.

<table>
<thead>
<tr>
<th>Mesh</th>
<th># of levels</th>
<th>full complexity</th>
<th>sub-block complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$30 \times 30$</td>
<td>2</td>
<td>1.673</td>
<td>1.607</td>
</tr>
<tr>
<td>$60 \times 60$</td>
<td>3</td>
<td>1.815</td>
<td>1.716</td>
</tr>
<tr>
<td>$90 \times 90$</td>
<td>3</td>
<td>1.650</td>
<td>1.583</td>
</tr>
<tr>
<td>$120 \times 120$</td>
<td>4</td>
<td>1.699</td>
<td>1.621</td>
</tr>
</tbody>
</table>

6.4.1 Application in 3D

The methods developed in this work are primarily motivated by three dimensional simulations where iterative methods are essential due to the high costs associated with direct solvers. However, the main concern in 3D is that cracks become essentially two-dimensional i.e. surface entities, and tips become one-dimensional. To validate the applicability of the proposed AMG solution process for three dimensional problems, a sample block with three planar cracks is considered as shown in Figure 6.5. In this case, only Heaviside enrichments are considered. Table 6.5 shows the number of preconditioned CG iterations for various mesh-sizes and the number of AMG levels employed. The AMG operator complexities - full (FC) and sub-block (SC), are presented in Ta-
From the table, it may be observed that iteration counts remain nicely behaved, though there is a slight growth as we refine. While the AMG operator complexities are quite reasonable, they are higher than in the two dimensional case. The bulk of this is due to a technical issue in how
sparsity patterns are chosen and a somewhat inefficient aggregation algorithm. In particular, the coarse operators have six degrees-of-freedom per node as opposed to three degrees-of-freedom on the fine mesh due to the presence of six near null space vectors in three dimensions. This six degrees-of-freedom per node is certainly not required and was only used to mimic smoothed aggregation sparsity patterns. Based on prior non-fracture experimentation, it is suspected that a more judicious pattern choice could noticeably lower complexities without significantly affecting convergence, though further experimentation would be needed to verify this.

6.5 Extensions

The algorithm proposed in this work only applies coarsening to the $A_{rr}$ block of the XFEM linear system. When the percentage of elements with enrichment functions is relatively modest, then this type of algorithm is indeed practical (though there may be some parallelization concerns). However, when the dimensions of $A_{xx}$ are relatively large, the lack of coarsening on the enrichment degrees-of-freedom can become prohibitively expensive. This often happens when many micro cracks are modeled or in three dimensions when cracks are surface entities. Preliminary experience indicates that coarsening degrees-of-freedom associated with Heaviside enrichments is not problematic. In particular, a block diagonal prolongator can be developed using $\mathcal{P}$ for regular dofs, $\hat{\mathcal{P}}$ for Heaviside enrichment dofs, and $I_s$ for tip enrichment dofs where $s$ is the number of dofs with tip enrichments. Somewhat standard AMG methods can then be used to develop $\hat{\mathcal{P}}$. This is due to the fact that the associated basis functions (which are the product of Heaviside functions and standard nodal basis functions) are quite similar to typical finite element basis functions. Additionally, Heaviside dofs in $A_{xx}$ on one side of a crack do not contain direct couplings to Heaviside dofs on the other side of a crack and so the crack crossing issues do not arise. These couplings are illustrated in Figure 5.5 where the first narrow banded section corresponds to the lower crack interface while the second narrow band corresponds to the upper crack interface. Coarsening of tip functions appears to be much more problematic as they are fairly nonstandard and highly ill-conditioned basis functions.
Chapter 7

AMG for XFEM - II

In this chapter, a new algebraic multigrid method that is suitable for linear systems obtained from discretizing fracture mechanics problems using the extended finite element method is presented. This method is based on the transformation of the XFEM linear system into an alternate representation based on the Phantom-node variant of XFEM. Three-dimensional cracks, nonlinear material properties and MPI based parallel implementation on the parFEAP and Trilinos platforms are considered. The inputs from authors of the paper (Gerstenberger et. al. [25]) which forms the basis for this chapter are gratefully acknowledged.

7.1 Introduction

Similar to Chapter 6, here an alternative approach will be discussed to adapting algebraic multigrid (AMG) such that it is suitable for linear systems arising from fracture mechanics problems modeled using the extended finite element method (XFEM). In this chapter, an extension of the method proposed in Gerstenberger and Tuminaro [25] to fracture problems in three dimensions is presented. In addition, nonlinear materials and parallel implementation so that the method can be scaled to studying fracture problems at a larger scale are considered.

Here’s a brief outline of this chapter: In Section 7.2.1 a phantom node representation of XFEM for modeling cracks is discussed. It is shown that AMG applied on the phantom node representation of the XFEM problem does retain its optimal convergence properties. The choice of XFEM enrichment functions play an important role in this chapter and this is discussed in Section 7.2.1.
In Section 7.2.2 a simple transformation that exists between the XFEM system modeled with a particular type of enrichment and the phantom node representation is presented. Finally some implementation aspects are discussed in Section 7.3 and the effectiveness of this approach demonstrated via some numerical examples presented in Section 7.4.

### 7.2 Phantom node representation

A phantom node approach for modeling discontinuities has been studied by many researchers [27, 63, 61, 19]. It is a variant of XFEM in the sense that it also incorporates discontinuous shape functions. The key idea is to model the domains on both sides of the crack separately with overlapping discontinuous elements as shown in Fig. 7.1. The discontinuous elements have the nodes on the opposite sides of the respective discontinuities. The discrete equation for displacement in the phantom node approach is given as:

$$u^p = \sum N_I u_I + N_I \Psi_I u_I^a + N_I \Psi_I u_I^b$$

(7.1)

where the enrichment functions $\Psi_I$ take the value of 1 in domain $\Omega_\alpha$ and 0 otherwise. In the above equation, $\alpha$ corresponds to the subdomains $a$ and $b$ on either side of the crack. A graphical description of the shape functions for a two-dimensional quadrilateral element are shown in Fig. 7.2. Let us consider a domain that is cut through with a planar crack. If the crack cuts through the entire domain, one would expect that the stiffnesses from either side of the domain would be
Figure 7.2: Shape functions for a phantom node representation

completely decoupled. This decoupling is naturally represented if regular finite are used elements with double nodes. The decoupling of the stiffness is also seen if a phantom node representation is adopted as can be seen in the sparsity pattern of the global stiffness matrix in Fig. 7.3. Since the stiffness matrix entries are decoupled across the crack, if we use AMG on the phantom node representation, the coarsening in the grid transfer process will retain the discontinuities across all levels.

Figure 7.3: Sparsity pattern of the stiffness matrix for phantom node representation
7.2.1 Choice of enrichment form

Now a transformation is sought that commutes the linear system represented in XFEM into an equivalent linear system obtained from the phantom node representation. In other words, a transformation is sought that decouples the stiffness matrix across strong discontinuities. To this end, some choices for the XFEM enrichment functions to model strong discontinuities are first analyzed.

For the modeling of crack opening, the discrete form for displacement originally proposed in XFEM literature is of the following form:

\[ u(x) = \sum_{i=1}^{nel} N_i(x) u_i + \sum_{i=1}^{nel} N_i(x) \Psi(x) a_i \]  \hspace{1cm} (7.2)

where \( \Psi(x) \) is the enrichment function that models the jump in displacement. It may be noted that in the above form, the enrichment functions are non-zero at the nodes. Therefore the original nodal unknown quantities \( u_i \) do not directly correspond to physical displacements at the nodes and post-processing of the solution is required to extract the physical displacement values.

Therefore an alternative formulation was proposed that retains the discontinuities at the crack interface but is zero at the nodes. In this case, the nodal degrees of freedom \( u_i \) correspond directly to the physical displacements (though post-processing of the entire unknown solution would be required for displacements in the interior of element). This formulation is called the “shifted enrichment” and is written as:

\[ u(x) = \sum_{i=1}^{nel} N_i(x) u_i + \sum_{i=1}^{nel} N_i(x) \left( \Psi_i - \Psi(x) \right) a_i \]  \hspace{1cm} (7.3)

where \( \Psi_i \) corresponds to the value of enrichment function \( \Psi(x) \) at node \( i \).

In this work, an absolute value of the shifted enrichment multiplied by \( \frac{1}{2} \) is used to model the displacement jumps. The reason this form is used is because a simple transformation may be then employed to obtain an equivalent phantom node described in the previous section:

\[ u(x) = \sum_{i=1}^{nel} N_i(x) u_i + \sum_{i=1}^{nel} N_i(x) \frac{1}{2} \left( |\Psi_i - \Psi(x)| \right) a_i \]  \hspace{1cm} (7.4)
The various shape functions and enrichment for a two-dimensional quadrilateral element is shown in Fig. 7.4.

Figure 7.4: Shape functions and enrichments (a) Standard bilinear shape function (b) Original XFEM Heaviside enrichment (c) Shifted enrichment (d) Modified shifted enrichment
7.2.2 Transformation

A simple transformation exists between the XFEM representation with modified shifted enrichments and the phantom node representation. For complete details about this transformation, the reader is referred to [25]. In brief, the coefficients in the transformation matrix corresponding to the regular degrees of freedom \( u_i \) is equal to -1 and the coefficient corresponding to the enriched degree of freedom itself is 1. The rows corresponding to non-enriched degrees of freedom remain unchanged. Thus for a six node element in which node 2 and 3 are enriched, the transformation matrix can be written as follows:

\[
G = \begin{bmatrix}
1 & 1 & 1 & -1 & 1 & 1 \\
1 & 1 & -1 & 1 & -1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
u_3 \\
a_3 \\
u_4 \\
a_4 \\
u_5 \\
u_6
\end{bmatrix}
\]

(7.5)

It may be recalled that the original linear system is of the form:

\[
A u = f
\]

(7.6)

where \( A \) is the XFEM stiffness matrix using the modified shifted enrichment. Using a global transformation matrix \( G \) developed as described herein, the modified linear system is written as follows:

\[
\left( G^T A G \right) \left( G^{-1} u \right) = \left( G^T u \right)
\]

(7.7)

\[
\Rightarrow \bar{A} \bar{u} = \bar{f}
\]

(7.8)
The transformation involves two matrix-matrix multiplies for the stiffness matrix and one matrix-vector multiply for the right hand side. Once the solution for the modified linear system is obtained, to get the solution back in the original representation involves a second matrix-vector multiply:

\[ u = G\tilde{u} \quad (7.9) \]

It may be noted the the transformation matrix \( G \) is extremely sparse i.e., it is essentially an identity matrix with only a few off-diagonal -1 values corresponding to the enriched degrees of freedom.

### 7.2.3 Modified null space

To develop the tentative prolongators \( P^{tent} \) for the smoothed aggregation AMG, the nullspace vectors corresponding to the stiffness matrices are provided. For three dimensional elasticity, the nullspace vectors correspond to a translation and a rotation about each of the three spacial axes.

In general, the six nullspace vectors can be written as follows:

\[
N_\Theta = \begin{bmatrix}
1 & 0 & 0 & -y_1 & 0 & z_1 \\
0 & 1 & 0 & x_1 & -z_1 & 0 \\
0 & 0 & 1 & 0 & y_1 & -x_1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
1 & 0 & 0 & -y_i & 0 & z_i \\
0 & 1 & 0 & x_i & -z_i & 0 \\
0 & 0 & 1 & 0 & y_i & -x_i \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots 
\end{bmatrix}
\begin{bmatrix}
u^x_1 \\
u^y_1 \\
u^z_1 \\
\vdots \\
u^x_i \\
u^y_i \\
u^z_i \\
a^x_i \\
a^y_i \\
a^z_i \\
\vdots 
\end{bmatrix}
\quad (7.10)
\]

where the first three vectors are the translations and the next three vectors are the rotations. The quantities \( x_i, y_i \) and \( z_i \) are the spatial coordinates at node \( i \). The rows corresponding to the en-
riched degrees of freedom are all zero since they do not correspond to physical displacements. In the current approach, even the nullspace vectors need to be transformed to the phantom node representation. This is done through the following operation:

\[ \tilde{N}_\Theta = G^{-1}N_\Theta \]  

(7.11)

It may be noted that this operation is not very expensive because of the block-diagonal nature and extreme sparsity of the transformation matrix. The \( 2 \times 2 \) blocks containing the off-diagonal -1 term can be inverted very inexpensively.

### 7.3 Implementation

Both serial and parallel implementations of XFEM and AMG routines have been developed. The XFEM routines are implemented in the finite element program FEAP [65]. ParFEAP is the MPI based parallel implementation of the program and the XFEM user element routines are also developed in a manner amenable to parallelization. FEAP comes by default with an interface to PETSC [2] for the use of many solver routines. The AMG based solution schemes are developed on the Trilinos [28] platform through a user-solve interface for FEAP. Trilinos is a family of numerical packages developed by Sandia National Laboratories. A new solver class of libraries called XICE is developed and it interfaces with FEAP and calls the relevant Trilinos packages.

The Trilinos package ML provides the AMG routines for the preconditioners while Aztec00 provides the interface to the conjugate gradient implementation. The global stiffness matrix is developed on Trilinos as an Epetra CrsMatrix. In the serial implementation, the assembly is performed directly in FEAP and the global stiffness matrix is passed on to a Trilinos Epetra CrsMatrix in directly in terms of compressed sparse row format. In the parallel implementation, the stiffness matrix assembly is performed within each processor.
7.4 Numerical Results

In this section, some numerical results are presented that compare the computational time and effort in different solvers.

7.4.1 Linear elastic solves in serial

In Table 7.1, a comparison of different solvers for linear elastic solves in serial is provided. Nine different cases are considered. Case 1 corresponds to a non-XFEM case and the rest involve XFEM systems. Figures 7.5 through ?? show some sample cases considered. Case 1 and 2 use the same mesh with the only difference being that case 1 does not have a crack and is therefore a non-XFEM example whereas the rest of the examples consider XFEM systems with different mesh sizes and crack configurations. The column “Direct” corresponds to the case of a direct solver. The Amesos package of Trilinos was used to provide an interface to the KLU algorithm that first provides a factorization of the matrix (which takes the most effort) and then solves the linear system (relatively faster compared to factorization). For this solver, only the CPU time in seconds is shown.

The other columns correspond to iterative solvers and both the CPU time and the number of iterations are mentioned. All the iterative solves use the conjugate gradient method. The column “CG-brute” corresponds to the case where no preconditioner is used. The columns AMG-S1 and AMG-S5 correspond to the conjugate gradient using a “black-box” AMG preconditioner. S1 and S5 refer to the number of pre- and post- smoother sweeps in the AMG preconditioner. The columns XAMG-S1 and XAMG-S5 correspond to the proposed AMG scheme involving the phantom node transformation. Even in this case, the number of pre- and post- sweeps of the smoother is varied. For both of these preconditioning schemes, a symmetric block Gauss-Seidel smoother was used. The number of iterations to reduce the residual norm (relative to the initial state) by a factor of 1e-8 is reported.

It is clear from the results that XAMG preconditioning provides the fastest and most efficient convergence both in terms of CPU time and in terms of the number of Krylov iterations. The num-
Figure 7.5: Representative examples. Tip enrichments are not used in any of the examples shown. Stress contours correspond to Von Mises stress.
Table 7.1: Comparison of different solvers: Direct, CG (unpreconditioned, “black-box” AMG pre-
conditioned and “XAMG“ preconditioned). The number of pre- and post- sweeps of symmetric
Gauss-Seidel smoother is also varied (S1 and S5) for the preconditioner. “nits” is the number of
CG iterations and cpu time is measured in seconds

<table>
<thead>
<tr>
<th>Case</th>
<th>Direct</th>
<th>CG-brute</th>
<th>AMG-S1</th>
<th>AMG-S5</th>
<th>XAMG-S1</th>
<th>XAMG-S5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>time</td>
<td>nits</td>
<td>time</td>
<td>nits</td>
<td>time</td>
<td>nits</td>
</tr>
<tr>
<td>1</td>
<td>0.11</td>
<td>-</td>
<td>58 0.15</td>
<td>31 0.32</td>
<td>25 0.07</td>
<td>15 0.16</td>
</tr>
<tr>
<td>2</td>
<td>0.13</td>
<td>-</td>
<td>80 0.25</td>
<td>43 0.54</td>
<td>27 0.09</td>
<td>18 0.24</td>
</tr>
<tr>
<td>3</td>
<td>4.16</td>
<td>51 2.49</td>
<td>50 0.58</td>
<td>30 1.45</td>
<td>22 0.29</td>
<td>14 0.76</td>
</tr>
<tr>
<td>4</td>
<td>5.44</td>
<td>192 4.17</td>
<td>69 0.86</td>
<td>38 1.95</td>
<td>29 0.42</td>
<td>20 1.17</td>
</tr>
<tr>
<td>5</td>
<td>77.76</td>
<td>102 11.88</td>
<td>70 3.04</td>
<td>40 7.18</td>
<td>23 1.17</td>
<td>16 3.27</td>
</tr>
<tr>
<td>6</td>
<td>100.05</td>
<td>191 17.03</td>
<td>101 4.82</td>
<td>55 10.84</td>
<td>27 1.49</td>
<td>19 4.24</td>
</tr>
<tr>
<td>7</td>
<td>292.88</td>
<td>77 25.43</td>
<td>57 4.46</td>
<td>34 11.14</td>
<td>25 2.23</td>
<td>16 5.78</td>
</tr>
<tr>
<td>8</td>
<td>345.65</td>
<td>78 25.90</td>
<td>64 4.99</td>
<td>36 11.60</td>
<td>26 2.32</td>
<td>17 6.13</td>
</tr>
<tr>
<td>9</td>
<td>313.90</td>
<td>-</td>
<td>138 12.25</td>
<td>81 29.20</td>
<td>35 3.45</td>
<td>24 9.58</td>
</tr>
</tbody>
</table>

The number of smoother sweeps has a positive effect on the number of iterations but it is more expensive
to apply and therefore has a detrimental effect on total solution time. It may be noted that all the
cases considered here involve a single crack. It was shown in Table 6.3, the behaviour of standard
AMG deteriorates for larger number of cracks. The proposed transformation approach will be
further advantageous in such cases and further experiments will be conducted to illustrate this
advantage.

7.4.2 Plasticity solves in serial

Here an example including plasticity is discussed. In this case, as the crack opens, the stress
concentrations near the tip of the crack induce plasticity. Since the behaviour of the material is
nonlinear, this problem involves loading in a series of steps. For each load step, a number of
sub-iterations are performed depending on the state of stress. If the material is still in the elastic
limit, the solution converges in two steps, whereas when the plasticity yield limit is reached, the
simulation involves multiple Newton iterations to find the tangent stiffness and the solution state
that brings to stress state to converge to the yield surface. The plastic zone is localized near the tip
and since the conditioning of the stiffness matrix changes with each the number of iterations also
change. In this case, the stopping criteria was such that the absolute residual norm was reduced
to a magnitude of 1e-7.
Table 7.2 shows the results using the conjugate gradient method without any preconditioner. The column “plets” indicates the number of Newton iteration subsolves required for convergence to be achieved. Each Newton iteration corresponds to a number of AMG iterations where the material tangent stiffness is updated. The column “cumits” refer to to the cumulative total number of AMG iterations for that particular load step. In this case, the CG was limited to a maximum of 500 iterations and in many cases even though the sub-solve iterations did not converge, the approximate solution was sufficient for the Newton method to update the stiffness. Table 7.3 show the results using a “black-box” AMG preconditioner and Table 7.4 shows the results using the AMG preconditioner on the transformed XFEM linear system. Similar to the elastic case, the results show that transformation of the linear system to the phantom node representation greatly enhances the effectiveness of the AMG preconditioner. Typically for amg-preconditioned iterative solves, the computational time scales proportionally with the number of iterations. This is not observed from comparing Tables 7.3 and 7.4 indicating scope for further profiling and code optimization. This effort is currently in progress and it is expected that the reported computational time results for the transformed system will be revised and the relative advantage of the proposed approach would be better demonstrated.

![Simulation involving plasticity: Von Mises stress contours](image)

Figure 7.6: Simulation involving plasticity: Von Mises stress contours
Table 7.2: Solution of a fracture problem involving plasticity using CG-brute

<table>
<thead>
<tr>
<th>Load step</th>
<th># Iterations</th>
<th>Residual norm</th>
<th>Energy Norm</th>
<th>CPU Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Initial Final</td>
<td>Initial Final</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3 501</td>
<td>8.94E+00</td>
<td>5.28E-08</td>
<td>1.36E-02 0.00E+00</td>
</tr>
<tr>
<td>2</td>
<td>3 501</td>
<td>8.94E+00</td>
<td>5.28E-08</td>
<td>1.36E-02 0.00E+00</td>
</tr>
<tr>
<td>3</td>
<td>3 501</td>
<td>8.94E+00</td>
<td>5.28E-08</td>
<td>1.36E-02 0.00E+00</td>
</tr>
<tr>
<td>4</td>
<td>3 501</td>
<td>8.94E+00</td>
<td>5.28E-08</td>
<td>1.36E-02 0.00E+00</td>
</tr>
<tr>
<td>5</td>
<td>4 1006</td>
<td>8.94E+00</td>
<td>7.77E-08</td>
<td>1.36E-02 0.00E+00</td>
</tr>
<tr>
<td>6</td>
<td>10 1445</td>
<td>8.94E+00</td>
<td>3.00E-08</td>
<td>1.36E-02 0.00E+00</td>
</tr>
<tr>
<td>7</td>
<td>10 1767</td>
<td>8.94E+00</td>
<td>8.90E-08</td>
<td>1.36E-02 0.00E+00</td>
</tr>
<tr>
<td>8</td>
<td>12 3582</td>
<td>8.95E+00</td>
<td>6.22E-08</td>
<td>1.36E-02 0.00E+00</td>
</tr>
<tr>
<td>9</td>
<td>24 3181</td>
<td>8.94E+00</td>
<td>1.95E-07</td>
<td>1.36E-02 6.69E-19</td>
</tr>
<tr>
<td>10</td>
<td>26 3654</td>
<td>8.95E+00</td>
<td>2.55E-07</td>
<td>1.36E-02 1.14E-18</td>
</tr>
</tbody>
</table>

Table 7.3: Solution of a fracture problem involving plasticity using “black-box” AMG-S1

<table>
<thead>
<tr>
<th>Load step</th>
<th># Iterations</th>
<th>Residual norm</th>
<th>Energy Norm</th>
<th>CPU Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Initial Final</td>
<td>Initial Final</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2 77</td>
<td>8.94E+00</td>
<td>1.06E-07</td>
<td>1.36E-02 1.79E-19</td>
</tr>
<tr>
<td>2</td>
<td>2 77</td>
<td>8.94E+00</td>
<td>1.06E-07</td>
<td>1.36E-02 1.78E-19</td>
</tr>
<tr>
<td>3</td>
<td>2 77</td>
<td>8.94E+00</td>
<td>1.06E-07</td>
<td>1.36E-02 1.79E-19</td>
</tr>
<tr>
<td>4</td>
<td>2 77</td>
<td>8.94E+00</td>
<td>1.06E-07</td>
<td>1.36E-02 1.79E-19</td>
</tr>
<tr>
<td>5</td>
<td>4 153</td>
<td>8.94E+00</td>
<td>9.86E-08</td>
<td>1.36E-02 0.00E+00</td>
</tr>
<tr>
<td>6</td>
<td>10 475</td>
<td>8.94E+00</td>
<td>7.09E-08</td>
<td>1.36E-02 0.00E+00</td>
</tr>
<tr>
<td>7</td>
<td>10 499</td>
<td>8.94E+00</td>
<td>1.14E-07</td>
<td>1.36E-02 1.20E-19</td>
</tr>
<tr>
<td>8</td>
<td>12 665</td>
<td>8.95E+00</td>
<td>9.35E-08</td>
<td>1.36E-02 0.00E+00</td>
</tr>
<tr>
<td>9</td>
<td>24 774</td>
<td>8.94E+00</td>
<td>2.29E-07</td>
<td>1.36E-02 5.74E-19</td>
</tr>
<tr>
<td>10</td>
<td>26 921</td>
<td>8.95E+00</td>
<td>2.89E-07</td>
<td>1.36E-02 1.07E-18</td>
</tr>
</tbody>
</table>

Table 7.4: Solution of a fracture problem involving plasticity using XAMG-S1

<table>
<thead>
<tr>
<th>Load step</th>
<th># Iterations</th>
<th>Residual norm</th>
<th>Energy Norm</th>
<th>CPU Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Initial Final</td>
<td>Initial Final</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2 31</td>
<td>8.94E+00</td>
<td>8.53E-08</td>
<td>1.36E-02 0.00E+00</td>
</tr>
<tr>
<td>2</td>
<td>2 31</td>
<td>8.94E+00</td>
<td>8.70E-08</td>
<td>1.36E-02 0.00E+00</td>
</tr>
<tr>
<td>3</td>
<td>2 31</td>
<td>8.94E+00</td>
<td>8.90E-08</td>
<td>1.36E-02 0.00E+00</td>
</tr>
<tr>
<td>4</td>
<td>2 31</td>
<td>8.94E+00</td>
<td>8.71E-08</td>
<td>1.36E-02 0.00E+00</td>
</tr>
<tr>
<td>5</td>
<td>4 59</td>
<td>8.94E+00</td>
<td>8.86E-08</td>
<td>1.36E-02 0.00E+00</td>
</tr>
<tr>
<td>6</td>
<td>10 183</td>
<td>8.94E+00</td>
<td>4.11E-08</td>
<td>1.36E-02 0.00E+00</td>
</tr>
<tr>
<td>7</td>
<td>10 192</td>
<td>8.94E+00</td>
<td>1.17E-07</td>
<td>1.36E-02 1.18E-19</td>
</tr>
<tr>
<td>8</td>
<td>12 251</td>
<td>8.95E+00</td>
<td>7.21E-08</td>
<td>1.36E-02 0.00E+00</td>
</tr>
<tr>
<td>9</td>
<td>23 306</td>
<td>8.94E+00</td>
<td>3.35E-07</td>
<td>1.36E-02 1.36E-18</td>
</tr>
<tr>
<td>10</td>
<td>26 357</td>
<td>8.95E+00</td>
<td>2.94E-07</td>
<td>1.36E-02 9.89E-19</td>
</tr>
</tbody>
</table>
7.5 Summary and Conclusions

In this chapter, a new method to adapt AMG for linear systems arising from modeling fracture problems using XFEM is discussed. This method involves a simple transformation to an equivalent phantom node representation. Some implementation aspects have been discussed and the efficacy of this method has been demonstrated on a few numerical examples.
Part IV

Conclusions
Chapter 8

Summary

8.1 Main Contributions

The author’s main contributions in the field of computational modeling of discontinuities in solids presented in this dissertation are summarized here:

i. **XFEM for Inclusions**: The extended finite element method is adapted to model multiple elliptical inclusions within a single element domain. A new and simple formulation for the enrichment function suitable for elliptical shaped inclusions is presented.

ii. **XFEM - Monte Carlo**: The extended finite element method is coupled with a Monte Carlo approach to quantify the uncertainty in the homogenized effective elastic properties of multiphase materials having random microstructures. The methodology allows for an arbitrary number, aspect ratio, location and orientation of elliptic inclusions within a matrix, without the need for fine meshes in the vicinity of tightly packed inclusions and especially without the need to remesh for every different generated realization of the microstructure. Moreover, the number of degrees of freedom in the enriched elements is dynamically reallocated for each Monte Carlo sample run based on the given volume fraction. The main advantage of the proposed XFEM-based methodology is a major reduction in the computational effort in extensive Monte Carlo simulations compared to the standard FEM approach. Monte Carlo and XFEM appear to work extremely efficiently together.

iii. **XFEM for Cracks**: The extended finite element method for modeling both two and three
dimensional cracks is implemented. Further a parallel implementation of the three dimensional crack problem involving general nonlinear material properties has been implemented. A Trilinos interface to the finite element program FEAP has been developed.

iv. **AMG for XFEM - I:** It is shown the the linear systems arising from the discretization of fracture mechanics problems using the XFEM are not suitable for a “black-box” application of AMG. A new quasi-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. Numerical experiments have been presented to 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.

v. **AMG for XFEM - II:** An alternative method to improve the convergence properties for AMG applied to XFEM systems that does require invasive modifications to the AMG has been presented. Here the XFEM system is transformed into an alternate phantom-node representation that makes it more suitable for a direct application of AMG. This transformation automatically decouples the graph of the global stiffness matrix across strong discontinuities ensuring their accurate representation in the multigrid hierarchy. Numerical examples involving linear and nonlinear materials and implementations in serial and parallel are presented.

vi. **Collaborative contributions:** The appendices that follow detail two collaborative efforts that the author has been involved in. The first effort pertains to the development of a Schwarz domain decomposition method that allows the AMG method to be applied to fracture mechanics simulations while preserving its optimal convergence properties. The second collaborative effort involves the use of XFEM coupled with genetic algorithms for flaw detection in structures.
Bibliography


Appendix A

Schwarz-AMG for XFEM

In this chapter, another algebraic multigrid preconditioner based on the concept of Domain Decomposition and suitable for fracture problems is presented. This work is reproduced in part from Berger-Vergiat et. al. [8]. Inputs from the co-authors are gratefully acknowledged.

A.1 Introduction

In this article a simple domain decomposition approach is reproduced that retains the AMG advantages on well behaved domains by avoiding the coarsening of enriched degrees of freedom. The idea is to employ a multiplicative Schwarz preconditioner where the physical domain is partitioned into a healthy (or unfractured) and cracked subdomains. First, the healthy subdomain containing only standard degrees of freedom, is solved approximately by one AMG V-cycle, followed by concurrent direct solves of cracked subdomains. This strategy alleviates the need to redesign special AMG coarsening strategies that can handle XFEM discretizations. Numerical examples on various crack problems clearly illustrate the superior performance of this approach over a brute force AMG preconditioner applied to the linear system.

In the present work domain decomposition concepts based on the multiplicative Schwarz method are employed to reformulate the problem so that AMG would retain its convergence features when applied to the bulk of the domain that contains no cracks. Hence, the problem is partitioned in a way that separates the enriched degrees of freedom from the nodes that are not enriched. The formulation and the proposed preconditioner are outlined in the following subsec-
A.2 Domain Decomposition formulation for cracks: A multiplicative Schwarz approach

The proposed multiplicative Schwarz preconditioner begins with a special domain decomposition. As shown in Fig. A.1, two possible partitioning strategies are considered: (i) a single subdomain containing all cracks (see Fig. A.1a) and (ii) multiple crack subdomains (see Fig. A.1b), where each crack owns its own subdomain. In many cases the physics of the problem will determine the partitioning scheme. For example, clusters of cracks and microcracks, e.g., formed due to a localized impact or indentation loads, can be aggregated into a single subdomain, while more isolated cracks, nucleating at far distances from each other, e.g., formed due to fatigue loads applied to the whole structure, may be too far apart and will require their own subdomains. In any case, the two strategies lead to the same type of systems.

In Fig. A.1, the subdomain $\Omega^h_1$ that does not contain any enriched nodes (or cracks) is referred to as a “healthy” subdomain (for simplicity of the presentation only one healthy subdomain is considered). The other subdomains containing at least one crack, and hence all enriched degrees of freedom associated with that crack, are referred to as “cracked” subdomains and denoted by $\Omega^c_i$. Cracked subdomains may also contain one or several layers of elements around cracks that constitute the overlap with the healthy subdomain.

The general formulation leads to a coupled set of linear systems associated with the healthy and cracked subdomains. The coupling occurs through the boundary conditions and overlapping elements. Note that in this formulation, cracked subdomains are decoupled from each other and are only coupled to the healthy subdomain. This property is not an essential feature of the proposed approach, but is adopted in the current work to keep the description of the method simple, without any loss of generality.
Figure A.1: Schematic representation of the “healthy” and “cracked” subdomains in the formulation of domain decomposition. [a] multiple cracks share a single cracked subdomain [b] each crack is assigned to a different cracked subdomain.

Mathematically, the decomposition into subdomains is written in the following way:

\[
\begin{align*}
\begin{cases}
K_1^h u_1^h &= f_1^h & \text{in } \Omega_1^h \text{ with } u_1^h = u_E \text{ on } \Gamma \\
& & \text{and with } u_1^c = u_1^h \text{ on } \Gamma_i^c \\
K_1^c u_1^c &= f_1^c & \text{in } \Omega_1^c \text{ with } u_1^c = u_1^h \text{ on } \Gamma_1^c \\
K_2^c u_2^c &= f_2^c & \text{in } \Omega_2^c \text{ with } u_2^c = u_1^h \text{ on } \Gamma_2^c \\
& \vdots & \\
K_n^c u_n^c &= f_n^c & \text{in } \Omega_n^c \text{ with } u_n^c = u_1^h \text{ on } \Gamma_n^c
\end{cases}
\end{align*}
\]  
(A.1)

The superscript $h$ is employed to denote an operator or variable defined on a healthy subdomain and the superscript $c$ denotes operators or variables defined on a cracked subdomain. Note that the healthy subdomain is solved first, followed by cracked subdomains. $\Omega = \bigoplus_{i=1}^{n_c} \Omega_i^h \oplus \bigoplus_{i=1}^{n_c} \Omega_i^c$ is the finite element space associated with the problem, $n_c$ is the number of cracked subdomains, $u_E$ are the values of the displacement field where essential boundary are imposed, $\Gamma$ is the boundary of $\Omega$ and $\Gamma_i$ is the boundary of subdomain $\Omega_i$. The solution over the full problem domain (henceforth referred to as global) is formed by assembling the solutions from the subdomain problems.

Once the problem is discretized by finite elements, in each subdomain, the stiffness matrix, force vector and the unknown displacement vector are denoted by $K,f$ and $u$, respectively.
A simple procedure to generate a cracked subdomain employs the levelset formulation and is described as follows. Start by forming a set which is the union of all elements having specified overlap for a given crack with the healthy subdomain. This is easily generated by selecting all the nodes that have levelset values smaller than $n_o \times h^e$, where $h^e$ is the characteristic dimension of an element. If $n_o$ is chosen equal to 1, only the elements containing the crack are selected. Choosing $n_o = 2$ selects all elements containing at least one enriched node. This would create a non-overlapping partition of the domain between cracked subdomains and the healthy subdomain, and could be well suited for application of a FETI algorithm. An illustration of the domain decomposition and the overlapping elements for XFEM are shown in Fig. A.2.

![Overlapping elements](image)

Figure A.2: Two overlapping domains employed in the Schwarz method. The following color legend is used: black squares represent Schwarz essential boundary conditions, the black triangles represent clamped nodes, the red circles represent pulled nodes, the green zone represent the elements belonging to the same subdomain, the blue zone represent the elements that are part of the overlapping layer.

The Schwarz algorithm may be written in the following way. The linear system on subdomain $i$ (healthy or cracked) is extracted from the global system by employing a restriction operator $R_i : \Omega^{N \times N} \rightarrow \Omega_{i}^{N_i \times N}$, where $N$ denotes the overall number of degrees of freedom and $N_i$ the
number of degrees of freedom associated with subdomain $i$. The restriction operator is also the transpose of the prolongation operator $R_i = (P_i)^T$ which will be used in subsequent notation. The prolongator matrix $P_i$ is a Boolean matrix constructed such that each row corresponds to a global degree of freedom and each column corresponds to a degree of freedom belonging to the particular subdomain $\Omega_i$. The entries of $P_i$ are as follows:

$$P_i(j, k) = \delta_{jk} \quad \& \quad k \in \Omega_i \quad (A.2)$$

where $\delta_{jk}$ denotes the Kronecker symbol, $j$ being the current global degree of freedom and $k$ is the global degree of freedom in subdomain $\Omega_i$ that corresponds to local column index $\hat{k}$.

The application of the prolongation operator to the inverse of the restricted stiffness matrix $P_i^T K P_i$ is defined as:

$$B_i = P_i \left( P_i^T K P_i \right)^{-1} P_i^T \quad (A.3)$$

For notation simplicity but without the loss of generality, a decomposition into two subdomains is assumed, one healthy and one cracked subdomain, as illustrated in Fig. A.2. For this domain decomposition, the Schwarz preconditioning iterate may be written by

$$\begin{align*}
\begin{array}{c|c}
\text{input} & r_n \\
\hline
u_n^h & = B^h r_n \\
u_n^c & = B^c \left( r_n - K u_E^h \right) \\
u_{n+1} & = u_n^h + u_n^c
\end{array}
\end{align*} \quad (A.4)$$

where $r^n$ is the residual at iteration $n$ given as $r^n = f - Ku^n$ and $u_E^h$ is the essential boundary conditions applied to the cracked subdomain which are obtained by solving (approximately) the healthy subdomain (note that these values are changing with every solver iteration as opposed to $u_E$ which are the given boundary conditions of the problem). $u_n^h$ and $u_n^c$ are the updated terms of the solution vectors corresponding to healthy and cracked subdomains, respectively. In a more
compact notation, Eq. (A.4) becomes:

\[ u^{n+1} = \left( B^h + B^c - B^c K B^h \right) r_n = M r_n \]  \hspace{1cm} (A.5)

Where \( M \) is the preconditioning operator. Note that \( M \) defined in Eq. (A.5) is not symmetric. One could symmetrize the preconditioner by adding an additional solve of \textit{cracked} subdomains before the \textit{healthy} part, as follows:

\[
\begin{align*}
\text{input} & \quad r_n \\
\text{input} & \quad c^1_n = B^c r_n \\
\text{input} & \quad h_n = B^h \left( r_n - K c^1_k \right) \\
\text{input} & \quad c^2_n = B^c \left( r_n - K h_k \right) \\
\text{output} & \quad u_{n+1} = c^1_n + h_n + c^2_n
\end{align*}
\]  \hspace{1cm} (A.6)

In a compact matrix form it reads:

\[ M^{sym} = B^c + B^h - B^h K B^c - B^c K B^h + B^c K B^h K B^c \]  \hspace{1cm} (A.7)

\[ = B^c + \left( I - B^c K \right) B^h \left( I - K B^c \right) \]  \hspace{1cm} (A.8)

Normally, a symmetric form of the preconditioner allows for a bigger choice of iterative solvers. However, it also must be noted that it requires the application of the operator \( B^c \) twice, which is more expensive, and hence is not be pursued in this work.

Another way of creating a symmetric system is to use an additive Schwarz algorithm instead of a multiplicative one. The idea of the additive Schwarz algorithm is to solve all subdomains concurrently. In contrast with the multiplicative Schwarz approach, the non symmetric term in
Eq. A.5 is dropped in this formulation. The additive Schwarz algorithm has the following form:

\[
\begin{align*}
\text{input} & & r_n \\
\mathbf{u}_n^h & = & \mathbf{B}^h r_n \\
\mathbf{u}_n^c & = & \mathbf{B}^c r_n \\
\mathbf{u}_{n+1} & = & \mathbf{u}_n^h + \mathbf{u}_n^c \\
\text{output} & & \mathbf{u}_{n+1}
\end{align*}
\]  

(A.9)

or in a compact matrix form:

\[
\mathbf{u}_{n+1} = \left( \mathbf{B}^h + \mathbf{B}^c \right) r_n = \mathbf{M} r_n
\]

(A.10)

One obvious advantage of this approach is the ease of parallelization of this algorithm. In addition, it is also much simpler to symmetrize the system of equations in this case without introducing new matrix operations. In the additive Schwarz approach, since the residual is not updated after each subdomain solve, convergence is slower compared to the multiplicative Schwarz approach. In the numerical examples presented in Section 5, a GMRES solver is used with the preconditioner scheme given in Eq. A.5 and employ only a single smoothed aggregation AMG cycle during the solution phase.

### A.3 Preconditioner setup and algorithm flow

The proposed approach uses a domain decomposition algorithm with an inexact multiplicative Schwarz method as the preconditioner for the residual obtained at each iteration of a global GMRES solver. This partitioning is only performed once and is reused over successive iterations. The healthy subdomain is approximately solved using one AMG V-cycle and cracked subdomains are solved concurrently with a direct solver. The two solutions are then assembled back to be returned to the global GMRES solver. A schematic representation of the algorithm is illustrated in Fig. A.3 and shown in a condensed form in Alg. A.3.

While one could argue that the use of a direct solver on the cracked subdomains may be computationally too expensive, it is pointed out that cracked subdomains are relatively small compared
Exact Solves on cracked domains

Inexact AMG solve on $\Omega^h$

$\Omega_1^c$

$\Omega_2^c$

$\Omega_3^c$

$\Omega_n^c$

GMRES

Figure A.3: Schematic description of the inexact Schwarz-AMG preconditioner.

to the overall size of the problem and therefore, this step is fairly inexpensive. Moreover, linear systems associated with cracked subdomains are factored before starting GMRES iterations, and reused in successive iterations of the preconditioner. Nonetheless in the case of propagating cracks, a new partition may need to be built at every crack propagation step. Hence, the linear systems associated with cracked subdomains must also be re-factored after each crack propagation step.

Algorithm A.3.1 GMRES preconditioned by an inexact Schwarz-AMG preconditioner

Step 1: Apply one AMG V cycle to the healthy subdomain

$$u^h \leftarrow \text{AMG}(r^h, K^h)$$ for the healthy subdomain

Step 2: solve all cracked subdomains

$$u^c_i \leftarrow B_i(\mathbf{r}^h - K^h u^h)$$ for all cracked subdomains $i$

Step 3: Update the residual and return to the GMRES solver

$$u \leftarrow \text{assemble}(u^h, u^c_1, \ldots, u^c_n)$$
A.4 Representative Example

A.4.1 Multiple cracks with different lengths and orientations

In this example the convergence of the preconditioners on a plate containing three cracks of different lengths and orientations is investigated. Two strategies, illustrated in Fig. A.1, are used to partition the domain into healthy and cracked subdomains. In the first case the cracked subdomain owns all the cracks (a single cracked subdomain) while in the second approach each crack owns its own local subdomain (multiple cracked subdomains). In addition an Additive Schwarz method is investigated and compared with the other methods. The mesh and the partitioned domains are shown in Fig. A.4, and the convergence results are plotted in Fig. A.5.

![Figure A.4: Domain decomposition and mesh of a plate with three cracks with different lengths and orientations](image)

(a) Decomposition with multiple cracked subdomains
(b) Decomposition with a single cracked subdomain.

It is clear that both domain decomposition strategies give excellent results compared to the AMG brute force preconditioner. The AMG performance is poor, which is mainly attributed to the cracks having two sets of tip functions inside the domain, and in close proximity to each other. Moreover, the different orientation of the cracks makes it significantly harder for the AMG to generate
Figure A.5: Comparison of the convergence rate for the decomposition strategies shown in Fig. A.4.
appropriate aggregates and the coarsening of these special functions, significantly deteriorates its performance.

As expected, the multiplicative inexact Schwarz method with a single cracked subdomain gives slightly better performance than its counterpart with multiple cracked subdomains. The single cracked subdomain has converged in 50 iterations whereas it converged in 63 iterations when multiple subdomains are introduced. This behavior is due to the fact that all the cracks are solved concurrently in the single cracked subdomain case whereas solving them in a sequential manner introduce a small delay in the coupling of these cracks. The convergence of the additive Schwarz method is slightly worse than the multiplicative Schwarz as has been noticed before.

Table A.1: Summary of the convergence results for the problem considered in Fig. A.4.

<table>
<thead>
<tr>
<th></th>
<th>AMG brute force</th>
<th>Additive Schwarz-AMG</th>
<th>Multiplicative Schwarz-AMG</th>
<th>Exact Schwarz-AMG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single &quot;crack&quot; subdomain</td>
<td>190</td>
<td>68</td>
<td>50</td>
<td>48</td>
</tr>
<tr>
<td>Multiple &quot;crack&quot; subdomains</td>
<td>190</td>
<td>74</td>
<td>63</td>
<td>59</td>
</tr>
</tbody>
</table>
Appendix B

Applications to Structural Health Monitoring

In this chapter, the synergy of XFEM and Genetic Algorithms is leveraged to build an efficient method to detect flaws in structures. This work is reproduced in part from Chatzi et. al. [16]. Inputs from the co-authors are gratefully acknowledged.

B.1 Introduction

The extended finite element formulation (XFEM) combined with genetic algorithms (GAs) have previously been shown to be very effective in the detection of flaws in structures. By this approach, the XFEM is used to model the forward problem and a GA is used as the optimization scheme, converging to the true flaw. The convergence is obtained by minimizing the error between sensor measurements and data obtained by solving the forward problem.

The current study proposes several advances of this XFEM-GA algorithm, more specifically: (i) a novel genetic algorithm that accelerates the convergence of the scheme and alleviates entrapment in local optima, (ii) a generic XFEM formulation of an elliptical hole which is utilized to detect any type of flaw (cracks or holes) of any shape, and (iii) experimental verification of the approach for an arbitrary crack in a 2D plate. Convergence studies on various benchmark problems including the experimental verification clearly show the potential of this approach to detection of arbitrary flaws.
B.2 XFEM approach for solution of the forward problem

In XFEM, the effect of internal features such as cracks, voids or material interfaces are captured by introducing special functions, which locally enrich the span of basis functions in the discretized system.

The general form of the discretized weak form approximation for the solution variable $u(x)$ in XFEM is given as follows:

$$
\begin{align*}
    u^h(x) &= \sum_{I \in \mathcal{N}} N_I(x) u_I + \sum_{I \in \mathcal{N}_{enr}} \left( \sum_{I \in \mathcal{F}} \Phi_I(x) a_{IJ} \right) \\
    & \quad (B.1)
\end{align*}
$$

Here $\mathcal{N}$ refers to the all the nodes in the mesh and $\mathcal{N}_{enr}$ refers to the subset of $\mathcal{N}$ that contains all the nodes enriched by the functions $\Phi_I$, called the support of the basis $\Phi_I$. This support is made up of only those elements that contain within their domains, the discontinuity interface modeled by $\Phi_I$. If an element-wise discretized weak form is considered, both $\mathcal{N}$ and $\mathcal{N}_{enr}$ (for an enriched element) refer to the number of element nodes $\mathcal{N}_e$. The enrichments are controlled by degrees of freedom $a_{IJ}$ which are additional unknowns to be solved for, in the global system of equations.

When considering XFEM for linear elastic fracture mechanics, the number of enrichment functions $\mathcal{F}$ generally equals 1 for elements with a crack interface and the Heavyside step function $\mathcal{H}(x)$ is used for $\Phi$. For elements with a crack-tip, $\mathcal{F}$ generally equals 4 and the span of $\Phi_I$ is suitably chosen to model tip singularity. In the case of XFEM for modeling weak-discontinuities (such as those caused by material inclusions), a single enrichment function which is $C^0$ continuous is used. The discretized weak form for an enriched element is given as:

$$
\begin{align*}
    u^*_e(x) &= \sum_{I \in \mathcal{N}_e} N_I(x) (u_I + \Phi(x) a_I) \\
    & \quad (B.2)
\end{align*}
$$

It may be seen from equation $B.2$ that the physical displacement at an enriched node $I$ is provided in terms of both the standard dof $u_I$ and the enriched dof $a_I$. To ensure that the physical displacement solution is completely defined by the standard dof $u_I$, a shifted-basis form of equation $(B.2)$ may be written as follows:
\[ u_h^b(x) = \sum_{l \in \mathcal{N}_e} N_l(x) \left( u_l + [\hat{\Phi}(x) - \hat{\Phi}(x_l)] a_l \right) \] (B.3)

**B.2.1 Convergence of elliptical enrichment**

The convergence of the elliptical approximation to the modeling of a straight crack is studied. Figure B.1(a) represents a rectangular mesh grid with a straight crack shown as a red line, subjected to load and boundary conditions as shown. An FEM analysis on a mesh conforming to the internal boundaries with the presence of double nodes along the crack was performed. The results from this analysis (Displacement solution in Fig. B.1(b)) was used as a benchmark to compare against the XFEM ellipse approximation. The XFEM approximation involved an ellipse with a major radius equal to the crack length and oriented in the same direction. The variable parameter in this study was the ellipse aspect ratio \( r = \text{minor/major radius} \), which varied from 1 (circle) to 0.1 (elongated ellipse). As shown in Fig. B.1(c), the norm of the relative error in nodal displacements between the XFEM and FEM analyses shows a linear trend decreasing to substantially low magnitudes as the ellipse minor radius is reduced. Since the FEM nodes did not overlap with the XFEM nodes, the FEM nodal displacements were first interpolated to the XFEM node locations before the error was computed. The results confirm that an ellipse with a sufficiently small aspect ratio can be used as a good approximation to a crack.

**B.3 GA - XFEM based Identification**

Genetic Algorithms have been chosen as the optimization tool for this non destructive detection scheme. GAs have long been used as an efficient tool for search, optimization and machine learning problems. The main concept associated with this particular method is the mimicking of the biological processes of natural evolution and survival of the fittest candidates. According to Holland’s Genetic Algorithm each potential optimal solution to a problem can be seen as an individual that can be coded by a set of genes. Usually these genes are chosen to be binary bits and the binary string or alternatively the “individual” is also known as a chromosome. A random
Figure B.1: Convergence study of crack modeled with elliptical enrichment
generation of such chromosomes provides us with the initial population. At each cycle of the evolutionary process a new set of offsprings is produced from the fittest individuals of the previous generation. Reproduction takes place through the recombination of the bit strings or simple bit flips that occur with some probability. The purpose of this evolutionary procedure is to eventually lead to the survival of the fittest individual, as this will be the one to produce the largest number of offsprings and thus has the best chances for survival. All genetic algorithms are based on the following scheme (Figure B.2 - classic GA):

- **Representation:** Depending on the application the parameters of the problem can be either integer or real numbers. Usually, the set of parameters can be appropriately encoded into a finite length binary string. Once a representation is decided a number of different chromosomes is randomly generated to form the initial population.

- **Fitness Evaluation - Selection:** The evaluation of the fitness for each member of the population is carried out through the use of an objective function associated with each problem. The suitability of each chromosome is the criterion based on which this individual will be selected for reproduction. Selection can be performed through various schemes like “roulette wheel” selection or “tournament” selection. The roulette wheel selection is solely dependent upon the performance (fitness) of each individual. In the case of the second method, the probability of a string selection is also dependent upon the fitness of the “competitive” strings chosen to participate in a tournament round. The latter contributes toward the preservation of the diversity of the population.

- **Crossover:** This operator is applied with a certain probability, to the pairs of the previously selected individuals (parents). In general the crossover procedure randomly selects a position in the binary string and mutually exchanges the parts of the chromosome before and after this location in order to produce two offsprings.

- **Mutation:** The use of the crossover procedure as the only means for generating new individuals could result in the loss of the diversity of the population. This problem can be
overcome using the mutation operator which involves the random flip of the binary genes of the chromosome for the case of jump mutation, or the alteration of the phenotypic (real) representation of the design parameters by a small increment or decrement for the case of creep mutation.

In a nutshell, the basic GA consists of coding each potentially optimal solution to a problem, as a set of genes. Usually these genes are chosen to be binary bits, and the binary string or alternatively the “individual”, is also known as a chromosome. Random generation of such chromosomes provides the initial population. At each cycle of the evolutionary process a new set of offsprings is produced from the fittest individuals of the previous generation (Selection). Reproduction takes place through the recombination of the bit strings (Crossover) or simple bit flips (Mutation) that occur with some probability. The purpose of this evolutionary procedure is to eventually lead to the survival of the fittest individual, as this will be the one to produce the largest number of offsprings and thus has the best chances for survival (Figure B.2).

![Figure B.2: The XFEM-GA algorithm flowchart.](image)

In the context of the current work, there are several noted advantages to using genetic algorithms: (i) they have been fairly well studied in the literature and have proven to be very efficient on difficult optimization problems (converging to a global minima), (ii) the current problem is ex-
tremely difficult as the forward model keeps changing and updating. Hence there is no straight-
forward analytical expression connecting the strain field to the crack parameters which may cause
traditional methods to stagnate, (iii) they are not limited by constraints imposed by other ana-
lytical methods which usually require linear or differentiable objective functions and a proper
estimation of the initial parameter interval, and (iv) can easily be implemented and adjusted to
any user source codes. The problem described in Section 2, consists of finding the parameters de-
scribing the flaw, i.e., for linear cracks-the tip coordinates, for circular holes-the radius and center
coordinates, or for elliptical holes-the center, minor and major axes and orientation angle. This
is achieved through minimizing the residual of the XFEM-GA estimate with respect to the actual
data provided at specific sensor locations, usually along the boundaries (equation (B.4)):

\[ r(\beta_i) = \frac{\| \epsilon_\gamma (\beta_i) - \epsilon_\gamma^0 \|}{\| \epsilon_\gamma^0 \|} \]  (B.4)

where \( \| \cdot \| \) denotes the \( L_2 \) norm, \( \epsilon_\gamma^0 \) is the real measured data (in most cases, this is the mea-
sured strain obtained from the damaged model) at some locations along the surface and \( \epsilon_\gamma \) are the
computed strains at the exact same points. These computed strains are a function of the optimiza-
tion parameters \( \beta_i \) which change with the forward problem as the outer (optimization) iteration
proceeds.

The problem setup involves the definition of the number of unknown parameters as well as the
discretized search space (upper, lower bounds and number of discrete possibilities per param-
ter). The XFEM-GA identification process is initiated with the generation of binary bit individuals
(chromosomes), which are randomly selected from the search space, serving as the initial popu-
lation. The next step involves the fitness evaluation of each individual in the population group.
The fitness function to be minimized is the one defined in equation (B.4). At the next cycle a new
set of offsprings is produced from the fittest individuals of the previous generation using the tour-
nament selection scheme. The mechanisms of reproduction (mutation and crossover) are applied
and the evolutionary cycle continues for a specific number of total generations.
B.3.1 Applied Weighted Average Mutation GA (WAM-GA)

Most population-based, reproductive, optimization algorithms such as genetic algorithms, ant colony optimization, and particle swarm optimization (PSO) often deal with the premature convergence problem. This problem occurs when highly fit parents in a population pool rapidly dominate the breeding process, commonly leading to confinement in local optima. The crossover operator, is not enough to circumvent this hindrance since it uses only acquired information. The mutation operator on the other hand, can move into broader areas within the search space in contrast to the crossover, but it also can not change many bits in individuals because the mutation rate is too low. If the mutation rate is set to a high value, then genetic algorithms approach the global optimum very slowly. As a result, it is very difficult to escape this premature convergence problem.

This method suggests the use of a guided mutation scheme applied at the phenotype, meaning the real representation of the individual (creep mutations). The standard implementation for the creep mutation scheme is for a child to differ from its parent by only a small increment or decrement, usually toward a random direction. The novel Weighted Average Mutation (WAM-GA) implemented here, takes advantage of the large amount of information obtained throughout a course of GA iterations. Each set of candidate parameters (individual) is linked to a fitness value, thus providing a pseudo probability density function (pdf). The WAM-GA method utilizes a weighted average approach to mutate the design parameters of the current population toward areas of increased fitness.

B.4 Representative Example

B.4.1 Identification of a crack using the elliptical inclusion XFEM-GA scheme

In order to study the validity of the proposed algorithm in the presence of model error, the following scheme is applied. The numerical results obtained from the XFEM analysis of a 45° oriented crack are utilized as the strain measurements in order to identify the elliptical shaped hole that would best approximate this straight crack. Thus, the GA process makes use of an elliptical en-
richment XFEM code for the solution of the forward problem, while the reference measurements are obtained from the strain of an actual embedded crack, which is modeled with a crack formulation with XFEM. For this particular problem, the number of parameters to be identified is five; namely, the Cartesian coordinates \((x, y)\) of the center, the major radius \((\alpha)\), the minor radius \((\beta)\) and the major axis orientation \((\psi)\). The search space for the elliptical hole fitting problem is defined follows: \(\{x, y\} \in [0.5, 9.5], \alpha \in [0, 2], \alpha \in [0, 2], \beta \in [0, 2], \psi \in [0, 2]\). The flaw in each case lies within the interior of a rectangular plate of dimension 10 by 10 (units). Fixed boundary conditions are applied on the left vertical boundary of the plate which is in tension along the horizontal direction. Displacement sensors are this time assumed to be uniformly distributed along the three free boundaries. Parametric runs have been performed for different crack sizes and locations within the interior of the plate, as shown in Figure B.3. Four different crack sizes have been assumed: \(l_1 = 1\sqrt{2}, l_2 = 0.75\sqrt{2}, l_3 = 0.5\sqrt{2}\) and \(l_4 = 0.25\sqrt{2}\). The XFEM - WAM-GA scheme was set to run for 200 generations for a mean population size of 7 individuals, a creep mutation probability \(p_c = 0.30\) and a weighted average mutation probability \(p_{wam} = 0.80\). The performance of the XFEM-GA identification scheme is satisfactory even in this case of artificial noise presence. Figure B.4 provides the finally estimated elliptical shapes that best approximate a crack of size \(l_1 = 1\sqrt{2}\), for two mesh cases; a coarser mesh of \(41 \times 41\) nodes and a finer mesh of \(81 \times 41\) nodes. It is obvious that the convergence results are improved for a finer mesh since this leads to higher accuracy and thus closer finite element solutions for the linear and elliptical code. However, a finer mesh corresponds to a higher computational cost both for the XFEM forward problem and the GA optimization, hence it is sometimes preferable to utilize a coarser mesh and obtain cruder estimates.
Figure B.3: Mesh generation, loading configuration, sensor placement and assumed linear crack locations.

Figure B.4: Estimated elliptical approximations for the assumed linear cracks of size $l_1 = 1\sqrt{2}$
Table B.1: WAM-GA scheme

<table>
<thead>
<tr>
<th>Mutation Pseudocode</th>
</tr>
</thead>
<tbody>
<tr>
<td>for ( j = 1, \text{npop} )</td>
</tr>
<tr>
<td>{ for ( k = 1, \text{nparam} )</td>
</tr>
<tr>
<td>{ if ( \text{rand}_1 \leq p_c ) /<em>perform mutation</em>/</td>
</tr>
<tr>
<td>{ if ( \text{rand}<em>2 \leq p</em>{wam} ) /<em>perform weighted average mutation</em>/</td>
</tr>
<tr>
<td>( \text{child}(k,j) = \text{child}(k,j) + \text{rand}_3 \times (\text{wmean}(j) - \text{child}(k,j)) )</td>
</tr>
<tr>
<td>else /<em>(perform standard creep mutation)</em>/</td>
</tr>
<tr>
<td>( \text{child}(k,j) = \text{child}(k,j) + g_1(k) \times \text{crs} )</td>
</tr>
<tr>
<td>} }</td>
</tr>
</tbody>
</table>

where,

- \( \text{npop} \) is the number of individuals per population
- \( \text{nparam} \) is the number of parameters per candidate solution (individual)
- \( \text{rand}_{1,2,3} \) are random numbers uniformly distributed \( \in [0, 1] \)
- \( p_c \) is the creep mutation probability
- \( g_1(k) \) is the incremental value between the upper and lower bound for each parameter \( k \)
- \( \text{crs} \) is equal to \( \pm 1 \) with a uniform probability
- \( p_{wam} \) is the weighted average mutation probability
- \( \text{child}(k,j) \) is the real value corresponding to parameter \( k \) of the \( j \)-th individual, and
- \( x_m(k) \) is the weighted average for that parameter value by the end of the previous generation obtained from:

\[
x_m(k) = \frac{\sum_i w_i \text{child}(k,i)}{\sum_i w_i}, \quad w_i = \frac{1}{\|\text{fitness}(i)\| + \epsilon}
\]  

(B.5)

where, \( i \) is an index corresponding to each of the formed individuals up to that point of the evolutionary process, \( w_i \) is the associated weight and \( \epsilon \) is a sufficiently “small” numerical value added for the case where the fitness of the individual approaches 0. This particular expression has been chosen for the type of problems considered herein aiming at minimizing a residual (error) function toward zero. The formula yielding the weights can be properly modified for different classes of problems.
Appendix C

FEAP & Trilinos Implementations

In this chapter, the implementation details of the program XICE is presented.

C.1 FEAP User Element for 3D XFEM

A user element UEL is developed within the finite element program FEAP to model three dimensional cracks. Listed below is a sample user input file and a description of using the XFEM user elements. Here the solid elements to be checked are all assigned to Material 1 (XFEM elements). A user mesh-manipulation command MANI is then called upon to identify the enriched elements based on the nodal levelsets. This function automatically assigns all the non-enriched elements to Material 2 which is the FEAP default solid elements.

The second line of the UEL input calls for the user element defined by ELMT01 (XFEM) to be used. The third line of input requires four parameters to be defined. They are:

i. ncr: Number of crack planes

ii. gporder: Order of gauss quadrature for each tetrahedral subquadrature region for an enriched element

iii. tipon: A flag set to 0 or 1 to toggle tip enrichment on or off

iv. opflag: A flag set to 0 or 1 to toggle screen output from XFEM elements.

From the next line onwards, four sets of (x,y,z) points are specified and define the vertices of each crack plane. For example, if ncr is 3, then 12 sets of points need to be specified.
Lastly the material properties to be used for the XFEM element are specified. Any standard material definition that can be used in FEAP for solid continuum elements (small displacements) can be specified. Please refer to the FEAP manual for details about this specification.

### C.1.1 Input format

```plaintext
! MATERIAL 1 is XFEM user element
! -----------------------------------
MATERIAL 1
USER 1
  1 11 0 0
  x1 y1 z1
  x1 y2 z2
  x2 y2 z2
  x2 y1 z1
SOLID
  ELASTIC ISOTROPIC e nu
  PLASTIC MISES ys

! MATERIAL 2 is regular solid element
! -----------------------------------
MATERIAL 2
SOLID
  ELASTIC ISOTROPIC e nu
  PLASTIC MISES ys
```

### C.1.2 Stiffness generation for jump-enriched element

The code related to the generation of the element stiffness matrix for an XFEM element with jump-enriched nodes is reproduced below.

```c
subroutine XFEMnonlinstiff(d,ul,xl,tl,S,R,isw)
  c SUBROUTINE TO COMPUTE THE STIFFNESS MATRIX OF A USER ELEMENT
  use xfemvars
  implicit none
  include 'sdata.h' !ndf,ndm,nen1,nst,nneq,ndl,nnlm,nadd
  include 'eldata.h' !dm,n,ma,mct,iel,nel,pstyp
  include 'pointer.h' !np,up
  include 'comblk.h' !hr(1),mr(1)
  include 'cdata.h' !numnp,numel,nummat,nen,neq,ipr
  include 'hdata.h' !nh1,nh2,nh3,ht1,ht2,ht3
  include 'iofile.h'
  include 'bdata.h'
```
integer :: maxgp
parameter ( maxgp = 1728 )
integer :: l,i1,i,j1,j,ii,jj,m,k,ismw
integer :: ngp,lint,istr,n,nn,nhv
integer :: crnum,sctr(nel),nodetype(nel)
real*8 :: sw(4,1728),shp(4,nel),xjac,t1(*)
real*8 :: eps(9,3),sig(10,maxgp),dd(6,6,5),epsv(maxgp)
real*8 :: s(nst,*),r(*),ul(ndf,nel,*),xl(ndm,*)
real*8 :: epsv(maxgp)
real*8 :: s(nst,*),r(*),ul(ndf,nel,*),xl(ndm,*)
real*8 :: a(6,6),x,yn,zn,ta,th(125),fin(nst)
real*8 :: a(6,6),x,yn,zn,ta,th(125),fin(nst)
real*8 :: phiel(nel),psiel(nel),phi,psi,H,Hi,Hj,Fenr
real*8 :: phiel(nel),psiel(nel),phi,psi,H,Hi,Hj,Fenr
real*8 :: alam,ha,psig(3),epp(6),peps(3)
logical :: NAT = .true.
character :: sfi*12

save

data alam,ha / 2*0.0d0 /

c OBTAIN NODAL PHI/PSI
sctr = elems(1:8,n)
crnum = elenr(n,2)
if (crnum.ne.0) then
  phiel = xfemphi(sctr,crnum)
  psiel = xfempsi(sctr,crnum)
endif

c OBTAIN QUADRATURE POINTS
if (elenr(n,1).ne.0) then
  ngp = xngp
  call XFEMquadraturetet(xl,phiel,psiel,ngp,nat,lint,sw)
else
  ngp = 2
  call int3d(ngp,lint,sw)
endif

write(*,'(A,I2,6E12.1)') 'XFEM EL# ',n,(d(i1),i1=1,6)

c ==============================
if (ismw.eq.3 .or. isw.eq.6) then

c ==============================
do l=1,lint

c Compute shape functions
call shp3d(sw(l,1),xjac,shp,xl,ndm,nel)

c Compute enrichment function at quadrature-point
phi = 0.0
psi = 0.0
do i = 1, nel
   phi = phi + shp(4,i)*phiel(i)
   psi = psi + shp(4,i)*psiel(i)
   nodetype(i) = ndenr(sctr(i),1)
endo

c Compute strain at point
   call XFEMstrn3d(d,xl,ul,th,shp,eps,ta,H,phiel)

c Compute stress at point
   call modlsd(d,ta,eps,hr(nh1+nn),hr(nh2+nn),nhv,istrt,
             & dd,sig(1,l),alam,ha,isw)

c Compute d * b matrix = adb
   i1 = 1
   do i = 1, nel
      xn = shp(1,i)*xjac*sw(4,l)
      yn = shp(2,i)*xjac*sw(4,l)
      zn = shp(3,i)*xjac*sw(4,l)
      call Heaviside(Hi,xfemphi(sctr(i),crnum))

      a(1,1) = xn*dd(1,1,1) + yn*dd(4,1,1) + zn*dd(6,1,1)
      a(1,2) = xn*dd(1,2,1) + yn*dd(4,2,1) + zn*dd(6,2,1)
      a(1,3) = xn*dd(1,3,1) + yn*dd(4,3,1) + zn*dd(6,3,1)
      a(1,4) = xn*dd(1,4,1) + yn*dd(4,4,1) + zn*dd(6,4,1)
      a(1,5) = xn*dd(1,5,1) + yn*dd(4,5,1) + zn*dd(6,5,1)
      a(1,6) = xn*dd(1,6,1) + yn*dd(4,6,1) + zn*dd(6,6,1)

      a(2,1) = xn*dd(4,1,1) + yn*dd(2,1,1) + zn*dd(5,1,1)
      a(2,2) = xn*dd(4,2,1) + yn*dd(2,2,1) + zn*dd(5,2,1)
      a(2,3) = xn*dd(4,3,1) + yn*dd(2,3,1) + zn*dd(5,3,1)
      a(2,4) = xn*dd(4,4,1) + yn*dd(2,4,1) + zn*dd(5,4,1)
      a(2,5) = xn*dd(4,5,1) + yn*dd(2,5,1) + zn*dd(5,5,1)
      a(2,6) = xn*dd(4,6,1) + yn*dd(2,6,1) + zn*dd(5,6,1)

      a(3,1) = xn*dd(6,1,1) + yn*dd(5,1,1) + zn*dd(3,1,1)
      a(3,2) = xn*dd(6,2,1) + yn*dd(5,2,1) + zn*dd(3,2,1)
      a(3,3) = xn*dd(6,3,1) + yn*dd(5,3,1) + zn*dd(3,3,1)
      a(3,4) = xn*dd(6,4,1) + yn*dd(5,4,1) + zn*dd(3,4,1)
      a(3,5) = xn*dd(6,5,1) + yn*dd(5,5,1) + zn*dd(3,5,1)
      a(3,6) = xn*dd(6,6,1) + yn*dd(5,6,1) + zn*dd(3,6,1)

! Heaviside-enriched-nodes a(4:6,1:6) = a(1:3,1:6)*Fenr
Fenr = dabs(H-Hi)
if (nodetype(i).eq.1) then
do ii = 1,3
    do jj = 1,6
        a(ii + 3,jj) = a(ii,jj)*Fenr
    enddo
enddo
endif

c Compute stiffness matrix
j1 = 1
do j = 1,nel

    xn = shp(1,j)
    yn = shp(2,j)
    zn = shp(3,j)

    s(i1 ,j1   ) = s(i1 ,j1   ) + a(1,1)*xn + a(1,4)*yn + a(1,6)*zn
    s(i1 ,j1+1) = s(i1 ,j1+1) + a(1,4)*xn + a(1,2)*yn + a(1,5)*zn
    s(i1 ,j1+2) = s(i1 ,j1+2) + a(1,6)*xn + a(1,5)*yn + a(1,3)*zn

    s(i1+1,j1   ) = s(i1+1,j1   ) + a(2,1)*xn + a(2,4)*yn + a(2,6)*zn
    s(i1+1,j1+1) = s(i1+1,j1+1) + a(2,4)*xn + a(2,2)*yn + a(2,5)*zn
    s(i1+1,j1+2) = s(i1+1,j1+2) + a(2,6)*xn + a(2,5)*yn + a(2,3)*zn

    s(i1+2,j1   ) = s(i1+2,j1   ) + a(3,1)*xn + a(3,4)*yn + a(3,6)*zn
    s(i1+2,j1+1) = s(i1+2,j1+1) + a(3,4)*xn + a(3,2)*yn + a(3,5)*zn
    s(i1+2,j1+2) = s(i1+2,j1+2) + a(3,6)*xn + a(3,5)*yn + a(3,3)*zn

    if (nodetype(j).eq.1) then
        ! bottom left
        do m = 4,6
            k = m - 1
            s(i1+k,j1   ) = s(i1+k,j1   ) + a(m,1)*xn + a(m,4)*yn + a(m,6)*zn
            s(i1+k,j1+1) = s(i1+k,j1+1) + a(m,4)*xn + a(m,2)*yn + a(m,5)*zn
            s(i1+k,j1+2) = s(i1+k,j1+2) + a(m,6)*xn + a(m,5)*yn + a(m,3)*zn
        enddo

        ! dNd(X,Y,Z) * (H-Hj)
        call Heaviside(Hj,xfemphi(sctr(j),crnum))
        Fenr = dabs(H-Hj)
        xn = xn*Fenr
        yn = yn*Fenr
        zn = zn*Fenr

    do m = 1,6
        ! Top/bottom-right
        k = m-1
        s(i1+k,j1+3) = s(i1+k,j1+3) + a(m,1)*xn + a(m,4)*yn + a(m,6)*zn
s(i1+k,j1+4) = s(i1+k,j1+4) + a(m,4)*xn + a(m,2)*yn + a(m,5)*zn
s(i1+k,j1+5) = s(i1+k,j1+5) + a(m,6)*xn + a(m,5)*yn + a(m,3)*zn
enddo
endif
j1 = j1 + ndf
end do   ! j
i1 = i1 + ndf
end do   ! i
enddo   ! Quadrature points

do i = 1,nst
   fin(i) = r(i)
endo

call XFEMresid3d(s,ul,fin,r,nst)
c
elseif (isw.eq.4 .or. isw.eq.8 .or. isw.eq.16) then
c
   nn = 0
   do l = 1,lint
      c Compute shape functions
      call shp3d(sw(1,l),xjac,shp,xl,ndm,nel)
c Compute enrichment function at quadrature-point
      phi = 0.0
      psi = 0.0
      do i = 1,nel
         phi = phi + shp(4,i)*phiel(i)
         psi = psi + shp(4,i)*psiel(i)
         nodetype(i) = ndenr(sctr(i),1)
      enddo
      call Heaviside(H,phi)
c Compute strain at point
      call XFEMstrn3d(d,xl,ul,th,shp,eps,ta,H,phiel)
c Compute stress at point
      call modlsd(d,ta,eps,hr(nh1+nn),hr(nh2+nn),nhv,istrt,
                  dd,sig(1,l),alam,ha,isw)
c
C Output values
if(isw.eq.4) then
C Compute coordinates
$xn = 0.0$
$yn = 0.0$
$zn = 0.0$
do j = 1, nel
    $xn = xn + shp(4,j)*xl(1,j)$
    $yn = yn + shp(4,j)*xl(2,j)$
    $zn = zn + shp(4,j)*xl(3,j)$
end do ! j

c Compute principal stress values
do i = 1,3
    epp(i ) = eps(i ,1)
    epp(i+3) = eps(i+3,1)*0.5d0
end do ! i
call pstr3d(sig(1,l),psig)
call pstr3d(epp ,peps)

mct = mct - 5
if(mct.le.0) then
    write(iow,2010) o,head
    if(ior.lt.0) write(*,2010) o,head
    mct = 50
endif
write(iow,2011) n,ma,xn,yn,zn,
&   (sig(i,l),i=1,6),(epp(i),i=1,6),
&   (psig(i),i=1,3),(peps(i),i=1,3)
if(ior.lt.0) then
    write(*,2011) n,ma,xn,yn,zn,
&   (sig(i,l),i=1,6),(epp(i),i=1,6),
&   (psig(i),i=1,3),(peps(i),i=1,3)
endif
endif
nn = nn + nhv
end do ! l

c Plot stress values
if(isw.eq.8) then
call slcn3d(sig, r,s, nel)
c Compute J-integrals and material forces
elseif(isw.eq.16) then
call pjint3d(ul,tl,epsv,sig,r,ndf,ndm)
endif
endif
c Formats

2010 format(a1,20a4//5x,'Element Stresses'//' Elmt Mat',
& ' 1-coord 2-coord 3-coord'/12x,
& ' 11-stress 22-stress 33-stress 12-stress',
& ' 23-stress 31-stress'/12x,
& ' 11-strain 22-strain 33-strain 12-strain',
& ' 23-strain 31-strain'/12x,
& ' 1-stress 2-stress 3-stress',
& ' 1-strain 2-strain 3-strain',/39('-'))

2011 format(/i9,i4,1p,3e11.3/(12x,1p,6e11.3))

end subroutine XFEMnonlinstiff