

RESEARCH ARTICLE

Open Access



A hierarchical non-intrusive algorithm for the generalized finite element method

Travis B. Fillmore* and C. Armando Duarte *

*Correspondence:
travis.b.fillmore@usace.army.mil;
caduarte@illinois.edu
Department of Civil and
Environmental Engineering,
University of Illinois at
Urbana-Champaign, 61801
Urbana, USA

Abstract

An algorithm for non-intrusively coupling a commercial finite element software with a research code implementing a hierarchical enrichment of finite element spaces is presented. Examples of hierarchical methods supported by the algorithm are the Generalized or eXtended FEM (GFEM), the scale-bridging GFEM with numerically defined enrichment functions (GFEM^{gl}), and the p -version of the FEM. The proposed hierarchical non-intrusive algorithm (HNA) combines the vast library of classical elements available in commercial FEM platforms with the ability of the GFEM^{gl} to analyze localized phenomena like cracks and spot welds, on coarse meshes. The algorithm does not require iterations between the standard and Generalized FEM platforms and is simple to implement. Examples showing the application of the HNA to the coupling of Abaqus with a 3-D GFEM^{gl} software are presented. They also demonstrate the benefits of combining finite elements available only in a commercial platform with a GFEM.

Keywords: Finite element method, Generalized FEM, GFEM, eXtended FEM, XFEM, Non-intrusive, Global–local

Mathematics Subject Classification: Primary 65N30

Introduction

The finite element method (FEM) is used in the industry, research laboratories and academia for the simulation of a broad class of problems and phenomena. Industrial applications of the method usually rely on implementations provided by commercial software like Abaqus [1], NASTRAN [2], ANSYS [3], Radioss [4], among many others. These finite element analysis (FEA) platforms provide vast libraries of elements and solution algorithms. However, they suffer the inherent limitations of the standard FEM. An example is the analysis of localized phenomena like propagating fractures, spot welds, and localized material non-linearities in large structures. Highly refined meshes that need to change in the course of a simulation are often needed for the simulation of these phenomena. This leads to high computational costs and many times requires user intervention. New discretization methods able to accurately and efficiently simulate these phenomena are currently available in research software developed in academia and research laboratories. However, their implementation in commercial software is often difficult, time-consuming and therefore expensive. In turn, research software that implement these novel methods and algorithms are typically limited in scope of applications. One strategy to address this

issue is to couple commercial and research software and thus provide to the end user simulation and modeling capabilities not available in any single software.

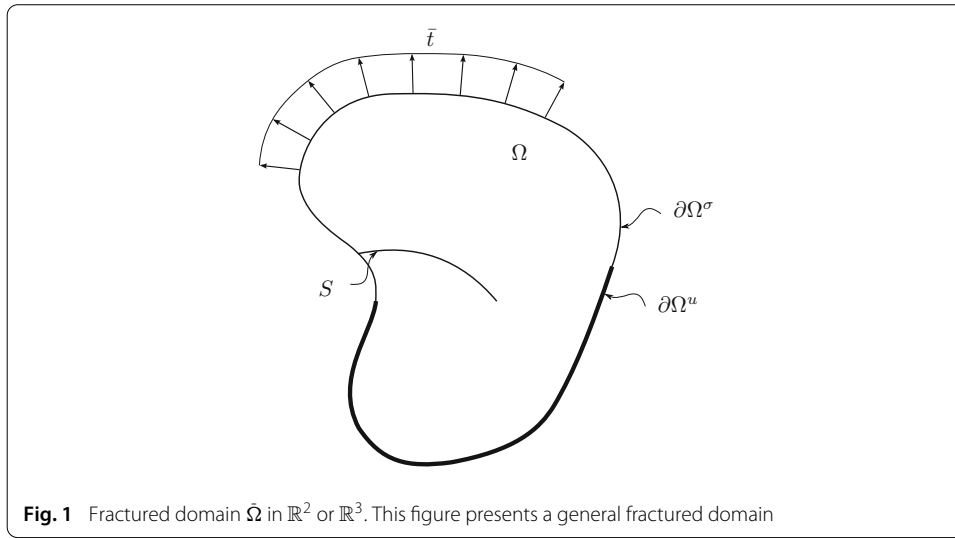
This paper presents an algorithm for non-intrusively coupling a commercial (or not) finite element software, with a research code implementing a hierarchical enrichment of finite element spaces. Examples of hierarchical methods supported by the proposed algorithm are the Generalized or eXtended FEM (GFEM) [5–9] and the GFEM with numerically defined enrichment functions [10,11]. The only requirement on the FEM software is that it outputs the global FEM stiffness matrix \mathbf{K}^0 and load vector \mathbf{f}^0 in a format readable by other software. The global FEM stiffness matrix \mathbf{K}^0 and load vector \mathbf{f}^0 can be assembled from any finite element available in the FEM solver. This includes structural elements like beams and shells, as well as continuum elements for 2-D and 3-D problems. The proposed hierarchical non-intrusive algorithm (HNA), like the non-intrusive algorithm presented in [12,13], exploits the hierarchic structure of the global matrix and vector in the GFEM: it contains the global FEM stiffness matrix \mathbf{K}^0 and load vector \mathbf{f}^0 associated with the underlying FEM mesh. This is demonstrated in “Discrete system of equations of the enriched global problem and the hierarchic non-intrusive algorithm” section. The HNA overcomes the memory requirement issues of the static condensation algorithm presented in [12,13] while not compromising its accuracy and flexibility. The implementation of the HNA is also simpler than the algorithm presented in [12,13]. The hierarchical non-intrusive algorithm is also related to the approach to embed arbitrary approximation spaces into classical finite element spaces recently proposed by Schweitzer and Ziegenhagel [14]. A brief review of approaches for the implementation of the GFEM in commercial FEA software can be found in [12].

The commercial FEM software adopted for all examples presented in this paper is Abaqus [1] while the research code is the Illinois Scientific and Engineering Toolbox (ISET). ISET implements the Generalized FEM and the GFEM with global–local enrichments (GFEM^{gl}). This method combines the GFEM with a classical global–local or sub-modeling analysis [15,16]. The enrichment functions in the GFEM^{gl} are the solution of local problems discretized with a GFEM enriched with analytically defined functions and a fine mesh around features of interest like fractures, material interfaces, etc. Further details on this GFEM are presented in “A scale-bridging GFEM for linear elastic fracture mechanics” section. The HNA combines the vast library of classical elements available in commercial FEM platforms with the ability of the GFEM^{gl} to analyze localized phenomena like cracks and spot welds, on coarse meshes. Details of the HNA are presented in “Discrete system of equations of the enriched global problem and the hierarchic non-intrusive algorithm” section.

The proposed Abaqus-ISET coupling is extensively verified in “Numerical experiments with the hierarchical non-intrusive algorithm”. The accuracy of the solution is also assessed using reference solutions for three-dimensional fracture problems.

Problem definition

The proposed hierarchical non-intrusive algorithm is not restricted to a particular problem. However, in this paper, the focus is on linear elastic fracture mechanics problems. 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

$$\nabla \cdot \boldsymbol{\sigma} = \mathbf{0} \quad \text{in } \Omega, \quad (1)$$

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

$$\mathbf{u} = \bar{\mathbf{u}} \text{ on } \partial\Omega^u \quad \boldsymbol{\sigma} \cdot \mathbf{n} = \bar{\mathbf{t}} \text{ on } \partial\Omega^\sigma, \quad (2)$$

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

$$\boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\varepsilon}, \quad (3)$$

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

$$\boldsymbol{\varepsilon} = \nabla_s \mathbf{u} \quad \text{in } \Omega, \quad (4)$$

where $\boldsymbol{\varepsilon}$ is the linear strain tensor and ∇_s is the symmetric part of the gradient operator.

We seek to find an approximation to the solution \mathbf{u} of the problem defined by Eqs. (1)–(4) using a FEM implemented in a commercial code like Abaqus [1], and a GFEM implemented in a research code.

GFEM approximations

A brief review of generalized FEM approximations is given in this section. Further details can be found in [5, 17–21].

The GFEM test and trial space \mathbb{S}^{GFEM} is obtained by hierarchically enriching a low-order standard finite element approximation space \mathbb{S}^{FEM} , with special functions related to the given problem and belonging to the enrichment space \mathbb{S}^{ENR} . Consider a finite element

mesh covering the domain of interest $\tilde{\Omega}$. Let $\varphi_\alpha(\mathbf{x})$, $\alpha \in I_h = \{1, \dots, NN\}$, be the standard linear finite element shape function associated with node \mathbf{x}_α and with support ω_α . The *patch or cloud* ω_α is given by the union of the finite elements sharing node \mathbf{x}_α . The test/trial space of the GFEM is given by

$$\mathbb{S}^{\text{GFEM}} = \mathbb{S}^{\text{FEM}} + \mathbb{S}^{\text{ENR}}, \quad (5)$$

where

$$\begin{aligned} \mathbb{S}^{\text{FEM}} &= \sum_{\alpha \in I_h} \hat{\mathbf{u}}_\alpha \varphi_\alpha, \quad \hat{\mathbf{u}}_\alpha \in \mathbb{R}^d, \quad d = 2, 3, \\ \mathbb{S}^{\text{ENR}} &= \sum_{\alpha \in I_h^e} \varphi_\alpha \chi_\alpha, \quad \text{and} \quad \chi_\alpha(\omega_\alpha) = \text{span}\{E_{\alpha i}\}_{i=1}^{m_\alpha}. \end{aligned} \quad (6)$$

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 $i = \{1, \dots, m_\alpha\}$ is the index of the enrichment function at the node with m_α being the total number of enrichments associated with node \mathbf{x}_α . The functions $E_{\alpha i} \in \chi_\alpha(\omega_\alpha)$ are chosen such that they approximate the unknown solution \mathbf{u} of the problem locally in ω_α . Examples of enrichment functions are polynomials, the Heaviside function, and numerically generated functions (cf. “A scale-bridging GFEM for linear elastic fracture mechanics” section). The spaces $\chi_\alpha(\omega_\alpha)$ are called *patch* approximation spaces, and \mathbb{S}^{ENR} is referred to as the *global enrichment space* of the GFEM. The functions in \mathbb{S}^{ENR}

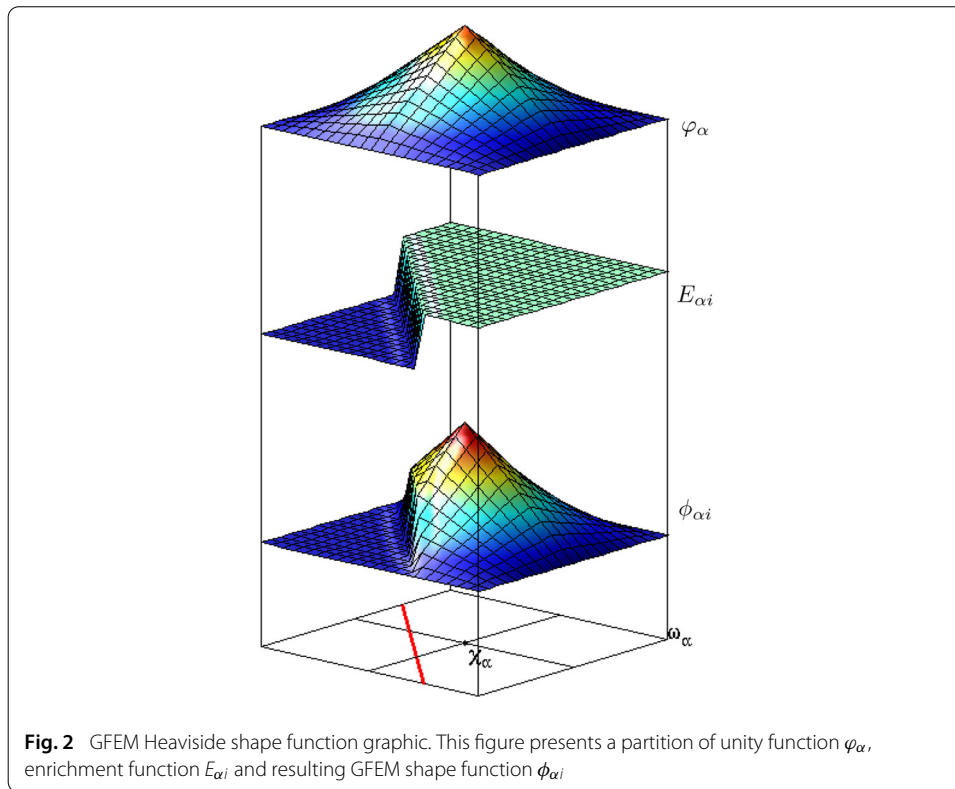
$$\phi_{\alpha i}(\mathbf{x}) = \varphi_\alpha(\mathbf{x}) E_{\alpha i}(\mathbf{x}), \quad \alpha \in I_h^e, \quad i = 1, \dots, m_\alpha, \quad (7)$$

are denoted GFEM shape functions. They are built from the product of Finite Element shape functions, $\varphi_\alpha(\mathbf{x})$, $\alpha \in I_h^e$, and enrichment functions, $E_{\alpha i}$, $i = 1, \dots, m_\alpha$. There are m_α GFEM shape functions at a node \mathbf{x}_α , $\alpha \in I_h^e$, of a finite element mesh. These nodes also have a standard FE shape function $\varphi_\alpha \in \mathbb{S}^{\text{FEM}}$. Nodes not in the set I_h^e have only one function—the FE shape function φ_α . Figure 2 illustrates the construction of a GFEM shape function using a step enrichment function.

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

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

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 the cornerstone of the hierarchical non-intrusive algorithm (HNA) presented in this paper.



In this work, the polynomial degree of a FEM or GFEM shape function is denoted by p . In all numerical experiments performed herein, the same polynomial order is adopted for all coordinate directions. Polynomial order $p = 1$ at a node \mathbf{x}_α implies that the node has only the FEM shape function φ_α .

Available enrichment functions for linear elastic fracture problems [5–9] 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 [10,11]. This allows the GFEM to use coarse meshes while delivering accurate solutions. This facilitates the non-intrusive implementation of the GFEM in a commercial FEM software as demonstrated in the next section. The so-called GFEM with global–local enrichments (GFEM^{gl}) is reviewed in the next section and used as a basis of the proposed non-intrusive algorithm.

A scale-bridging GFEM for linear elastic fracture mechanics

The GFEM with global–local enrichment functions (GFEM^{gl}) [10,11] combines the GFEM and the global–local FEM [15,16]. The enrichment functions in the GFEM^{gl} are the solution of local problems discretized with a GFEM enriched with analytically defined functions and a fine mesh around features of interest like fractures, material interfaces, etc. The GFEM^{gl} has been formulated and applied to various classes of problems including transient heat transfer [22], linear fracture [23,24], cohesive fracture [25], local plasticity [26], material heterogeneity [27], and localized thermoplasticity [28]. A GFEM^{gl} for linear

elastic fracture mechanics problems and suitable for a non-intrusive implementation of the method is described next.

Initial global problem of the GFEM^{gl}

Let \mathbf{u}^0 denote the standard FEM solution of the problem described in “Problem definition” section but *without* cracks. This is hereafter denoted as the *initial global problem*. The approximation \mathbf{u}^0 is the solution of the following problem: Find $\mathbf{u}^0 \in \mathbb{S}^0(\Omega)$ such that, $\forall \mathbf{v}^0 \in \mathbb{S}^0(\Omega)$

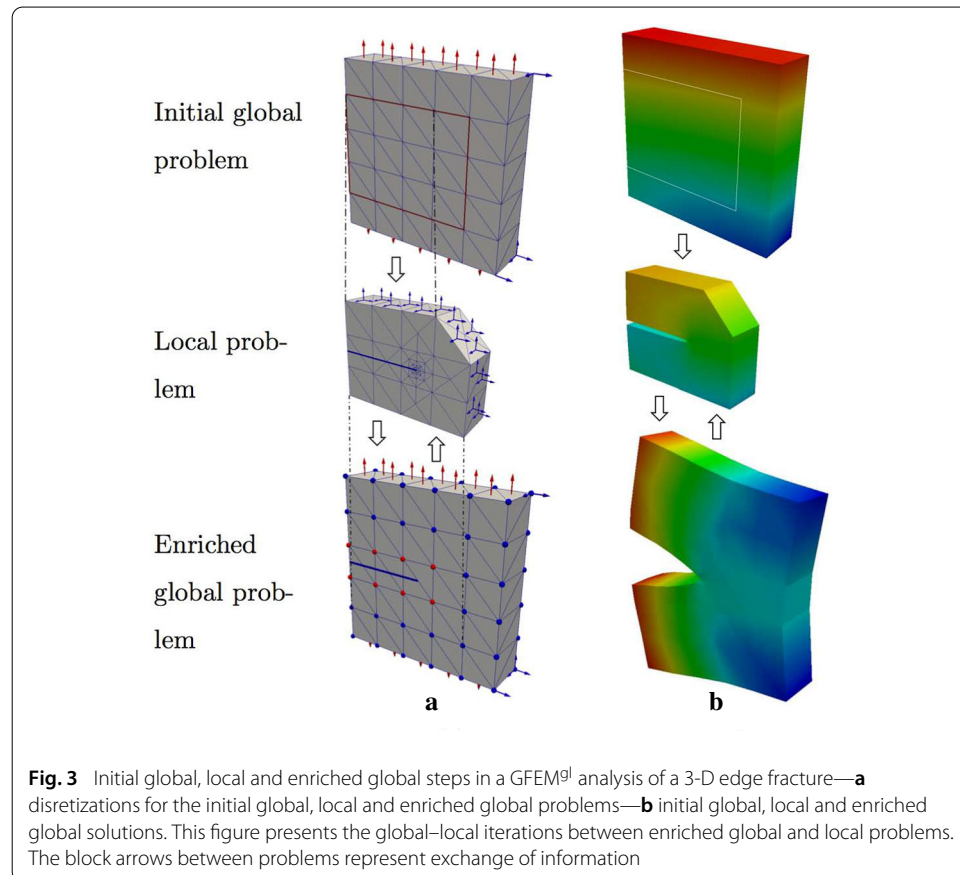
$$\int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}^0) : \boldsymbol{\varepsilon}(\mathbf{v}^0) d\Omega = \int_{\partial\Omega^\sigma} \bar{\mathbf{t}} \cdot \mathbf{v}^0 d\partial\Omega \quad (9)$$

where $\mathbb{S}^0(\Omega)$ is standard FEM space—the span of shape functions available in a commercial software like Abaqus [1]. Accurate solutions can be computed using a commercial software and quasi-uniform meshes, like the one shown in Fig. 3, since cracks are not considered in the initial global problem.

Crack-scale local problem

Let the local domain $\tilde{\Omega}^L = \Omega^L \cup \partial\Omega^L$ where $\Omega^L \subset \Omega$, be selected from Ω as the union of the support of nodes in the set \mathcal{I}^{gl} , to wit,

$$\Omega^L = \bigcup_{\alpha \in \mathcal{I}^{gl}} \omega_\alpha \quad (10)$$



Nodes in set \mathcal{I}^{gl} are denoted “seed” nodes. They are the only set of nodes from the global mesh enriched with global–local functions. As an example, seed nodes are shown in red in Fig. 3a.

Let $\mathbf{u}^L \in \mathbb{S}^L(\Omega^L)$ denote the GFEM approximation of the local solution computed on Ω^L . Fine meshes and GFEM shape functions for fractures [5–9, 29] are used in the definition of $\mathbb{S}^L(\Omega^L)$. These include Heaviside and singular crack front enrichments. Therefore, the local problem must be solved by a GFEM research code. An example of a local mesh is shown in Fig. 3a. The weak statement of the local problem is as follows: Find $\mathbf{u}^L \in \mathbb{S}^L(\Omega^L)$ such that, $\forall \mathbf{v}^L \in \mathbb{S}^L(\Omega^L)$

$$\begin{aligned} & \int_{\Omega^L} \boldsymbol{\sigma}(\mathbf{u}^L) : \boldsymbol{\varepsilon}(\mathbf{v}^L) \, d\Omega^L + \kappa \int_{\partial\Omega^L \setminus (\partial\Omega^L \cap \partial\Omega)} \mathbf{u}^L \cdot \mathbf{v}^L \, d\partial\Omega^L \\ &= \int_{\partial\Omega^L \cap \partial\Omega^\sigma} \bar{\mathbf{t}} \cdot \mathbf{v}^L \, d\partial\Omega^L + \int_{\partial\Omega^L \setminus (\partial\Omega^L \cap \partial\Omega)} (\mathbf{t}(\mathbf{u}^0) + \kappa \mathbf{u}^0) \cdot \mathbf{v}^L \, d\partial\Omega^L \end{aligned} \quad (11)$$

A key aspect of problem (11) is the use of the GFEM solution of the initial global problem \mathbf{u}^0 , as boundary conditions on $\partial\Omega^L \setminus (\partial\Omega^L \cap \partial\Omega)$, the portion of the local boundary that does not intersect the boundary of the global problem.

The traction vector $\mathbf{t}(\mathbf{u}^0)$ on $\partial\Omega^L \setminus (\partial\Omega^L \cap \partial\Omega)$ is computed from the initial global solution using

$$\mathbf{t}(\mathbf{u}^0) = \boldsymbol{\sigma}(\mathbf{u}^0) \cdot \mathbf{n} \quad (12)$$

with \mathbf{n} being the outward unit normal vector to $\partial\Omega^L \setminus (\partial\Omega^L \cap \partial\Omega)$. The parameter κ is a spring stiffness. When $\kappa = 0$, the boundary condition on $\partial\Omega^L \setminus (\partial\Omega^L \cap \partial\Omega)$ becomes a Neumann boundary condition. When κ equals to a large value (i.e., a penalty), the boundary condition becomes a Dirichlet boundary condition. An intermediate value leads to a spring boundary condition. In all problems solved in this paper, the spring stiffness is automatically selected using [24]

$$\kappa = \frac{E}{\sqrt[n]{V_0 J}}, \quad (13)$$

where n is the number of the spatial dimensions of the problem and V_0 is the volume of the master element. This choice of κ yields a spring stiffness that is proportional to the element stiffness. For further details on the choice of spring stiffness, please refer to [24]. It is noted that an in-depth study of the choice of the spring value (or impedance) in mixed non-intrusive coupling is presented in [30].

The local solution \mathbf{u}^L defined in this section is the same solution provided by the classical global–local or sub-modeling analysis [31] with local solutions computed by a GFEM instead of the standard FEM. As such, the local solution may have large errors due to the use of inexact boundary conditions on $\partial\Omega^L \setminus (\partial\Omega^L \cap \partial\Omega)$. This is addressed in the GFEM by re-solving the global problem with GFEM shape functions defined using \mathbf{u}^L . This is described in the next section.

Enriched global problem of the GFEM^{gl}

The solution \mathbf{u}^L of the local problem (11) yields the so-called global–local enrichments for the global problem. They are used to define global–local GFEM shape functions using

$$\phi_{\alpha}^{gl}(\mathbf{x}) = \varphi_{\alpha}(\mathbf{x})\mathbf{u}^L(\mathbf{x}), \quad \alpha \in \mathcal{I}^{gl} \quad (14)$$

These functions are hierarchically added to the seed nodes \mathcal{I}^{gl} defined earlier. They enrich the approximation space $\mathbb{S}^0(\Omega)$ of the initial global problem while keeping the global mesh unchanged. The global GFEM space containing these functions is given by

$$\mathbb{S}^E(\Omega) = \mathbb{S}^0(\Omega) + \left\{ \sum_{\alpha \in \mathcal{I}^{gl}} \varphi_{\alpha}(\mathbf{x})\mathbf{u}_{\alpha}^{gl}(\mathbf{x}) \right\}, \quad (15)$$

where for a 3-D problem

$$\mathbf{u}_{\alpha}^{gl}(\mathbf{x}) = \{u_{\alpha}u_u^L(\mathbf{x}), v_{\alpha}u_v^L(\mathbf{x}), w_{\alpha}u_w^L(\mathbf{x})\}^T \quad (16)$$

with u_{α} , v_{α} , w_{α} being global dofs and $u_u^L(\mathbf{x})$, $u_v^L(\mathbf{x})$, $u_w^L(\mathbf{x})$ are corresponding components of the local solution vector $\mathbf{u}^L(\mathbf{x})$. Therefore, only three enrichments are added to global (seed nodes), regardless of the number of dofs of the local problem. The 2-D case is analogous. Problem (9) with space $\mathbb{S}^E(\Omega)$ is called the *enriched global problem*. The solution of this problem is denoted \mathbf{u}^E . By definition, it belongs to $\mathbb{S}^E(\Omega)$.

Overall GFEM^{gl} algorithm

An example with all the steps of a GFEM^{gl} analysis is illustrated in Fig. 3. It shows the domain, mesh, and boundary condition of the initial global, local and enriched global problems on the left side and the solution of each problem on the right side. The domain is a panel with an edge-crack on the left edge, as shown in the local and enriched global problems. It is noted that the crack cuts elements in the global problem. The red arrows represent Neumann boundary conditions, while the blue arrows indicate either face or point Dirichlet boundary conditions. The latter are used to prevent rigid body motions. Seed nodes in the global problem are shown in red. They are enriched with the local solution while blue nodes are not enriched—they have only the standard FEM shape functions for tetrahedron elements. The initial global problem shows a maroon box outlining where the local domain Ω^L is located. The crack is discretized in the local domain and is shown as a dark blue line in the figure. The local domain is subjected to Dirichlet boundary conditions (blue arrows) provided by the solution \mathbf{u}^0 of the initial global problem. The local problem mesh is refined so that the characteristic length of the elements intersected by the crack front is about 5% of the crack size, unless stated otherwise. The polynomial order of the GFEM shape functions in the local problem is taken as $p = 3$. Singular crack front and discontinuous Heaviside enrichments are used as well. The block arrows shown between problems represent exchange of information: Boundary conditions from global to local problem and enrichments from local to global problem. Note that the enriched global problem solution, \mathbf{u}^E , can also be used as boundary conditions for the local problem, leading to *global–local iterations* between these two problems [11, 32]. These iterations are used to improve the quality of the local solutions (global–local enrichments) and in turn, the accuracy of \mathbf{u}^E . While the proposed non-intrusive implementation of the GFEM^{gl} supports global–local iterations, they are not used in the problems solved in this paper.

The above presentation assumes that all the steps in a GFEM^{gl} simulation are performed by a single GFEM solver. This does not have to be the case. The hierarchical structure of

the enriched global space $\mathbb{S}^E(\Omega)$ can be used to split the algorithm between two solvers: a classical FEM solver that implements the global space $\mathbb{S}^0(\Omega)$ and a GFEM solver the implements the global–local GFEM shape functions. Details are provided in the next section.

Discrete system of equations of the enriched global problem and the hierarchic non-intrusive algorithm

The solution $\mathbf{u}^E \in \mathbb{S}^E$ of the enriched global problem defined in “Enriched global problem of the GFEM^{gl}” section can be written in matrix format as [12,13]

$$\mathbf{u}^E = \underbrace{\mathbf{u}^0}_{\text{std FEM approx.}} + \underbrace{\mathbf{u}^{\text{gl}}}_{\text{GFEM}^{\text{gl}} \text{ enriched approx.}} = \mathbf{N}^E \underline{\mathbf{u}}^E = \begin{bmatrix} \mathbf{N}^0 & \mathbf{N}^{\text{gl}} \end{bmatrix} \begin{bmatrix} \underline{\hat{\mathbf{u}}}^0 \\ \underline{\hat{\mathbf{u}}}^{\text{gl}} \end{bmatrix} \quad (17)$$

where \mathbf{N}^0 is a matrix with shape functions used in the FEM discretization of the initial global problem, \mathbf{N}^{gl} has the global–local GFEM shape functions defined in (14), $\underline{\hat{\mathbf{u}}}^0$ are standard FEM degrees of freedom associate with shape functions in \mathbf{N}^0 , and $\underline{\hat{\mathbf{u}}}^{\text{gl}}$ are degrees of freedom associated with (hierarchical) global–local GFEM shape functions in \mathbf{N}^{gl} . An analogous matrix equation can be written for the GFEM approximation $\mathbf{u}^{\text{GFEM}} \in \mathbb{S}^{\text{GFEM}}$ given in (8). Therefore, the derivation that follows can be used for both \mathbf{u}^E and \mathbf{u}^{GFEM} .

Using decomposition (17) of the displacement field, the strain-displacement matrix of the enriched global problem can be written as

$$\mathbf{B}^E = \mathbf{L} \begin{bmatrix} \mathbf{N}^0 & \mathbf{N}^{\text{gl}} \end{bmatrix} = \begin{bmatrix} \mathbf{B}^0 & \mathbf{B}^{\text{gl}} \end{bmatrix} \quad (18)$$

where for a 3-D problem

$$\mathbf{L} = \begin{bmatrix} \partial/\partial x & 0 & 0 \\ 0 & \partial/\partial y & 0 \\ 0 & 0 & \partial/\partial z \\ \partial/\partial y & \partial/\partial x & 0 \\ 0 & \partial/\partial z & \partial/\partial y \\ \partial/\partial z & 0 & \partial/\partial x \end{bmatrix}$$

This leads to the following decomposition of the stiffness matrix and load vector of the enriched global problem

$$\mathbf{K}^E \underline{\mathbf{u}}^E = \begin{bmatrix} \mathbf{K}^0 & \mathbf{K}^{0,\text{gl}} \\ \mathbf{K}^{\text{gl},0} & \mathbf{K}^{\text{gl}} \end{bmatrix} \begin{bmatrix} \underline{\hat{\mathbf{u}}}^0 \\ \underline{\hat{\mathbf{u}}}^{\text{gl}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}^0 \\ \mathbf{f}^{\text{gl}} \end{bmatrix} = \mathbf{f}^E \quad (19)$$

where

$$\mathbf{K}^0 = \int_{\Omega} (\mathbf{B}^0)^T \mathbb{C} \mathbf{B}^0 \, d\Omega$$

is the global stiffness matrix of the initial global problem (9), and \mathbb{C} is the Hooke tensor \mathbf{C} in matrix format. Stiffness matrix \mathbf{K}^0 can be computed by a standard FE software like Abaqus [1]. The matrix

$$\mathbf{K}^{0,\text{gl}} = (\mathbf{K}^{\text{gl},0})^T = \int_{\Omega^L} (\mathbf{B}^0)^T \mathbb{C} \mathbf{B}^{\text{gl}} \, d\Omega^L \quad (20)$$

represents the coupling between standard and GFEM shape function, and

$$\mathbf{K}^{\text{gl}} = \int_{\Omega^L} (\mathbf{B}^{\text{gl}})^T \mathbb{C} \mathbf{B}^{\text{gl}} d\Omega^L \quad (21)$$

represents the global stiffness associated with GFEM shape functions. The initial global problem stiffness matrix is nested in the enriched global problem stiffness matrix since the global–local GFEM shape functions (14) are hierarchically added to the initial problem discretization space $\mathbb{S}^0(\Omega)$.

Static condensation algorithm The hierarchic structure of \mathbf{K}^E was explored in [12,13] to solve the system of equations (19) using static condensation of the degrees of freedom $\tilde{\mathbf{u}}^{\text{gl}}$. The algorithm was formulated such that it could be implemented non-intrusively using a commercial software like Abaqus [1] and a GFEM code. It basically involves sending a set of pseudo loads from the GFEM to the FEM solver which solves, using \mathbf{K}^0 , for a set of pseudo solution vectors. These are sent back to the GFEM solver which then computes solution vectors $\tilde{\mathbf{u}}^0$ and $\tilde{\mathbf{u}}^{\text{gl}}$. The procedure is repeated in the case of crack propagation. It provides exactly the same solution as when solving the global system (19) in the GFEM solver and it does not involve iterations between the two solvers. However, the memory requirements of the algorithm can be quite high for large problems. One load and one solution vector is defined for each degree of freedom in $\tilde{\mathbf{u}}^{\text{gl}}$. The dimension of these vectors is equal to the dimension of $\tilde{\mathbf{u}}^0$ which is usually high. While the pseudo load vectors are quite sparse, the pseudo solution vectors in general are not, requiring a dense rectangular matrix of dimensions $\dim(\tilde{\mathbf{u}}^0) \times \dim(\tilde{\mathbf{u}}^{\text{gl}})$ to store them. Furthermore, when applying the algorithm to heat conduction problems, Abaqus re-factorizes matrix \mathbf{K}^0 for every right hand side vector [13]. This makes the algorithm more expensive than if the enriched global problem is solved entirely by a GFEM solver.

Hierarchical non-intrusive algorithm—basic idea The proposed HNA algorithm seeks to overcome the memory requirement issues of the static condensation algorithm while not compromising its accuracy and flexibility. The HNA achieves this by utilizing the ability of FEM solvers (commercial or not) to output the FEM stiffness matrix \mathbf{K}^0 and load vector \mathbf{f}^0 to files readable by other software. These files store \mathbf{K}^0 and \mathbf{f}^0 in a sparse format and thus \mathbf{K}^0 and \mathbf{f}^0 can be efficiently loaded into a GFEM solver. Furthermore, since \mathbf{K}^0 and \mathbf{f}^0 do not change during, for example, the simulation of a fracture propagation problem, they only need to be read once at the beginning of the simulation. A step-by-step description of the HNA algorithm is provided in the next section.

The global FEM stiffness matrix \mathbf{K}^0 and load vector \mathbf{f}^0 can be assembled from any finite element available in a commercial solver. This includes structural elements like beams and shells, as well as continuum elements for 2-D and 3-D problems. Thus the HNA combines the vast library of classical elements available in a commercial code with the ability of the GFEM^{gl} to analyze localized phenomena like cracks [23,24], spot welds [33], material heterogeneity [27], high thermal gradients [22], etc., on a coarse mesh provided by the commercial code. One practical requirement of the HNA is that the finite elements adopted over the subdomain $\Omega^L \subset \Omega$ must be available in both solvers. I.e., the FEM mesh for the global domain Ω must adopt elements over the subdomain $\Omega^L \subset \Omega$ that are also implemented in the GFEM solver. Alternatively, user elements implemented in the commercial solver can be used in the subdomain. This type of element is supported by many commercial solvers. This requirement on the type of element is necessary for the computation of the coupling global matrix $\mathbf{K}^{0,\text{gl}}$ which requires knowledge of \mathbf{B}^0 by the

GFEM solver when integrating $\mathbf{K}^{0,gl}$. It is noted that any type of element can be used in the local domain Ω^L when solving the local problem. They don't have to match the type of the element used in the global subdomain $\Omega^L \subset \Omega$. One example is presented in [33] where hexahedron elements are used in the global domain while the local problems adopt tetrahedron elements.

The commercial FEM software adopted for all examples in this paper is Abaqus [1]. The GFEM code used in this paper is hereafter called the Illinois Scientific and Engineering Toolbox (ISET). Standard 2-D finite elements implemented in ISET include TRI3, TRI6, QUAD4, and QUAD8. Three-dimensional elements include TET4, TET10, HEX8, HEX20, and HEX27. All elements can be used for linear and material non-linear elasticity and heat conduction. Multi-physics problems like thermo-elasticity, thermo-plasticity and hydraulic fracturing, are also supported.

The HNA algorithm can be used with any method that hierarchically *enriches* a FEM space. However, for many classes of problems, in particular fracture mechanics problems, mesh *refinement* is required for acceptable accuracy. This is also the case of the GFEM with analytically defined enrichment functions. While singular crack tip enrichments significantly improves the accuracy of the solution, a sufficiently fine mesh is still necessary, in particular when solving 3-D problems. In contrast, the GFEM^{gl} can provide accurate solutions using fairly coarse 3-D meshes composed of tetrahedral or hexahedral elements at the global problem. Thus, the GFEM^{gl} is ideally suitable for the HNA since it enables the analysis of fractures of a *coarse* FEM mesh defined in a commercial solver. This allows the *same* FEM mesh to be used for the simulation of fractures anywhere in the domain of interest. ISET implements the GFEM^{gl} described in "A scale-bridging GFEM for linear elastic fracture mechanics" section and the HNA described next.

Remark While the ability of FEM solvers of dumping \mathbf{K}^0 and \mathbf{f}^0 is explored here in the context of the GFEM^{gl}, it can also be applied in the implementation of the Iterative Global–Local FEM [34,35]. This would avoid the potential re-factorization by the commercial solver of the global matrix \mathbf{K}^0 at every global–local iteration. By reading \mathbf{K}^0 and \mathbf{f}^0 at the beginning of the iterative process it needs to be factorized only once. This is particularly appealing when the global problem is assumed to remain linear while still allowing a non-linear material response in the local problem. It is noted that not all implementations of the Iterative Global–Local FEM require re-factorization of the global matrix as shown, for example, in Duval et. al [36].

Implementation of the hierarchic non-intrusive algorithm

All the steps of the hierarchical non-intrusive algorithm presented in the previous section are listed in Algorithm 1. The algorithm has been implemented and tested with Abaqus [1] but it does not assume a particular FEM solver. The global matrix and load vector in (19) are assembled using \mathbf{K}^0 and \mathbf{f}^0 provided by the FEM solver and \mathbf{K}^{gl} , $\mathbf{K}^{0,gl}$, and \mathbf{f}^{gl} provided by the GFEM solver. Dirichlet boundary conditions are imposed by the GFEM code once \mathbf{K}^E and \mathbf{f}^E are assembled. A suitable sparse solver is used to factorize \mathbf{K}^E and solve for \mathbf{u}^E . In the computations presented in this paper, the Pardiso solver [37,38] is adopted. Therefore, the available factorization of submatrix \mathbf{K}^0 is not used when solving (19). There are however several block iterative solvers that have been proposed to solve system (19) in a more efficient manner. Examples are presented in [39–41].

Algorithm 1 Hierarchical Non-Intrusive Algorithm

```

1: procedure HNA
2:   FEM code computes  $\mathbf{K}^0$  and  $\mathbf{f}^0$ 
3:   FEM code outputs  $\mathbf{K}^0$  and  $\mathbf{f}^0$  into files (see A for Abaqus commands)
4:   GFEM code reads  $\mathbf{K}^0$  and  $\mathbf{f}^0$ 
5:   if using the GFEMgl then
6:     GFEM code imposes Dirichlet boundary conditions on  $\mathbf{K}^0$  and  $\mathbf{f}^0$ 
7:     GFEM code solves initial global problem  $\mathbf{K}^0 \mathbf{u}^0 = \mathbf{f}^0$ 
8:     GFEM code enforces boundary conditions on local problem using  $\mathbf{u}^0$ 
9:     GFEM code solves local problem
10:    GFEM code enriches seed nodes of the FEM mesh and integrates  $\mathbf{K}^{gl}$ ,  $\mathbf{K}^{0,gl}$ , and  $\mathbf{f}^{gl}$ 
11:  else
12:    GFEM code computes analytical enrichment functions (Heaviside and crack tip enrichments for crack problems), and integrates  $\mathbf{K}^{gl}$ ,  $\mathbf{K}^{0,gl}$ , and  $\mathbf{f}^{gl}$ 
13:  end if
14:  GFEM code imposes Dirichlet boundary conditions on  $\mathbf{K}^E$  and  $\mathbf{f}^E$ 
15:  GFEM code solves the enriched global problem  $\mathbf{K}^E \mathbf{u}^E = \mathbf{f}^E$ 
16:  GFEM code post-processes solution over subdomain  $\Omega^L$  of problem domain  $\Omega$ 
17:  To visualize element-wise quantities in  $\Omega \setminus \Omega^L$ , GFEM code outputs to a file Dirichlet boundary conditions over the boundary of this subdomain
18:  FEM code reads file with Dirichlet boundary conditions and post-processes element-wise quantities in  $\Omega \setminus \Omega^L$ 
19: end procedure

```

It is important to state that the HNA supports the any type of enrichment at nodes of a FEM mesh, derived from a local problem or analytically defined. All that it relies on is the hierarchical nature of the enrichment and thus of the global stiffness matrix and global vector.

The post-processing of the problem solution over elements in the subdomain $\Omega^L \subset \Omega$ poses no problem since these elements and their enrichments are available in the GFEM solver. Nodal displacement and rotations at nodes in $\Omega \setminus \Omega^L$ are available after the GFEM solver solves for the global vector \mathbf{u}^E . However, the post-processing of the problem solution over elements which are available only in the FEM solver must be addressed. By assumption, these elements are only in the subdomain $\Omega \setminus \Omega^L$. Since the details of the formulation of elements used in $\Omega \setminus \Omega^L$ are in general not known, quantities like stress and strain in these elements cannot be computed by the GFEM solver even after it solves for the global vector \mathbf{u}^E . The following approach is adopted in this work in order to address this issue. A new job, identical to the original problem, but with additional Dirichlet boundary conditions along the boundary of $\Omega \setminus \Omega^L$, is submitted to the FEM solver. Entries from the global solution vector \mathbf{u}^E are used to setup Dirichlet boundary conditions on the boundary of $\Omega \setminus \Omega^L$. This FEM solution over $\Omega \setminus \Omega^L$ together with the GFEM solution over Ω^L provides the solution of the original problem over the analysis domain Ω .

For the examples in this paper, Abaqus operates as the FEM code in Algorithm 1 while ISET is the GFEM code. The specific steps where each solver is used are listed in Algorithm 1. A HNA simulation based on these two solvers is denoted hereafter an HNA Abaqus-ISET simulation. If ISET (only) is used to solve a problem, this is denoted an ISET simulation. In both cases, the enrichment may be analytically or numerically defined. They correspond to the GFEM and the GFEM^{gl}, respectively.

Numerical experiments with the hierarchical non-intrusive algorithm

This section presents the solution of 2-D and 3-D fracture mechanics problems using the proposed hierarchical non-intrusive algorithm. The algorithm is used with

both the GFEM (analytical enrichments only) and the GFEM^{gl} (numerically defined enrichments).

Two-dimensional edge crack

Consider the 2-D square domain $\Omega \subset \mathbb{R}^2$ with a crack and loaded as shown in Fig. 4. The dimensions are $b = 4$, $l = 2$, and $a = 2$. A state of plane strain is assumed. The Young's modulus $E = 1$, the Poisson's ratio $\nu = 0.3$ and the magnitude of the tractions is taken as $\sigma = 1$.

The problem is solved using the hierarchical non-intrusive algorithm implemented with Abaqus and ISET solvers. Different FEM discretizations provided by Abaqus are adopted as described below. Analytical enrichments in form of crack tip functions [5–7] are used by the GFEM implemented in ISET. The FEM nodes enriched with these functions is shown in Figs. 5 and 7. Thus, in this section, only analytically defined enrichments are adopted.

Verification of HNA with TRI3 elements

In this section, the problem is solved with the FEM mesh of linear triangular elements shown in Fig. 5 and the crack tip enrichments shown in Fig. 5. This triangular element is denoted by TRI3 in Abaqus. The ISET implementation of TRI3 is identical to the one in Abaqus. Thus, the setup shown in Fig. 5 can be used to verify the HNA implementation with Abaqus-ISET against a solution computed with ISET only.

In the case of Abaqus-ISET solver, the mesh shown in Fig. 5 is used by Abaqus to produce \mathbf{K}^0 and \mathbf{f}^0 . ISET then uploads them and computes the additional matrices $\mathbf{K}^{0,gl}$, \mathbf{K}^{gl} and vector \mathbf{f}^{gl} used in (19). This system is then solved for \mathbf{u}^E .

Remark The notation $\mathbf{K}^{0,gl}$, \mathbf{K}^{gl} , and \mathbf{f}^{gl} is adopted here even though in this section these quantities are defined using analytical enrichments, not global–local functions. This

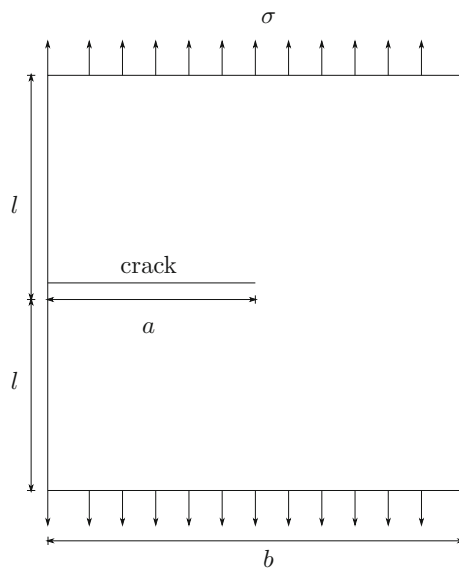


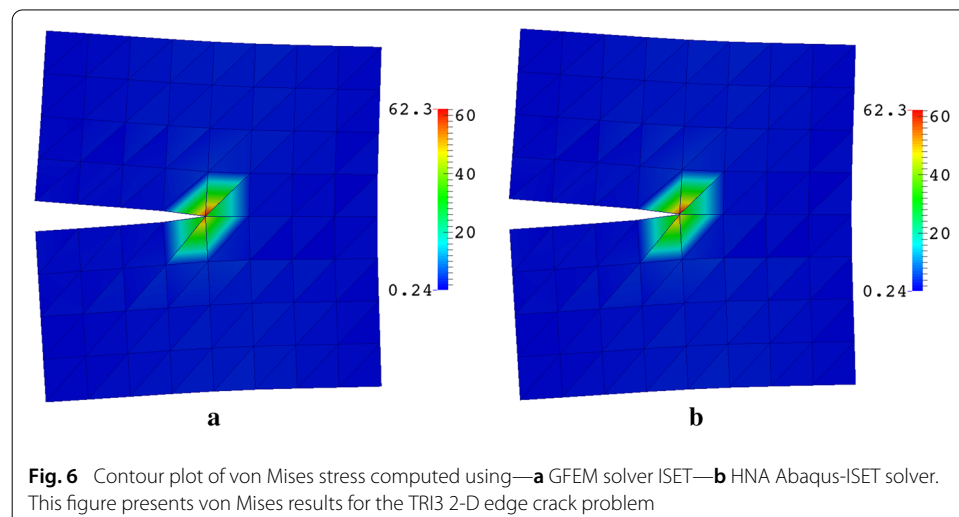
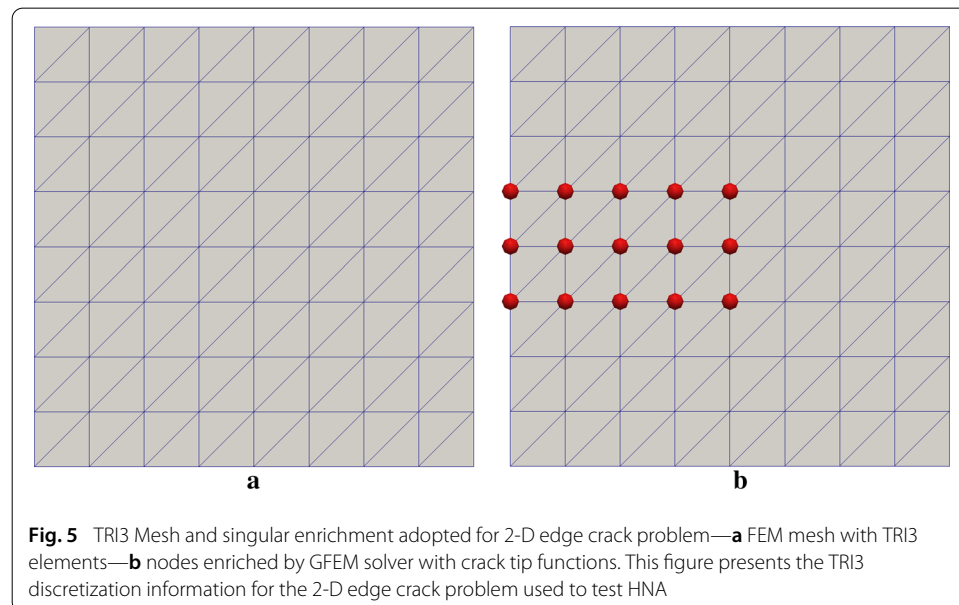
Fig. 4 Two-dimensional edge crack. This figure presents the geometry for a two-dimensional crack used to test HNA

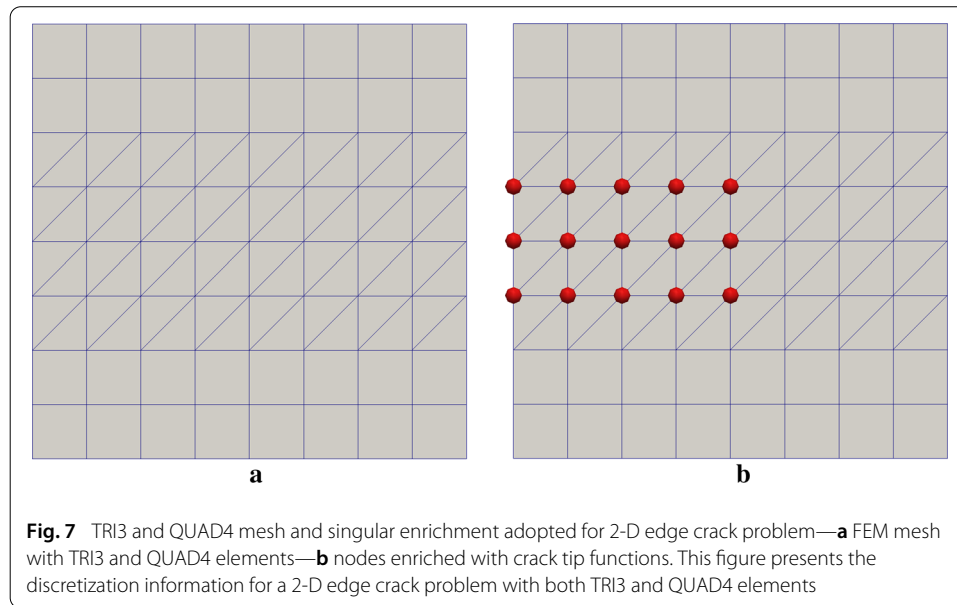
is justified since the difference in the type of enrichment is irrelevant for the proposed HNA.

Figure 6 shows contour plots of the von Mises stress computed with both solution strategies. As expected, the two solutions look identical. This confirms that the HNA does not introduce any approximation beyond the underlying GFEM or GFEM^{gl} discretizations adopted. It simply partitions the solution process based on the hierarchical nature of the GFEM and GFEM^{gl} spaces. Of course there are minor implementation differences between Abaqus and ISET which might lead to some small differences in results even when the approximation spaces adopted by both solvers are theoretically the same. In the case of this problem, the strain energy computed using Abaqus-ISET and ISET has a relative difference of only $5.6E-14$.

Improvement of solution using Abaqus QUAD4 elements

The 2-D edge crack is solved again in this section using the FEM mesh shown in Fig. 7. This mesh adopts TRI3 and QUAD4 elements. All other parameters are unchanged. All





elements that are enriched with singular crack tip functions are TRI3 elements as in the previous section (cf. Fig. 7). The Abaqus QUAD4 element adopts reduced integration when computing the element stiffness matrix. This leads to more accurate solutions than the standard bi-linear quadrilateral element with full integration. They are used when solving the problem with the Abaqus-ISET solver.

It is noted that the QUAD4 Abaqus element is not available in ISET. Thus, when solving the problem with ISET only, standard bi-linear quadrilateral elements with full integration are used. Therefore, the ISET and the Abaqus-ISET solutions are different.

The strain energy of the solution computed by ISET (only) is equal to 28.63 while in the case of the Abaqus-ISET solver it is equal to 29.24. As expected there is a clear difference between these values because of the difference in quadrilateral elements.

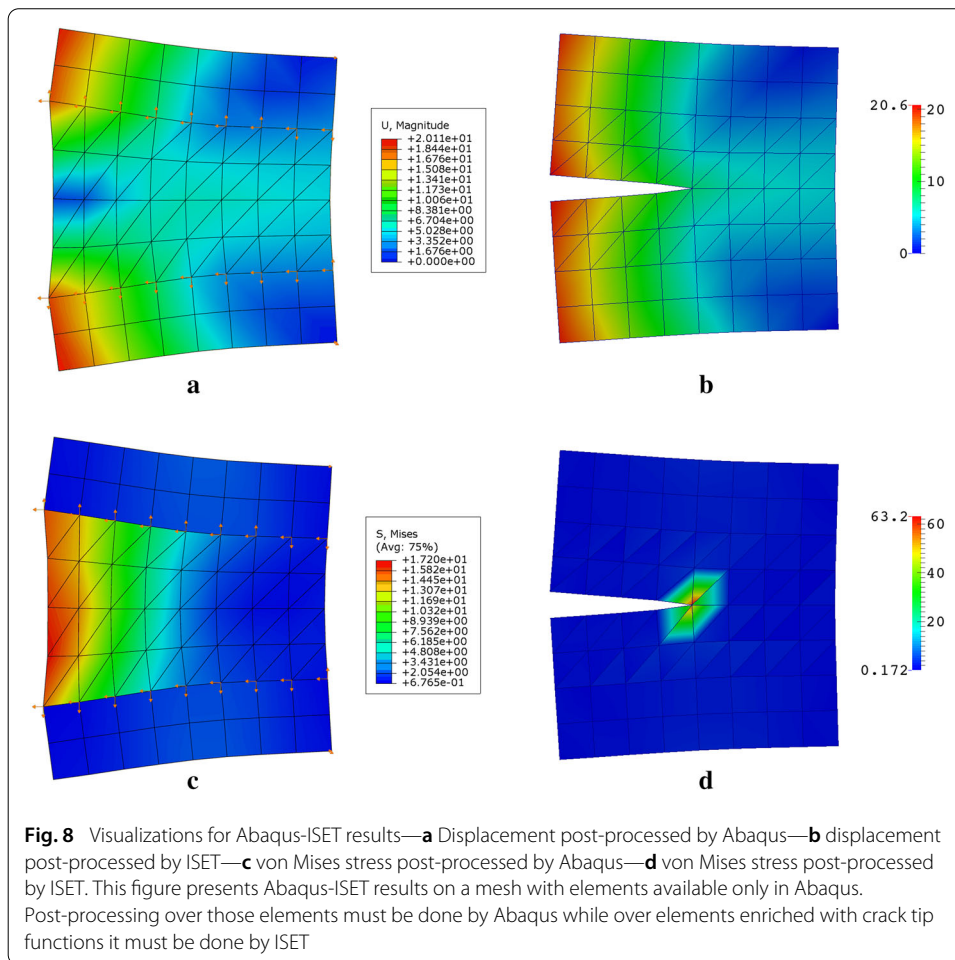
A reference strain energy equal to 36.97 was computed using a much refined mesh of triangles, GFEM shape functions of degree three, and the same crack tip enrichments shown in Fig. 5. Using this reference value, the Abaqus-ISET strain energy has a relative error of 20.9%, while the ISET strain energy has an a relative error of 22.6%, attesting the benefit of using the improved QUAD4 elements available in Abaqus.

The Mode I stress intensity factor extracted from the reference, Abaqus-ISET, and ISET only solutions are $K_I^{\text{ref}} = 7.542$, $K_I^{\text{Abaqus-ISET}} = 6.573$, and $K_I^{\text{ISET}} = 6.534$, respectively. The relative error of $K_I^{\text{Abaqus-ISET}}$ and K_I^{ISET} are 12.85 and 13.37%, respectively.

The above results illustrate one of the benefits of the HNA which combines the library of elements available in Abaqus with the GFEM and GFEM^{sl} provided by ISET or another software that implements these methods.

Post-processing of solution Here the post-processing strategy described in “Implementation of the hierarchic non-intrusive algorithm” section is necessary since the solution cannot be post-processed by ISET over the QUAD4 elements implemented in Abaqus.

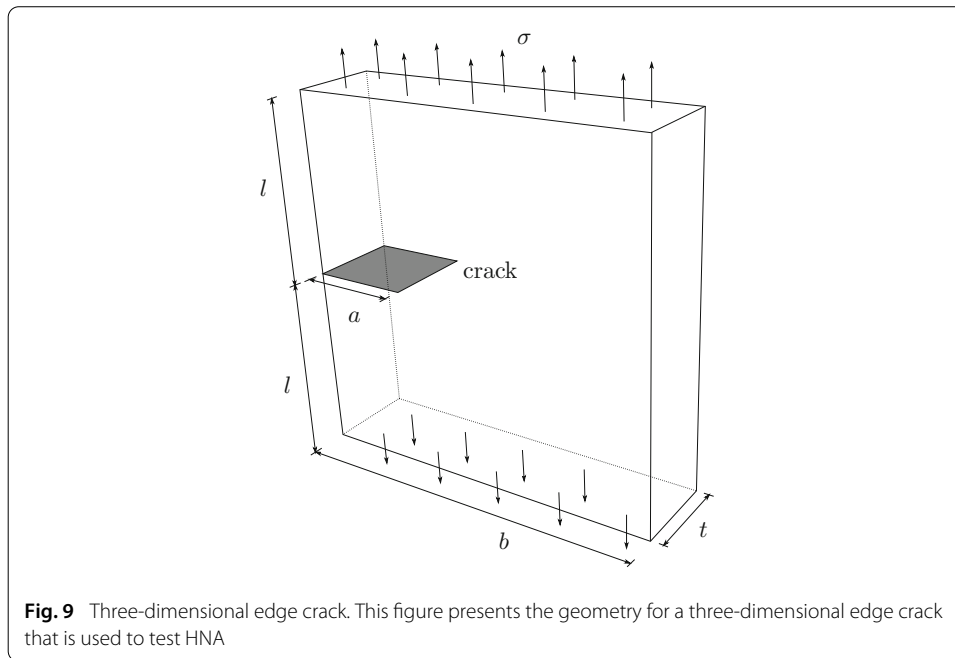
Figure 8a shows the post-processing in Abaqus of the displacement field computed by Abaqus-ISET. Dirichlet boundary conditions provided by the Abaqus-ISET solution are prescribed at the edges between triangular and quadrilateral elements. This allows



Abaqus to correctly post-process the solution over all QUAD4 elements. The displacement over triangular elements shown in Fig. 8a are obviously incorrect since Abaqus can not post-process the solution over elements that have GFEM enrichments. The correct displacement is shown in Fig. 8b which was post-processed by ISET. The discontinuous displacement field can be properly visualized. It is noted that the contour plots over quadrilateral elements shown in Fig. 8b are only approximate since they were performed using the quadrilateral elements available in ISET, which as discussed earlier, are not the same as those used by Abaqus in the computation on K^0 . However, for this problem and type of elements used in Abaqus and ISET, the difference in the post-processed values are visually identical. This is not always the case. Figure 8c, d show contour plots of von Mises stress computed from the HNA Abaqus-ISET solution. They are analogous to Fig. 8a, b. The only difference is the quantity being post-processed.

Three-dimensional edge crack

The 3-D edge crack shown in Fig. 9 is solved in this section using the hierarchical non-intrusive algorithm implemented with Abaqus-ISET and with ISET only for verification purposes. The dimensions shown in Fig. 9 are $b = 4$, $l = 2$, $t = 1$, and $a = 1$. The Young's modulus $E = 200,000$, the Poisson's ratio $\nu = 0.3$ and the magnitude of the tractions is taken as $\sigma = 1$. Point displacement boundary conditions are assigned to selected nodes of the FEM mesh to prevent rigid body motions. The GFEM^{gl} is used to numerically define



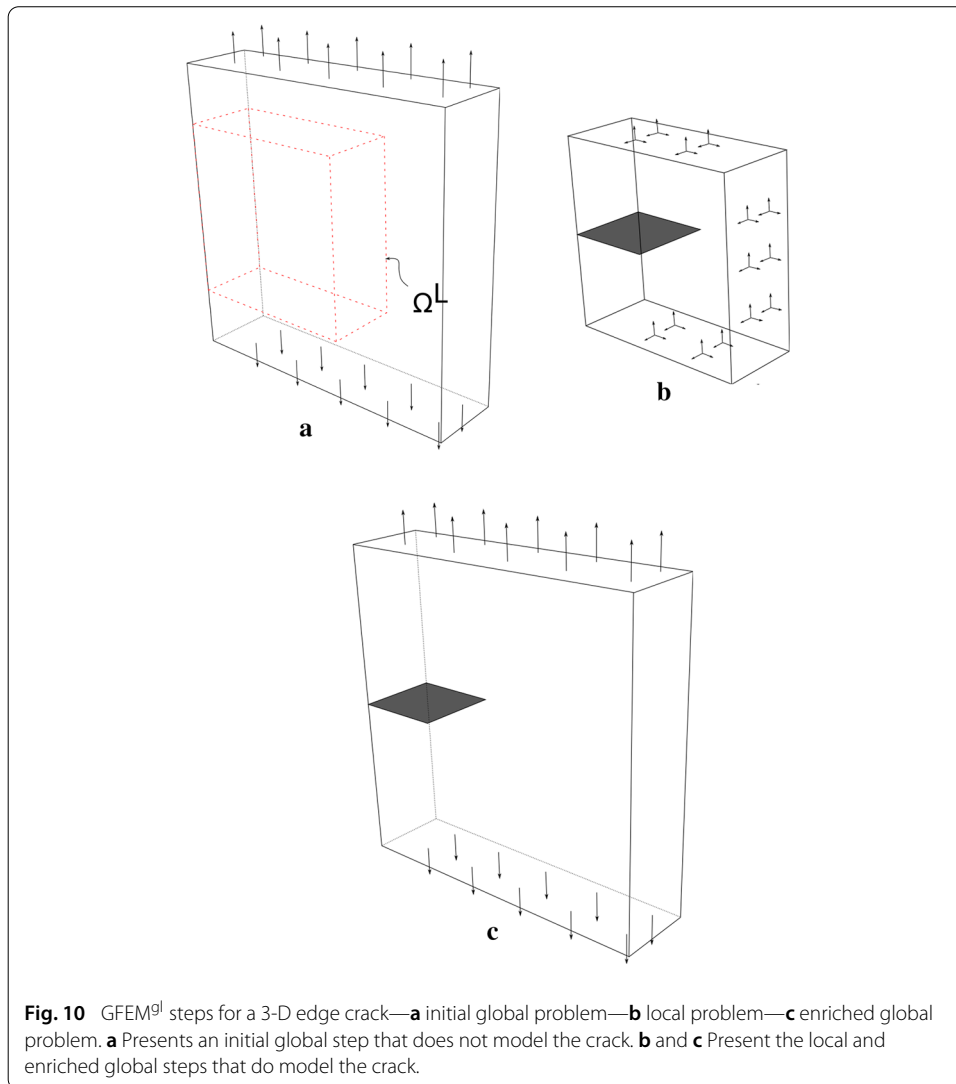
the enrichment functions adopted in the global problem. The accuracy of the Abaqus-ISET solution is also assessed using a reference solution. The three steps of the GFEM^{gl} analysis of this problem are illustrated in Fig. 10.

This problem has also been solved in Section 7.1 of [12] using a non-intrusive GFEM^{gl} algorithm based on static condensation of the global–local enrichments as briefly discussed in “Discrete system of equations of the enriched global problem and the hierarchic non-intrusive algorithm” section. The problem and discretization parameters adopted here are the same as in [12].

Three element types provided by Abaqus are adopted for the global problem. They are linear and quadratic tetrahedron elements (TET4 and TET10) and quadratic serendipity hexahedral element (HEX20). The global meshes with these elements are shown in Fig. 11. Uniform and fairly coarse meshes are adopted for the global problem. The figure also shows the mesh adopted for the local problem. The same local discretization is used regardless of the element type adopted in the global problem. The mesh is locally refined to capture the singularity at the crack front. The characteristic length of the elements near the crack front is taken as 5% of the crack length. Cubic GFEM shape functions are adopted in the local problem. Singular crack tip and Heaviside enrichment are also used in the local problem. These discretization parameters for the local problem are the same as in Section 7.1 of [12]. The local solution is the only enrichment adopted in the global problem. The global nodes enriched with the local solution are shown in red in Fig. 11. There are three G-L enrichments in each one of those nodes while no enrichment is used at other nodes of the global meshes. Spring boundary conditions are adopted at the local boundary $\partial\Omega^L \setminus (\partial\Omega^L \cap \partial\Omega)$. Details on this choice of boundary condition can be found in [24].

Verification of hierarchical non-intrusive algorithm with 3-D elements

The ISET and Abaqus formulations for TET4, TET10, and HEX20 elements are identical. Therefore the solutions computed using the hierarchical non-intrusive algorithm implemented with Abaqus-ISET and with ISET only should be nearly identical. The only differ-



ence between the two solution strategies is that in one case the global matrix K^0 and the global vector f^0 are provided by Abaqus while in the other case they are provided by ISET. Thus, this section provides a verification of the HNA in the case of 3-D fracture problems. This is analogous to the results presented in “Verification of HNA with TRI3 elements” section but here the GFEM^{gl} is adopted instead of the GFEM used in that section.

Two quantities are used to verify the HNA Abaqus-ISET solution: The strain energy of the solution and K_I , the Mode I Stress Intensity Factor (SIF) extracted at the center of the crack front. Table 1 presents the results. The relative difference between the values provided by the Abaqus-ISET and by ISET is also listed in the table. The agreement between the HNA Abaqus-ISET and the ISET only solutions is clear. The relative difference between the strain energy is at most of $\mathcal{O}(10^{-6})$. This is likely due to minor differences between the implementation of the 3-D elements in Abaqus and ISET.

Accuracy of the HNA Abaqus-ISET solution

The accuracy of the Abaqus-ISET solution against a reference solution is assessed in this section. A comparison with solutions computed with the non-intrusive GFEM^{gl} algorithm

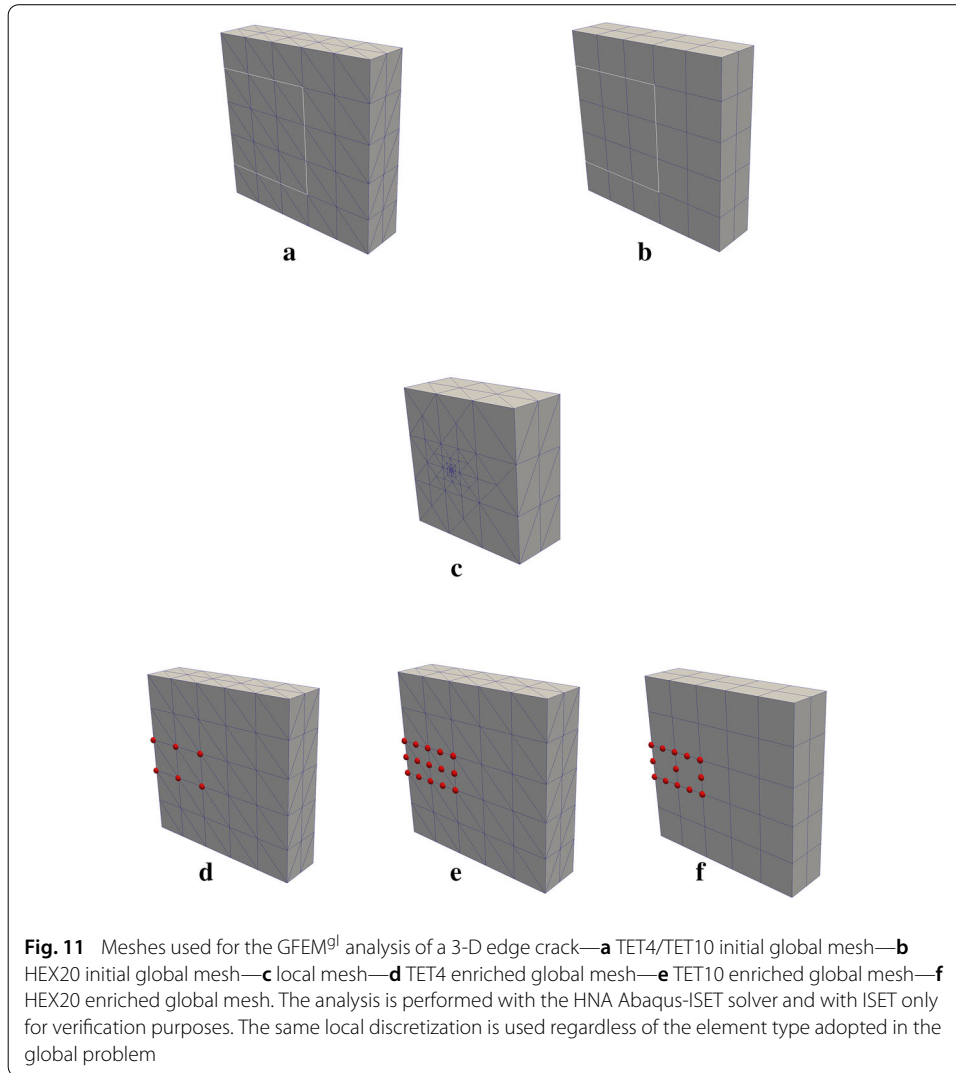


Table 1 Strain energy (U) and mode I stress intensity factor (K_I) computed with HNA Abaqus-ISET and ISET only

Solver	Element type	K_I	K_I rel. diff.	$U(E - 5)$	U rel. diff.
ISET only	TET4	2.229905	–	5.108473	–
HNA Abaqus-ISET	TET4	2.229903	$8.97E - 7$	5.108471	$3.32E - 7$
ISET only	TET10	2.87537	–	5.3423454250874	–
HNA Abaqus-ISET	TET10	2.87537	0	5.3423454250871	$4.97E - 14$
ISET only	HEX20	3.05214	–	5.37984	–
HNA Abaqus-ISET	HEX20	3.05217	$7.536E - 6$	5.37986	$3.604E - 6$

The GFEM^{9l} is adopted in both cases

proposed in [12] is also presented. The parameters adopted in this section are the same as Section 7.1 of [12]. Thus, the HNA Abaqus-ISET solution should be very close to the results presented in [12].

The reference value for the strain energy and Mode I SIF are taken as $U = 5.55154E - 5$ and $K_I = 3.0796$, respectively. The relative errors are computed using

$$e^r(K_I) = \frac{K_I^{\text{ref}} - \tilde{K}_I}{K_I^{\text{ref}}} \quad \text{and} \quad e^r(U) = \frac{U^{\text{ref}} - \tilde{U}}{U^{\text{ref}}}$$

where \tilde{U} and \tilde{K}_I are approximations provided by either the HNA proposed here or the non-intrusive static condensation method of [12]. The results are presented in Table 2. The HNA and the static condensation solutions are very close as expected. This serves as another verification for the proposed non-intrusive algorithm. The errors with linear tetrahedrons are quite large. Significant improvement is observed when TET10 elements are adopted. The most accurate results are obtained with HEX20 elements, in particular for K_I which has an error of less than 1% when these elements are adopted in the global problem. It is noted that the G-L enrichments are exactly the same in all cases since the solution of the initial global problem, which does not have a crack, is the same for all three element types.

Figure 12 shows contour plots of the von Mises stress computed with Abaqus-ISET and HEX20 elements in the global problems. The plots are shown on the deformed configuration of each problem. The initial global problem shows a constant stress field since the crack is ignored when solving the problem.

Three-dimensional surface crack

The surface crack shown in Fig. 13 is solved in this section using the HNA Abaqus-ISET and with ISET only for verification purposes. The HNA Abaqus-ISET solution is also verified against the non-intrusive algorithm of Gupta et al. [12]. The accuracy of the stress intensity factor along the crack front is assessed using the reference values from Walters et al. [42].

The dimensions shown in Fig. 13 are $2b = 2h = 2.0$, domain thickness $t = 1$, and the crack radius $r = 0.2$. Adopted material properties are Young's modulus $E = 1$ and Poisson's ratio $\nu = 0.25$. The magnitude of the tractions is taken as $\sigma = 1$. Point displacement boundary conditions are assigned to selected nodes of the FEM mesh to prevent rigid body motions. The GFEM^{gl} is used to numerically define the enrichment functions adopted at the global problem. The initial global problem and the local problem are illustrated in Fig. 14.

Like in the previous section, three element types provided by Abaqus are adopted for the global problem: TET4, TET10 and HEX20. The global meshes with these elements are shown in Fig. 15. The stiffness matrices associated with these meshes are computed by Abaqus when solving the problem with the hierarchical non-intrusive algorithm. The figure also shows the mesh adopted for the local problem. The characteristic length of the elements near the crack front is taken as 2% of the crack radius. The same local discretization is used regardless of the element type adopted in the global problem. The

Table 2 Strain energy (U) and mode I stress intensity Factor (K_I) computed with HNA Abaqus-ISET and the non-intrusive static condensation algorithm presented in [12]

Method	Element type	K_I	$U (E = 5)$	$e^r(K_I)(\%)$	$e^r(U)(\%)$
Reference	–	3.0796	5.55154	–	–
Static Cond.	TET4	2.244	5.114	27.143	7.8729
HNA Abaqus-ISET	TET4	2.230	5.108	27.59	7.981
Static Cond.	TET10	2.885	5.348	6.3266	3.6659
HNA Abaqus-ISET	TET10	2.875	5.342	6.632	3.77
HNA Abaqus-ISET	HEX20	3.052	5.380	0.891	3.093

The element type used in the global problems of the GFEM^{gl} is also listed

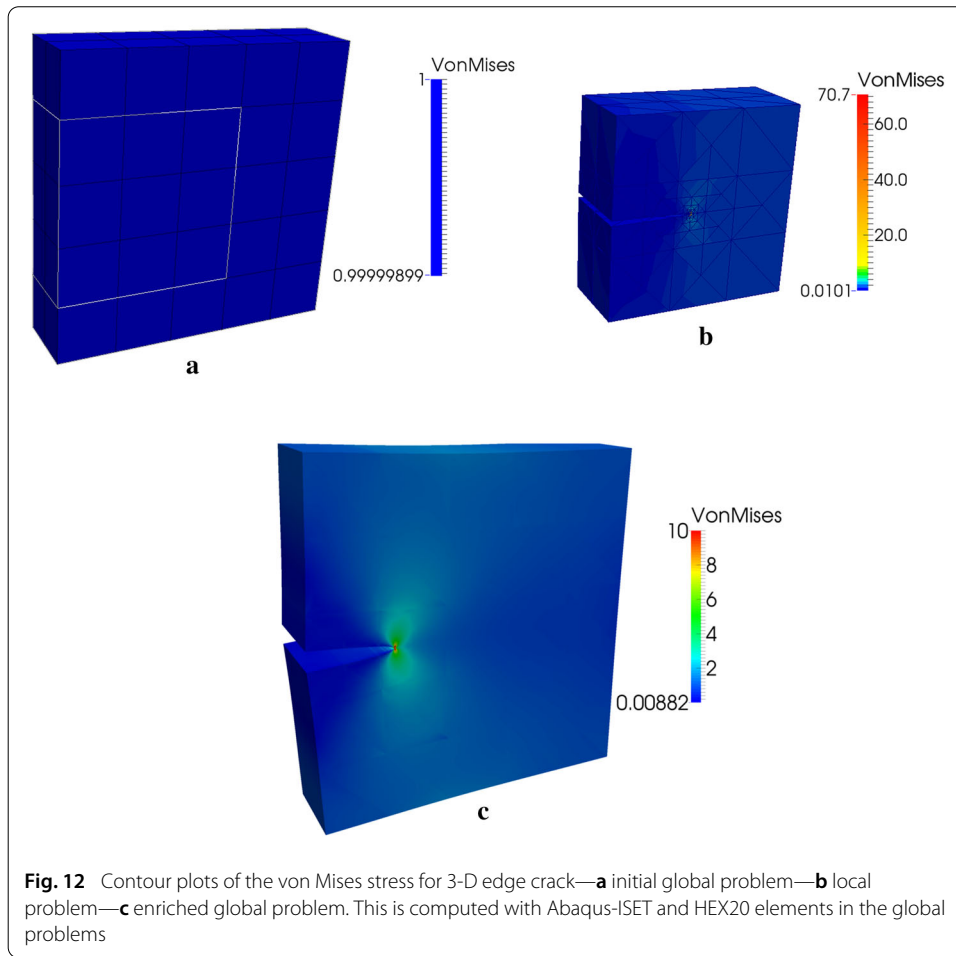


Fig. 12 Contour plots of the von Mises stress for 3-D edge crack—**a** initial global problem—**b** local problem—**c** enriched global problem. This is computed with Abaqus-ISET and HEX20 elements in the global problems

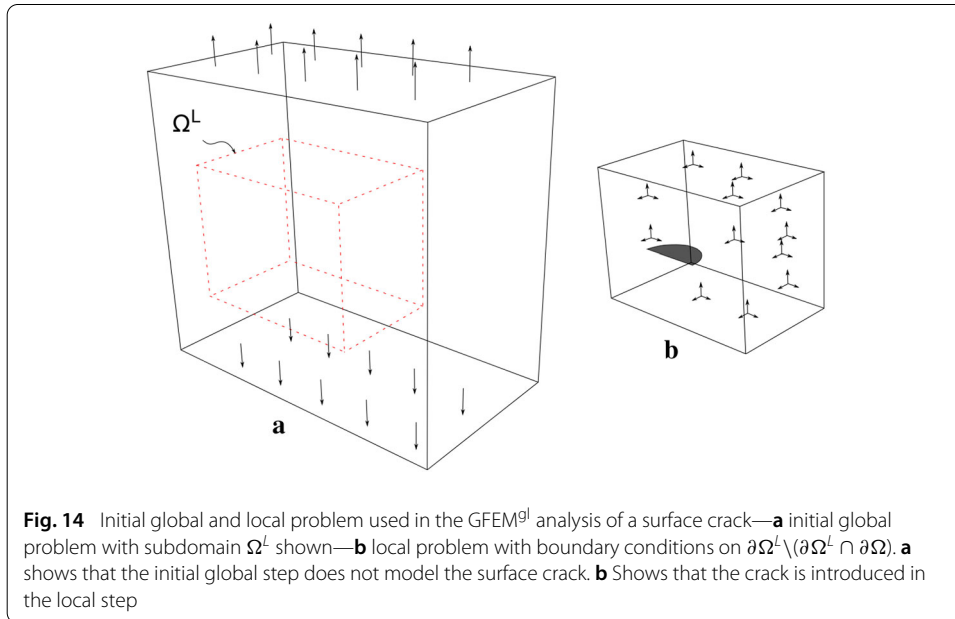
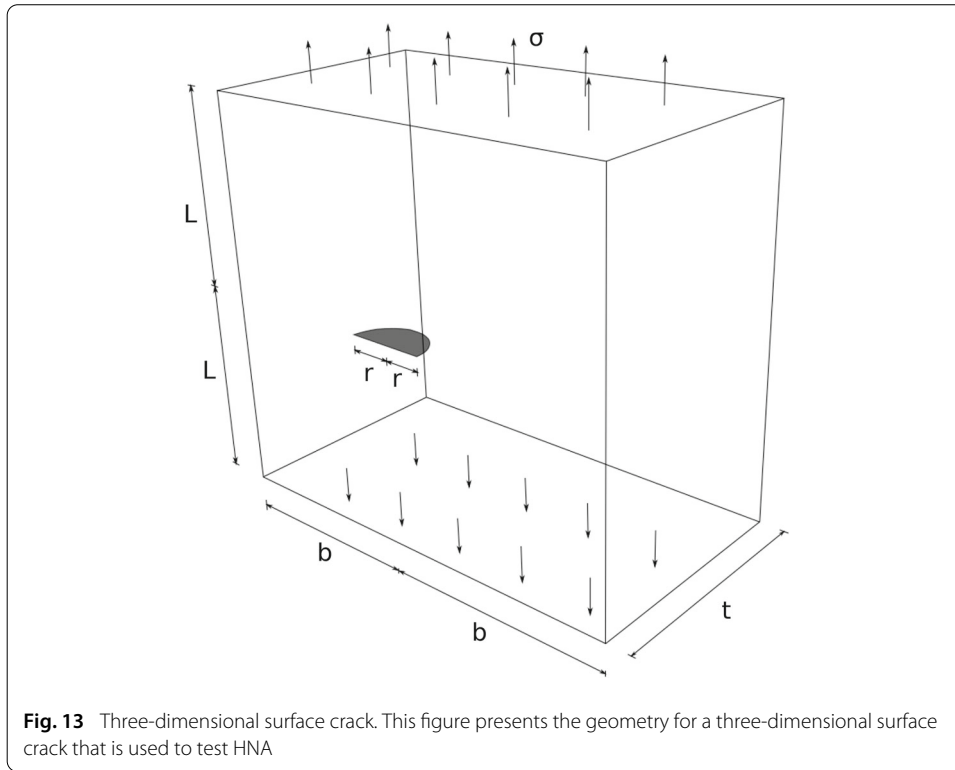
same discretization parameters as in previous section are adopted for the local problem: Cubic GFEM shape functions, singular crack tip and Heaviside enrichment, and spring boundary conditions at the local boundary $\partial\Omega^L \setminus (\partial\Omega^L \cap \partial\Omega)$. The global nodes enriched with the local solution are shown in red in Fig. 15.

Verification of hierarchical non-intrusive algorithm with 3-D elements

As discussed earlier, the HNA Abaqus-ISET solution and the ISET only solutions should be nearly identical if TET4, TET10, or HEX20 elements are used in the global problem. This is verified for the case of the surface crack problem. Two quantities are used to verify the HNA Abaqus-ISET solution: The strain energy U of the solution, and K_I , the mode I stress intensity factor (SIF) extracted along the curved crack front. The relative difference between Abaqus-ISET and ISET only K_I and U are computed using

$$e^r(K_I) = \frac{\sqrt{\sum_{j=1}^{N_{\text{ext}}} (K_{I,j}^{\text{ref}} - \tilde{K}_{I,j})^2}}{\sqrt{\sum_{j=1}^{N_{\text{ext}}} (K_{I,j}^{\text{ref}})^2}} \quad (22)$$

$$e^r(U) = \frac{U^{\text{ref}} - \tilde{U}}{U^{\text{ref}}} \quad (23)$$



where N_{ext} is the number of extraction points along the crack front, $K_{I,j}^{\text{ref}}$ and $\tilde{K}_{I,j}$ are the reference and computed stress intensity factor values at crack front point j , respectively. In this section, the reference values are taken as those computed with ISET only. Table 3 lists the relative difference between strain energy ($e^r(U)$) and stress intensity factor ($e^r(K_I)$) computed with HNA Abaqus-ISET and ISET only. Like in the previous problem, the agreement between the HNA Abaqus-ISET and the ISET only quantities is excellent. The relative difference between the strain energy is at most of $\mathcal{O}(10^{-7})$.

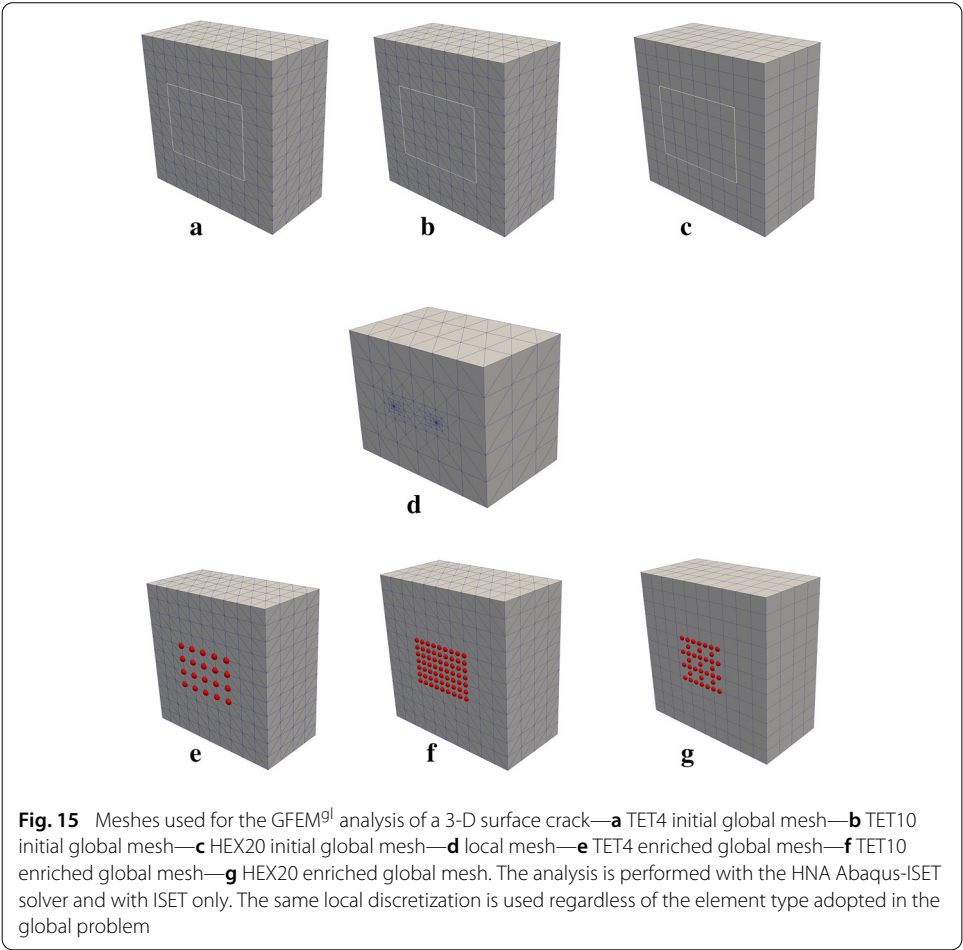


Fig. 15 Meshes used for the GFEM^{9l} analysis of a 3-D surface crack—**a** TET4 initial global mesh—**b** TET10 initial global mesh—**c** HEX20 initial global mesh—**d** local mesh—**e** TET4 enriched global mesh—**f** TET10 enriched global mesh—**g** HEX20 enriched global mesh. The analysis is performed with the HNA Abaqus-ISET solver and with ISET only. The same local discretization is used regardless of the element type adopted in the global problem

Table 3 Relative difference between strain energy (U) and stress intensity factor (K_I) computed with HNA Abaqus-ISET and ISET only

Element type	$e'(K_I)$	$e'(U)$
TET4	0	$4.71E-14$
TET10	$1.37E-4$	$1.17E-7$
HEX20	$9.34E-5$	$1.93E-9$

The GFEM^{9l} is adopted in both cases

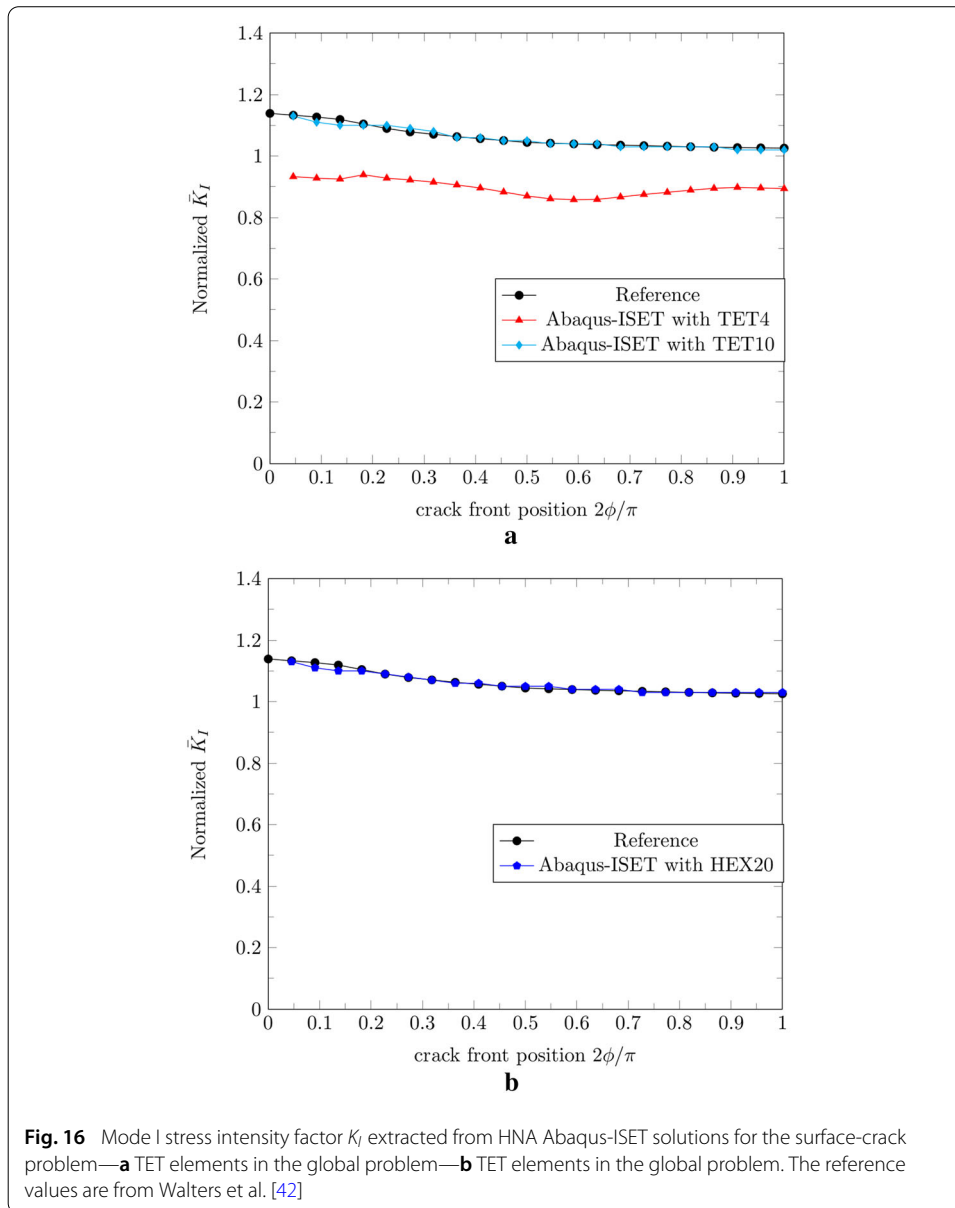
Table 4 Relative error of stress intensity factor K_I extracted from the HNA Abaqus-ISET solution of the surface crack problem

Element type	$e'(K_I)$ (%)
TET4	15.5
TET10	0.445
HEX20	0.505

The element type used in the initial and enriched global problems is indicated

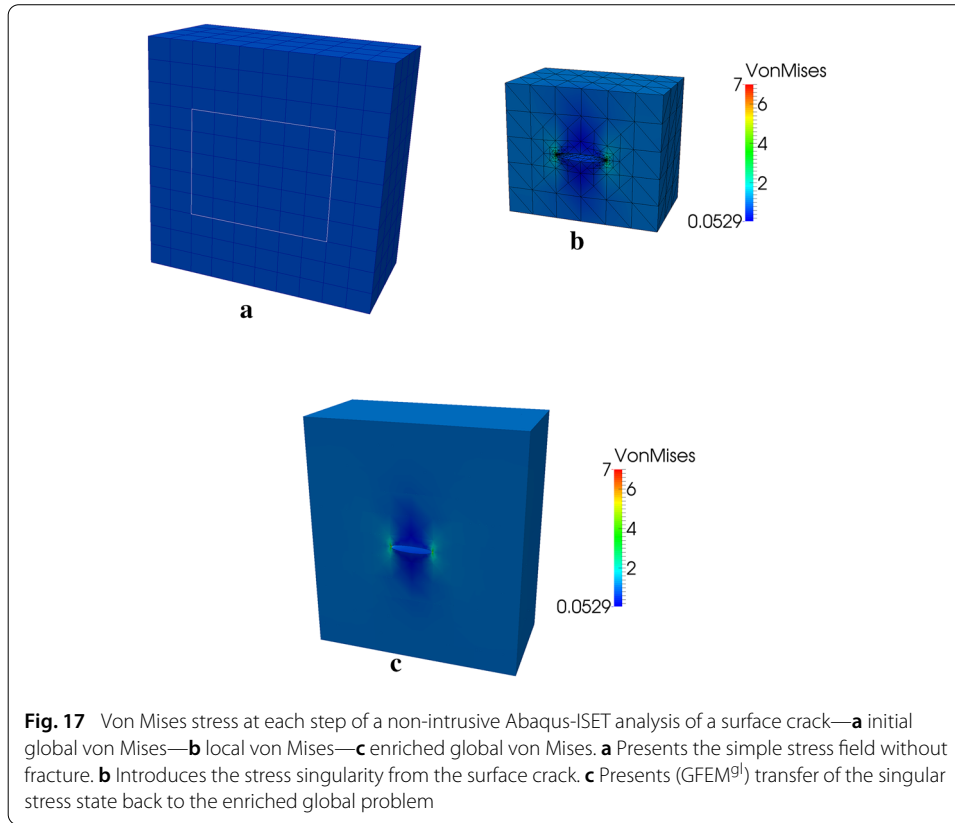
Accuracy of the HNA Abaqus-ISET solution

The accuracy of the stress intensity factor K_I extracted from the HNA Abaqus-ISET solution is assessed in this section. The cut-off function method (CFM) [43] is used to extract K_I at several points along the circular crack front. The reference values for K_I are from Walters et al. [42]. The relative error of the HNA Abaqus-ISET stress intensity factor



K_I is computed using (22). The relative error for each element type used in the initial and enriched global problems are listed in Table 4. Quadratic elements provide significantly more accurate K_I than linear tetrahedron elements. The relative error for both quadratic elements listed in the table is close to 0.5% even though a very coarse mesh is used in the global problem, attesting the good performance of the GFEM^{gl}. The results for TET4 and TET10 elements are very similar to corresponding ones listed in Table 3 of [12].

Figure 16 shows the normalized Mode I SIF extracted from the HNA Abaqus-ISET solutions computed with different element types in the global problem. The position along the crack front is defined by the polar angle ϕ where $\phi = 0$ and $\phi = \pi/2$ correspond to a crack front point on the domain boundary and to the center of the crack front, respectively. The SIF at each extraction point is normalized using [42]



$$\tilde{K}_I = \frac{\tilde{K}_I}{\sigma \sqrt{\frac{\pi r}{Q}}}$$

where $Q = 2.464$ for a circular crack, σ is the tensile stress applied to the specimen, and r is the radius of the crack. The SIF extracted from solutions computed with quadratic elements agrees very well with the reference values.

Figure 17 shows contour plots of the von Mises stress computed with Abaqus-ISET and HEX20 elements in the global problems. The plots are shown on the deformed configuration of each problem. The initial global problem shows a constant stress field since the crack is ignored when solving the problem.

Conclusions

A non-intrusive algorithm for coupling standard and Generalized FEM platforms was presented. The proposed algorithm can be used with any method that hierarchically enriches a finite element space. Examples are the Generalized FEM [5, 17–21] and the GFEM^{gl} [10, 11]. The advantage of using the latter is that it can deliver accurate solutions using coarse meshes. The proposed hierarchical non-intrusive algorithm (HNA) overcomes the memory requirement issues of the static condensation algorithm presented in [12, 13] while not compromising its accuracy and flexibility. The implementation of the HNA is also simpler than the algorithm presented in [12, 13].

The HNA combines the vast library of classical elements available in commercial FEM platforms with the ability of the GFEM^{gl} to analyze localized phenomena, like cracks and

spot welds, on coarse meshes. Examples demonstrating the application of the hierarchical non-intrusive algorithm to 2- and 3-D fracture mechanics problems were presented. They show that the HNA provides the same solution as the GFEM or the GFEM^{gl} implemented in a single software. Furthermore, it does not require iterations between the standard and Generalized FEM platforms. The examples also demonstrate the benefits of combining finite elements available only in a commercial platform with the GFEM.

While the commercial FEM software adopted for the examples presented in this paper is Abaqus, the proposed HNA can be used with other commercial software. The only requirement is that the FEM platform outputs the global matrix and load vector for a problem. Furthermore, while this capability of FEM platforms was explored here in the context of the GFEM and the GFEM^{gl}, it can also be applied to the implementation of other non-intrusive algorithms like the Iterative Global–Local FEM [34,35] and the method proposed by Schweitzer and Ziegenhagel [14].

Authors' contributions

CAD proposed the HNA algorithm. TBF implemented the method and conducted the numerical experiments. Both authors read and approved the final manuscript.

Acknowledgements

The authors gratefully acknowledge the contributions of collaborators at U.S. Air Force Research Laboratory, Wright-Patterson Air Force Base.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

Not applicable.

Consent for publication

Not applicable.

Ethics approval and consent to participate

Not applicable.

Funding

This work was funded under contract number AF Sub OSU 60038238 provided by the Collaborative Center in Structural Sciences (C²S²) at the Ohio State University, supported by the U.S. Air Force Research Laboratory.

Appendix A: Abaqus commands to output the global matrix and load vector

Abaqus can generate and output a global stiffness matrix using the following keywords in *.inp files

```
*MATRIX GENERATE, STIFFNESS
```

and

```
*MATRIX OUTPUT, STIFFNESS
```

These keywords must be located within a loading “*STEP” that is not “*STATIC”. The matrix is stored in a file that reflects the name of the job.

While multiple keywords can be used in the same “*STEP”, it is generally advisable to use different steps to output different things. For example, a second step uses

```
*MATRIX GENERATE, LOAD
```

and

```
*MATRIX OUTPUT, LOAD
```

to output the global load vector. The rest of an .inp file can be understood by creating Abaqus models using their graphical user interface. When a job is submitted for analysis,

Abaqus will create a .inp file that includes all needed keywords to run that specific example.

In a Linux system, an .inp file can be run using the syntax

```
./(executable name) job=(job name) inp=(input file name)
```

Further details can be found in the Abaqus keyword user manual [44].

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 8 December 2017 Accepted: 23 January 2018

Published online: 12 February 2018

References

1. Dassault Systemes Simulia Corporation. Abaqus/CAE user's manual. Providence: Dassault Systemes Simulia Corporation; 2016.
2. MSC Software Corporation. MSC nastran 2016 reference manual. Newport Beach: MSC Software Corporation; 2016.
3. ANSYS, Inc. ANSYS user's manual. Canonsburg: ANSYS, Inc.; 2010.
4. Altair Engineering Inc. Altair RADIOSS reference guide. Troy: Altair Engineering Inc.; 2014.
5. Oden JT, Duarte CA. Clouds, cracks and FEMs. In: Reddy BD, editor. Recent developments in computational and applied mechanics. Barcelona: International Center for Numerical Methods in Engineering, CIMNE; 1997. p. 302–21. <http://citeseer.ist.psu.edu/17994.html>.
6. Duarte CA, Babuška I, Oden JT. Generalized finite element methods for three dimensional structural mechanics problems. *Comput Struct*. 2000;77:215–32. [https://doi.org/10.1016/S0045-7949\(99\)00211-4](https://doi.org/10.1016/S0045-7949(99)00211-4).
7. Duarte CA, Hamzeh ON, Liszka TJ, Tworzydło WW. A generalized finite element method for the simulation of three-dimensional dynamic crack propagation. *Comput Methods Appl Mech Eng*. 2001;190(15–17):2227–62. [https://doi.org/10.1016/S0045-7825\(00\)00233-4](https://doi.org/10.1016/S0045-7825(00)00233-4).
8. Belytschko T, Black T. Elastic crack growth in finite elements with minimal remeshing. *Int J Numer Methods Eng*. 1999;45:601–20.
9. Moës N, Dolbow J, Belytschko T. A finite element method for crack growth without remeshing. *Int J Numer Methods Eng*. 1999;46:131–50.
10. Duarte CA, Kim D-J, Babuška I. A global-local approach for the construction of enrichment functions for the generalized FEM and its application to three-dimensional cracks. In: Leitão VMA, Alves CJS, Duarte CA, editors. *Advances in Meshfree techniques. Computational methods in applied sciences*, vol. 5. Dordrecht: Springer; 2007. p. 1–26. https://doi.org/10.1007/978-1-4020-6095-3_1.
11. Duarte CA, Kim D-J. Analysis and applications of a generalized finite element method with global-local enrichment functions. *Comput Methods Appl Mech Eng*. 2008;197(6–8):487–504. <https://doi.org/10.1016/j.cma.2007.08.017>.
12. Gupta P, Pereira JP, Kim D-J, Duarte CA, Eason T. Analysis of three-dimensional fracture mechanics problems: a non-intrusive approach using a generalized finite element method. *Eng Fract Mech*. 2012;90:41–64. <https://doi.org/10.1016/j.engfracmech.2012.04.014>.
13. Plews J, Duarte CA, Eason T. An improved non-intrusive global-local approach for sharp thermal gradients in a standard FEA platform. *Int J Numer Methods Eng*. 2012;91(4):426–49. <https://doi.org/10.1002/nme.4279>.
14. Schweitzer MA, Ziegenhagen A. Embedding enriched partition of unity approximations in finite element simulations. In: Griebel M, Schweitzer MA, editors. *Meshfree methods for partial differential equations VIII. Lecture notes in computational science and engineering*, vol. 115. Berlin: Springer; 2017. p. 199–208. https://doi.org/10.1007/978-3-319-51954-8_10.
15. Noor AK. Global-local methodologies and their applications to nonlinear analysis. *Finite Elem Anal Design*. 1986;2:333–46.
16. Diamantoudis AT, Labeas GN. Stress intensity factors of semi-elliptical surface cracks in pressure vessels by global-local finite element methodology. *Eng Fract Mech*. 2005;72:1299–312.
17. Babuška I, Melenk JM. The partition of unity method. *Int J Numer Methods Eng*. 1997;40:727–58.
18. Melenk JM, Babuška I. The partition of unity finite element method: basic theory and applications. *Comput Methods Appl Mech Eng*. 1996;139:289–314.
19. Oden JT, Duarte CA, Zienkiewicz OC. A new cloud-based hp finite element method. *Comput Methods Appl Mech Eng*. 1998;153:117–26. [https://doi.org/10.1016/S0045-7825\(97\)00039-X](https://doi.org/10.1016/S0045-7825(97)00039-X).
20. Duarte CAM, Oden JT. An hp adaptive method using clouds. *Comput Methods Appl Mech Eng*. 1996;139:237–62. [https://doi.org/10.1016/S0045-7825\(96\)01085-7](https://doi.org/10.1016/S0045-7825(96)01085-7).
21. Duarte CAM, Oden JT. hp clouds - An hp meshless method. *Numer Methods Partial Differ Equ*. 1996;12:673–705. [https://doi.org/10.1002/\(SICI\)1098-2426\(199611\)12:6<673::AID-NUM3>3.0.CO;2-P](https://doi.org/10.1002/(SICI)1098-2426(199611)12:6<673::AID-NUM3>3.0.CO;2-P).
22. O'Hara P, Duarte CA, Eason T. Transient analysis of sharp thermal gradients using coarse finite element meshes. *Comput Methods Appl Mech Eng*. 2011;200(5–8):812–29. <https://doi.org/10.1016/j.cma.2010.005>.
23. Kim D-J, Duarte CA, Pereira JP. Analysis of interacting cracks using the generalized finite element method with global-local enrichment functions. *J Appl Mech*. 2008;75(5):1–12. <https://doi.org/10.1115/1.2936240>.
24. Kim D-J, Pereira JP, Duarte CA. Analysis of three-dimensional fracture mechanics problems: a two-scale approach using coarse generalized FEM meshes. *Int J Numer Methods Eng*. 2010;81(3):335–65. <https://doi.org/10.1002/nme.2690>.
25. Kim J, Duarte CA. A new generalized finite element method for two-scale simulations of propagating cohesive fractures in 3-D. *Int J Numer Methods Eng*. 2015;104(13):1139–72. <https://doi.org/10.1002/nme.4954>.

26. Kim D-J, Duarte CA, Proenca SP. A generalized finite element method with global-local enrichment functions for confined plasticity problems. *Comput Mech.* 2012;50(5):563–78. <https://doi.org/10.1007/s00466-012-0689-7>.
27. Plews JA, Duarte CA. Bridging multiple structural scales with a generalized finite element method. *Int J Numer Methods Eng.* 2015;102(3–4):180–201. <https://doi.org/10.1002/nme.4703>.
28. Plews JA, Duarte CA. A two-scale generalized finite element approach for modeling localized thermoplasticity. *Int J Numer Methods Eng.* 2016;108(10):1123–58. <https://doi.org/10.1002/nme.5241>.
29. Pereira JP, Duarte CA, Guoy D, Jiao X. *Hp*-Generalized FEM and crack surface representation for non-planar 3-D cracks. *Int J Numer Methods Eng.* 2009;77(5):601–33. <https://doi.org/10.1002/nme.2419>.
30. Gendre L, Allix O, Gosselet P. A two-scale approximation of the Schur complement and its use for non-intrusive coupling. *Int J Numer Methods Eng.* 2011;87(9):889–905. <https://doi.org/10.1002/nme.3142>.
31. Felippa CA. Introduction to finite element methods. Course Notes. Department of Aerospace Engineering Sciences, University of Colorado at Boulder; 2004. <http://www.colorado.edu/engineering/Aerospace/CAS/courses.d/IFEM.d>.
32. Gupta V, Kim D-J, Duarte CA. Analysis and improvements of global-local enrichments for the generalized finite element method. *Comput Methods Appl Mech Eng.* 2012;245–246:47–62. <https://doi.org/10.1016/j.cma.2012.06.021>.
33. Li H, Duarte CA. A two-scale generalized finite element method for parallel simulations of spot welds in large structures. San Diego, CA: ASCE EMI 2017 Conference; 2017.
34. Whitcomb JD. Iterative global/local finite element analysis. *Comput Struct.* 1991;40(4):1027–31. [https://doi.org/10.1016/0045-7949\(91\)90334-I](https://doi.org/10.1016/0045-7949(91)90334-I).
35. Gendre L, Allix O, Gosselet P, Comte F. Non-intrusive and exact global/local techniques for structural problems with local plasticity. *Comput Mech.* 2009;44(2):233–45. <https://doi.org/10.1007/s00466-009-0372-9>.
36. Duval M, Passieux J-C, Salaün M, Guinard S. Non-intrusive coupling: recent advances and scalable nonlinear domain decomposition. *Arch Comput Methods Eng.* 2016;23(1):17–38. <https://doi.org/10.1007/s11831-014-9132-x>.
37. Schenk O, Gärtner K. Two-level dynamic scheduling in pardiso: improved scalability on shared memory multiprocessing systems. *Parallel Comput.* 2002;28(2):187–97. [https://doi.org/10.1016/S0167-8191\(01\)00135-1](https://doi.org/10.1016/S0167-8191(01)00135-1).
38. Schenk O, Gärtner K. Solving unsymmetric sparse systems of linear equations with PARDISO. *Future Gener Comput Syst.* 2004;20(3):475–87. <https://doi.org/10.1016/j.future.2003.07.011>.
39. Kim D-J, Hong S-G, Duarte CA. Generalized finite element analysis using the preconditioned conjugate gradient method. *Appl Math Model.* 2015;39(19):5837–48. <https://doi.org/10.1016/j.apm.2015.04.002>.
40. Fillmore T. Multi-scale problems: an improved non-intrusive algorithm that enhances fea platforms with the generalized finite element method; an improved preconditioned conjugate gradient solver for hierarchical generalized finite element systems of equations. Master's thesis, Champaign: University of Illinois at Urbana; 2016.
41. Kergrene K, Babuska I, Banerjee U. Stable generalized finite element method and associated iterative schemes; application to interface problems. *Comput Methods Appl Mech Eng.* 1999;305:1–36. <https://doi.org/10.1016/j.cma.2016.02.030>.
42. Walters MC, Paulino GH, Dodds RH Jr. Stress-intensity factors for surface cracks in functionally graded materials under mode-I thermomechanical loading. *Int J Solids Struct.* 2004;41:1081–118.
43. Pereira JP, Duarte CA. Extraction of stress intensity factors from generalized finite element solutions. *Eng Anal Bound Elem.* 2005;29:397–413. <https://doi.org/10.1016/j.enganabound.2004.09.007>.
44. Dassault Systèmes Simulia Corporation. Abaqus analysis user's manual, vol. I–V. Providence: Dassault Systèmes Simulia Corporation; 2010.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)