Adaptive smoothed stable extended finite element method for weak discontinuities for finite elasticity

Chintan Jansari[a], Sundararajan Natarajan[a,1,*], Lars Beex[b], K Kannan[a]

[a] *Department of Mechanical Engineering, Indian Institute of Technology Madras, Chennai-600036, India.*

[b] *Faculté des Sciences, de la Technologie et de la Communication, University of Luxembourg, Luxembourg.*

Abstract

In this paper, we propose a smoothed stable extended finite element method (S^{2}XFEM) by combining the strain smoothing with the stable extended finite element method (SXFEM) to efficiently treat inclusions and/or voids in hyperelastic matrix materials. The interface geometries are implicitly represented through level sets and a geometry based error indicator is used to resolve the geometry. For the unknown fields, the mesh is refined based on a recovery based error indicator combined with a quadtree decomposition guarantee the method's accuracy with respect to the computational costs. Elements with hanging nodes (due to the quadtree meshes) are treated as polygonal elements with mean value coordinates as the basis functions. The accuracy and the convergence properties are compared to similar approaches for several numerical examples. The examples indicate that S^{2}XFEM is computationally the most efficient without compromising the accuracy.

Keywords: Adaptive meshing, Hyperelastic material, Mean-value shape functions, Numerical integration, Quadtree mesh, Smoothed finite element method, Stable extended finite element method.

1. Introduction

The solutions of many real world problems in engineering and other physical fields vary spatially. However, the solutions to many problems are non-smooth. Problems governed by Helmholtz and Laplace equations are globally non-smooth for instance, whereas problems involving material interfaces, cracks, shear bands, to name a few, are locally non-smooth. The conventional FEM has difficulties to treat non-smooth solutions due to (a) its polynomial interpolation functions which cannot capture steep gradients in the solution, (b) the requirement of spatially conforming interpolations, and

[*] Corresponding author

[1] Department of Mechanical Engineering, Indian Institute of Technology-Madras, Chennai - 600036, India. E-mail:sundararajan.natarajan@gmail.com; snatarajan@iitm.ac.in

(c) the requirement of substantially fine interpolations if the solution changes spatially over time.

To circumvent these challenges, meshless methods [1–4] have emerged as an alternative to the FEM. Partition of unity methods (PUMs) [5–9] on the other hand are enhancements of the conventional FEM, which incorporate non-smooth interpolations of the spatial solution. PUMs have consequently attracted much attention, which may for instance be recognised by the fact that the eXtended FEM (XFEM) has found its place in the commercial FE software Abaqus.

The salient feature of the PUMs is that the geometric morphology and the underlying FE discretization can be independent of each other. Consequently, the computational burden is substantially reduced. The basic idea of the PUMs is to extend the classical approximate solution basis by a set of additional functions that carry information about the character of the solution. The price to pay for this flexibility is the increased complexity to numerically integrate the additional functions and an enlarged system of equations to solve.Recently, PUMs have been combined with local mesh refinement to treat problems with weak and strong discontinuities both in two and three dimensions [10–13].

On another front, based on the seminal work of Chen *et al.* [14], Liu *et al.* [15] formulated a series of methods, called the smoothed finite element method (SFEM). The SFEM is aimed at improving the performance of conventional finite elements, viz., triangles, quadrilaterals, tetrahedra, hexahedra and in general, arbitrary polytopes. Since its inception, the different variants of the SFEM (such as the edge based SFEM, node based SFEM, cell based SFEM), have been applied to wide variety of problems, such as to fracture mechanics [16–20], incompressible elasticity [21, 22], visco-elastoplastic analysis [23–25] and impact problems [26], amongst others. The cell based and the edge based SFEM were combined with the XFEM in [17, 27–29]. This greatly simplified the numerical integration of the terms in the bilinear form, especially, the enriched part. It alleviated the need to sub-triangulate the elements intersected by the discontinuity for the purpose of numerical integration. Amrita *et al.* [30] and Surendran *et al.* [31] improved the performance of cell-based SFEM over arbitrary polytopes by modifying the smoothing function. For a comprehensive overview, interested readers are referred to the recent review of Zeng *et al.* [32]. The above work has been restricted to small deformation problems. To the best of authors' knowledge, the SFEM is scarcely used to treat nonlinear problems. We know about the work of Duong in which the face based SFEM is employed to model nonlinear soft tissue growth [33]. Lee *et al.* [34] employed the node based SFEM and edge based SFEM for compressible and nearly incompressible hyperelastic materials on the other hand. These studies were restricted to single phase material however.

In this paper, we propose the smoothed stable extended finite element method (S^{2}XFEM) by combining the strain smoothing with the stable extended finite element method (SXFEM) to solve interface problems in hyperelastic materials. It is noted that the stable XFEM employed here is similar to the ridge enrichment function [35]. The salient features of this work are:

- The strain smoothing does not require an explicit form of the interpolation functions to compute the contributions of the quadrature points to the stiffness matrices and internal force columns. It therefore requires less computational efforts. Moreover, strain smoothing requires simple integration schemes.
- The adaptive refinement of the interpolation discretisation reduces the computational efforts by local remeshing. Interpolation elements with hanging nodes are treated as polygonal elements, thereby ensuring compatibility between the elements.

The accuracy and the convergence properties are demonstrated with two numerical examples in two dimensions. It is inferred that the S^{2}XFEM employs fewer integration points to compute the terms in the bilinear form and yields comparable results when compared to the SXFEM.

The paper is organized as follows: Section 2 presents the governing equations for a hyperelastic body, the weak formulation of the problem and a description of the stable extended finite method. The smoothed variant of the stable extended finite element method is presented in Section 3. A simple a-posteriori error indicator and an adaptive meshing procedure is discussed in Section 4. The accuracy, robustness and the convergence behaviour of the proposed approach is detailed in Section 5, followed by the main conclusions in the last section.

2. Governing equations and weak form

Consider a hyperelastic body with an internal discontinuity Γ_0^d as shown in Figure 1. $\Omega_0 \subset \mathbb{R}^2$ bounded by $\Gamma \in \mathbb{R}$, with $\Gamma_0^u \cap \Gamma_0^t = \emptyset$. The boundary is uniquely decomposed into $\Gamma = \Gamma_0^u \cup \Gamma_t$, where Dirichlet and Neumann boundary conditions are applied on Γ_u and Γ_t, respectively. Let $\boldsymbol{\chi}$ be the motion function that maps the reference coordinate $\mathbf{X} \in \Omega_0$ onto the current coordinate $\mathbf{x} \in \Omega$, given by:

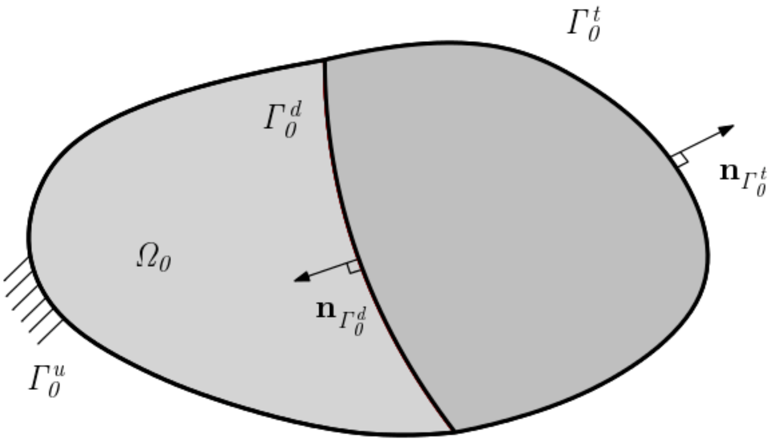


Figure 1: Domain description of the problem

$$\mathbf{x} = \boldsymbol{\chi}(\mathbf{X}). \tag{1}$$

The deformation gradient tensor can then be written as:

$$\mathbf{F} = \frac{\partial \boldsymbol{\chi}}{\partial \mathbf{X}}, \tag{2}$$

and the Lagrangian strain tensor as:

$$\mathbf{E} = \frac{1}{2}\big(\mathbf{F}^{\mathrm{T}}\mathbf{F} - \mathbf{I}\big). \tag{3}$$

The governing equations and the boundary conditions in the reference configuration (in the absence of inertia and body forces) are given by:

$$\begin{aligned} \nabla \cdot \mathbf{P} &= \mathbf{0} \quad \text{in} \quad \varOmega_0 \\ \mathbf{u} &= \overline{\mathbf{u}} \;\; \text{on} \;\; \varGamma_0^u \\ \mathbf{P} \cdot \mathbf{n}_{\varGamma_0^t} &= \overline{\mathbf{t}}_0 \;\; \text{on} \;\; \varGamma_0^t \\ [\![\mathbf{P} \cdot \mathbf{n}_{\varGamma_0^d}]\!] &= 0 \;\; \text{on} \;\; \varGamma_0^d \end{aligned} \tag{4}$$

where $\mathbf{P}$ denotes the first Piola Kirchhoff stress, $\overline{\mathbf{t}}_0$ the externally applied traction on boundary $\varGamma_0^t$, $\overline{\mathbf{u}}$ the prescribed displacement on boundary $\varGamma_0^u$, $\mathbf{n}_{\varGamma_0^t}$ the unit normal on boundary $\varGamma_0^t$ and $\mathbf{n}_{\varGamma_0^d}$ the unit normal on boundary $\varGamma_0^d$.

In this paper, compressible hyperelastic models are formulated to treat large deformations. To achieve this, the constitutive relations are nonlinear in terms of the deformation gradient tensor. In this study, the Neo-Hookean material is employed, which has the following stored energy function:

$$W = \frac{\kappa}{2}(\det\mathbf{F} - 1)^2 + \frac{\mu}{2}(\mathrm{Tr}\,\overline{\mathbf{C}} - 3) \tag{5}$$

where

$$\overline{\mathbf{C}} = (\det\mathbf{F})^{-\frac{2}{3}}\mathbf{C} \quad \text{and} \quad \mathbf{C} = \mathbf{F}^{\mathrm{T}} \cdot \mathbf{F}.$$

Here, $\mathrm{Tr}(\cdot)$ denotes the trace of a tensor and κ and μ denote the bulk modulus and the shear modulus, respectively. Due to the presence of material and geometric nonlinearities, the resulting boundary value problem is nonlinear and consequently, needs to be solved by the nonlinear finite element method. In this study, the Total Lagrangian formulation is employed. The domain occupied by the body is partitioned into non-overlapping elements $\varOmega^h$ and the functions that span at least the linear space are employed as the trial (N_a) and the test functions (N_b) as: $\mathbf{u}^h = \sum\limits_a N_a \mathbf{u}_a$ and $\mathbf{v}^h = \sum\limits_b N_b \mathbf{v}_b$, respectively. Following the standard Galerkin procedure, the weak form is given by: Find $\mathbf{u}^h \in \mathscr{U}^h$, such that,

$$\int_{\varOmega_0} \delta \mathbf{E}^{\mathrm{T}} \mathbf{S} \mathrm{d}\varOmega_0 - \int_{\varGamma_0} \delta \mathbf{u}^{\mathrm{T}} \overline{\mathbf{t}}_0 \, \mathrm{d}\varGamma_0 = 0, \tag{6}$$

where

$$\mathscr{U}^h \subset \mathscr{U} = \big\{ \mathbf{u}(\mathbf{x}) \in \mathbb{H}^1(\varOmega), \mathbf{u} = \bar{\mathbf{u}} \text{ on } \varGamma_0^u \big\} \tag{7a}$$

$$\mathscr{V}^h \subset \mathscr{V} = \left\{ \mathbf{v}(\mathbf{x}) \in \mathbb{H}_0^1(\varOmega), \mathbf{v} = 0 \text{ on } \varGamma_0^u \right\} \tag{7b}$$

and $\mathbf{S} = \mathbf{F}^{-1}\mathbf{P}$ denotes the second Piola Kirchhoff stress, related to stored energy function W by:

$$\mathbf{S} = \frac{\partial W}{\partial \mathbf{E}} = 2\frac{\partial W}{\partial \mathbf{C}} \tag{8}$$

Equation (6) is a nonlinear equation and the unknown fields are computed by adopting the Newton-Raphson method, which leads to the following system of algebraic equations:

$$\mathbf{K}_{\mathrm{T}} \varDelta \mathbf{d} = \mathbf{f}_{ext} - \mathbf{f}_{int} \tag{9}$$

Here, $\mathbf{K}_{\mathrm{T}}$ is the tangent stiffness matrix, which is a sum of the material tangent stiffness matrix $\mathbf{K}_{\mathrm{M}}$ and the geometric tangent stiffness matrix $\mathbf{K}_{\mathrm{G}}$, given by:

$$\mathbf{K}_{\mathrm{M}} = \sum_e \mathbf{K}_{\mathrm{M}}^h = \sum_e \int_{\varOmega_0} \mathbf{B}_{\mathrm{L}}^{\mathrm{T}} \; \mathbf{D} \; \mathbf{B}_{\mathrm{L}} \, \mathrm{d}\varOmega_0 \tag{10a}$$

$$\mathbf{K}_{\mathrm{G}} = \sum_e \mathbf{K}_{\mathrm{G}}^h = \sum_e \int_{\varOmega_0} \mathbf{B}_{\mathrm{NL}}^{\mathrm{T}} \; \mathbf{S} \; \mathbf{B}_{\mathrm{NL}} \, \mathrm{d}\varOmega_0 \tag{10b}$$

where the following matrix forms are implied:

$$\mathbf{S} = \begin{bmatrix} S_{xx} & S_{xy} & 0 & 0 \\ S_{yx} & S_{yy} & 0 & 0 \\ 0 & 0 & S_{xx} & S_{xy} \\ 0 & 0 & S_{yx} & S_{yy} \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} D_{xx} & D_{xy} & 0 \\ D_{yx} & D_{yy} & 0 \\ 0 & 0 & D_{zz} \end{bmatrix}$$

with the tensor form of $\mathbf{D}$ to be computed following:

$$\mathbf{D} = \frac{\partial^2 W}{\partial \mathbf{E} \partial \mathbf{E}} = 4\frac{\partial^2 W}{\partial \mathbf{C} \partial \mathbf{C}}.$$

The linear and the nonlinear strain-displacement matrices in Equation (10) are given by:

$$\mathbf{B}_{\mathrm{L1}(I)} = \begin{bmatrix} N_{I,X} & 0 \\ 0 & N_{I,Y} \\ N_{I,Y} & N_{I,X} \end{bmatrix} \tag{11a}$$

$$\mathbf{B}_{\mathrm{L2}(I)} = \begin{bmatrix} N_{I,X} u_{X,X} & N_{I,X} u_{Y,X} \\ N_{I,Y} u_{X,Y} & N_{I,Y} u_{Y,Y} \\ N_{I,Y} u_{X,X} + N_{I,X} u_{X,Y} & N_{I,X} u_{Y,Y} + N_{I,Y} u_{Y,X} \end{bmatrix} \tag{11b}$$

$$\mathbf{B}_{\mathrm{NL}(I)} = \begin{bmatrix} N_{I,X} & 0 \\ N_{I,Y} & 0 \\ 0 & N_{I,X} \\ 0 & N_{I,X} \end{bmatrix} \tag{11c}$$

The right-hand side of Equation (9) is the residual due to lack of strict fulfillment of equilibrium condition. It is the difference between internal force $\mathbf{f}_{int}$ and the external force $\mathbf{f}_{ext}$ vector, given by:

$$\mathbf{f}_{int} = \sum_e \mathbf{f}^h_{int} = \sum_h \int_{\Omega_0} \mathbf{B}_\mathrm{L}^\mathrm{T} \, \mathbf{S}_v \, \mathrm{d}\Omega_0 \tag{12a}$$

$$\mathbf{f}_{ext} = \sum_e \mathbf{f}^h_{ext} = \sum_e \int_{\Gamma_0} \mathbf{N}^\mathrm{T} \bar{\mathbf{t}}_0 \, \mathrm{d}\Gamma_0 \tag{12b}$$

where $\mathbf{S}_v$ in column form reads:

$$\mathbf{S}_v = \left\{ \begin{array}{ccc} S_{xx} & S_{yy} & S_{xy} \end{array} \right\}^\mathrm{T}.$$

3. Smoothed stable extended finite element method

3.1. Stable extended finite element method

The trial and the test functions used to derive the weak form in the previous section are continuous and hence cannot capture the internal discontinuity, unless a conforming discretization is used. In this work, the internal discontinuity is represented independent of the underlying discretization using level sets. The local behaviour is captured within the framework of the extended finite element method [6, 8]. The basic idea is to augment the FE function space with 'additional functions' that capture the local behavior. Of the different variants of the XFEM available in the literature, we adopt the Stable eXtended Finite Element (SXFEM) [36]. This is because, the SXFEM: (a) alleviates the blending problem; (b) reduces the conditioning number of the resulting system and (c) does not require extra layer of enrichment (which results in additional dofs) like the corrected XFEM [37]. Within this framework, the augmented (also referred to as the enriched) displacement field is given by:

$$\mathbf{u}(\mathbf{X}) = \underbrace{\sum_{I=1}^{n} N_I(\mathbf{X})\mathbf{u}_I}_{\mathbf{u}^{std}} + \underbrace{\sum_{J=1}^{m} N_J(\mathbf{X}) \left[\psi(\mathbf{X}) - \mathcal{I}_h\psi(\mathbf{X})\right] \mathbf{a}_J}_{\mathbf{u}^{enr}} \tag{13}$$

where

- n, m are the set of all the nodes in the domain and the set of all enriched nodes, respectively.

- $\mathbf{u}_I$ and $\mathbf{a}_J$ are the standard and the enriched displacement unknowns.

- $N_I(\mathbf{X})$ are the standard finite element shape functions and $\psi(\mathbf{X})$ is the enrichment function.

- $\mathcal{I}_h\psi(\mathbf{X}) = \sum_I N_I(\mathbf{X})\psi(\mathbf{X}_I)$ is a piecewise linear interpolant of $\psi(\mathbf{X})$.

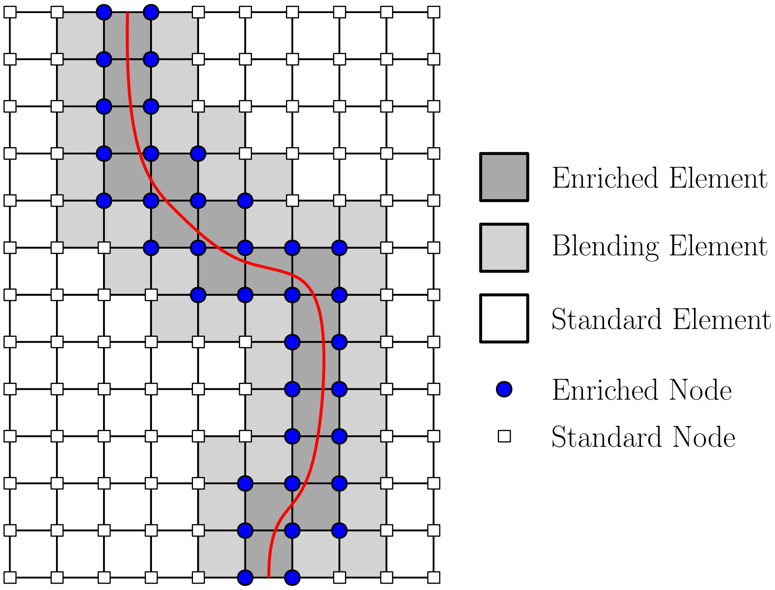


Figure 2: Types of elements in the XFEM: the weak discontinuity is presented by the red line.

Figure 2 shows an interface represented independent of the underlying FE mesh and the corresponding standard and enriched nodes. Two types of enrichment functions are generally used to model weak discontinuity. The first function is based on the absolute value of the level set function [38] and the second is a ridge function centered on the interface [35]. In this study, we use the absolute value of the level function as the enrichment function. The level set function required is given by the following signed distance function:

$$\psi(\mathbf{X}) = \min\|\mathbf{X} - \mathbf{X}^*\| \operatorname{sign}((\mathbf{X} - \mathbf{X}^*) \cdot \mathbf{n}) \tag{14}$$

where $\mathbf{X}^*$ denotes the point on the interface boundary closest to point $\mathbf{X}$, and $\mathbf{n}$ denotes the unit normal vector to the interface at a point $\mathbf{X}$. The level set function is stored at the nodes and interpolated using the conventional FE basis functions. With the displacement given by Equation (13), the strain-displacement matrix for the augmented approximation can be expressed as follows:

$$\begin{aligned} \mathbf{B}_{\mathrm{L}} &= \begin{bmatrix} \mathbf{B}_{\mathrm{L}}^{std} & \mathbf{B}_{\mathrm{L}}^{enr} \end{bmatrix} \\ \mathbf{B}_{\mathrm{NL}} &= \begin{bmatrix} \mathbf{B}_{\mathrm{NL}}^{std} & \mathbf{B}_{\mathrm{NL}}^{enr} \end{bmatrix}. \end{aligned} \tag{15}$$

The standard part (for both linear and nonlinear) of the strain-displacement matrix is given by Equation (11), whereas the enriched part reads:

$$\mathbf{B}_{\mathrm{L1}(I)}^{enr} = \begin{bmatrix} N_{I,X}^{enr} & 0 \\ 0 & N_{I,Y}^{enr} \\ N_{I,Y}^{enr} & N_{I,X}^{enr} \end{bmatrix} \quad \mathbf{B}_{\mathrm{L2}(I)}^{enr} = \begin{bmatrix} N_{I,X}^{enr} u_{X,X} & N_{I,X}^{enr} u_{Y,X} \\ N_{I,Y}^{enr} u_{X,Y} & N_{I,Y}^{enr} u_{Y,Y} \\ N_{I,Y}^{enr} u_{X,X} + N_{I,X}^{enr} u_{X,Y} & N_{I,X}^{enr} u_{Y,Y} + N_{I,Y}^{enr} u_{Y,X} \end{bmatrix}.$$

$$\mathbf{B}^{enr}_{\mathrm{NL}(I)} = \begin{bmatrix} N^{enr}_{I,X} & 0 \\ N^{enr}_{I,Y} & 0 \\ 0 & N^{enr}_{I,X} \\ 0 & N^{enr}_{I,X} \end{bmatrix}.$$

The addition of custom tailored functions to capture the local behaviour poses a challenge in computing the stiffness matrix and the residual force vector. Standard Gauss quadrature can, after all, not be employed, as the discontinuity cuts the elements arbitrarily. This has attracted considerable attentions. Some of the key contributions are: the use of higher order Gauss quadrature [39, 40], equivalent polynomial approach [41, 42], conformal mapping [43], strain smoothing technique [28, 30, 31], moment fitting [44, 45] and Euler's homogeneous function approach [46].

3.2. Overview of the smoothed finite element method

To alleviate the difficulty associated with the numerical integration of the stiffness matrix and the force vector, in this study, we propose to use the cell based smoothed finite element method (CSFEM). Here, only the key equations of the method are summarized. For a detailed derivation and the fundamental concepts of this method, the reader is referred to the open literature [15, 28] and the references therein.

The SFEM relies on computing a modified strain field, which is a weighted average of the standard compatible strain field $\boldsymbol{\varepsilon}$. In the present study however, the deformation gradient tensor is modified instead:

$$\begin{aligned} \widetilde{F}_{ij}(\mathbf{X}_c) &= \int_{\Omega^h_C} F_{ij}(\mathbf{X}) f(\mathbf{X})\, \mathrm{d}V \\ \delta_{ij} + \widetilde{u}_{i,j}(\mathbf{X}_c) &= \int_{\Omega^h_C} \left[\delta_{ij} + u_{i,j}(\mathbf{X})\right] f(\mathbf{X})\, \mathrm{d}V \end{aligned} \tag{16}$$

where $f(\mathbf{X})$ denotes the smoothing function which satisfies $f(\mathbf{X}) \geq 0$ and $\int_{\Omega^h_C} f(\mathbf{X})\, \mathrm{d}V = 1$. Our expression for $f(\mathbf{X})$ reads:

$$f(\mathbf{X}) = \begin{cases} \dfrac{1}{A_c} & \text{if } \mathbf{X} \in \Omega^h_C \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

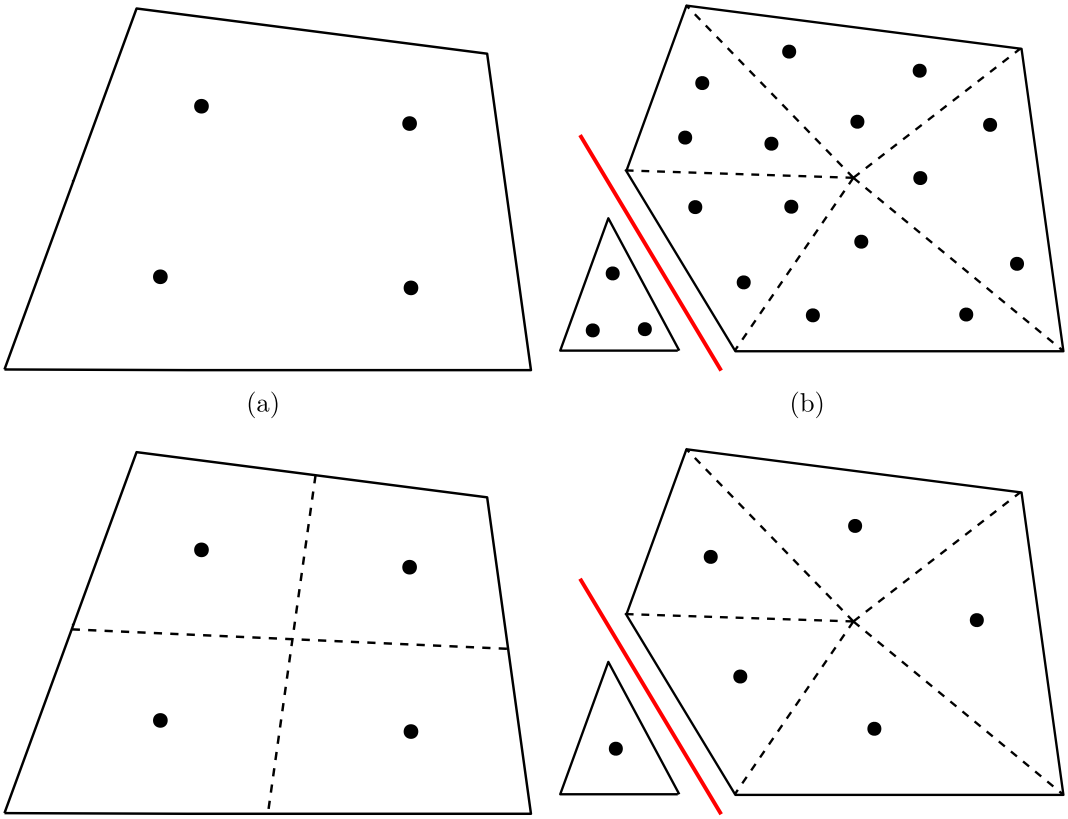
Substitution of Equation (17) in Equation (16) and by the use of the divergence theorem then yields:

$$\widetilde{u}_{i,j}(\mathbf{X}_c) = \frac{1}{A_c} \int_{\Gamma^h_C} u_i(\mathbf{X}) n_j(\mathbf{X})\, dV. \tag{18}$$

This integral can be evaluated using a one dimensional numerical integration rule, as:

$$\widetilde{u}_{i,j}(\mathbf{X}_c) = \frac{1}{A_c} \sum_{k=1}^{nb} \sum_{g=1}^{ngpt_k} u_i({}^k\mathbf{X}_g) {}^k n_j {}^k w_g. \tag{19}$$

where $\mathbf{X}_c$ denotes the internal Gauss point of the smoothing cell, A_c the area of the smoothing cell, nb the number of edges of a particular smoothing cell, $ngpt_k$ the number of Gauss points for the k^{th} edge, ${}^k n_j$ the j^{th} component of the unit normal of the k^{th} edge, and ${}^k\mathbf{X}_g$ and ${}^k w_g$ the coordinate vector and weight of the g^{th} Gauss point of the k^{th} edge, respectively. Figure 3 shows a schematic of the location of the interior Gauss points, for a standard element and an element intersected by a discontinuity. In case of the augmented displacement field, the above procedure is applied to both the standard


(a)

(b)

(c)

(d)

Figure 3: Schematic representation of location of Gauss points: (a-b) standard and enriched element in SXFEM and (c-d) standard and enriched element in S^{2}XFEM. The red line represents the discontinuity, the black circle represents the internal Gauss points and the 'dotted' line represents the sub-division. Note that the sub-division is solely for the purpose of numerical integration and does not introduce any additional dofs.

and the enriched part as:

$$\begin{aligned}
\widetilde{u}^{std}_{i,j}(\mathbf{X}_c) &= \frac{1}{A_c}\sum_{k=1}^{nb}\sum_{g=1}^{ngpt_k} u_i^{std}({}^k\mathbf{X}_g){}^k n_j{}^k w_g \\
\widetilde{u}^{enr}_{i,j}(\mathbf{X}_c) &= \frac{1}{A_c}\sum_{k=1}^{nb}\sum_{g=1}^{ngpt_k} u_i^{enr}({}^k\mathbf{X}_g){}^k n_j{}^k w_g
\end{aligned} \tag{20}$$

With these definitions, the modified strain-displacement matrix yields:

$$\widetilde{\mathbf{B}}^{std}_{\mathrm{L1}(I)} = \begin{bmatrix} \widetilde{N}^{std}_{I,X} & 0 \\ 0 & \widetilde{N}^{std}_{I,Y} \\ \widetilde{N}^{std}_{I,Y} & \widetilde{N}^{std}_{I,X} \end{bmatrix} \qquad \widetilde{\mathbf{B}}^{std}_{\mathrm{L2}(I)} = \begin{bmatrix} \widetilde{N}^{std}_{I,X}\widetilde{u}_{X,X} & \widetilde{N}^{std}_{I,X}\widetilde{u}_{Y,X} \\ \widetilde{N}^{std}_{I,Y}\widetilde{u}_{X,Y} & \widetilde{N}^{std}_{I,Y}\widetilde{u}_{Y,Y} \\ \widetilde{N}^{std}_{I,Y}\widetilde{u}_{X,X} + \widetilde{N}^{std}_{I,X}\widetilde{u}_{X,Y} & \widetilde{N}^{std}_{I,X}\widetilde{u}_{Y,Y} + \widetilde{N}^{std}_{I,Y}\widetilde{u}_{Y,X} \end{bmatrix}$$

$$\widetilde{\mathbf{B}}^{enr}_{\mathrm{L1}(I)} = \begin{bmatrix} \widetilde{N}^{enr}_{I,X} & 0 \\ 0 & \widetilde{N}^{enr}_{I,Y} \\ \widetilde{N}^{enr}_{I,Y} & \widetilde{N}^{enr}_{I,X} \end{bmatrix} \qquad \widetilde{\mathbf{B}}^{enr}_{\mathrm{L2}(I)} = \begin{bmatrix} \widetilde{N}^{enr}_{I,X}\widetilde{u}_{X,X} & \widetilde{N}^{enr}_{I,X}\widetilde{u}_{Y,X} \\ \widetilde{N}^{enr}_{I,Y}\widetilde{u}_{X,Y} & \widetilde{N}^{enr}_{I,Y}\widetilde{u}_{Y,Y} \\ \widetilde{N}^{enr}_{I,Y}\widetilde{u}_{X,X} + \widetilde{N}^{enr}_{I,X}\widetilde{u}_{X,Y} & \widetilde{N}^{enr}_{I,X}\widetilde{u}_{Y,Y} + \widetilde{N}^{enr}_{I,Y}\widetilde{u}_{Y,X} \end{bmatrix}$$

$$\widetilde{\mathbf{B}}^{std}_{\mathrm{NL}(I)} = \begin{bmatrix} \widetilde{N}^{std}_{I,X} & 0 \\ \widetilde{N}^{std}_{I,Y} & 0 \\ 0 & \widetilde{N}^{std}_{I,X} \\ 0 & \widetilde{N}^{std}_{I,X} \end{bmatrix} \qquad \widetilde{\mathbf{B}}^{enr}_{\mathrm{NL}(I)} = \begin{bmatrix} \widetilde{N}^{enr}_{I,X} & 0 \\ \widetilde{N}^{enr}_{I,Y} & 0 \\ 0 & \widetilde{N}^{enr}_{I,X} \\ 0 & \widetilde{N}^{enr}_{I,X} \end{bmatrix}.$$

and the following smoothed elemental stiffness matrix and the smoothed elemental internal force vector:

$$\begin{aligned}
\widetilde{\mathbf{K}}^e_{\mathrm{M}} &= \sum_{C=1}^{nc}\left(\sum_{q=1}^{ngp}\widetilde{\mathbf{B}}^{\mathrm{T}}_{\mathrm{L}}({}^q\mathbf{X}_C)\widetilde{\mathbf{D}}({}^q\mathbf{X}_C)\widetilde{\mathbf{B}}_{\mathrm{L}}({}^q\mathbf{X}_C){}^q w_C\right) \\
\widetilde{\mathbf{K}}^e_{\mathrm{G}} &= \sum_{C=1}^{nc}\left(\sum_{q=1}^{ngp}\widetilde{\mathbf{B}}^{\mathrm{T}}_{\mathrm{NL}}({}^q\mathbf{X}_C)\widetilde{\mathbf{S}}_m({}^q\mathbf{X}_C)\widetilde{\mathbf{B}}_{\mathrm{NL}}({}^q\mathbf{X}_C){}^q w_C\right) \\
\widetilde{\mathbf{f}}^h_{int} &= \sum_{C=1}^{nc}\left(\sum_{q=1}^{ngp}\widetilde{\mathbf{B}}^{\mathrm{T}}_{\mathrm{L}}({}^q\mathbf{X}_C)\widetilde{\mathbf{S}}_v({}^q\mathbf{X}_C){}^q w_C\right).
\end{aligned} \tag{21}$$

where *nc* is the number of smoothing cells and *ngp* is the number of Gauss points within the smoothing cell. Note that in evaluating the smoothed strain-displacement matrices and the corresponding stiffness matrix, no explicit form of the shape functions is required and it eliminates the need for an isoparametric mapping.

4. Recovery based error indicator and quadtree decomposition

4.1. Recovery based error indicator

The recovery based error indicator proposed by Bordas and Duflot [47, 48] is employed to assess the error and list the elements for refinement. In this method, the enhanced strain field is computed using the standard nodal solution through the eXtended Moving Least Square (XMLS) derivative recovery process. The Moving Least Square (MLS) shape functions of the n_x points whose domain of influence contain points $\mathbf{x}$ is given by:

$$\mathbf{\Phi}^T(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x}) \tag{22}$$

where $\mathbf{p}(\mathbf{x}) = \begin{bmatrix} 1 & x & y \end{bmatrix}$ denotes the reproducing polynomial and

$$\begin{aligned} \mathbf{A}(\mathbf{x}) &= \sum_{I=1}^{n_x} w_I(\mathbf{x})\mathbf{p}(\mathbf{x}_I)\mathbf{p}^T(\mathbf{x}_I) \\ \mathbf{B}(\mathbf{x}) &= \begin{bmatrix} w_1(\mathbf{x})\mathbf{p}(\mathbf{x}_1) & w_2(\mathbf{x})\mathbf{p}(\mathbf{x}_2) & \dots & w_{n_x}(\mathbf{x})\mathbf{p}(\mathbf{x}_{n_x}) \end{bmatrix}. \end{aligned}$$

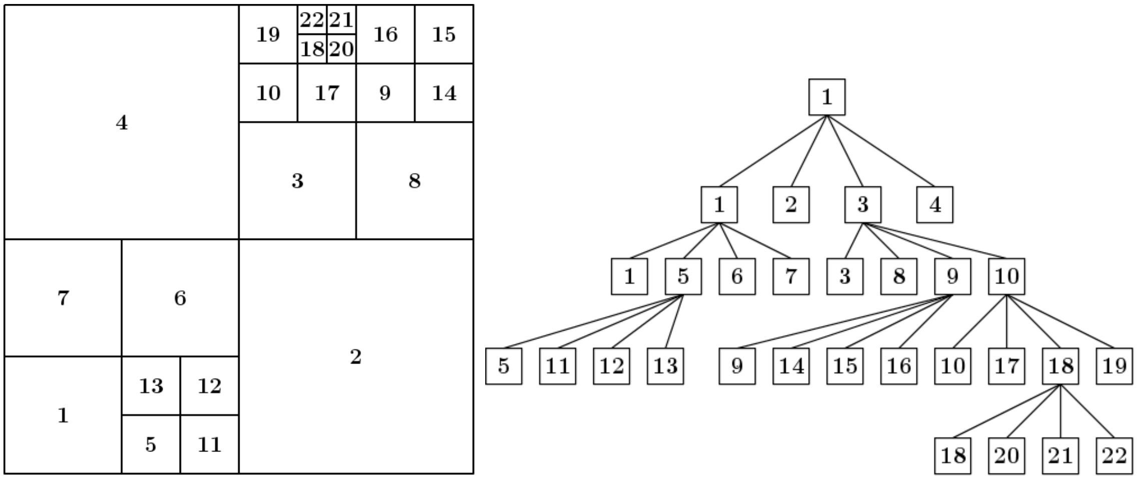
Here, the weight function of a node is calculated by the diffraction method with a circular domain of influence for node $\mathbf{x}_k$. In this work, a fourth order spline is taken as the weighting function:

$$w_k(s) = \begin{cases} 1 - 6s^2 + 8s^3 - 3s^4 & \text{if} \quad |s| \leq 1 \\ 0 & \text{if} \quad |s| > 1 \end{cases}. \tag{23}$$

where $s = \frac{\|\mathbf{x}-\mathbf{x}_k\|}{d_k}$ and d_k denotes the support domain of node $\mathbf{x}_k$. The function takes 0 value when the parameter s takes value less than 1, which means nodes whose domain of influence does not contain the point $\mathbf{x}$ will not contribute to the recovery process. Since the strain is discontinuous across the discontinuity, nodes hidden behind the discontinuity are also neglected. The enhanced derivative of the shape functions are computed by finding the derivatives of the MLS shape functions (see Equation (22)). Using the enhanced derivatives, the enhanced deformation gradient $\mathbf{F}_s$ and the enhanced Lagrangian strain $\mathbf{E}_s$ can be found. The error between the enhanced strain field and the standard strain field is considered as the error. The tolerance is considered based on the bulk error criteria, where the fixed fraction of the total error creating elements are refined in the next level. The elements selected for the refinement are chosen by the descending order of their individual elemental error.

4.2. Quadtree decomposition

In this meshing technique, one specific criterion named as the stopping criterion is chosen to decide which element requires to be further refined. If the given element does not satisfy the stopping criterion within the user specified tolerance limit, it will be divided into four child elements. A quadtree structure and a mesh with element numbering are shown in Figure 4. This criterion can be a geometry based factor or



(a) Quadtree mesh

(b) Tree structure for the mesh

Figure 4: Quadtree decomposition: (a) representative quadtree mesh and (b) tree structure employed to store the mesh details.

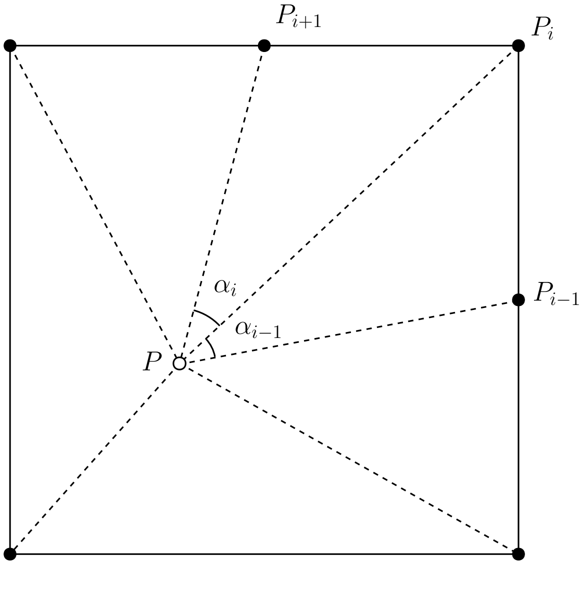


Figure 5: Schematic representation of an element with hanging node and the construction of mean value coordinates.

any error indicator. The above decomposition, leads to elements with hanging nodes and the conventional FE approach cannot handle such elements without additional work. This is because of lack of compatibility. In this work, the elements with hanging nodes are considered as the polygons (see Figure 5) and the mean-value coordinates [49] are used to approximate the unknown fields, that are compatible across the element boundaries. The mean-value coordinates for an arbitrary polygon is given by:

$$
\begin{aligned}
N_i(\mathbf{x}) &= \frac{W_i(\mathbf{x})}{\sum_{j=1}^{n} W_i(\mathbf{x})}, \quad i = 1, \cdots, n \\
W_i(\mathbf{x}) &= \frac{\tan(\alpha_{i-1}/2) + \tan(\alpha_i/2)}{\|\mathbf{x} - \mathbf{x}_i\|}
\end{aligned}
\tag{24}
$$

where n is the number of nodes in an element, $\mathbf{x}_i$ is the coordinate of point P_i and α_i's are the internal angle.

5. Numerical Examples

In this section, two 2-dimensional examples of a plane strain nature are solved to demonstrate the accuracy and the robustness of the proposed method. Here, the results of the proposed framework, i.e. Smoothed SXFEM (S^{2}XFEM), are compared with those of the stable XFEM (SXFEM) in terms of error norms and computational time. All the computations were performed on a i7-7700K CPU@4.2GHZ and 32GB RAM workstation. As analytical solutions are not available for the problems due to the nonlinearity of the constitutive law, the standard FEM solution using an overkill discretisation from Abaqus is used as the reference solution. Note that the overkill FE solution uses a conforming mesh. For both SXFEM and S^{2}XFEM, the adaptive refinement combined with the quadtree decomposition is used. We measure the accuracy by determining the L_2-norms of the displacements and the Cauchy stress as follows:

$$
\begin{aligned}
||\mathbf{u}||_{L_2} &= \frac{\int\limits_{\Omega_0} (\mathbf{u} - \mathbf{u}_h) \cdot (\mathbf{u} - \mathbf{u}_h) \, \mathrm{d}V}{\int\limits_{\Omega_0} \mathbf{u} \cdot \mathbf{u} \, \mathrm{d}V} \\
||\boldsymbol{\sigma}||_{L_2} &= \frac{\int\limits_{\Omega_0} (\boldsymbol{\sigma} - \boldsymbol{\sigma}_h) \cdot (\boldsymbol{\sigma} - \boldsymbol{\sigma}_h) \, \mathrm{d}V}{\int\limits_{\Omega_0} \boldsymbol{\sigma} \cdot \boldsymbol{\sigma} \, \mathrm{d}V}
\end{aligned}
\tag{25}
$$

where $\mathbf{u}$ and $\boldsymbol{\sigma}$ denote the reference displacement and stress values, respectively, and $\mathbf{u}_h$ and $\boldsymbol{\sigma}_h$ denote the computed displacement and stress values, respectively. An outline of the nonlinear XFEM implementation with the adaptive refinement is shown in Algorithm 1. For the purpose of numerical integration, the elements that are inter-

Algorithm 1: Pseudo-code for adaptive smoothed stable XFEM

```
Input  : Material parameters, boundary conditions, inclusion information
Output: Displacement field
while level <= maxLevel or number of slave elements > 0 do
    update the mesh by refining the elements ;
    find enrichment function at the nodes of the new mesh ;
    initialize parameters for Newton iteration: err, count, $\mathbf{d}$ ;
    for steps $\leftarrow 1$ to numsteps do
        count = 1; convergence = 1; factor = step/numsteps; ;
        compute external force $\mathbf{f}_{ext}$ (at $\mathbf{d}$) ;
        while err > tolerance do
            find stiffness matrix $\mathbf{K}_\mathrm{T}$ and internal force $\mathbf{f}_{int}$ (at $\mathbf{d}$) ;
            find the residual $\mathbf{R} = \mathbf{f}_{ext} - \mathbf{f}_{int}$ ;
            solve for incremental displacement: $\Delta\mathbf{d} = \mathbf{K}_\mathrm{T}^{-1}\mathbf{R}$ ;
            update $\mathbf{d} = \mathbf{d} + \Delta\mathbf{d}$ ;
            find $err = ||\Delta\mathbf{d}/\mathbf{d}||$ ;
            update internal boundary conditions;
    estimate the enhanced Lagrangian strain and compute the error in the FE
     solution ;
    find the elements with error greater than the user defined tolerance ;
```

sected by the discontinuity are sub-divided into triangles and higher order quadrature is employed in case of SXFEM, whilst the modified derivative is computed at the center of the triangle in case of S^{2}XFEM, as mentioned in Section 3.2. Note that the purpose of the subdivision is for numerical integration and does not introduce additional dofs. Table 1 presents the quadrature rule employed for the SXFEM and S^{2}XFEM.

Table 1: Number of quadrature points (per element) employed to compute the terms in the stiffness matrix and the residual force vector for standard and enriched elements for SXFEM and S^{2}XFEM.

Type of element	SXFEM	S^{2}XFEM
Standard elements	4	4
Split enriched elements (per triangle)	3	1
Blending elements	4	4

5.1. Square plate with a circular inclusion

The first example considers a stiff circular inclusion in the center of a square plate (see Figure 6). The radius of the inclusion is taken as 0.21 mm and each side of the plate has a length of 1 mm. Both the matrix and the inclusion are considered as compressible Neo-Hookean materials given by the stored energy function, see Equation (5). Table 2 lists the material properties of the matrix and the inclusion. The vertical displacements at the bottom surface are restrained and a vertical displacement of 0.15mm is applied at the top surface in the negative $y-$direction. This loading is applied in five increments and each increment is stopped when the absolute residual $< 1{\times}10^{-11}$. As mentioned earlier, the geometry of the inclusion is implicitly defined and a coarse discretization is adopted. Based on the error indicator, the elements with an error larger than a user specified tolerance (70% for the current study) are listed in its ascending order. By maintaining a 2:1 ratio, the elements in the list are divided into four ‘child’ elements. Figure 7 shows the mesh obtained after the third level of refinement and the corresponding error distribution; the mesh at the fourth level is also shown for comparison. Based on this, five levels of refinements is found to be adequate. Table 3 compares

Table 2: Square plate with a rigid circular inclusion: material properties for the matrix and the inclusion

Method	Shear modulus (MPa)	Bulk modulus (MPa)
	μ	κ
Matrix	2.5	12.5
Inclusion	2.5×10^2	12.5×10^2

Table 3: Comparison of the degrees of freedom, relative error in L_2 norm between the SXFEM and the S^{2}XFEM for different levels of refinement. The total dofs of the conventional FEM is also given, the solution of which is used as the reference solution.

Method	Number	Number	Total	Number	Relative	Relative
	of	of	dofs	of	error in	error in
	standard	enriched		Gauss	displacement	Cauchy stress
	nodes	nodes		points	L_2 norm	L_2 norm
S^{2}XFEM						
Level 1:	121	36	314	440	1.4736×10^{-2}	5.7303×10^{-1}
Level 2:	341	72	826	1,292	5.6756×10^{-3}	3.9826×10^{-1}
Level 3:	853	136	1,978	3,280	1.9793×10^{-3}	2.4261×10^{-1}
Level 4:	1,877	264	4,282	7,368	1.0397×10^{-3}	1.2765×10^{-1}
Level 5:	4,977	536	11,026	10,256	3.8898×10^{-4}	4.7072×10^{-2}
SXFEM						
Level 1:	121	36	314	1,160	1.7910×10^{-2}	1.9109×10^{0}
Level 2:	341	72	826	3,476	5.6402×10^{-3}	6.7000×10^{-1}
Level 3:	881	136	2,034	8,588	1.9492×10^{-3}	3.4003×10^{-1}
Level 4:	2,283	264	5,094	19,928	8.0782×10^{-4}	1.4061×10^{-1}
Level 5:	6,195	536	13,462	46,252	3.1209×10^{-4}	6.7708×10^{-2}
FEM	138,683		277,366	552,064		

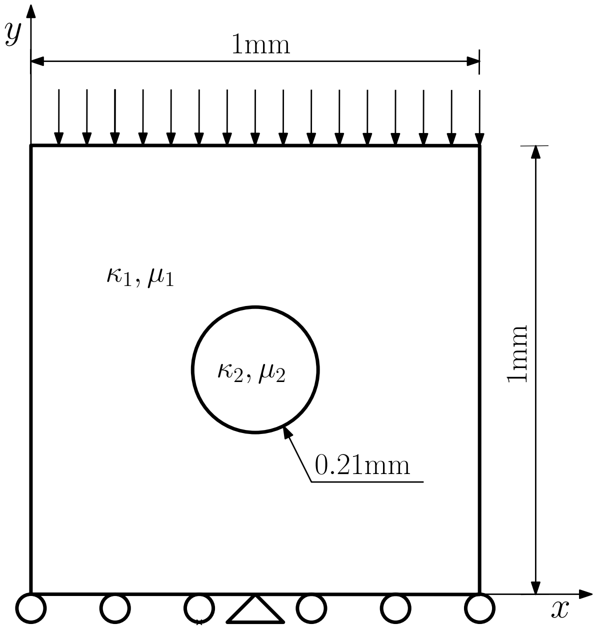
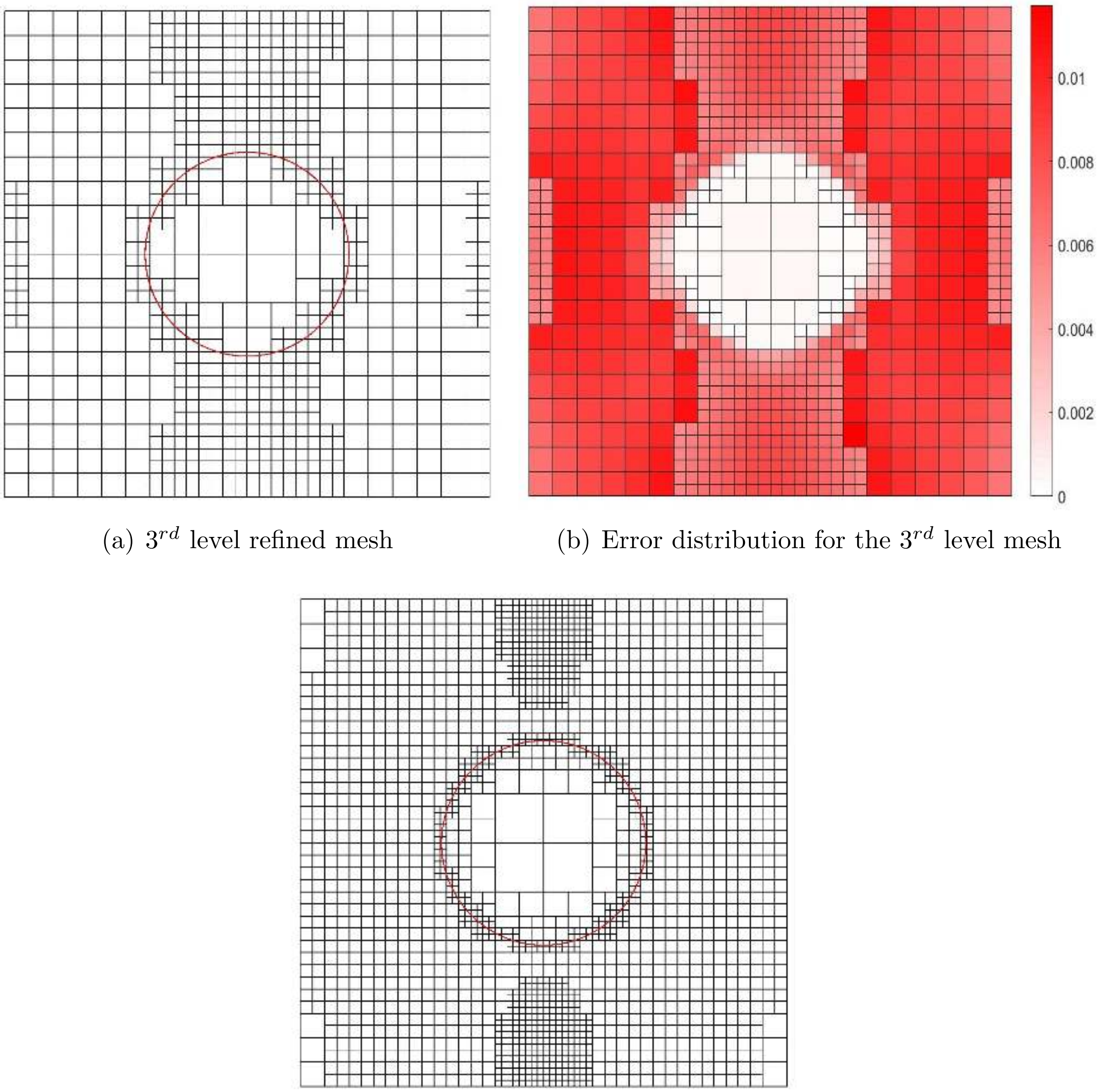


Figure 6: Domain description of the problem

the total number of dofs, number of Gauss points required to evaluate the bilinear and linear form and the relative error in the L_2 norm of the displacement and the Cauchy stress for SXFEM and S^{2}XFEM. An overkill FE solution is used as reference solution and the error in the L_2 norm is obtained by projection of meshes. It can be seen that with adaptive refinement, both the methods converge asymptotically with optimal convergence rate as shown in Figure 8. From Table 3, it is opined that the total dofs for SXFEM and S^{2}XFEM are almost similar, however, S^{2}XFEM requires fewer number of Gauss points (roughly 4 times less) than the SXFEM without compromising the accuracy of the solution. Figures 9-10 show the displacement and Cauchy stress comparison between the proposed framework, the SXFEM and the conventional FEM. It is seen that a good comparison is obtained. A detailed comparison of computational time is presented in the next example.



(a) 3^{rd} level refined mesh

(b) Error distribution for the 3^{rd} level mesh

(c) 4^{th} level refinement

Figure 7: Refinement from 3^{rd} level mesh to 4^{th} level mesh using the error distribution given by the error indicator: (a) shows the mesh at level 3 and the corresponding error plot and (b) the mesh at level 4.

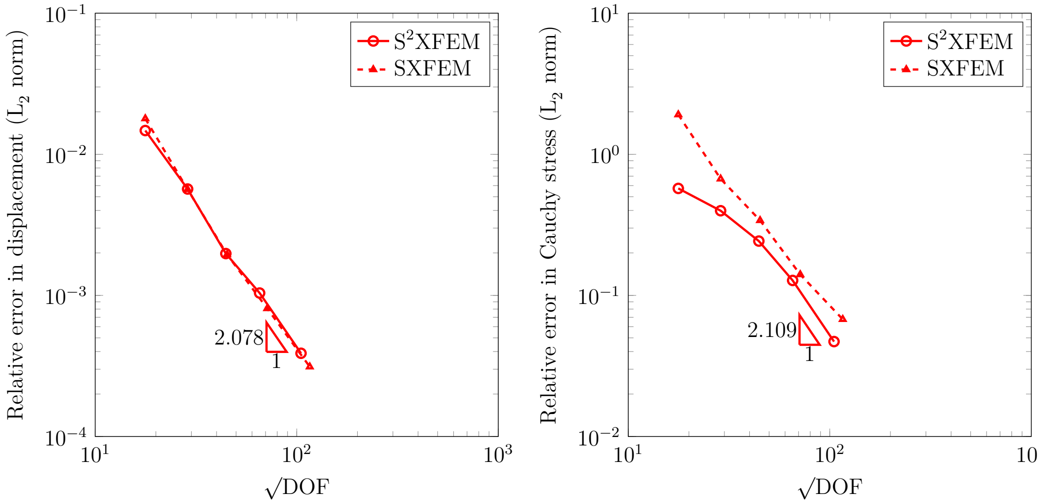
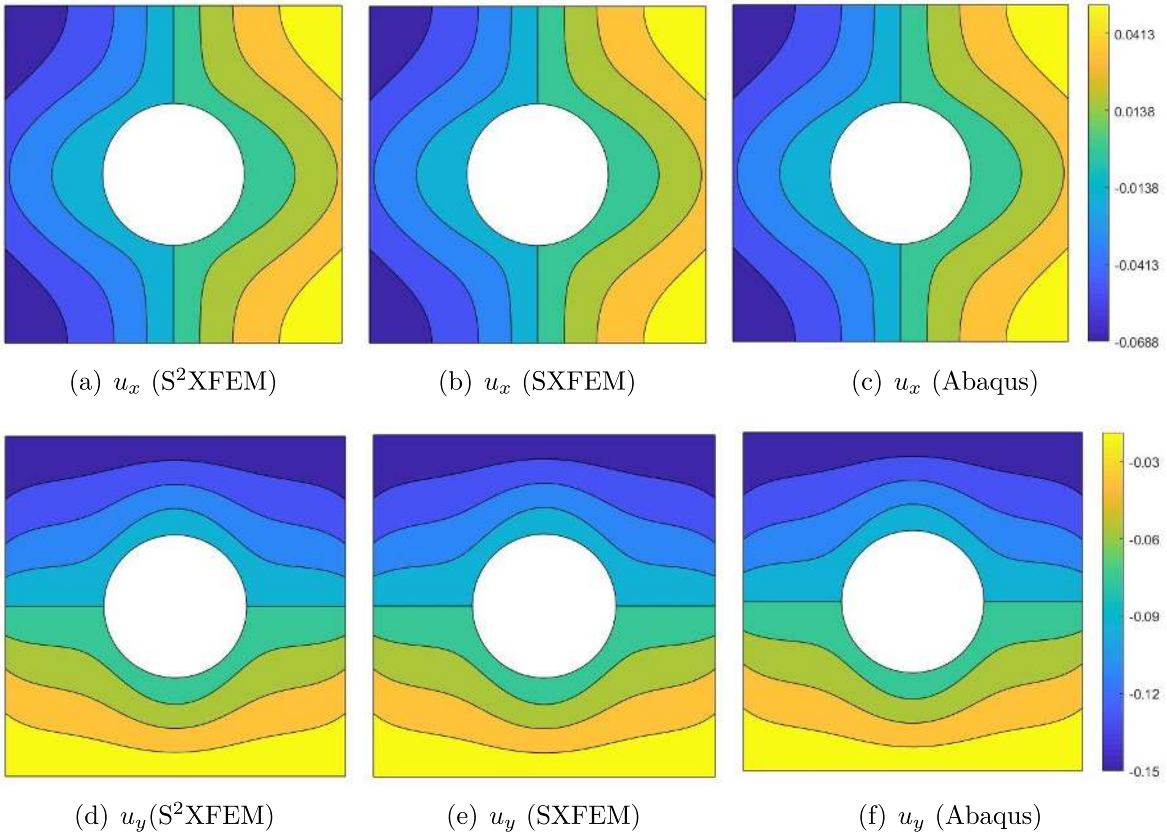


Figure 8: Square plate with a single inclusion: convergence of the relative error in the L_2 norm of (a) the displacement and (b) Cauchy stress with mesh refinement for SXFEM and S^{2}XFEM.



(a) u_x (S^{2}XFEM)

(b) u_x (SXFEM)

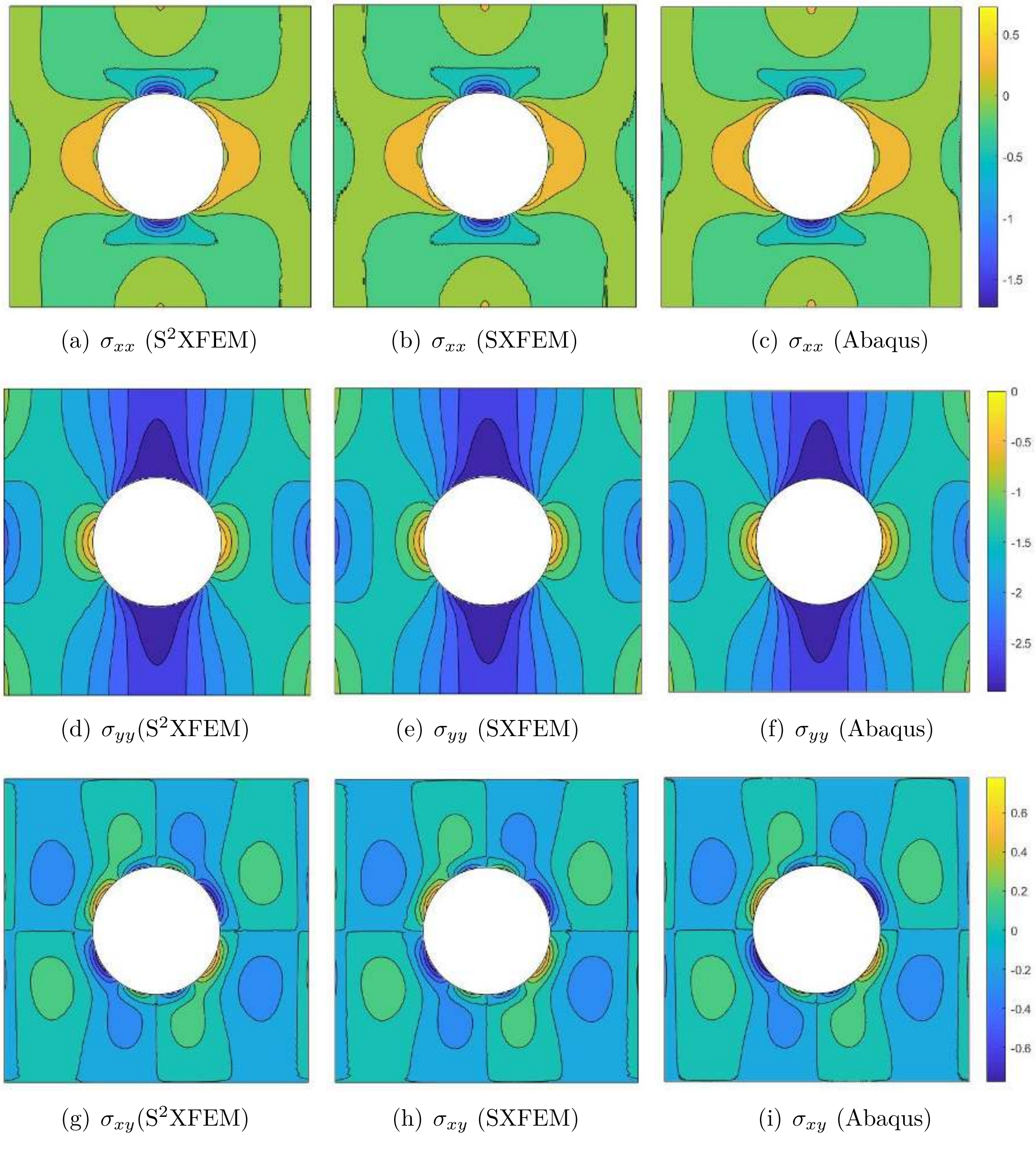
(c) u_x (Abaqus)

(d) u_y(S^{2}XFEM)

(e) u_y (SXFEM)

(f) u_y (Abaqus)

Figure 9: Comparison of displacement contours along x and y direction between S^{2}XFEM, SXFEM and Abaqus.



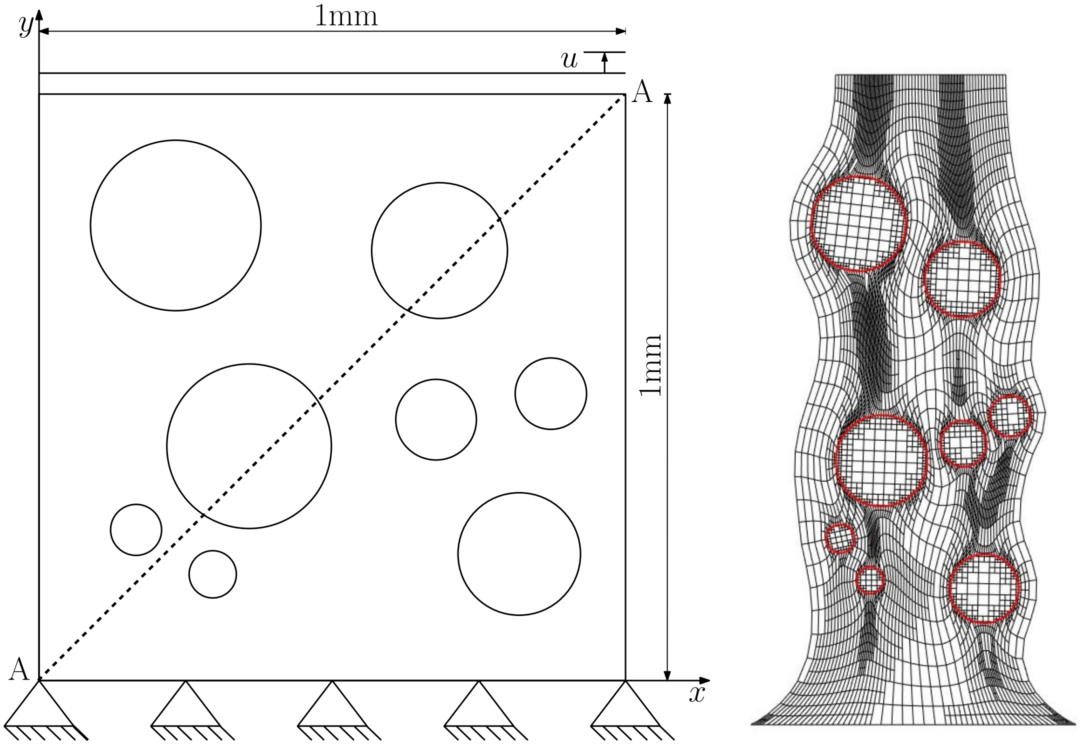
(a) σ_{xx} (S^{2}XFEM) (b) σ_{xx} (SXFEM) (c) σ_{xx} (Abaqus)

(d) σ_{yy}(S^{2}XFEM) (e) σ_{yy} (SXFEM) (f) σ_{yy} (Abaqus)

(g) σ_{xy}(S^{2}XFEM) (h) σ_{xy} (SXFEM) (i) σ_{xy} (Abaqus)

Figure 10: Cauchy stress distribution in the matrix for a single inclusion: a comparison between the S^{2}XFEM, SXFEM and Abaqus.

5.2. Square domain with multiple random inclusions - Dirichlet Boundary Condition



(a) Domain description

(b) Final deformed configuration

Figure 11: Square domain with multiple random inclusions. Note that along section A-A, the results will discussed.

The second example considers 8 stiff inclusions of different sizes, randomly placed in a matrix material (Figure 11). The radius of the smallest inclusion is approximately 0.041 mm and that of the largest inclusion is approximately 0.1458 mm. The random inclusions are generated using a random sequential adsorption algorithm [50]. The domain is fixed on the bottom edge and top edge is given a displacement of 1 mm in the positive y direction as shown Figure 11. Table 4 lists the material properties of the matrix and the inclusions. The reference solution is again obtained using a

Table 4: Material properties for the matrix and the inclusion

Material	Shear modulus (MPa)	Bulk modulus (MPa)
	μ	κ
Matrix	$^7/_9 \times 10^5$	$^7/_3 \times 10^5$
Inclusion	$^7/_9 \times 10^9$	$^7/_3 \times 10^9$

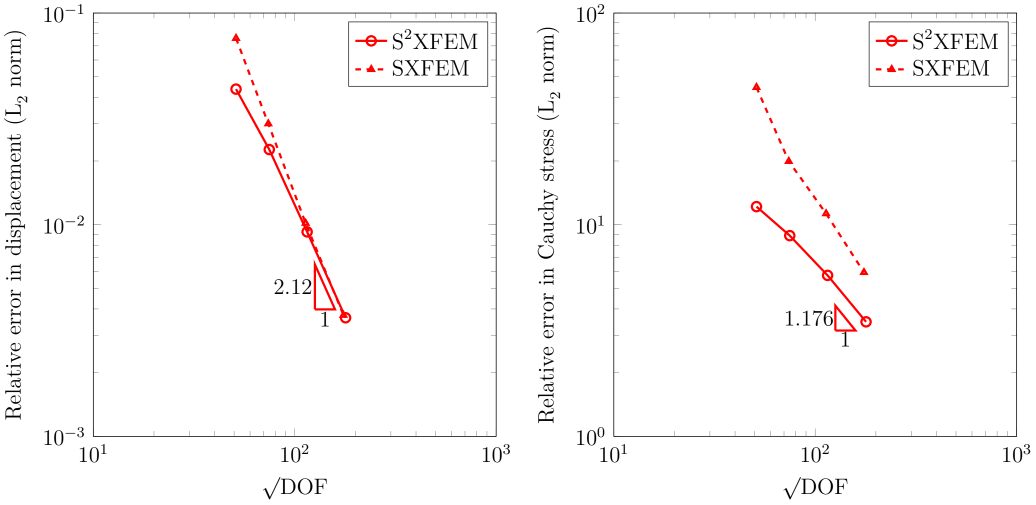


Figure 12: Square plate with multiple inclusions: convergence of relative error in (a) displacement and (b) Cauchy stress.

conforming mesh in Abaqus, this time with 133,965 nodes. The final mesh employed for SXFEM and S^{2}XFEM consists of 12,464 standard nodes (+ 2780 enriched nodes) and 13,209 standard nodes (+ 2780 enriched nodes), respectively and the final deformed configuration is shown in Figure 11(b). Figure 12 shows the convergence of the relative error in the L_2 norm of the displacement and Cauchy stress with mesh refinement and it is seen that both the methods converge asymptotically with mesh refinement. Table 5 shows the time taken by the error indicator procedure at different levels of refinement for SXFEM and S^{2}XFEM, respectively and it is seen that the time taken is slightly higher for SXFEM than the S^{2}XFEM, although the SXFEM has fewer DOFs. This is because, SXFEM requires 'more' integration points to evaluate the terms in the bilinear and linear form. Next, the relative error in Cauchy stress along the section A-A and the corresponding DOFs for different levels of refinement is shown for SXFEM and S^{2}XFEM in Table 6. For comparison, the DOFs and the relative error in case of FEM is also given. In case of FEM, the built-in error indicator (based on Mises equivalent stress) with adaptive refinement available in Abaqus with default setting is employed.

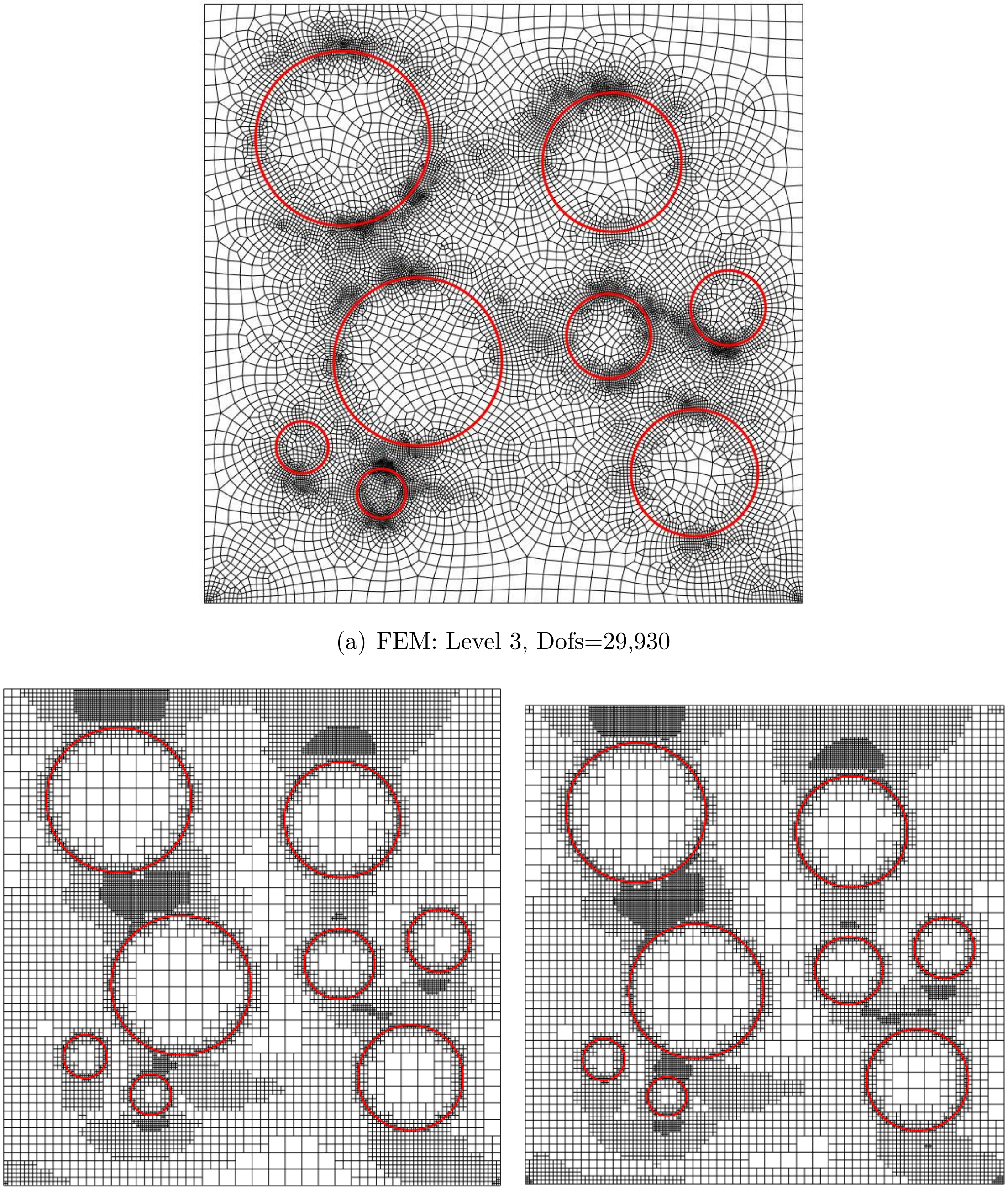
Table 5: Error indicator computation time for SXFEM and S^{2}XFEM

Level	DOFs		CPU time (sec)	
	S^{2}XFEM	SXFEM	S^{2}XFEM	SXFEM
Level 1	2,612	2,612	11.976	48.627
Level 2	5,588	5,464	31.050	126.240
Level 3	13,272	12,822	121.95	383.727

Table 6: Relative error in Cauchy Stress and DOFs at different levels of refinement for FEM, SXFEM and S^{2}XFEM along the section AA

Method	Level	DOFs	Relative
			Stress Error
FEM	Level 1	2,924	7.9043×10^{-2}
	Level 2	8,514	5.8445×10^{-2}
	Level 3	29,930	1.1230×10^{-2}
SXFEM	Level 1	2,612	3.1239×10^{0}
	Level 2	5,464	1.4728×10^{0}
	Level 3	12,822	3.2782×10^{-1}
	Level 4	30,488	9.5557×10^{-2}
S^{2}XFEM	Level 1	2,612	4.4418×10^{0}
	Level 2	5,588	4.2432×10^{-1}
	Level 3	13,272	2.3028×10^{-1}
	Level 4	31,978	5.1661×10^{-2}

Figure 13 shows the mesh adopted at level 3 in case of FEM and level 4 in case of SXFEM and S^{2}XFEM. It is seen that the SXFEM and S^{2}XFEM yields similar order of accuracy, with the advantage that the geometry can be independent of the underlying discretization.



(a) FEM: Level 3, Dofs=29,930

(b) SXFEM: Level 4, Dofs=30,488

(c) S^{2}XFEM: Level 4, Dofs=31,978

Figure 13: Mesh details for FEM (level 3) and for SXFEM and S^{2}XFEM (level 4). The 'red' line denotes the interface.

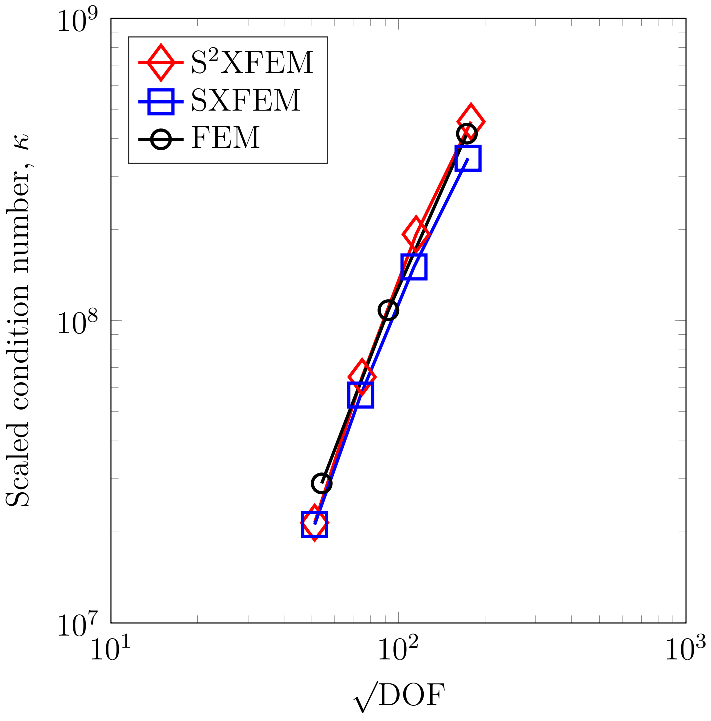


Figure 14: Scaled conditioning number as a function of mesh size for the S^{2}XFEM, SXFEM and the FEM.

The scaled conditioning number κ of the transformed matrix $\tilde{\mathbf{K}} = \mathbf{P}^{\mathrm{T}}\mathbf{K}\mathbf{P}$ with increasing total DOFs is shown in Figure 14, where $\mathbf{P} = \sqrt{\text{diag}(\mathbf{K})^{-1}}$ is the preconditioner, chosen such that the conditioning number of the transformed matrix $\tilde{\mathbf{K}}$ is smaller than the original matrix. It is inferred from Figure 14 that the scaled conditioning number from the proposed method increases at the same rate as the SXFEM and the conventional FEM with increasing DOFs. Figure 15 shows the computational effort required for SXFEM and the S^{2}XFEM for evaluating the stiffness matrix and the subsequent solution of linear systems. The subscript 's' refers to the proposed framework. It is evident from Figure 15 that the proposed framework requires less computational time when compared to SXFEM. The Cauchy stress along the section AA (see Figure 11) is calculated. The results obtained from SXFEM and S^{2}XFEM are compared with the results obtained from Abaqus (FEM) in Figure 16. It is opined that both SXFEM and S^{2}XFEM yields accurate results.

6. Conclusions

This paper proposes to combine the stable XFEM with strain smoothing. A recovery based error indicator together with a quadtree decomposition ensures an effective mesh refinement. The elements with hanging nodes are treated as polygonal elements, which ensures compatibility between elements and the mean value coordinates as the basis functions. The results from the proposed framework are compared with those of the

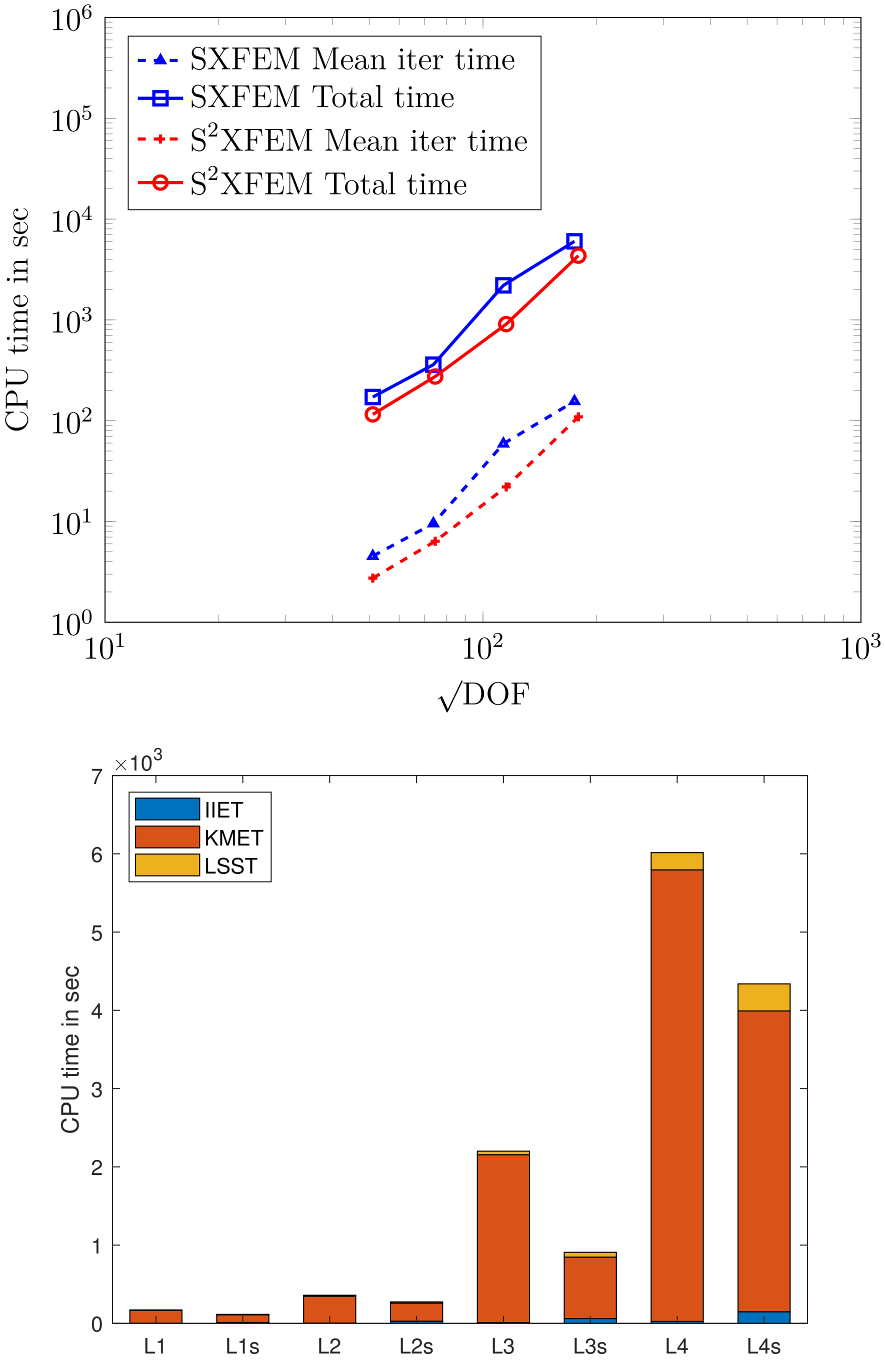


Figure 15: CPU time comparison between SXFEM and S^2XFEM in terms of (a) total time and mean iteration time (b) IIET (Initial Information Evaluation Time), KMET (Stiffness Matrix Evaluation Time) and LSST (Linear System Solving Time) (where, Ln = SXFEM Level n , Lns= S^2XFEM Level n

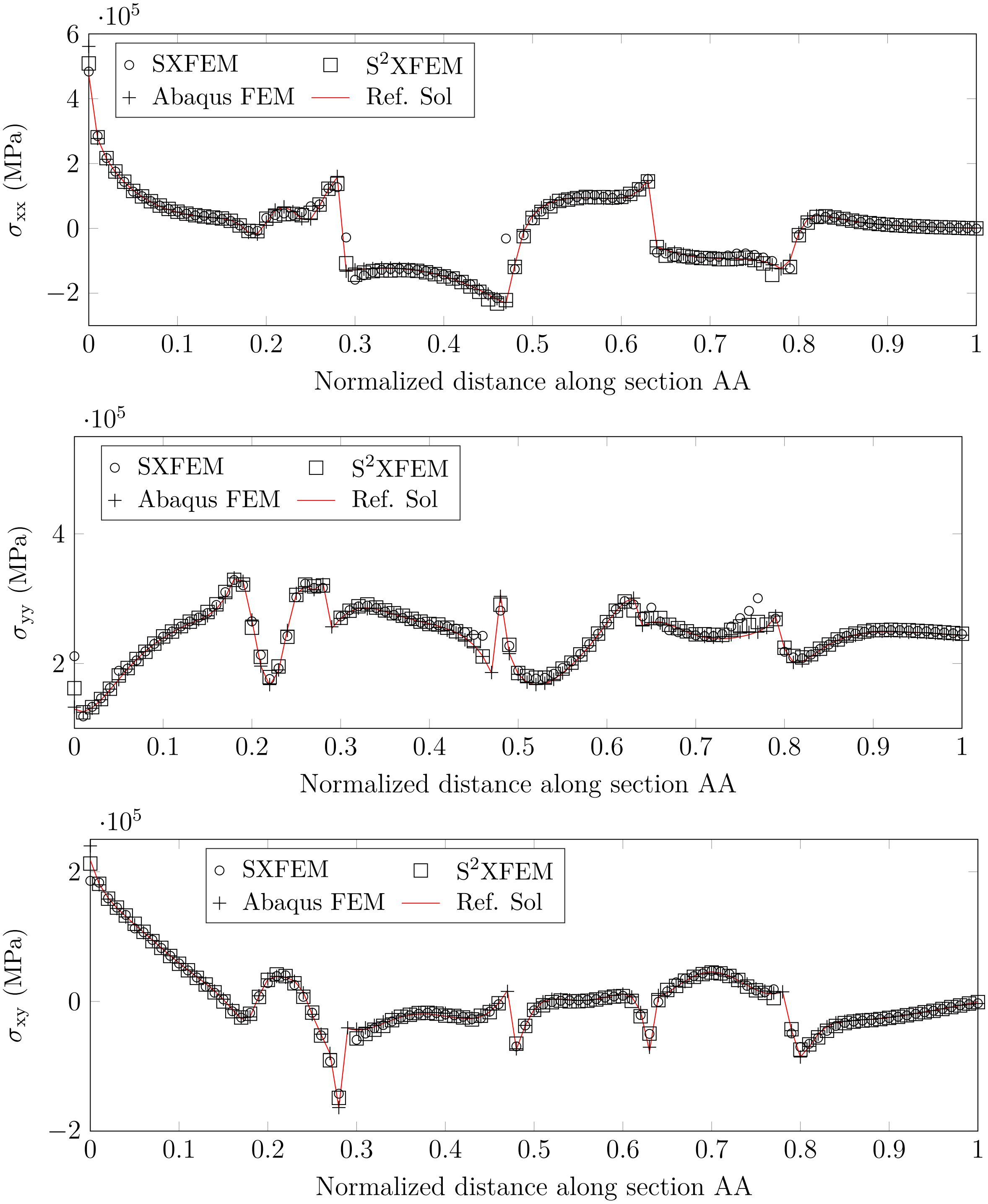


Figure 16: Cauchy stress along the section AA. Note that the Abquas FEM is based on adaptive refinement and the Ref. Solution is obtained using an uniformly FE refined mesh (conforming mesh).

stable XFEM and those of a conventional FEM with a substantially fine discretisation. It is seen that the S^2XFEM yields comparable results. Moreover, the proposed method does not require the derivative of the shape functions, hence, the isoparametric mapping is alleviated and requires fewer integration points to achieve similar level of accuracy as that of the stable XFEM.

Acknowledgements

Krishna Kannan and Sundararajan Natarajan would like to thank MHRD and MoRTH, Government of India, for the financial support under the IMPRINT India initiative (Pr.No. MEE/1617/357 /MIMP/KRIA), and the industrial partner MRF.

Reference

[1] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 1977.

[2] T. Belytschko, Y. Y. Lu, and L. Gu. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 1994.

[3] Wing Kam Liu, Sukky Jun, and Yi Fei Zhang. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids*, 20(8-9):1081–1106, 1995.

[4] C. Armando Duarte and J. Tinsley Oden. H-p clouds: an h-p meshless method. *Numerical Methods for Partial Differential Equations*, 12(6):673–705, 1996.

[5] J. M. Melenk and I. Babuška. Partition of unity finite element method: basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):289–314, 1996.

[6] I. Babuška and J. M. Melenk. The partition of unity method. *International Journal for Numerical Methods in Engineering*, 40(4):727–758, 1997.

[7] N. Moës, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46(1):131–150, 1999.

[8] T. Belytschko and T. Black. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, 45(5):601–620, 1999.

[9] Zuoyi Kang, Tinh Quoc Bui, Du Dinh Nguyen, Takahiro Saitoh, and Sohichi Hirose. An extended consecutive-interpolation quadrilateral element (XCQ4) applied to linear elastic fracture mechanics. *Acta Mechanica*, 2015.

[10] Zhen Wang, Tiantang Yu, Tinh Quoc Bui, Ngoc Anh Trinh, Nguyen Thi Hien Luong, Nguyen Dinh Duc, and Duc Hong Doan. Numerical modeling of 3-D inclusions and voids by a novel adaptive XFEM. *Advances in Engineering Software*, 2016.

[11] Tiantang Yu and Tinh Quoc Bui. Numerical simulation of 2-D weak and strong discontinuities by a novel approach based on XFEM with local mesh refinement. *Computers and Structures*, 2018.

[12] Jiming Gu, Tiantang Yu, Le Van Lich, Thanh Tung Nguyen, and Tinh Quoc Bui. Adaptive multi-patch isogeometric analysis based on locally refined B-splines. *Computer Methods in Applied Mechanics and Engineering*, 2018.

[13] Jiming Gu, Tiantang Yu, Le Van Lich, Thanh Tung Nguyen, Satoyuki Tanaka, and Tinh Quoc Bui. Multi-inclusions modeling by adaptive XIGA based on LR B-splines and multiple level sets. *Finite Elements in Analysis and Design*, 2018.

[14] Chen, Jiun-Shyan and Wu, Cheng-Tang and Yoon, Sangpil and You, Yang. A stabilized conforming nodal integration for galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 50(2):435–466, 2001.

[15] G. R. Liu, K. Y. Dai, and T. T. Nguyen. A smoothed finite element method for mechanics problems. *Computational Mechanics*, 39(6):859–877, 2007.

[16] P. Liu, T. Q. Bui, Ch Zhang, T. T. Yu, G. R. Liu, and M. V. Golub. The singular edge-based smoothed finite element method for stationary dynamic crack problems in 2D elastic solids. *Computer Methods in Applied Mechanics and Engineering*, 2012.

[17] L. Wu, P. Liu, C. Shi, Z. Zhang, Tinh Quoc Bui, and D. Jiao. Edge-based smoothed extended finite element method for dynamic fracture analysis. *Applied Mathematical Modelling*, 2016.

[18] Zuoyi Kang, Tinh Quoc Bui, Takahiro Saitoh, and Sohichi Hirose. Quasi-static crack propagation simulation by an enhanced nodal gradient finite element with different enrichments. *Theoretical and Applied Fracture Mechanics*, 2017.

[19] Zuoyi Kang, Tinh Quoc Bui, Du Dinh Nguyen, and Sohichi Hirose. Dynamic stationary crack analysis of isotropic solids and anisotropic composites by enhanced local enriched consecutive-interpolation elements. *Composite Structures*, 2017.

[20] Minh Ngoc Nguyen, Tinh Quoc Bui, Nha Thanh Nguyen, Thien Tich Truong, and Le Van Lich. Simulation of dynamic and static thermoelastic fracture problems by extended nodal gradient finite elements. *International Journal of Mechanical Sciences*, 2017.

[21] H. Nguyen-Xuan and G. R. Liu. An edge-based smoothed finite element method softened with a bubble function (bES-FEM) for solid mechanics problems. *Computers and Structures*, 2013.

[22] Thanh Hai Ong, Claire E. Heaney, Chang Kye Lee, G. R. Liu, and H. Nguyen-Xuan. On stability, convergence and accuracy of bES-FEM and bFS-FEM for nearly incompressible elasticity. *Computer Methods in Applied Mechanics and Engineering*, 2015.

[23] T. Nguyen-Thoi, G. R. Liu, H. C. Vu-Do, and H. Nguyen-Xuan. An edge-based smoothed finite element method for visco-elastoplastic analyses of 2D solids using triangular mesh. *Computational Mechanics*, 2009.

[24] T. Nguyen-Thoi, G.R. Liu, H.C. Vu-Do, and H. Nguyen-Xuan. A face-based smoothed finite element method (FS-FEM) for visco-elastoplastic analyses of 3D solids using tetrahedral mesh. *Computer Methods in Applied Mechanics and Engineering*, 198(41):3479 – 3498, 2009.

[25] T. Nguyen-Thoi, H. C. Vu-Do, T. Rabczuk, and H. Nguyen-Xuan. A node-based smoothed finite element method (NS-FEM) for upper bound solution to visco-elastoplastic analyses of solids using triangular and tetrahedral meshes. *Computer Methods in Applied Mechanics and Engineering*, 2010.

[26] C. Thai-Hoang, N. Nguyen-Thanh, H. Nguyen-Xuan, and T. Rabczuk. An alternative alpha finite element method with discrete shear gap technique for analysis of laminated composite plates. *Applied Mathematics and Computation*, 2011.

[27] Stéphane P. A. Bordas, Timon Rabczuk, Nguyen Xuan Hung, Vinh Phu Nguyen, Sundararajan Natarajan, Tino Bog, Do Minh Quan, and Nguyen Vinh Hiep. Strain smoothing in FEM and XFEM. *Computers and Structures*, 88(23-24):1419–1443, 2010.

[28] Stéphane P. A. Bordas, Sundararajan Natarajan, Pierre Kerfriden, Charles Edward Augarde, D. Roy Mahapatra, Timon Rabczuk, and Stefano Dal Pont. On the performance of strain smoothing for quadratic and enriched finite element approximations (XFEM/GFEM/PUFEM). *International Journal for Numerical Methods in Engineering*, 86(4-5):637–666, 2011.

[29] Y. Jiang, T. E. Tay, L. Chen, and X. S. Sun. An edge-based smoothed XFEM for fracture in composite materials. *International Journal of Fracture*, 2013.

[30] Amrita Francis, Alejandro Ortiz-Bernardin, Stéphane P. A. Bordas, and Sundararajan Natarajan. Linear smoothed polygonal and polyhedral finite elements. *International Journal for Numerical Methods in Engineering*, 109(9):1263–1288, 2017. nme.5324.

[31] M. Surendran, Sundararajan Natarajan, Stéphane P. A. Bordas, and G. S. Palani. Linear smoothed extended finite element method. *International Journal for Numerical Methods in Engineering*, 112(12):1733–1749, 2017.

[32] W. Zeng and G. R. Liu. Smoothed Finite Element Methods (S-FEM): An Overview and Recent Developments. *Archives of Computational Methods in Engineering*, 25(2):397–435, 2018.

[33] M. T. Duong. Hyperelastic Modeling and Soft-Tissue Growth Integrated with the Smoothed Finite Element Method-SFEM. *PhD Thesis,RWTH Aachen University, Germany*, 2014.

[34] Chang Kye Lee, L. Angela Mihai, Jack S. Hale, Pierre Kerfriden, and Stéphane P. A. Bordas. Strain smoothing for compressible and nearly-incompressible finite elasticity. *Computers and Structures*, 182:540–555, 2017.

[35] N. Moës, M. Cloirec, P. Cartraud, and J. F. Remacle. A computational approach to handle complex microstructure geometries. *Computer Methods in Applied Mechanics and Engineering*, 192:3163–3177, 2003.

[36] I. Babuška and U. Banerjee. Stable Generalized Finite Element Method (SGFEM). *Computer Methods in Applied Mechanics and Engineering*, 201-204:91–111, 2012.

[37] Thomas Peter Fries. A corrected XFEM approximation without problems in blending elements. *International Journal for Numerical Methods in Engineering*, 75(5):503–532, 2008.

[38] N. Sukumar, D. L. Chopp, N. Moës, and T. Belytschko. Modeling holes and inclusions by level sets in the extended finite-element method. *Computer Methods in Applied Mechanics and Engineering*, 2001.

[39] Éric Béchet, Hans Minnebo, Nicolas Moës, and B Burgardt. Improved implementation and robustness study of the X-FEM for stress analysis around cracks. *International Journal for Numerical Methods in Engineering*, 64(8):1033–1056, 2005.

[40] Patrick Laborde, Julien Pommier, Yves Renard, and Michel Salaün. High-order extended finite element method for cracked domains. *International Journal for Numerical Methods in Engineering*, 64(3):354–381, 2005.

[41] Giulio Ventura. On the elimination of quadrature subcells for discontinuous functions in the eXtended Finite-Element Method. *International Journal for Numerical Methods in Engineering*, 66(5):761–795, 2006.

[42] Giulio Ventura, Robert Gracie, and Ted Belytschko. Fast integration and weight function blending in the extended finite element method. *International journal for numerical methods in engineering*, 77(1):1–29, 2009.

[43] Sundararajan Natarajan, Stéphane Bordas, and D Roy Mahapatra. Numerical integration over arbitrary polygonal domains based on Schwarz–Christoffel conformal mapping. *International Journal for Numerical Methods in Engineering*, 80(1):103–134, 2009.

[44] S. E. Mousavi and N. Sukumar. Numerical integration of polynomials and discontinuous functions on irregular convex polygons and polyhedrons. *Computational Mechanics*, 47(5):535–554, 2011.

[45] Y. Sudhakar and Wolfgang A. Wall. Quadrature schemes for arbitrary convex/concave volumes and integration of weak form in enriched partition of unity methods. *Computer Methods in Applied Mechanics and Engineering*, 258:39 – 54, 2013.

[46] E. B. Chin, J. B. Lasserre, and N. Sukumar. Modeling crack discontinuities without element-partitioning in the extended finite element method. *International Journal for Numerical Methods in Engineering*, 110(11):1021–1048, 2016.

[47] Stéphane Bordas and Marc Duflot. Derivative recovery and a posteriori error estimate for extended finite elements. *Computer Methods in Applied Mechanics and Engineering*, 196(35-36):3381–3399, 2007.

[48] Stéphane P. A. Bordas, Marc Duflot, and Phong Le. A simple error estimator for extended finite elements. *Communications in Numerical Methods in Engineering*, 24(11):961–971, 2008.

[49] Michael S. Floater. Mean value coordinates. *Computer Aided Geometric Design*, 20(1):19–27, 2003.

[50] Joaquín Moraleda, Javier Segurado, and Javier LLorca. Finite deformation of incompressible fiber-reinforced elastomers: A computational micromechanics approach. *Journal of the Mechanics and Physics of Solids*, 57(9):1596 – 1613, 2009.