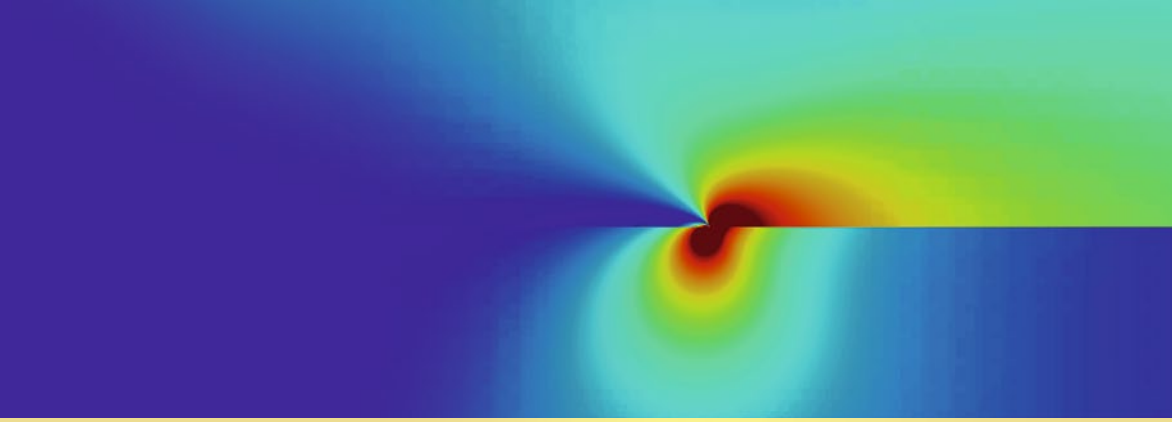

LECTURE NOTES IN COMPUTATIONAL SCIENCE AND ENGINEERING 129

Michael Griebel
Marc Alexander Schweitzer *Editors*

Meshfree Methods for Partial Differential Equations IX

Springer

Lecture Notes in Computational Science and Engineering **129**

Editors:

More information about this series at http://www.springer.com/series/3527

Michael Griebel • Marc Alexander Schweitzer
Editors

Meshfree Methods for Partial Differential Equations IX

Springer

Editors
Michael Griebel
Institut für Numerische Simulation
Universität Bonn
Bonn, Germany

Fraunhofer-Institut für Algorithmen
und Wissenschaftliches
Rechnen SCAI,
Sankt Augustin, Germany

Marc Alexander Schweitzer
Institut für Numerische Simulation
Universität Bonn
Bonn, Germany

Fraunhofer-Institut für Algorithmen
und Wissenschaftliches
Rechnen SCAI,
Sankt Augustin, Germany

ISSN 1439-7358 ISSN 2197-7100 (electronic)
Lecture Notes in Computational Science and Engineering
ISBN 978-3-030-15118-8 ISBN 978-3-030-15119-5 (eBook)
https://doi.org/10.1007/978-3-030-15119-5

Mathematics Subject Classification (2010): 65N30, 65N75, 65M60, 65M75, 65Y99

Cover illustration: Image reprinted with kind permission from Albert Ziegenhagel and Matthias Birner.

This Springer imprint is published by the registered company Springer Nature Switzerland AG.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The Ninth International Workshop on *Meshfree Methods for Partial Differential Equations* was held from September 18 to September 20, 2017, in Bonn, Germany. Meshfree methods have a diverse and rich mathematical background and their flexibility renders them particularly interesting for challenging applications in which classical mesh-based approximation techniques struggle or even fail. This workshop series was established in 2001 to bring together European, American, and Asian researchers working in this exciting field of interdisciplinary research on a regular basis.

To this end, Ivo Babuška, Jiun-Shyan Chen, Michael Griebel, Wing Kam Liu, Marc Alexander Schweitzer, C. T. Wu, and Harry Yserentant invited scientists from all over the world to Bonn to strengthen the mathematical understanding and analysis of meshfree discretizations and to promote the exchange of ideas on their implementation and application.

The workshop was again hosted by the Institut für Numerische Simulation at the Rheinische Friedrich-Wilhelms-Universität Bonn with the financial support of the Sonderforschungsbereich 1060 *The Mathematics of Emergent Effects* and the Hausdorff Center for Mathematics.

Bonn, Germany
Bonn, Germany
December 2018

Michael Griebel
Marc Alexander Schweitzer

Contents

Preconditioned Conjugate Gradient Solvers for the Generalized Finite Element Method

Travis B. Fillmore, Varun Gupta, and Carlos Armando Duarte

Abstract This paper focuses on preconditioners for the conjugate gradient method and their applications to the Generalized FEM with global-local enrichments (GFEM^{gl}) and the Stable GFEM^{gl}. The preconditioners take advantage of the hierarchical struture of the matrices in these methods and the fact that most of the matrix does not change when simulating for example, the evolution of interfaces and fractures. The performance of the conjugate gradient method with the proposed preconditioner is investigated. A 3-D fracture problem is adopted for the numerical experiments.

1 Introduction

The Generalized or Extended FEM (GFEM/XFEM) [3, 4, 6, 12, 26, 28, 31] has successfully been applied to problems involving moving interfaces, crack propagation, material discontinuities, and many others. These applications rely on a-priori knowledge of the solution in order to define enrichment functions. Several assumptions are usually required for the derivation of these enrichments. As a result, refinement of the FEM mesh is usually required for acceptable accuracy. One strategy to address this issue is to define the enrichments numerically as the solution of auxiliary boundary value problems [10]. This leads to the so-called Generalized FEM with global-local enrichments (GFEM^{gl}). Another limitation of the GFEM is the ill-conditioning of the resulting system of equations which may lead to severe round-off errors of direct solvers or to the lack convergence of iterative solvers.

T. B. Fillmore · C. A. Duarte (✉)
Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA
e-mail: travis.b.fillmore@usace.army.mil; caduarte@illinois.edu

V. Gupta
Pacific Northwest National Laboratory, Richland, WA, USA
e-mail: varun.gupta@pnnl.gov

M. Griebel, M. A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations IX*, Lecture Notes in Computational Science and Engineering 129,
https://doi.org/10.1007/978-3-030-15119-5_1

Preconditioning schemes for the GFEM can be found in the works of Kim et al. [23] which proposes a Block-Jacobi preconditioner for the conjugate gradient method, Waisman et al. [7, 36], Menk and Bordas [27], Béchet et al. [5], and several others.

The Stable GFEM (SGFEM), initially proposed in [1, 2] and extended to 2- and 3-D fracture mechanics in [17, 18], provides a robust and yet simple solution to the problem of ill-conditioning of the GFEM/XFEM. It is shown in [2] that the SGFEM yields matrices with a condition number which is orders of magnitude lower than in the GFEM/XFEM.

Block Gauss-Seidel iterative solution algorithms for the SGFEM are proposed in Kergrene et al. [20] and [15]. This paper proposes preconditioners for the SGFEM and, in particular to the SGFEM$^{\text{gl}}$ –an application of SGFEM ideas to the GFEM$^{\text{gl}}$ which was first proposed in [15]. The key idea of the preconditioners is to explore the hierarchical structure of the system of equations in Generalized FEMs like the SGFEM$^{\text{gl}}$. For example, when simulating the propagation of fractures in a domain using the SGFEM$^{\text{gl}}$, only the enrichments change between propagation steps. The entries of the matrix associated with the FEM space—a sub-space of the SGFEM$^{\text{gl}}$ space—remains constant throughout the entire simulation regardless of the complexity of the fracture. This has been demonstrated in [32]. Two preconditioners for the conjugate gradient method are investigated: The Block Jacobi (BJ-PCG) and the Block Gauss-Seidel (BGS-PCG). They are defined in Sect. 4 and their performance investigated in Sect. 5. The numerical experiments involve the solution of a 3-D fracture problem using the SGFEM$^{\text{gl}}$. This method is briefly reviewed in Sect. 3. The model problem adopted in this paper—the linear elastic fracture mechanics problem—is summarized in Sect. 2. The main conclusions of this work are presented in Sect. 6.

2 Model Problem

The iterative solvers investigated in this paper are not restricted to a particular problem. However, we focus on linear elastic fracture mechanics problems in 2- and 3-D. Consider a cracked domain $\bar{\Omega} = \Omega \cup \partial\Omega$ in $\mathbb{R}^d$, $d = 2$ or 3, as illustrated in Fig. 1. The boundary is decomposed as $\partial\Omega = \partial\Omega^u \cup \partial\Omega^\sigma$ with $\partial\Omega^u \cap \partial\Omega^\sigma = \emptyset$. The crack surface $S \subset \partial\Omega^\sigma$ is assumed to be traction-free. We consider the linear elasticity problem on this domain. The equilibrium equations are given by

$$\nabla \cdot \boldsymbol{\sigma} = \mathbf{0} \qquad \text{in } \Omega, \tag{1}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor. The following boundary conditions are prescribed on $\partial\Omega$

$$\boldsymbol{u} = \bar{\boldsymbol{u}} \ \text{ on } \partial\Omega^u \qquad \boldsymbol{\sigma} \cdot \boldsymbol{n} = \bar{\boldsymbol{t}} \ \text{ on } \partial\Omega^\sigma, \tag{2}$$

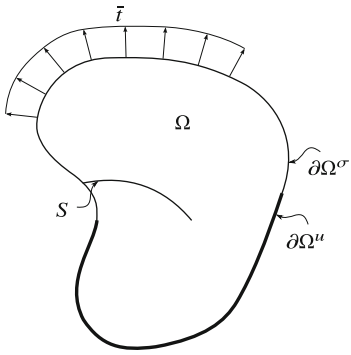

Fig. 1 Fractured domain $\bar{\Omega}$ in $\mathbb{R}^2$ or $\mathbb{R}^3$

where $\boldsymbol{n}$ is the outward unit normal vector to $\partial\Omega^{\sigma}$ and $\bar{\boldsymbol{t}}$ and $\bar{\boldsymbol{u}}$ are prescribed tractions and displacements, respectively. Without loss of generality, we assume hereafter that $\bar{\boldsymbol{u}} = \boldsymbol{0}$. The constitutive relations are given by the generalized Hooke's law,

$$\boldsymbol{\sigma} = \boldsymbol{C} : \boldsymbol{\varepsilon}, \tag{3}$$

where $\boldsymbol{C}$ is Hooke's tensor. The kinematic relations are given by

$$\boldsymbol{\varepsilon} = \nabla_s \boldsymbol{u} \qquad \text{in } \Omega, \tag{4}$$

where $\boldsymbol{\varepsilon}$ is the linear strain tensor and ∇_s is the symmetric part of the gradient operator. We seek to find a GFEM approximation to the solution $\boldsymbol{u}$ of the problem defined by Eqs. (1)–(4).

3 GFEM and GFEM$^{\mathbf{gl}}$ Approximations

A brief review of generalized FEM approximations is given in this section. Further details can be found in, for example, [3, 11, 26, 31, 34].

The GFEM test and trial space $\mathbb{S}^{\text{GFEM}}$ is obtained by hierarchically enriching a low-order standard finite element approximation space $\mathbb{S}^{\text{FEM}}$, with special functions related to the given problem and belonging to the enrichment space $\mathbb{S}^{\text{ENR}}$. Consider a finite element mesh covering the domain of interest $\bar{\Omega}$. Let $N_\alpha(\boldsymbol{x})$, $\alpha \in I_h = \{1, \cdots, \text{nnod}\}$, be the standard linear finite element shape function associated with node $\boldsymbol{x}_\alpha$ and with support ω_α. The *patch or cloud* ω_α is given by the union of the finite elements sharing node $\boldsymbol{x}_\alpha$. The test/trial space of the GFEM is given by

$$\mathbb{S}^{\text{GFEM}} = \mathbb{S}^{\text{FEM}} + \mathbb{S}^{\text{ENR}}, \tag{5}$$

where

$$\mathbb{S}^{\text{FEM}} = \sum_{\alpha \in I_h} \underline{\hat{\boldsymbol{u}}}_\alpha N_\alpha, \quad \underline{\hat{\boldsymbol{u}}}_\alpha \in \mathbb{R}^d,\ d = 2, 3,$$

$$\mathbb{S}^{\text{ENR}} = \sum_{\alpha \in I_h^e} N_\alpha \chi_\alpha, \text{ and } \quad \chi_\alpha(\omega_\alpha) = \text{span}\{E_{\alpha i}\}_{i=1}^{m_\alpha}. \tag{6}$$

The basis function $E_{\alpha i}$ is called an *enrichment function*, $\alpha \in I_h^e \subset I_h$ is the index of the node with this enrichment, and $i = \{1, \cdots, m_\alpha\}$ is the index of the enrichment function at the node with m_α being the total number of enrichments associated with node $\boldsymbol{x}_\alpha$. The functions $E_{\alpha i} \in \chi_\alpha(\omega_\alpha)$ are chosen such that they approximate the unknown solution $\boldsymbol{u}$ of the problem locally in ω_α. Examples of enrichment functions are polynomials, the Heaviside function, crack tip singular functions, and numerically generated functions (cf. Sect. 3.1). The spaces $\chi_\alpha(\omega_\alpha)$ are called *patch* approximation spaces, and $\mathbb{S}^{\text{ENR}}$ is referred to as the *global enrichment space* of the GFEM. The functions in $\mathbb{S}^{\text{ENR}}$

$$\phi_{\alpha i}(\boldsymbol{x}) = N_\alpha(\boldsymbol{x}) E_{\alpha i}(\boldsymbol{x}), \qquad \alpha \in I_h^e,\ i = 1, \ldots, m_\alpha, \tag{7}$$

are denoted GFEM shape functions. They are built from the product of Finite Element shape functions, $N_\alpha(\boldsymbol{x})$, $\alpha \in I_h^e$, and enrichment functions, $E_{\alpha i}$, $i = 1, \ldots, m_\alpha$. There are m_α GFEM shape functions at a node $\boldsymbol{x}_\alpha$, $\alpha \in I_h^e$, of a finite element mesh. These nodes also have a standard FE shape function $N_\alpha \in \mathbb{S}^{\text{FEM}}$. Nodes not in the set I_h^e have only one function—the FE shape function N_α.

Based on Eqs. (5)–(7), the GFEM approximation $\boldsymbol{u}^{\text{GFEM}}$ of a vector field $\boldsymbol{u}$ (e.g., displacements) can be written as

$$\begin{aligned} \boldsymbol{u}^{\text{GFEM}}(\boldsymbol{x}) &= \boldsymbol{u}^{\text{FEM}}(\boldsymbol{x}) + \boldsymbol{u}^{\text{ENR}}(\boldsymbol{x}) \\ &= \underbrace{\sum_{\alpha \in I_h} \underline{\hat{\boldsymbol{u}}}_\alpha N_\alpha(\boldsymbol{x})}_{\text{Standard FEM approx.}} + \underbrace{\sum_{\alpha \in I_h^e} N_\alpha(\boldsymbol{x}) \sum_{i=1}^{m_\alpha} \underline{\tilde{\boldsymbol{u}}}_{\alpha i} E_{\alpha i}(\boldsymbol{x})}_{\text{GFEM enriched approx.}}, \quad \underline{\hat{\boldsymbol{u}}}_\alpha,\ \underline{\tilde{\boldsymbol{u}}}_{\alpha i} \in \mathbb{R}^d,\ d = 2, 3. \end{aligned} \tag{8}$$

The above equation clearly shows that a GFEM approximation is obtained by hierarchically enriching a standard finite element approximation. As a consequence, any GFEM stiffness matrix is given by a FEM matrix augmented with entries associated with GFEM enrichments. This property of GFEM matrices is used in the proposed preconditioners for the GFEM.

3.1 The Generalized FEM with Global-Local Enrichments

Available enrichment functions for linear elastic fracture problems [6, 12, 13, 28, 30] are based on the expansion of the elasticity solution in the neighborhood of a straight crack front in an infinite domain. They also assume a planar fracture surface. These assumptions are not valid in most practical fracture mechanics problems, in particular for the case of 3-D problems. As a result, refinement of the FEM mesh is required for acceptable accuracy. Alternatively, the enrichments can be defined numerically as the solution of auxiliary boundary value problems [10, 14]. This so-called Generalized FEM with Global-Local Enrichments (GFEM$^{\text{gl}}$), combines the GFEM and the global-local FEM [9, 29]. This allows the GFEM to use coarse meshes while delivering accurate solutions. The GFEM$^{\text{gl}}$ has been formulated and applied to various classes of problems. In Sect. 5, the method is used to discretize a 3-D linear elastic fracture problem. The resulting discrete system of equations is solved using the iterative solvers described in Sect. 4. Further details on GFEM$^{\text{gl}}$ in the context of linear elastic fracture mechanics, can be found in [10, 21, 22].

3.2 Stable GFEM and Stable GFEM^{gl}

It can be shown that the growth of the condition number for the GFEM is $O(h^{-4})$ with mesh refinement [2]. In contrast, the condition number of the *standard* FEM stiffness matrix for a 3-D elasticity problem subjected to Neumann boundary conditions is $O(h^{-2})$ [8]. It is noted that if point constraints are adopted in 3-D to eliminate the rigid body motions, the condition number of the FEM matrix with these point constraints is $O(h^{-3})$ [8]. Condition number in this paper is taken to mean the condition number computed using the non-zero eigenvalues of a matrix scaled such that its diagonal entries are 1 or $O(1)$. This is also known as the scaled condition number.

The Stable GFEM (SGFEM) [2] was proposed to address this ill-conditioning issue of the GFEM. In the SGFEM, the enrichment functions are locally modified to construct the patch approximation spaces $\widetilde{\chi}_\alpha,\ \alpha \in I_h^e$. The modified SGFEM enrichment functions $\tilde{E}_{\alpha i}(\boldsymbol{x}) \in \widetilde{\chi}_\alpha(\omega_\alpha)$ are given by

$$\tilde{E}_{\alpha i}(\boldsymbol{x}) = E_{\alpha i}(\boldsymbol{x}) - \mathrm{I}_{\omega_\alpha}(E_{\alpha i})(\boldsymbol{x}) \text{ and } \widetilde{\chi}_\alpha = \text{span}\{\tilde{E}_{\alpha i}\}_{i=1}^{m_\alpha} \tag{9}$$

where $\mathrm{I}_{\omega_\alpha}(E_{\alpha i})$ is the piecewise linear finite element interpolant of the enrichment function $E_{\alpha i}$ on the patch ω_α. The global enrichment space associated with $\widetilde{\chi}_\alpha$ is denoted by $\widetilde{\mathbb{S}}^{\text{ENR}}$. Therefore, the SGFEM trial space $\mathbb{S}^{\text{SGFEM}}$ is given by

$$\mathbb{S}^{\text{SGFEM}} = \mathbb{S}^{\text{FEM}} + \widetilde{\mathbb{S}}^{\text{ENR}}. \tag{10}$$

The SGFEM shape functions $\tilde{\phi}_{\alpha i}(\boldsymbol{x})$ belonging to $\widetilde{\mathbb{S}}^{\text{ENR}}$ are constructed using the same framework as GFEM and are given by

$$\tilde{\phi}_{\alpha i}(\boldsymbol{x}) = N_{\alpha}(\boldsymbol{x})\tilde{E}_{\alpha i}(\boldsymbol{x}). \tag{11}$$

The above procedure can also be applied to the GFEM$^{\text{gl}}$ [15, 25]. The resulting methodology is denoted as the SGFEM$^{\text{gl}}$. Further details about the SGFEM are given in [1, 2, 17]. The numerical implementation of the SGFEM is described in Section 4 of [17].

4 Iterative Solvers

The iterative solvers studied in this paper are the Block Gauss-Seidel (BGS), the Conjugate Gradient (CG), Block Jacobi Preconditioned CG (BJ-PCG), and Block Gauss-Seidel PCG (BGS-PCG). All of the "Block" iterative solvers take advantage of the hierarchical nature of the GFEM/SGFEM approximation spaces (5) and (10). This property leads to the following structure for the global stiffness matrix $\boldsymbol{K}$, displacement vector $\boldsymbol{d}$, and load vector $\boldsymbol{f}$ associated with a GFEM/SGFEM discretization of the problem described in Sect. 2:

$$\boldsymbol{K}\boldsymbol{d} = \begin{bmatrix} \boldsymbol{K}^0 & \boldsymbol{K}^{0,\text{gl}} \\ \boldsymbol{K}^{\text{gl},0} & \boldsymbol{K}^{\text{gl}} \end{bmatrix} \begin{bmatrix} \boldsymbol{d}^0 \\ \boldsymbol{d}^{\text{gl}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}^0 \\ \boldsymbol{f}^{\text{gl}} \end{bmatrix} = \boldsymbol{f}, \tag{12}$$

where $\boldsymbol{K}^0$ is associated with the FEM space $\mathbb{S}^{\text{FEM}}$, $\boldsymbol{K}^{\text{gl}}$ is associated with enrichment space $\mathbb{S}^{\text{ENR}}$ or $\widetilde{\mathbb{S}}^{\text{ENR}}$, and $\boldsymbol{K}^{\text{gl},0} = (\boldsymbol{K}^{0,\text{gl}})^{\text{T}}$ represents the coupling between the FEM and enrichment spaces.

Remark 1 The notation $\boldsymbol{K}^{0,gl}$, $\boldsymbol{K}^{gl}$, $\boldsymbol{d}^{\text{gl}}$, and $\boldsymbol{f}^{gl}$ is adopted since the enrichments used in this paper are computed through a global-local analysis as described in Sect. 3.1.

Remark 2 Matrix $\boldsymbol{K}^0$ does not change in a crack propagation simulation. Thus, it can be factorized once and re-used to define an efficient pre-conditioner for the GFEM. This factorization can also be re-used when solving the enriched global problem in the GFEM$^{\text{gl}}$ [10]. An iterative algorithm for the standard FEM can be used to solve a system of equations with coefficients given by $\boldsymbol{K}^0$ instead of a direct method. In this paper, however, a direct method is adopted.

4.1 Block Gauss-Seidel Algorithm

The Block Gauss-Seidel iterative method has been used in [15] and [20] to solve the system of equations (12). An SGFEM was adopted in these references. Rather, than

factorizing $\boldsymbol{K}$, the Block Gauss-Seidel (Block GS) method factorizes the diagonal blocks $\boldsymbol{K}^0$ and $\boldsymbol{K}^{gl}$. Algorithm 1 describes the method in details.

Algorithm 1: Block GS algorithm

Input: $\boldsymbol{K}, \boldsymbol{f}, \boldsymbol{d}$
Output: $\boldsymbol{d}$
for $i = 0$ *until convergence* **do**
  $\boldsymbol{r}^{gl} \leftarrow \boldsymbol{f}^{gl} - \boldsymbol{K}^{gl,0}\boldsymbol{d}^0$
  $\boldsymbol{d}^{gl} \leftarrow (\boldsymbol{K}^{gl})^{-1}\boldsymbol{r}^{gl}$
  $\boldsymbol{r}^0 \leftarrow \boldsymbol{f}^0 - \boldsymbol{K}^{0,gl}\boldsymbol{d}^{gl}$
  $\boldsymbol{d}^0 \leftarrow (\boldsymbol{K}^0)^{-1}\boldsymbol{r}^0$
return $\boldsymbol{d}$

4.2 Preconditioned Conjugate Gradient Method

The Preconditioned Conjugate Gradient (PCG) method is one of the most used iterative methods to solve symmetric positive-definite systems of equations. An excellent introduction to PCG can be found in [35]. The method finds new search directions through A-orthogonalization of previous search directions. It finds the magnitude of this direction by using the residual, a preconditioner, and matrix $\boldsymbol{K}$. The PCG algorithm as described in [35] follows in Algorithm 2.

The effectiveness of the PCG depends on the symmetric positive definite preconditioner $\boldsymbol{M}$ adopted. Matrix $\boldsymbol{M}$ is usually similar to $\boldsymbol{K}$ but easier to factorize. The lower the condition number of $\boldsymbol{M}^{-1}\boldsymbol{K}$, the faster the convergence of PCG. The most effective preconditioner is one that is easy to compute and factorize while leading to a better condition number than $\boldsymbol{K}$. The Block Jacobi and the Block Gauss-Seidel preconditioners are adopted in this paper. They are briefly described next.

4.2.1 Block Jacobi Preconditioner

The Block Jacobi PCG is used in [23] to solve the systems of equations (12). The following preconditioner is adopted in this algorithm

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{K}^0 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}^{gl} \end{pmatrix}. \tag{13}$$

This is potentially a good preconditioner for the SGFEM and SGFEM^{gl} since in these methods $\boldsymbol{K}^{gl}$ and $\boldsymbol{K}^0$ are near orthogonal [1]. The Block Jacobi Preconditioner proceeds as described in Algorithm 3.

Algorithm 2: PCG algorithm

Input: $\boldsymbol{K}, \boldsymbol{f}, \boldsymbol{d}$
Output: $\boldsymbol{d}$
$i \Leftarrow 0$
$\boldsymbol{r} \Leftarrow \boldsymbol{f} - \boldsymbol{K}\boldsymbol{u}$
$\boldsymbol{d} \Leftarrow \boldsymbol{M}^{-1}\boldsymbol{r}$
$\delta_{new} \Leftarrow \boldsymbol{r}^T\boldsymbol{d}$
for $i = 0$ *until convergence* **do**
  $\boldsymbol{q} \Leftarrow \boldsymbol{K}\boldsymbol{d}$
  $\alpha \Leftarrow \frac{\delta_{new}}{\boldsymbol{d}^T\boldsymbol{q}}$
  $\boldsymbol{d} \Leftarrow \boldsymbol{d} + \alpha\boldsymbol{d}$
  if i *is divisible by 50* **then**
    $\boldsymbol{r} \Leftarrow \boldsymbol{f} - \boldsymbol{K}\boldsymbol{u}$ `/* Reset r to exact value */`
  else
    $\boldsymbol{r} \Leftarrow \boldsymbol{r} - \alpha\boldsymbol{q}$ `/* r is typically not evaluated directly to save computations */`
  $\boldsymbol{s} \Leftarrow \boldsymbol{M}^{-1}\boldsymbol{r}$ `/* Preconditioner solution step */`
  $\delta_{old} \Leftarrow \delta_{new}$
  $\delta_{new} \Leftarrow \boldsymbol{r}^T\boldsymbol{s}$
  $\eta \Leftarrow \frac{\delta_{new}}{\delta_{old}}$
  $\boldsymbol{d} \Leftarrow \boldsymbol{s} + \eta\boldsymbol{d}$
return $\boldsymbol{d}$

Algorithm 3: BJ preconditioner algorithm

Input: $\boldsymbol{K}, \boldsymbol{r}$
Output: $\boldsymbol{s}$
$\boldsymbol{s}^{gl} \leftarrow (\boldsymbol{K}^{gl})^{-1}\boldsymbol{r}^{gl}$
$\boldsymbol{s}^0 \leftarrow (\boldsymbol{K}^0)^{-1}\boldsymbol{r}^0$
return $\boldsymbol{s}$

In this paper, the Cholesky factorization of $\boldsymbol{K}^0$ and $\boldsymbol{K}^{gl}$ are computed and stored at the start of the PCG iteration. See also Remark 2.

4.2.2 Block Gauss-Seidel Preconditioner

The Block Gauss-Seidel PCG uses the Block Gauss-Seidel Algorithm 4 as $\boldsymbol{M}$. The BGS method is similar computationally to the Block Jacobi method. It solves two systems of equations using the factorizations of $\boldsymbol{K}^0$ and $\boldsymbol{K}^{gl}$.

BGS-PCG is different from BJ-PCG because it involves sparse matrix multiplication of $\boldsymbol{K}^{0,gl}$ and its transpose, which is relatively inexpensive. This means that

Algorithm 4: Block GS algorithm

Input: $\boldsymbol{K}, \boldsymbol{r}, \boldsymbol{s}$
Output: $\boldsymbol{s}$
$\boldsymbol{s}^{gl} \leftarrow (\boldsymbol{K}^{gl})^{-1}\left(\boldsymbol{r}^{gl} - \boldsymbol{K}^{gl,0}\boldsymbol{s}^{0}\right)$
$\boldsymbol{s}^{0} \leftarrow (\boldsymbol{K}^{0})^{-1}\left(\boldsymbol{r}^{0} - \boldsymbol{K}^{0,gl}\boldsymbol{s}^{gl}\right)$
return $\boldsymbol{s}$

BGS-PCG considers more information about the coupling matrices. The BGS-PCG can also perform multiple iterations. In this paper, one iteration of the BGS is used in the preconditioner step of the PCG iteration. The initial value for $\boldsymbol{r}$ is $\mathbf{0}$.

5 Analysis of a 3-D Edge-Crack

The 3-D edge-crack shown in Fig. 2 is analyzed in this section using the GFEM^{gl} and the SGFEM^{gl}. The system of equations is solved using the iterative algorithms described in the previous section. The problem domain and dimensions are shown in Fig. 2. The dimensions are $b = 2$, $l = 4$, $t = 1$, and $a = 1$. The magnitude of the tractions is taken as $\sigma = 1$. The material parameters are Young's modulus $E = 200,000$, and Poisson's ratio $\nu = 0.3$. Point displacement boundary conditions are assigned to selected nodes of the FEM mesh to prevent rigid body motions. The GFEM^{gl} and the SGFEM^{gl} are used to numerically define the enrichment functions adopted in the global problem. The three steps of the (S)GFEM^{gl} analysis of this problem are illustrated in Fig. 3. It shows the domains for each stage in the GFEM^{gl} and SGFEM^{gl} solution process. Tetrahedron elements are used at both global and local problems. The local step of the GFEM^{gl} simulates the crack. Spring boundary

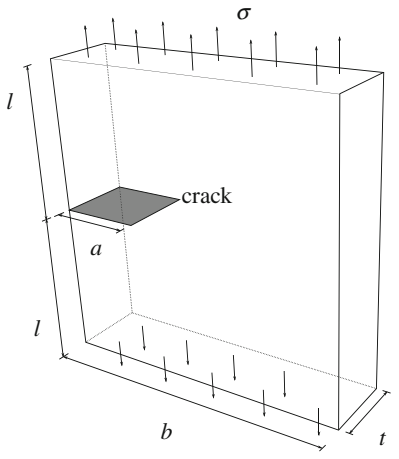

Fig. 2 Three-dimensional edge crack

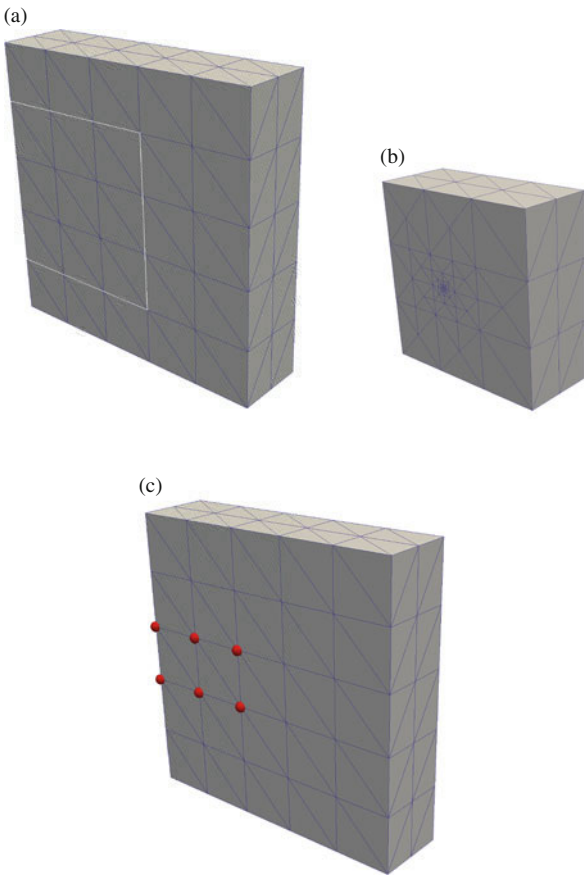

Fig. 3 GFEM^{gl} steps for a 3-D edge crack. The same steps are used in the SGFEM^{gl}. Red spheres are shown at global nodes enriched with the local solution. (**a**) Initial global mesh. (**b**) Local mesh. (**c**) Enriched global mesh

conditions are applied along the portion of the local boundary that does not intersect the boundary of the global problem. The local mesh is refined near the crack front, with the element length adjacent to the crack front being about 5% of the crack length. The polynomial order of the local problem is taken as 3. The local solution is used to generate enrichments for the enriched global problem. These global-local enrichments are the only enrichments in the global domain.

5.1 Condition Number Analysis

The condition number of the global stiffness matrix $\boldsymbol{K}$ of the GFEM^{gl} and the SGFEM^{gl} is compared in this section. The condition number of the sub-matrices in (12) is also compared. The following notation is adopted hereafter: $\kappa(\boldsymbol{K})$, $\kappa(\boldsymbol{K}^0)$, and $\kappa(\boldsymbol{K}^{gl})$ denote the condition number of global matrix $\boldsymbol{K}$, and sub-matrices $\boldsymbol{K}^0$

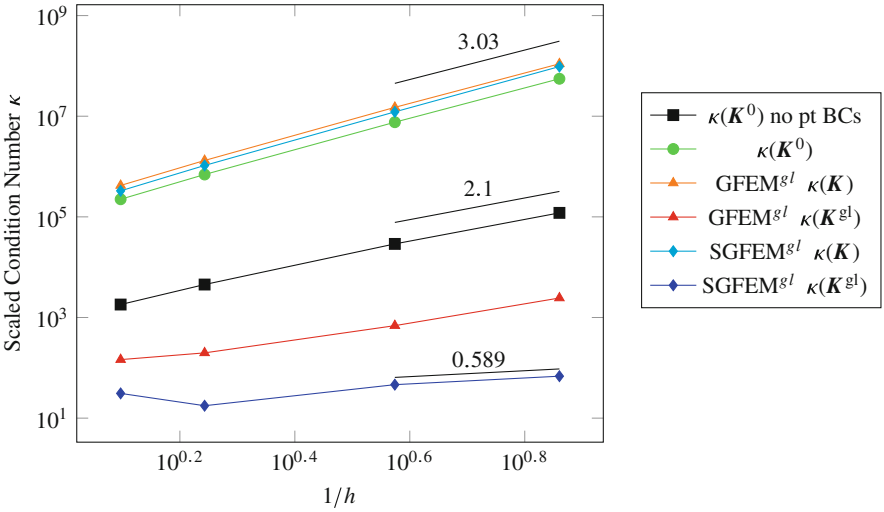

Fig. 4 Growth of condition numbers as the global mesh is refined in all three directions. Point Dirichlet boundary conditions are prescribed to the global problem to prevent rigid body motion

and $\boldsymbol{K}^{\mathrm{gl}}$, respectively. The condition number is computed for a sequence of global meshes with the same number of elements in all three directions. The local mesh is unchanged. This, and the boundary condition of the global problem, implies that the global-local enrichments do not change with refinement of the global mesh.

Figure 4 shows the condition number for the GFEM$^{\mathrm{gl}}$ and the SGFEM$^{\mathrm{gl}}$ with this sequence of meshes. The plots show that $\kappa(\boldsymbol{K}) = O(h^{-3})$ for both methods and that $\kappa(\boldsymbol{K}^0)$ is also of $O(h^{-3})$. This is surprising since one would expect that the conditioning of $\boldsymbol{K}$ would grow much quicker than $\kappa(\boldsymbol{K}^0)$, at least for the GFEM$^{\mathrm{gl}}$. The cause of this apparent contradiction is the point Dirichlet boundary conditions prescribed to prevent rigid body motion of the global problem. The condition number of the FEM stiffness matrix for a 3-D Neumann problem with point boundary conditions is $O(h^{-3})$ [8]. Thus, the condition number for both the GFEM$^{\mathrm{gl}}$ and the SGFEM$^{\mathrm{gl}}$ matrices is controlled by the effect of point constraints. Figure 4 also shows $\kappa(\boldsymbol{K}^0)$ when no point constraint is prescribed to the global problem. In this case $\kappa(\boldsymbol{K}^0) = O(h^{-2})$ as expected. It is noted that in the case of the GFEM$^{\mathrm{gl}}$, $\kappa(\boldsymbol{K})$ is expected to grow faster than $O(h^{-3})$ with further mesh refinement than shown in Fig. 4. This is proved in [1].

Another interesting feature shown in Fig. 4 is the noticeable decrease in SGFEM$^{\mathrm{gl}}$ $\kappa(\boldsymbol{K}^{\mathrm{gl}})$ relative to $\kappa(\boldsymbol{K}^{\mathrm{gl}})$ from GFEM$^{\mathrm{gl}}$. This is the case even though the SGFEM was designed to reduce the condition number $\kappa(\boldsymbol{K})$ by reducing the coupling between $\boldsymbol{K}^0$ and $\boldsymbol{K}^{\mathrm{gl}}$. This reduction in $\kappa(\boldsymbol{K}^{\mathrm{gl}})$ has an impact on the performance of iterative solvers as shown in the next section.

5.2 *Performance of Preconditioners for GFEM^{gl} and SGFEM^{gl}*

The performance of the iterative algorithms described in Sect. 4 is investigated in this section. The CG, BJ-PCG and BGS-PCG algorithms are used to solve the global system of equations (12) associated with the GFEM^{gl} and the SGFEM^{gl} discretizations. Each iterative solver is run until the relative error is less than $e^{\text{conv}} = 10^{-5}$, which is taken as the convergence tolerance. The relative error e^i of an iterative solver solution is calculated at each iteration i using

$$e^i = \frac{||\hat{\boldsymbol{d}} - \boldsymbol{d}^i||_2}{||\hat{\boldsymbol{d}}||_2}$$

where $\hat{\boldsymbol{d}}$ is a precalculated direct solver solution. The iteration i at which the solver converges is hereafter denoted i^{conv}.

Figure 5 shows the number of iterations for convergence (i^{conv}) of each solver and for the GFEM^{gl} and SGFEM^{gl}. The same sequence of global meshes adopted in the previous section is used. Point Dirichlet boundary conditions are used in the global problem to prevent rigid body motion. For any given mesh, the GFEM^{gl} i^{conv} is significantly higher than that for the SGFEM^{gl}. Also, the rate of growth of i^{conv} with respect to element size for the SGFEM^{gl} is less than half the rate for the GFEM^{gl}. This indicates that although the conditioning of the matrices for GFEM^{gl} and SGFEM^{gl} are for this problem fairly similar, iteratively solving (12) for the SGFEM^{gl} is faster than for the GFEM^{gl}. Both BGS-PCG and BJ-PCG benefit from the SGFEM^{gl}. The slopes of the curves in Fig. 5 is similar but i^{conv} is always less for the BGS-PCG than for BJ-PCG. The advantages of SGFEM versus GFEM in iterative solvers are also shown in [15] and [20].

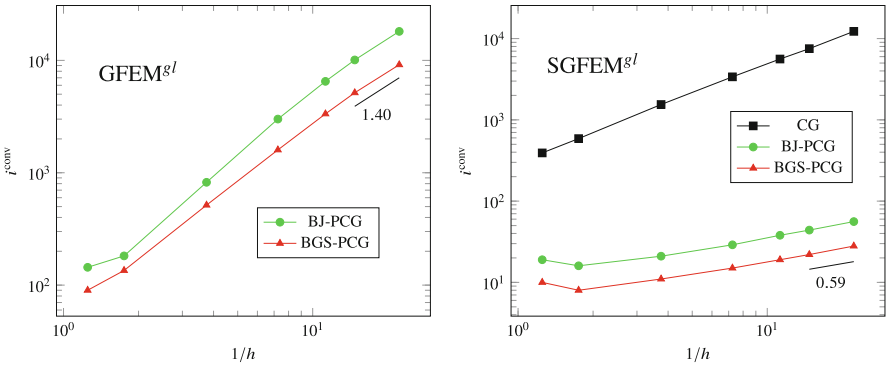

Fig. 5 Iterations to convergence of CG, BJ-PCG and BGS-PCG when solving global system (12) for GFEM^{gl} (left) and SGFEM^{gl} (right). Point Dirichlet boundary conditions are adopted in the global problem

5.3 Comparison of BGS-PCG with Pardiso

The performance of the proposed BGS-PCG algorithm with the SGFEM^{gl} is compared with the Intel Pardiso direct solver [24, 33]. The pure Neumann edge-crack problem with point Dirichlet boundary conditions is solved using both solvers and the sequence of uniform meshes described earlier. The largest global problem has about 2 million degrees of freedom. The CPU time required for convergence of the BGS-PCG when solving (12) and for the factorization of $\boldsymbol{K}$ by Pardiso is plotted against the number of degrees of freedom in Fig. 6. It shows that the BGS-PCG is, for this problem, always faster than Pardiso. The slope of the BGS-PCG curve is lower than the one for Pardiso which implies that the bigger the problem, the more efficient the BGS-PCG is relative to Pardiso. For the largest problem solved, BGS-PCG took 496 s for convergence while Pardiso required 7030 s for the factorization of $\boldsymbol{K}$ which is about 14 times slower than BGS-PCG.

The slope of 2.07 for Pardiso and 1.54 for BGS-PCG shown in Fig. 6 can be considered equivalent to the rate of increase of the total number of algebraic operations versus the number of degrees of freedom of the problem. The theoretical rate when solving 3-D elliptic boundary value problems using the adopted sparse direct solver is 2 [19]. This matches pretty well with the rate shown in Fig. 6. The theoretical rate when solving the same class of problems using the preconditioned conjugate gradient is 1.17 [19], which is lower than the rate for the BGS-PCG shown in Fig. 6. This can be traced to the computational effort required by the BGS preconditioner. The cost of the PCG is given by the number of CG iterations times the cost of each iteration, including the preconditioner. Figure 5 shows that the number of iterations required for convergence of the BGS-PCG grows at a rate of 0.59/3 with respect to the number of degrees of freedom in 3-D. This slow rate

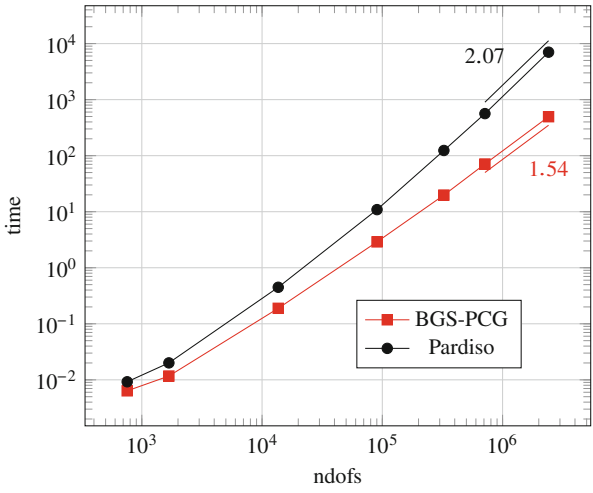

Fig. 6 Comparison of BGS-PCG with Pardiso for several problem sizes