An element-based formulation for ES-FEM and FS-FEM models for implementation in standard solid mechanics finite element codes for 2D and 3D static analysis

Daniele Colombo, Slah Drira, Ralf Frotscher, Manfred Staat

Received: 3 January 2022 | Accepted: 12 September 2022

DOI: 10.1002/nme.7126

RESEARCH ARTICLE

WILEY

An element-based formulation for ES-FEM and FS-FEM models for implementation in standard solid mechanics finite element codes for 2D and 3D static analysis

Daniele Colombo[1] | **Slah Drira**[2] | **Ralf Frotscher**[3] | **Manfred Staat**[3]

[1]Département Physique Numérique des Milieux Poreux, IFP Energies Nouvelles, Rueil-Malmaison, France

[2]Laboratoire d'informatique et de mécanique appliquées à la construction - IMAC, Ecole Polytechnique Fédérale de Lausanne – EPFL, Lausanne, Switzerland

[3]Faculty of Medical Engineering and Applied Mathematics, FH Aachen University of Applied Sciences, Jülich, Germany

Correspondence
Daniele Colombo, IFP Energies Nouvelles, 1-4 avenue de Bois-Préau, 92852 Rueil-Malmaison, France.
Email: daniele.colombo@ifpen.fr

Abstract

Edge-based and face-based smoothed finite element methods (ES-FEM and FS-FEM, respectively) are modified versions of the finite element method allowing to achieve more accurate results and to reduce sensitivity to mesh distortion, at least for linear elements. These properties make the two methods very attractive. However, their implementation in a standard finite element code is nontrivial because it requires heavy and extensive modifications to the code architecture. In this article, we present an element-based formulation of ES-FEM and FS-FEM methods allowing to implement the two methods in a standard finite element code with no modifications to its architecture. Moreover, the element-based formulation permits to easily manage any type of element, especially in 3D models where, to the best of the authors' knowledge, only tetrahedral elements are used in FS-FEM applications found in the literature. Shape functions for non-simplex 3D elements are proposed in order to apply FS-FEM to any standard finite element.

KEYWORDS

distorted element, ES-FEM, FS-FEM, non-simplex S-FEM elements, S-FEM, smoothed finite element method

1 | INTRODUCTION

The smoothed finite element method (S-FEM) is a modified version of the standard finite element method (FEM) characterized by interesting properties making it a very attractive method. For a given mesh, S-FEM allows to achieve more accurate results than FEM, is less sensitive to mesh distortion and can be easily applied to polygonal and polyhedral elements, since element shape function values have to be known at some particular points in an element and no analytical form is required.[1-3]

From a practical point of view, S-FEM is based on the smoothed strain field that is calculated by averaging standard FEM compatible strain field over so-called smoothing domains. These domains are obtained by subdividing an FEM mesh into nonoverlapping domains. Construction of smoothing domains can be accomplished in several ways. The features and properties of the resulting S-FEM method are strongly influenced by the adopted construction procedure. Four S-FEM

methods are currently available in the literature, each one using a different geometrical mesh entity to build the smoothing domains:[1,2] CS-FEM (cell-based S-FEM, using the whole element),[4] NS-FEM (nodal-based S-FEM, using mesh nodes),[5] ES-FEM (edge-based S-FEM, using 2D elements edges)[6-9] for 2D models only, and FS-FEM (face-based S-FEM, using 3D element faces) for 3D models only.[10]

Among these S-FEM methods, the ES-FEM (2D models) and the FS-FEM (3D models) are the most attractive because they achieve very accurate results and high mesh convergence rates compared to FEM.[2] However, their implementation in a standard finite element code is not straightforward. Discretization of solid mechanics equations by means of the S-FEM method results in a set of algebraic equations with the same form of the one obtained by the standard finite element procedure. The unknowns are still associated to mesh nodes and they are exactly the same as in the case of the standard finite element discretization. However, the stiffness matrix is no more calculated on an element basis but on a smoothing domain basis: a stiffness matrix is calculated for each smoothing domain and is assembled in the global stiffness matrix of the mesh by an assembly process similar to the one used in standard FEM.

The fact that in ES-FEM and FS-FEM methods a smoothing domain extends over two adjacent elements makes the implementation of S-FEM in a standard FEM code a challenging task, since extensive and substantial modifications of the code are required in order to account for the new entities that are used to calculate the global stiffness matrix contributions. In a standard FE code, element stiffness matrix calculations and their assembly in the global stiffness matrix of the model are done element by element. A new element formulation is added to the code by implementing all the calculations for one element only. In the ES-FEM and FS-FEM methods, stiffness matrix calculations are done in one smoothing domain, which extends over two adjacent elements. All the degrees of freedom of the two elements are used and information from the two adjacent elements are necessary. Therefore, it is impossible to implement an ES-FEM or FS-FEM element by simply adding a new element in a standard FE code because, due to the code architecture, this element has no access to information from the adjacent elements and only the degrees of freedom of the element are used for the calculation of the stiffness matrix.

In this article, an element-based formulation of ES-FEM and FS-FEM methods is presented. This formulation allows the implementation of ES-FEM and FS-FEM methods in a standard finite element code by using the standard procedure usually adopted to add a new standard FEM element. This is achieved by expressing the smoothing domain stiffness matrix as the sum of the contributions coming from two adjacent elements containing the smoothing domain and from an additional "auxiliary element" associated to each smoothing domain. Such a strategy permits to execute all the smoothing domain stiffness matrix calculations on an element basis, while avoiding any modification to the standard finite element code architecture.

The possibility to implement the element-based formulation in a standard FE code depends on the capabilities offered by the user interface of the code. For an open-source code, the user has access to any part of the code and she or he can freely implement the formulation. For a closed-source code, the user can successfully implement the formulation if the interface to the code allows to:

- Define additional elements starting from a given mesh in order to define the auxiliary elements used by the element-based formulation.
- Implement stiffness matrix and strain calculations for a user-defined element.
- Exchange information between the user defined elements in order to permit to one element to get data from the adjacent elements.

For example, the widely used Abaqus solver offers to the user powerful tools to extend code capabilities. The user could use Abaqus scripting to create auxiliary elements for a given input mesh, could define an Abaqus user element to implement the stiffness matrix calculation for the element-based formulation for the elements and auxiliary elements of the mesh and could use Abaqus scripting or Abaqus user subroutines to exchange data between the elements. Data used by the element-based formulation could be stored in state variables of the user-defined elements.

It is clear that if the possibility to implement the element-based S-FEM formulation in a closed-source standard FE code depends on the user interface capabilities of the code, the standard S-FEM formulation cannot be implemented because the user has no mean to modify the code architecture through the user interface of the code, independently of its capabilities. These modifications are necessary for the standard S-FEM formulation, as discussed above.

This article is organized as follows. In Section 2, the governing equations of the static solid mechanics problems are briefly reviewed. In Section 3, the basic equations produced by the standard finite element discretization are presented. In Section 4, these equations are compared to the equations produced by S-FEM discretization in order to highlight the ideas underlying the element-based formulation for ES-FEM and FS-FEM methods, which is presented in detail in Section 5 with the shape function formulation for non-simplex elements. Finally, the element-based formulation is applied to a 2D and a 3D case in Section 6 and the results are compared to FEM results and reference solutions available in literature.

2 | GOVERNING EQUATIONS

We consider a static solid mechanic problem defined in a domain $\Omega \in \mathbb{R}^m$ subjected to external body forces $\boldsymbol{b} = \begin{bmatrix} b_1 & \dots & b_m \end{bmatrix}^T$. Prescribed displacements $\boldsymbol{u}_\Gamma = \begin{bmatrix} u_{\Gamma_1} & \dots & u_{\Gamma_m} \end{bmatrix}^T$ and external surface forces $\boldsymbol{t}_\Gamma = \begin{bmatrix} t_{\Gamma_1} & \dots & t_{\Gamma_m} \end{bmatrix}^T$ are applied on the displacement and traction boundary on $\Gamma_u \in \Gamma = \partial\Omega$ and $\Gamma_t \in \Gamma = \partial\Omega$, respectively.

The equilibrium equation can be written as:[2,11,12]

$$\frac{\partial \sigma_{ij}}{\partial x_j} + b_i = 0, \quad i,j = 1, \dots, m \text{ for } x_j \in \Omega \setminus \partial\Omega, \tag{1}$$

where σ_{ij} are the stress tensor components and $\boldsymbol{x} = [x_1 \ \dots \ x_m]^T$ is the position vector defining point position. Equation (1) can be equivalently written in matrix form as follows:

$$\boldsymbol{L}_d^T \boldsymbol{\sigma} + \boldsymbol{b} = \boldsymbol{0}, \tag{2}$$

where $\boldsymbol{L}_d$ is the matrix of differential operators given by:

$$\boldsymbol{L}_d = \begin{bmatrix} \frac{\partial}{\partial x_1} & 0 \\ 0 & \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} \end{bmatrix} \text{ for 2D case } (m = 2),$$

$$\boldsymbol{L}_d = \begin{bmatrix} \frac{\partial}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial}{\partial x_3} \\ 0 & \frac{\partial}{\partial x_3} & \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_3} & 0 & \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} & 0 \end{bmatrix} \text{ for 3D case } (m = 3), \tag{3}$$

and $\boldsymbol{\sigma}$ is the stress vector given by:

$$\begin{aligned} \boldsymbol{\sigma} &= [\sigma_{11} \ \sigma_{22} \ \sigma_{12}]^T \text{ for 2D case } (m = 2), \\ \boldsymbol{\sigma} &= [\sigma_{11} \ \sigma_{22} \ \sigma_{33} \ \sigma_{23} \ \sigma_{13} \ \sigma_{12}]^T \text{ for 3D case } (m = 3). \end{aligned} \tag{4}$$

The strain field is defined by means of the compatibility equation under the assumption of small displacements:[2,11,12]

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \quad i,j = 1, \dots, m \text{ for } x_j \in \Omega \setminus \partial\Omega, \tag{5}$$

where u_i are the components of the displacement vector $\boldsymbol{u} = [u_1 \ \dots \ u_m]^T$. Equation (5) can be equivalently written in matrix form as:

$$\boldsymbol{\varepsilon} = \boldsymbol{L}_d \boldsymbol{u}, \tag{6}$$

where $\boldsymbol{\varepsilon}$ is the strain vector given by:

$$\begin{aligned}\boldsymbol{\varepsilon} &= [\varepsilon_{11}\ \varepsilon_{22}\ 2\varepsilon_{12}]^T = [\varepsilon_{11}\ \varepsilon_{22}\ \gamma_{12}]^T \ \text{ for 2D case } \ (m=2),\\ \boldsymbol{\varepsilon} &= [\varepsilon_{11}\ \varepsilon_{22}\ \varepsilon_{33}\ 2\varepsilon_{23}\ 2\varepsilon_{13}\ 2\varepsilon_{12}]^T = [\varepsilon_{11}\ \varepsilon_{22}\ \varepsilon_{33}\ \gamma_{23}\ \gamma_{13}\ \gamma_{12}]^T \ \text{ for 3D case } \ (m=3).\end{aligned} \tag{7}$$

The constitutive equation relating stress and strain is given by the generalized Hooke's law:[2,11,12]

$$\sigma_{ij} = c_{ijkl}\varepsilon_{kl}, \quad i,j,k,l \in [1,m] \tag{8}$$

or in equivalent matrix form:

$$\boldsymbol{\sigma} = \boldsymbol{C}\boldsymbol{\varepsilon}. \tag{9}$$

The fourth-order tensor C_{ijkl} refers to material stiffness moduli.
Finally, the boundary conditions read:

$$\begin{aligned}\boldsymbol{u} &= \boldsymbol{u}_{\Gamma_u} \quad \text{on } \Gamma_u,\\ \boldsymbol{L}_n^T\boldsymbol{\sigma} &= \boldsymbol{t}_{\Gamma_t} \quad \text{on } \Gamma_t,\end{aligned} \tag{10a}$$

where $\boldsymbol{L}_n(\boldsymbol{x})$ is the matrix of the outward normals $\boldsymbol{n}(\boldsymbol{x})$ to the domain boundary $\partial\Omega$ given by:

$$\boldsymbol{L}_n(\boldsymbol{x}) = \begin{bmatrix} n_x(\boldsymbol{x}) & 0 \\ 0 & n_y(\boldsymbol{x}) \\ n_y(\boldsymbol{x}) & n_x(\boldsymbol{x}) \end{bmatrix} \text{ for 2D case, } \quad \boldsymbol{n}(\boldsymbol{x}) = \begin{bmatrix} n_x(\boldsymbol{x}) & n_y(\boldsymbol{x}) \end{bmatrix}^T,$$

$$\boldsymbol{L}_n(\boldsymbol{x}) = \begin{bmatrix} n_x(\boldsymbol{x}) & 0 & 0 \\ 0 & n_y(\boldsymbol{x}) & 0 \\ 0 & 0 & n_z(\boldsymbol{x}) \\ 0 & n_z(\boldsymbol{x}) & n_y(\boldsymbol{x}) \\ n_z(\boldsymbol{x}) & 0 & n_x(\boldsymbol{x}) \\ n_y(\boldsymbol{x}) & n_x(\boldsymbol{x}) & 0 \end{bmatrix} \text{ for 3D case, } \quad \boldsymbol{n}(\boldsymbol{x}) = \begin{bmatrix} n_x(\boldsymbol{x}) & n_y(\boldsymbol{x}) & n_z(\boldsymbol{x}) \end{bmatrix}^T. \tag{10b}$$

3 | FEM DISCRETIZATION

A finite element mesh is built on the domain Ω. The mesh is composed by n_e elements and n_n nodes. Displacement degrees of freedom are assigned to each node i:

$$\boldsymbol{d}_i = \begin{bmatrix} d_1 & \dots & d_m \end{bmatrix}^T. \tag{11}$$

The global vector of the degrees of freedom of the whole mesh is given by:

$$\boldsymbol{d} = \begin{bmatrix} \boldsymbol{d}_1^T & \dots & \boldsymbol{d}_{n_n}^T \end{bmatrix}^T. \tag{12}$$

The displacement field is approximated by means of element shape functions:[11-13]

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{N}(\boldsymbol{x})\,\boldsymbol{d}, \tag{13}$$

where $\boldsymbol{N}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{N}_1 & \dots & \boldsymbol{N}_{n_n} \end{bmatrix}$ is the global shape function matrix, $\boldsymbol{N}_i(\boldsymbol{x}) = N_i(\boldsymbol{x})\,\boldsymbol{I}_m$ is the shape function matrix for node i, $N_i(\boldsymbol{x})$ is the shape function for the same node, and $\boldsymbol{I}_m$ is the identity matrix of size m. As a consequence of its definition, matrix $\boldsymbol{N}(\boldsymbol{x})$ is coarse.

The strain field is therefore approximated as follows (Equations 6 and 13):[11-13]

$$\boldsymbol{\varepsilon} = \boldsymbol{L}_d \boldsymbol{N}(\boldsymbol{x})\, \boldsymbol{d} = \boldsymbol{B}(\boldsymbol{x})\, \boldsymbol{d}, \tag{14}$$

where $\boldsymbol{B}(\boldsymbol{x}) = \boldsymbol{L}_d \boldsymbol{N}(\boldsymbol{x})$ is the strain-displacement matrix. For the following discussion, it is worth rewriting $\boldsymbol{B}(\boldsymbol{x})$ in a form where contributions of each node appear explicitly:

$$\boldsymbol{B}(\boldsymbol{x}) = \left[\boldsymbol{B}_1(\boldsymbol{x}) \ \ldots \ \boldsymbol{B}_{n_n}(\boldsymbol{x})\right], \tag{15}$$

where $\boldsymbol{B}_i(\boldsymbol{x}) = \boldsymbol{L}_d \boldsymbol{N}_i(\boldsymbol{x})$ is the strain-displacement matrix for node i.

FE method solves equilibrium equation (1) in weak form adopting the above discretized displacement and strain fields (Equations 13 and 14). The resulting system of algebraic equations can be written in matrix form as follow:[11-13]

$$\boldsymbol{K}\boldsymbol{d} = \boldsymbol{f}, \tag{16}$$

where $\boldsymbol{K}$ is the global stiffness matrix given by:

$$\boldsymbol{K} = \int_{\Omega} \boldsymbol{B}^T(\boldsymbol{x})\, \boldsymbol{C}\boldsymbol{B}(\boldsymbol{x})\, d\Omega, \tag{17}$$

and $\boldsymbol{f}$ is the vector of nodal forces given by:

$$\boldsymbol{f} = \int_{\Omega} \boldsymbol{N}^T(\boldsymbol{x})\, \boldsymbol{b} d\Omega + \int_{\Gamma_t} \boldsymbol{N}^T(\boldsymbol{x})\, \boldsymbol{t}_{\Gamma} d\Gamma. \tag{18}$$

For the following discussion, it is worth noting that contributions of each finite element to the global stiffness matrix can be easily separated. Indeed the global stiffness matrix is very coarse, which, as already mentioned, is due to the fact that matrix $\boldsymbol{N}(\boldsymbol{x})$ is coarse. Equation (17) can be easily rewritten on an element-basis:

$$\boldsymbol{K} = \sum_{i=1}^{n_e} \int_{\Omega_i} \boldsymbol{B}^T(\boldsymbol{x})\, \boldsymbol{C}\boldsymbol{B}(\boldsymbol{x})\, d\Omega. \tag{19}$$

Considering the fact that integration is now carried out on a single element Ω_i and that only the shape functions of the element nodes contribute to the integrand, the global stiffness matrix can be equivalently rewritten as the sum of the stiffness matrices of each element Ω_i ("summation" here means "assembly" since the involved matrices have different size, see below for details):

$$\boldsymbol{K} = \sum_{i=1}^{n_e} \boldsymbol{K}_{\Omega_i}, \tag{20}$$

where the element stiffness matrix is defined as:

$$\boldsymbol{K}_{\Omega_i} = \int_{\Omega_i} \boldsymbol{B}_{\Omega_i}^T(\boldsymbol{x})\, \boldsymbol{C}\boldsymbol{B}_{\Omega_i}(\boldsymbol{x})\, d\Omega. \tag{21}$$

Strain-displacement matrix $\boldsymbol{B}_{\Omega_i}(\boldsymbol{x})$ is the restriction to element Ω_i of the global strain-displacement matrix $\boldsymbol{B}(\boldsymbol{x})$ (Equation 15) and is given by:

$$\boldsymbol{B}_{\Omega_i}(\boldsymbol{x}) = \left[\boldsymbol{B}_1(\boldsymbol{x}) \ \ldots \ \boldsymbol{B}_{n_{\Omega_i}}(\boldsymbol{x})\right], \tag{22}$$

where n_{Ω_i} is the number of nodes of element Ω_i.

The size of element stiffness matrix $\boldsymbol{K}_{\Omega_i}$ is $(mn_{\Omega_i}) \times (mn_{\Omega_i})$, where n_{Ω_i} is the number of nodes of element Ω_i and m is the dimension of domain Ω. On the contrary, the size of the global stiffness matrix $\boldsymbol{K}$ is $(mn_n) \times (mn_n)$, where n_n is the

number of nodes of the model. Strictly speaking, the summation symbol in Equation (20) is used improperly, because the size of the matrices to sum is not the same. Equation (20) is just a convenient way to concisely express the assembly procedure commonly used in finite element codes: the element stiffness matrix is first calculated for an element and then added to the global stiffness matrix by selecting the rows and columns corresponding to the degrees of freedom of the finite element.[12,13]

The element-based assembly procedure will be exploited later to define the S-FEM implementation strategy.

4 | S-FEM DISCRETIZATION

S-FEM models work on the so-called smoothing domains obtained by subdividing the finite element mesh of domain Ω into several nonoverlapping and no gap subdomains. A domain can be contained entirely inside one element or can extend over two adjacent elements, depending on the type of S-FEM method. For each smoothing domain Ω_k, a smoothed strain is calculated at any point $\boldsymbol{x}_k \in \Omega_k$ by means of the strain/gradient smoothing technique:[2]

$$\boldsymbol{\varepsilon}_{\Omega_k}(\boldsymbol{x}_k) = \int_{\Omega_k} \boldsymbol{L}_d(\boldsymbol{x})\, \boldsymbol{u}(\boldsymbol{x})\, W(\boldsymbol{x}_k - \boldsymbol{x})\, d\Omega, \tag{23}$$

where $W(\boldsymbol{x}_k - \boldsymbol{x})$ is a smoothing function defined as:

$$W(\boldsymbol{x}_k - \boldsymbol{x}) = \begin{cases} \frac{1}{S_{\Omega_k}}, & \text{for } \boldsymbol{x} \in \Omega_k, \\ 0, & \text{for } \boldsymbol{x} \notin \Omega_k, \end{cases} \tag{24}$$

where S_{Ω_k} is the geometrical size of domain Ω_k (its area in 2D and its volume in 3D).

Because the displacement field $\boldsymbol{u}(\boldsymbol{x})$ is assumed continuous on $\partial\Omega_k$ and the smoothing function $W(\boldsymbol{x}_k - \boldsymbol{x})$ is differentiable over Ω_k, by applying Green's divergence theorem, the smoothed strain can be rewritten equivalently as:[2]

$$\boldsymbol{\varepsilon}_{\Omega_k}(\boldsymbol{x}_k) = \boldsymbol{\varepsilon}_{\Omega_k} = \frac{1}{S_{\Omega_k}} \int_{\partial\Omega_k} \boldsymbol{L}_n(\boldsymbol{x})\, \boldsymbol{u}(\boldsymbol{x})\, d\Gamma. \tag{25}$$

In S-FEM, it is assumed that the strain $\boldsymbol{\varepsilon}_{\Omega_k}$ in the smoothing domain Ω_k is constant and equal to $\boldsymbol{\varepsilon}_{\Omega_k}(\boldsymbol{x}_k)$ given by Equation (25).[2,6,10,14]

The integral in Equation (24) is restricted to the smoothing domain boundary $\partial\Omega_k$.

The discretization of the displacement field is identical to the one used in the FE method (Equation 13). The smoothed strain (Equation 24) can therefore be rewritten in matrix form as:[2,6,10,14]

$$\boldsymbol{\varepsilon}_{\Omega_k} = \boldsymbol{B}_{\Omega_k} \boldsymbol{d}, \tag{26}$$

where $\boldsymbol{B}_{\Omega_k}$ is the strain-displacement matrix associated to smoothing domain Ω_k, given by:

$$\boldsymbol{B}_{\Omega_k} = \left[\boldsymbol{B}_1^{\Omega_k} \ \dots \ \boldsymbol{B}_r^{\Omega_k} \right], \tag{27}$$

where r is the total number of nodes of the elements containing the smoothing domain. The contribution of each node is given by $\boldsymbol{B}_i^{\Omega_k}$, which is the strain-displacement matrix of node i associated to smoothing domain Ω_k, given by:

$$\boldsymbol{B}_i^{\Omega_k} = \frac{1}{S_{\Omega_k}} \int_{\partial\Omega_k} \boldsymbol{L}_n(\boldsymbol{x})\, \boldsymbol{N}_i(\boldsymbol{x})\, d\Gamma. \tag{28}$$

Matrix $\boldsymbol{B}_i^{\Omega_k}$ has the same structure as matrix $\boldsymbol{N}_i(\boldsymbol{x})$:[2,10]

$$\boldsymbol{B}_i^{\Omega_k} = \begin{bmatrix} b_{i1}^{\Omega_k} & 0 \\ 0 & b_{i2}^{\Omega_k} \\ b_{i2}^{\Omega_k} & b_{i1}^{\Omega_k} \end{bmatrix} \text{ for 2D case}, \quad \boldsymbol{b}_i^{\Omega_k} = \left[b_{i1}^{\Omega_k} \ b_{i2}^{\Omega_k} \right]^T,$$

$$\boldsymbol{B}_i^{\Omega_k} = \begin{bmatrix} b_{i1}^{\Omega_k} & 0 & 0 \\ 0 & b_{i2}^{\Omega_k} & 0 \\ 0 & 0 & b_{i3}^{\Omega_k} \\ 0 & b_{i3}^{\Omega_k} & b_{i2}^{\Omega_k} \\ b_{i3}^{\Omega_k} & 0 & b_{i1}^{\Omega_k} \\ b_{i2}^{\Omega_k} & b_{i1}^{\Omega_k} & 0 \end{bmatrix} \text{ for 3D case}, \quad \boldsymbol{b}_i^{\Omega_k} = \left[b_{i1}^{\Omega_k} \; b_{i2}^{\Omega_k} \; b_{i3}^{\Omega_k} \right]^T, \tag{29}$$

where

$$b_{ij}^{\Omega_k} = \frac{1}{S_{\Omega_k}} \int_{\partial\Omega_k} n_j(\boldsymbol{x}) N_i(\boldsymbol{x}) \, d\Gamma. \tag{30}$$

If linear elements are used, the shape functions are linear and the integration in Equation (30) is trivial:[2,10]

$$b_{ij}^{\Omega_k} = \frac{1}{S_{\Omega_k}} \sum_{p=1}^{n_\Gamma} n_j(\boldsymbol{x}_{\text{Gp}}) N_i(\boldsymbol{x}_{\text{Gp}}) \, l_p, \tag{31}$$

where n_Γ is the number of segments (2D case) or planar faces (3D case) constituting the smoothing domain boundary $\partial\Omega_k$, $\boldsymbol{x}_{\text{Gp}}$ is the position of the pth segment mid-point (2D case) or the pth planar face centroid (3D case), and l_p is its length (2D case) or its area (3D case).

It can be demonstrated that a smoothed Galerkin weak form of the equilibrium equation (1) based on smoothed strain can be written as done in FE case, leading to the following algebraic equation system:[2]

$$\boldsymbol{K}_{\text{SFEM}} \boldsymbol{d} = \boldsymbol{f}. \tag{32}$$

Equation (32) is formally identical to Equation (16) obtained for FEM discretization. Vector $\boldsymbol{f}$ is given by Equation (18) as in the FEM case. The global stiffness matrix $\boldsymbol{K}_{\text{SFEM}}$ is formally obtained by exactly the same assembly procedure as used in FEM but based on the contributions of each smoothing domain (Equation 20):

$$\boldsymbol{K}_{\text{SFEM}} = \sum_{k=1}^{n_\Omega} \boldsymbol{K}_{\Omega_k \text{SFEM}}, \tag{33}$$

where n_Ω is the number of smoothing domains. The smoothing domain stiffness matrix $\boldsymbol{K}_{\Omega_k \text{SFEM}}$ is given by the following equation, which is formally identical to Equation (17) used in the FEM case:

$$\boldsymbol{K}_{\Omega_k \text{SFEM}} = \int_{\Omega_k} \boldsymbol{B}_{\Omega_k}^T \boldsymbol{C} \boldsymbol{B}_{\Omega_k} \, d\Omega = \boldsymbol{B}_{\Omega_k}^T \boldsymbol{C} \boldsymbol{B}_{\Omega_k} S_{\Omega_k}. \tag{34}$$

Equation (34) shows that the stiffness matrix calculation does not need integration and it can be computed directly from the values of $\boldsymbol{B}_{\Omega_k} = \left[\boldsymbol{B}_1^{\Omega_k} \; \ldots \; \boldsymbol{B}_r^{\Omega_k} \right]$ and $\boldsymbol{C}$ matrices. Equation (31) shows that, if linear shape functions are used on the smoothing domain boundary $\partial\Omega_k$, integration of terms in matrix $\boldsymbol{B}_i^{\Omega_k}$ are done directly in the physical space and no isoparametric transformation and no evaluation of the Jacobian of the associated transformation matrix are needed, contrary to FEM (see Equation 21). This makes S-FEM method more insensitive to mesh distortion,[2] as it will be shown in in Section 6. Formulation of linear shape functions on the smoothing domain boundary $\partial\Omega_k$ for non-simplex elements will be done in Section 5.

Extension of the S-FEM formulation to large deformations and geometrically nonlinear problems can be done with a similar procedure to the one used in standard FEM, as detailed in References 2 and 10. A straightforward application of S-FEM formulation to this class of problems can be done by adopting an updated Lagrangian approach.[11] Discussion of this topic is out of the scope of this article and interested readers will find all the details in References 2, 10, and 11.

5 | AN ELEMENT-BASED FORMULATION FOR ES-FEM AND FS-FEM METHODS

Equations (20) and (33) are formally identical, except that the assembly procedure is carried out on smoothing domains for S-FEM, whereas it is carried out on finite elements for FEM. The architecture of finite element codes is usually based on elements and the calculations for the element stiffness matrices $\boldsymbol{K}_{\Omega_i}$ (Equation 21) are carried out element by element without the need of exchanging data between elements. Thus the parallelization of these calculations is straightforward.

The same approach can be used for S-FEM. However, all S-FEM methods except CS-FEM create smoothing domains that extend over two adjacent elements (Figures 1 and 2 for ES-FEM and FS-FEM methods, respectively) and data from the two elements are needed to calculate the smoothing domain stiffness matrix $\boldsymbol{K}_{\Omega_k \mathrm{SFEM}}$ in Equation (33). Consequently, contrary to the FEM, the degrees of freedom of all the nodes of the two adjacent elements are related in $\boldsymbol{K}_{\Omega_k \mathrm{SFEM}}$. Therefore the implementation of NS-FEM, ES-FEM and FS-FEM methods in an existing finite element code by means of Equation (33) requires modifications to the code architecture since FEM codes usually adopt an element-based formulation. These modifications are very complex and time-consuming.

In the following paragraphs, it will be shown that S-FEM equations can be equivalently rewritten using an element-based formulation for ES-FEM and FS-FEM methods. This equivalent formulation allows easy implementation of the S-FEM method into an existing FEM code because it does not require modifications of code architecture.

5.1 | Element-based formulation for smoothed stiffness matrix $\boldsymbol{K}_{\Omega_k \mathrm{SFEM}}$ for ES-FEM and FS-FEM methods

Equation (33) can be rewritten in a different way in order to identify the contribution of each element containing the smoothing domain.

We consider a smoothing domain Ω_k extending over two adjacent elements A and B. Some nodes are shared between these two elements. In 2D, the ES-FEM method builds the smoothing domain at the common edge between the two elements. Consequently, the two nodes defining the edge are shared between element A and B. In 3D, FS-FEM method builds the smoothing domain on the common face between the two elements. Therefore, for commonly used finite elements,

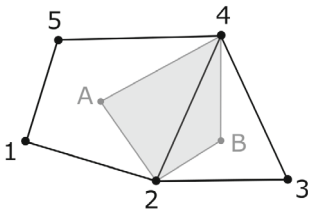

FIGURE 1 Example of construction of a smoothing domain for ES-FEM method. The smoothing domain is built on the edge defined by nodes 2 and 4 shared between the quadrangular and triangular elements by connecting these two nodes with the element centroids A and B.

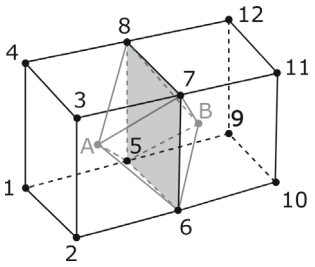

FIGURE 2 Example of construction of a smoothing domain for FS-FEM method. The smoothing domain is built on the face defined by nodes 5, 6, 7, and 8 shared between the two hexahedral elements by connecting these four nodes with the element centroids A and B.

three (triangular face) or four (quadrangular face) nodes are shared between element A and B. For polyhedral elements, a greater number of nodes is shared.

By rearranging the node order, contributions to the strain-displacement matrix $\boldsymbol{B}_{\Omega_k}$ of the nodes belonging to one element only and of the shared nodes can be separated in Equation (27):

$$\boldsymbol{B}_{\Omega_k} = \left[\overbrace{\boldsymbol{B}_1^{\Omega_k} \ \dots \ \boldsymbol{B}_{n_A}^{\Omega_k}}^{\text{element A only}} \middle| \overbrace{\boldsymbol{B}_{n_A+1}^{\Omega_k} \ \dots \ \boldsymbol{B}_{n_A+n_S}^{\Omega_k}}^{\text{shared A and B}} \middle| \overbrace{\boldsymbol{B}_{n_A+n_S+1}^{\Omega_k} \ \dots \ \boldsymbol{B}_{n_A+n_S+n_B}^{\Omega_k}}^{\text{element B only}} \right], \tag{35}$$

where n_A and n_B are the number of nodes belonging only to element A and B, respectively, and n_S is the number of shared nodes between the two elements.

We denote $\boldsymbol{B}_{\Omega_k}|_A$, $\boldsymbol{B}_{\Omega_k}|_B$, and $\boldsymbol{B}_{\Omega_k}|_S$ the smoothing domain strain-displacement matrix $\boldsymbol{B}_{\Omega_k}$ restricted to the nodes belonging only to elements A and B and shared between element A and B, respectively:

$$\begin{aligned} \boldsymbol{B}_{\Omega_k}|_A &= \left[\boldsymbol{B}_1^{\Omega_k} \ \dots \ \boldsymbol{B}_{n_A}^{\Omega_k} \right], \\ \boldsymbol{B}_{\Omega_k}|_S &= \left[\boldsymbol{B}_{n_A+1}^{\Omega_k} \ \dots \ \boldsymbol{B}_{n_A+n_S}^{\Omega_k} \right], \\ \boldsymbol{B}_{\Omega_k}|_B &= \left[\boldsymbol{B}_{n_A+n_S+1}^{\Omega_k} \ \dots \ \boldsymbol{B}_{n_A+n_S+n_B}^{\Omega_k} \right]. \end{aligned} \tag{36}$$

By substituting Equation (36) in Equation (35), matrix $\boldsymbol{B}_{\Omega_k}$ can be equivalently written as:

$$\boldsymbol{B}_{\Omega_k} = \left[\boldsymbol{B}_{\Omega_k}|_A \ \boldsymbol{B}_{\Omega_k}|_S \ \boldsymbol{B}_{\Omega_k}|_B \right]. \tag{37}$$

By substituting Equation (37) into Equation (34), the smoothing domain stiffness matrix $\boldsymbol{K}_{\Omega_k}$ can be written as:

$$\boldsymbol{K}_{\Omega_k} = \left[\boldsymbol{B}_{\Omega_k}|_A \ \boldsymbol{B}_{\Omega_k}|_S \ \boldsymbol{B}_{\Omega_k}|_B \right]^T \boldsymbol{C} \left[\boldsymbol{B}_{\Omega_k}|_A \ \boldsymbol{B}_{\Omega_k}|_S \ \boldsymbol{B}_{\Omega_k}|_B \right] S_{\Omega_k} = \begin{bmatrix} \boldsymbol{K}_{\mathrm{AA}}^{\Omega_k} & \boldsymbol{K}_{\mathrm{AS}}^{\Omega_k} & \boldsymbol{K}_{\mathrm{AB}}^{\Omega_k} \\ \boldsymbol{K}_{\mathrm{SA}}^{\Omega_k} & \boldsymbol{K}_{\mathrm{SS}}^{\Omega_k} & \boldsymbol{K}_{\mathrm{SB}}^{\Omega_k} \\ \boldsymbol{K}_{\mathrm{BA}}^{\Omega_k} & \boldsymbol{K}_{\mathrm{BS}}^{\Omega_k} & \boldsymbol{K}_{\mathrm{BB}}^{\Omega_k} \end{bmatrix} S_{\Omega_k}, \tag{38}$$

where $\boldsymbol{K}_{\mathrm{IJ}}^{\Omega_k}$ is defined as:

$$\boldsymbol{K}_{\mathrm{IJ}}^{\Omega_k} = \left(\boldsymbol{B}_{\Omega_k}|_I \right)^T \boldsymbol{C} \boldsymbol{B}_{\Omega_k}|_J, \quad \text{for } I, J \in \{A, B, S\}. \tag{39}$$

Contributions of each element to $\boldsymbol{K}_{\Omega_k}$ in Equation (38) can now be easily separated:

$$\boldsymbol{K}_{\Omega_k} = \overbrace{\begin{bmatrix} \boldsymbol{K}_{\mathrm{AA}}^{\Omega_k} & \boldsymbol{K}_{\mathrm{AS}}^{\Omega_k} & \boldsymbol{0} \\ \boldsymbol{K}_{\mathrm{SA}}^{\Omega_k} & \frac{\boldsymbol{K}_{\mathrm{SS}}^{\Omega_k}}{2} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}}^{\text{element A}} + \overbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \frac{\boldsymbol{K}_{\mathrm{SS}}^{\Omega_k}}{2} & \boldsymbol{K}_{\mathrm{SB}}^{\Omega_k} \\ \boldsymbol{0} & \boldsymbol{K}_{\mathrm{BS}}^{\Omega_k} & \boldsymbol{K}_{\mathrm{BB}}^{\Omega_k} \end{bmatrix}}^{\text{element B}} + \overbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{K}_{\mathrm{AB}}^{\Omega_k} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{K}_{\mathrm{BA}}^{\Omega_k} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}}^{\text{auxiliary element}}. \tag{40}$$

In Equation (40), $\boldsymbol{K}_{\Omega_k}$ is expressed as the sum of three matrices obtained by grouping the terms that can be calculated by accounting for the nodes of one element only: element A can calculate the terms $\boldsymbol{K}_{\mathrm{AA}}^{\Omega_k}$, $\boldsymbol{K}_{\mathrm{AS}}^{\Omega_k}$, $\boldsymbol{K}_{\mathrm{SA}}^{\Omega_k}$, and $\boldsymbol{K}_{\mathrm{SS}}^{\Omega_k}$ involving the nodes of element A only and the shared nodes, and similarly element B can calculate the terms $\boldsymbol{K}_{\mathrm{BB}}^{\Omega_k}$, $\boldsymbol{K}_{\mathrm{BS}}^{\Omega_k}$, $\boldsymbol{K}_{\mathrm{SB}}^{\Omega_k}$, and $\boldsymbol{K}_{\mathrm{SS}}^{\Omega_k}$. Equation (40) clearly shows that the two terms $\boldsymbol{K}_{\mathrm{AB}}^{\Omega_k}$ and $\boldsymbol{K}_{\mathrm{BA}}^{\Omega_k}$ cannot be calculated by considering the two elements separately: these terms relate nodes belonging to element A and B only. In order to calculate these terms, a new element must be added to the mesh. We will call this element the *auxiliary element* of smoothing domain Ω_k. A detailed discussion of its properties will be done later in Section 5.3.

Submatrix $\boldsymbol{K}_{\mathrm{SS}}^{\Omega_k}$ in Equation (38) needs a special treatment. Its elements are calculated from the contributions to the shared nodes of both elements A and B. This involves data exchange between the two elements because each element can only do a partial calculation, as it will be detailed in Section 5.2.1. Both elements A and B can evaluate $\boldsymbol{K}_{\mathrm{SS}}^{\Omega_k}$. Therefore the

calculated value must be divided by two, since the contributions of elements A and B are added to obtain the final stiffness matrix $\boldsymbol{K}_{\Omega_k}$, as shown in Equation (40). Alternatively, a more effective solution, which can be implemented without additional computational costs, would be to calculate $\boldsymbol{K}_{\mathrm{SS}}^{\Omega_k}$ in only one element, avoiding to make the same calculations twice. Any of the two elements A and B can do the calculation. The choice is arbitrary and can be done during the creation of the smoothing domains at the beginning of the simulation. For example, if element A has been chosen for the calculation of $\boldsymbol{K}_{\mathrm{SS}}^{\Omega_k}$ submatrix, Equation (40) would be rewritten as:

$$\boldsymbol{K}_{\Omega_k} = \overbrace{\begin{bmatrix} \boldsymbol{K}_{\mathrm{AA}}^{\Omega_k} & \boldsymbol{K}_{\mathrm{AS}}^{\Omega_k} & \boldsymbol{0} \\ \boldsymbol{K}_{\mathrm{SA}}^{\Omega_k} & \boldsymbol{K}_{\mathrm{SS}}^{\Omega_k} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}}^{\text{element A}} + \overbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{K}_{\mathrm{SB}}^{\Omega_k} \\ \boldsymbol{0} & \boldsymbol{K}_{\mathrm{BS}}^{\Omega_k} & \boldsymbol{K}_{\mathrm{BB}}^{\Omega_k} \end{bmatrix}}^{\text{element B}} + \overbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{K}_{\mathrm{AB}}^{\Omega_k} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{K}_{\mathrm{BA}}^{\Omega_k} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}}^{\text{auxiliary element}}. \tag{41}$$

5.2 | Evaluation of $\boldsymbol{K}_{\mathrm{IJ}}^{\Omega_k}$ submatrices

The stiffness matrices of elements A and B and auxiliary element (Equation 41) are built using submatrices $\boldsymbol{K}_{\mathrm{IJ}}^{\Omega_k}$ given by Equation (39). These submatrices are calculated from strain-displacement matrices $\boldsymbol{B}_{\Omega_k}|_A$, $\boldsymbol{B}_{\Omega_k}|_B$, and $\boldsymbol{B}_{\Omega_k}|_S$, which are a restriction of strain-displacement matrix $\boldsymbol{B}_{\Omega_k}$ to the nodes belonging to elements A and B exclusively and to the shared nodes, respectively (Equation 36). Matrices $\boldsymbol{B}_{\Omega_k}|_A$, $\boldsymbol{B}_{\Omega_k}|_B$, and $\boldsymbol{B}_{\Omega_k}|_S$ are therefore built using the strain-displacement matrices $\boldsymbol{B}_i^{\Omega_k}$ of the nodes of the elements containing the smoothing domain Ω_k (Equation 28).

The elements of $\boldsymbol{B}_i^{\Omega_k}$ are calculated by means of the shape function of node i on the boundary of the smoothing domain (Equations 28,30, and 31). As a consequence:

- If node i belongs exclusively to element E (element A or element B), $\boldsymbol{B}_i^{\Omega_k}$ can be calculated by considering that element only. This is a consequence of the fact that shape function $\boldsymbol{N}_i(\boldsymbol{x})$ is zero inside the element adjacent to element E and therefore its contributions to $\boldsymbol{B}_i^{\Omega_k}$ are zero. Matrices $\boldsymbol{B}_{\Omega_k}|_A$ and $\boldsymbol{B}_{\Omega_k}|_B$ can be calculated considering only element A and B, respectively, that is no information from the adjacent element are necessary.
- If node i is a shared node, both elements A and B contribute to $\boldsymbol{B}_i^{\Omega_k}$ because the shape function $\boldsymbol{N}_i(\boldsymbol{x})$ is nonzero everywhere on the smoothing domain boundary. Therefore, the evaluation of $\boldsymbol{B}_{\Omega_k}|_S$ needs information coming from the two elements A and B.

The element-based approach used in Equations (40) and (41) can therefore be only straightforwardly applied to the evaluation of $\boldsymbol{B}_{\Omega_k}|_A$ and $\boldsymbol{B}_{\Omega_k}|_B$ and further work is needed for $\boldsymbol{B}_{\Omega_k}|_S$.

5.2.1 | Evaluation of $\boldsymbol{B}_{\Omega_k}|_S$

Let us consider a node i that is shared between elements A and B. This node contributes to $\boldsymbol{B}_{\Omega_k}|_S$ through the matrix $\boldsymbol{B}_i^{\Omega_k}$ (Equation 36). Contributions of element A and B to $\boldsymbol{B}_i^{\Omega_k}$ can be easily separated (Equation 28):

$$\boldsymbol{B}_i^{\Omega_k} = \frac{1}{S_{\Omega_k|A} + S_{\Omega_k|B}} \left(\overbrace{\int_{\partial\Omega_k|A} \boldsymbol{L}_n(\boldsymbol{x}) \boldsymbol{N}_i(\boldsymbol{x})\, d\Gamma}^{\text{element A}} + \overbrace{\int_{\partial\Omega_k|B} \boldsymbol{L}_n(\boldsymbol{x}) \boldsymbol{N}_i(\boldsymbol{x})\, d\Gamma}^{\text{element B}} \right) = \frac{1}{S_{\Omega_k|A} + S_{\Omega_k|B}} (\boldsymbol{B}_i^{\Omega_k}|A + \boldsymbol{B}_i^{\Omega_k}|B), \tag{42}$$

where $\Omega_k|A$ and $\Omega_k|B$ denote the restriction of the smoothing domain Ω_k to elements A and B, respectively. $\partial\Omega_k|A$ and $\partial\Omega_k|B$ denote the part of the boundary of smoothing domain Ω_k that is contained inside elements A and B, respectively, and $\boldsymbol{B}_i^{\Omega_k}|A$ and $\boldsymbol{B}_i^{\Omega_k}|B$ are defined as:

$$\boldsymbol{B}_i^{\Omega_k}|E = \int_{\partial\Omega_k|E} \boldsymbol{L}_n(\boldsymbol{x}) \boldsymbol{N}_i(\boldsymbol{x})\, d\Gamma \quad \text{for} \quad E \in \{A, B\}. \tag{43}$$

Equation (42) clearly shows that element-based calculation of $\boldsymbol{B}_i^{\Omega_k}$ must be carried out in three steps:

- First, each element E calculates $S_{\Omega_k|E}$ and $\boldsymbol{B}_i^{\Omega_k}|E$. No information is needed to be exchanged between the two adjacent elements. This calculation can be straightforwardly parallelized by exploiting existing FEM code capabilities, since it is carried out on an element by element basis.
- Information is exchanged between the two adjacent elements: each element E sends $S_{\Omega_k|E}$ and $\boldsymbol{B}_i^{\Omega_k}|E$ to the adjacent element. This step cannot be parallelized by exploiting the element-based parallelization of the FEM code.
- Finally, each element E calculates the value of $\boldsymbol{B}_i^{\Omega_k}$ by means of Equation (42). This step can be parallelized by exploiting existing FEM code capabilities, since it is carried out on an element by element basis.

5.3 | Auxiliary element of smoothing domain Ω_k

Equation (40) shows that an auxiliary element must be defined for each smoothing domain in order to correctly calculate its stiffness matrix.

The auxiliary element is built using, in the order, the nodes belonging to element A only and the nodes belonging to element B only, without using any of the shared nodes. It is only used to complete the stiffness matrix calculation by providing the stiffness matrices $\boldsymbol{K}_{\mathrm{AB}}^{\Omega_k}$ and $\boldsymbol{K}_{\mathrm{BA}}^{\Omega_k}$ (Equation 40). These matrices can be promptly built from the values of $\boldsymbol{B}_i^{\Omega_k}$ coming from previous calculations done by elements A and B when evaluating $\boldsymbol{K}_{\mathrm{AA}}^{\Omega_k}$, $\boldsymbol{K}_{\mathrm{AS}}^{\Omega_k}$, $\boldsymbol{K}_{\mathrm{SA}}^{\Omega_k}$, $\boldsymbol{K}_{\mathrm{BB}}^{\Omega_k}$, $\boldsymbol{K}_{\mathrm{BS}}^{\Omega_k}$, $\boldsymbol{K}_{\mathrm{SB}}^{\Omega_k}$, and $\boldsymbol{K}_{\mathrm{SS}}^{\Omega_k}$:

- Matrix $\boldsymbol{C}$ and values of the vectors $\boldsymbol{b}_i^{\Omega_k}$ are copied from elements A and B (Equations 29 and 30).
- The values $\boldsymbol{b}_i^{\Omega_k}$ are used to build the matrices $\boldsymbol{B}_i^{\Omega_k}$ (Equation 29) and then the matrices $\boldsymbol{B}_{\Omega_k}|_A$ and $\boldsymbol{B}_{\Omega_k}|_B$ (Equation 36), which are used together with matrix $\boldsymbol{C}$ to build the matrices $\boldsymbol{K}_{\mathrm{AB}}^{\Omega_k}$ and $\boldsymbol{K}_{\mathrm{BA}}^{\Omega_k}$ (equations 36 and 39) used to finally build the auxiliary element stiffness matrix. This step can be straightforwardly parallelized by exploiting existing FEM code capabilities, since it is carried out on an element by element basis.

The geometry of the auxiliary element is never used in the calculations, which makes its construction straightforward for any type of adjacent elements A and B. It is important to point out the fact that auxiliary elements are not used for any other calculation. They are not used in post-processing operations and they can therefore be omitted when displaying simulation results.

5.4 | Calculation of strain tensor for smoothing domain Ω_k

Strain tensor is assumed to be constant over a smoothing domain Ω_k (Section 4). Its value is given by Equation (26), which can be rewritten as follows:

$$\boldsymbol{\varepsilon}_{\Omega_k} = \left[\boldsymbol{B}_{\Omega_k}|_A \; \boldsymbol{B}_{\Omega_k}|_S \; \boldsymbol{B}_{\Omega_k}|_B\right] \; \left[\boldsymbol{d}_A^T \; \boldsymbol{d}_S^T \; \boldsymbol{d}_B^T\right]^T, \tag{44}$$

where the vectors $\boldsymbol{d}_A$, $\boldsymbol{d}_B$, and $\boldsymbol{d}_S$ are the vectors containing the degrees of freedom of the nodes belonging to element A and element B exclusively and to shared nodes, respectively. They are defined as:

$$\begin{aligned}
\boldsymbol{d}_A &= \left[\boldsymbol{d}_1^T \; \dots \; \boldsymbol{d}_{n_A}^T\right], \\
\boldsymbol{d}_S &= \left[\boldsymbol{d}_{n_A+1}^T \; \dots \; \boldsymbol{d}_{n_A+n_S}^T\right], \\
\boldsymbol{d}_B &= \left[\boldsymbol{d}_{n_A+n_S+1}^T \; \dots \; \boldsymbol{d}_{n_A+n_S+n_B}^T\right].
\end{aligned} \tag{45}$$

The contributions of each element to the strain tensor can be easily separated in Equation (44):

$$\boldsymbol{\varepsilon}_{\Omega_k} = \overbrace{\left[\boldsymbol{B}_{\Omega_k}|_A \; \frac{1}{2}\boldsymbol{B}_{\Omega_k}|_S\right] \; \left[\boldsymbol{d}_A^T \; \boldsymbol{d}_S^T\right]^T}^{\text{element A}} + \overbrace{\left[\frac{1}{2}\boldsymbol{B}_{\Omega_k}|_S \; \boldsymbol{B}_{\Omega_k}|_B\right] \; \left[\boldsymbol{d}_S^T \; \boldsymbol{d}_B^T\right]^T}^{\text{element B}} \tag{46}$$

or in a more effective way, as done for stiffness matrix calculations (see Equation 41):

$$\boldsymbol{\varepsilon}_{\Omega_k} = \overbrace{\left[\boldsymbol{B}_{\Omega_k}|_A \; \boldsymbol{B}_{\Omega_k}|_S\right] \; \left[\boldsymbol{d}_A^T \; \boldsymbol{d}_S^T\right]^T}^{\text{element A}} + \overbrace{\left[\boldsymbol{0} \; \boldsymbol{B}_{\Omega_k}|_B\right] \; \left[\boldsymbol{d}_S^T \; \boldsymbol{d}_B^T\right]^T}^{\text{element B}}. \tag{47}$$

As already pointed out in Section 5.3, auxiliary elements are not used for strain calculations.

Strain-displacement matrices $\boldsymbol{B}_{\Omega_k}|_A$, $\boldsymbol{B}_{\Omega_k}|_S$, and $\boldsymbol{B}_{\Omega_k}|_B$ have already been evaluated during stiffness matrix calculations (Section 5.2). In particular, matrix $\boldsymbol{B}_{\Omega_k}|_S$ for shared nodes has already been built and its value has already been stored for both elements A and B and therefore no further data exchange is needed between these elements.

Equation (47) clearly shows that the evaluation of the strain tensor has to be done in two steps:

- Elements A and B calculate their contribution to the smoothed strain tensor. This step can be straightforwardly parallelized by exploiting existing FEM code capabilities, since it is carried out on an element by element basis.
- The contributions of each element is exchanged with the adjacent element. Each element finally calculates the smoothed strain tensor by adding its contribution to the one coming from the adjacent element.

5.5 | Computation of internal force vector

In finite element discretization, the internal force vector $\boldsymbol{f}_{\text{int}}$ for element Ω_i is calculated as:[11,12]

$$\boldsymbol{f}_{\text{int}} = \int_{\Omega_i} \boldsymbol{B}_{\Omega_i}^T \, \boldsymbol{\sigma}_{\Omega_i} \, d\Omega, \tag{48}$$

where $\boldsymbol{B}_{\Omega_i}$ is given by Equation (22).

In S-FEM discretization, the same equation holds for a smoothing domain Ω_k:

$$\boldsymbol{f}_{\text{int}} = \int_{\Omega_k} \boldsymbol{B}_{\Omega_k}^T \, \boldsymbol{\sigma}_{\Omega_k} \, d\Omega = \boldsymbol{B}_{\Omega_k}^T \, \boldsymbol{\sigma}_{\Omega_k} \, S_{\Omega_k}, \tag{49}$$

where $\boldsymbol{B}_{\Omega_k}$ is given by Equation (37). Stress tensor $\boldsymbol{\sigma}_{\Omega_k}$ is constant over the smoothing domain Ω_k.

By substituting Equation (37), the internal force vector can be equivalently rewritten as:

$$\boldsymbol{f}_{\text{int}} = \left[\boldsymbol{B}_{\Omega_k}|_A \; \boldsymbol{B}_{\Omega_k}|_S \; \boldsymbol{B}_{\Omega_k}|_B\right]^T \boldsymbol{\sigma}_{\Omega_k} \, S_{\Omega_k}. \tag{50}$$

Contributions of each element can now be easily separated:

$$\boldsymbol{f}_{\text{int}} = \overbrace{\left[\boldsymbol{B}_{\Omega_k}|_A \; \frac{1}{2}\boldsymbol{B}_{\Omega_k}|_S \; \boldsymbol{0}\right]^T \boldsymbol{\sigma}_{\Omega_k} \, S_{\Omega_k}}^{\text{element A}} + \overbrace{\left[\boldsymbol{0} \; \frac{1}{2}\boldsymbol{B}_{\Omega_k}|_S \; \boldsymbol{B}_{\Omega_k}|_B\right]^T \boldsymbol{\sigma}_{\Omega_k} \, S_{\Omega_k}}^{\text{element B}} \tag{51}$$

or in a more effective way, as already done for stiffness matrix and strain tensor calculations:

$$\boldsymbol{f}_{\text{int}} = \overbrace{\left[\boldsymbol{B}_{\Omega_k}|_A \; \boldsymbol{B}_{\Omega_k}|_S \; \boldsymbol{0}\right]^T \boldsymbol{\sigma}_{\Omega_k} \, S_{\Omega_k}}^{\text{element A}} + \overbrace{\left[\boldsymbol{0} \; \boldsymbol{0} \; \boldsymbol{B}_{\Omega_k}|_B\right]^T \boldsymbol{\sigma}_{\Omega_k} \, S_{\Omega_k}}^{\text{element B}}. \tag{52}$$

Equations (51) and (52) clearly show that the internal force vector can be calculated by the two elements independently since the strain-displacement matrices $\boldsymbol{B}_{\Omega_k}|_A$, $\boldsymbol{B}_{\Omega_k}|_S$, and $\boldsymbol{B}_{\Omega_k}|_B$ have already been evaluated during stiffness matrix evaluation (Section 5.2) and no further calculation and data exchange between the two elements are needed.

The two terms of the sum in Equation (51) can be restrained to the degrees of freedom of each element:

$$\begin{aligned} \boldsymbol{f}_{\text{int}} &= \boldsymbol{f}_{\text{int}}|_A + \boldsymbol{f}_{\text{int}}|_B, \\ \boldsymbol{f}_{\text{int}}|_I &= \left[\boldsymbol{B}_{\Omega_k}|_I \; \frac{1}{2}\boldsymbol{B}_{\Omega_k}|_S\right]^T \boldsymbol{\sigma}_{\Omega_k} \, S_{\Omega_k} \quad \text{for } I \in \{A, B\}, \end{aligned} \tag{53}$$

where the addition symbol "+" means "assembly."

It is worth to point out that no auxiliary element is used for internal force vector evaluation.

5.6 | Application of element-based FS-FEM formulation to non-tetrahedral elements and to adjacent elements of different type

Element-based FS-FEM formulation can be straightforwardly applied to any kind of finite element since it is independent of the element and smoothing domain geometries. The fact that the smoothing domain Ω_k is split in two subdomains $\Omega_k|A$ and $\Omega_k|B$ allows to compute smoothing domain related quantities inside each element ignoring the adjacent one. Elements A and B can therefore be of different type and the element-based FS-FEM formulation can be applied to meshes mixing different types of solid elements.

This is effectively another advantage of the element-based formulation since in the literature, to the best of the authors' knowledge, FS-FEM has been applied to tetrahedral meshes only.

In order to show the effectiveness of element-based FS-FEM formulation in managing a smoothing domain Ω_k extending over two elements of different type, the calculation of the stiffness matrix $\boldsymbol{K}_{\Omega_k}$ for the case shown in Figure 3 will be detailed in the following section.

5.6.1 | Example of calculation of the stiffness matrix of a smoothing domain extending over a hexahedral and a pyramidal element

The smoothing domain Ω_k is built on the quadrangular face 5-6-7-8 in Figure 3 shared by an hexahedral and a pyramidal element. It is split in two pyramidal subdomains, one for each of the two elements sharing the quadrangular face. An auxiliary element is built by using the nodes belonging to one element only (element defined by nodes 1-2-3-4-9 in Figure 3). Matrix $\boldsymbol{K}_{\Omega_k}$ is given by the sum of the contributions of hexahedral, pyramidal, and auxiliary elements (Equation 41):

$$\boldsymbol{K}_{\Omega_k} = \boldsymbol{K}_{\Omega_k \mathrm{HEXA}} + \boldsymbol{K}_{\Omega_k \mathrm{PYRA}} + \boldsymbol{K}_{\Omega_k \mathrm{AUX}}, \tag{54}$$

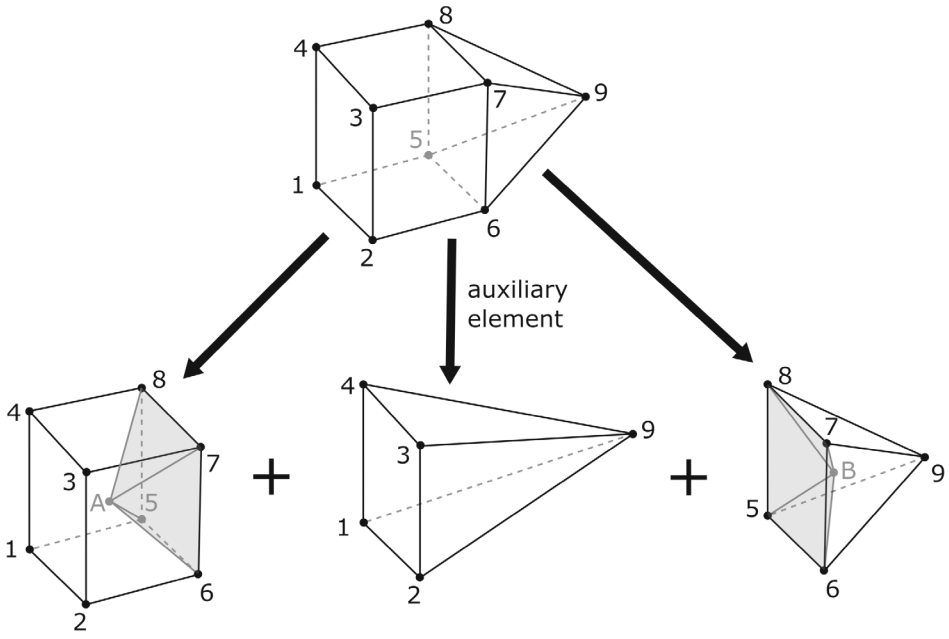

FIGURE 3 Example of smoothing domain extending over two elements of different type. The smoothing domain is built on the quadrangular face defined by nodes 5-6-7-8. The face is shared between the hexahedral and pyramidal elements. The smoothing domain is composed by the two pyramids shaded in the figure, one inside the hexahedral element and one inside the pyramidal element, obtained by connecting the centroid of each element with the nodes defining the shared face. The auxiliary element is defined by the nodes of the two elements not belonging to the shared face.

where the three contributions are given by:

$$\boldsymbol{K}_{\Omega_k \mathrm{HEXA}} = \begin{bmatrix} \boldsymbol{K}_{11}^{\Omega_k} & \cdots & \boldsymbol{K}_{14}^{\Omega_k} & \boldsymbol{K}_{15}^{\Omega_k} & \cdots & \boldsymbol{K}_{18}^{\Omega_k} & \boldsymbol{0} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{K}_{41}^{\Omega_k} & \cdots & \boldsymbol{K}_{44}^{\Omega_k} & \boldsymbol{K}_{45}^{\Omega_k} & \cdots & \boldsymbol{K}_{48}^{\Omega_k} & \boldsymbol{0} \\ \boldsymbol{K}_{51}^{\Omega_k} & \cdots & \boldsymbol{K}_{54}^{\Omega_k} & \boldsymbol{K}_{55}^{\Omega_k} & \cdots & \boldsymbol{K}_{58}^{\Omega_k} & \boldsymbol{0} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{K}_{81}^{\Omega_k} & \cdots & \boldsymbol{K}_{84}^{\Omega_k} & \boldsymbol{K}_{85}^{\Omega_k} & \cdots & \boldsymbol{K}_{88}^{\Omega_k} & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \tag{55}$$

$$\boldsymbol{K}_{\Omega_k \mathrm{PYRA}} = \begin{bmatrix} \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{K}_{59}^{\Omega_k} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{K}_{89}^{\Omega_k} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{K}_{95}^{\Omega_k} & \cdots & \boldsymbol{K}_{98}^{\Omega_k} & \boldsymbol{K}_{99}^{\Omega_k} \end{bmatrix}, \tag{56}$$

$$\boldsymbol{K}_{\Omega_k \mathrm{AUX}} = \begin{bmatrix} \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{K}_{19}^{\Omega_k} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{K}_{49}^{\Omega_k} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{K}_{91}^{\Omega_k} & \cdots & \boldsymbol{K}_{94}^{\Omega_k} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}. \tag{57}$$

Different colors have been used in Equations (55)–(57) to highlight the terms belonging to each submatrix in Equation (41).

Matrices $\boldsymbol{K}_{il}^{\Omega_k}$ $(i, l \in [1, 9])$ are evaluated by means of Equations (36) and (39):

$$\boldsymbol{K}_{il}^{\Omega_k} = \left(\boldsymbol{B}_i^{\Omega_k}\right)^T \boldsymbol{C} \boldsymbol{B}_l^{\Omega_k}, \quad \text{for } i, l \in [1, 9]. \tag{58}$$

The matrices $\boldsymbol{B}_i^{\Omega_k}$ are evaluated by means of Equations (28), (29), and (31) as follows.

First, we consider the hexahedral element. The terms $b_{ij}^{\Omega_k}$ $(i \in [1, 4])$ of matrix $\boldsymbol{B}_i^{\Omega_k}$ are calculated for nodes 1, 2, 3, and 4 by means of Equation (31). The calculation is done over the four triangular faces defining the boundary of smoothing domain Ω_k inside the hexahedral element only (triangles 5-6-A, 6-7-A, 7-8-A, and 8-5-A in Figure 3).

For node 9 of the pyramidal element, the same procedure applies for the evaluation of $\boldsymbol{B}_9^{\Omega_k}$, except that the calculations are done over the four triangular faces defining the boundary of smoothing domain Ω_k inside the pyramidal element only (triangles 5-6-B, 6-7-B, 7-8-B, and 8-5-B in Figure 3).

For shared nodes 5, 6, 7, and 8, the matrices $\boldsymbol{B}_i^{\Omega_k}$ $(i \in [5, 8])$ are evaluated by means of Equations (42) and (43) by adding the contributions $\boldsymbol{B}_i^{\Omega_k}|_{\mathrm{HEXA}}$ and $\boldsymbol{B}_i^{\Omega_k}|_{\mathrm{PYRA}}$ coming from the two adjacent elements. The calculations of these two contributions are done in exactly the same way as in the case of the calculations of $\boldsymbol{B}_i^{\Omega_k}$ for the nodes $i \in [1, 2, 3, 4, 9]$.

For the auxiliary element, the values of the terms $b_{ij}^{\Omega_k}$ used to calculate $\boldsymbol{B}_i^{\Omega_k}$ $(i \in [1, 2, 3, 4, 9])$ have already been calculated when evaluating $\boldsymbol{K}_{\Omega_k \mathrm{HEXA}}$ and $\boldsymbol{K}_{\Omega_k \mathrm{PYRA}}$. These values can be promptly reused to build the matrices $\boldsymbol{K}_{il}^{\Omega_k}$ $(i, l \in [1, 2, 3, 4, 9])$ (Equation 58) and finally $\boldsymbol{K}_{\Omega_k \mathrm{AUX}}$ (Equation 57). It is worth noting that auxiliary element geometry is ignored in the calculation of $\boldsymbol{K}_{\Omega_k \mathrm{AUX}}$.

It is clear that the above procedure does not take into account information concerning the element type and the same procedure applies for the two elements, hexahedral and pyramidal. The smoothing domain boundary is always decomposed in triangles, independently of the geometry of the shared face and of the two elements, and the calculations are carried out on these triangles only. Consequently, the above calculations can be carried out by the same routine in the code, independently of the type of the elements containing the smoothing domain.

5.7 | Shape functions for S-FEM elements

Contrary to FE method, S-FEM method does not use the derivatives of the element shape functions in its formulation ([2, 10]), as clearly shown by Equations (24)–(34). Moreover shape function values are needed only at some particular points of the boundary of the smoothing domain (Equation 31) and no explicit analytical form is required. This gives an enormous freedom in shape function definition.

Consequently, several different methods can be used to construct the shape functions provided that the following properties are satisfied at the discrete points of the element:[2,6-8]

1. Delta function: $N_i\left(\boldsymbol{x}_j\right) = \delta_{ij}$,
2. Partition of unity: $\sum_{i=1}^{n} N_i(\boldsymbol{x}) = 1$,
3. Linear compatibility: linear shape functions along element boundaries,
4. Linear consistency: $\sum_{i=1}^{n} N_i(\boldsymbol{x})\,\boldsymbol{x}_i = \boldsymbol{x}$,
5. $N_i(\boldsymbol{x}) \geqslant 0$,

where n is the number of nodes of the element.

In References 6-8, ES-FEM shape functions are successfully constructed by the simple averaging point interpolation method. The same method is adopted in this article for ES-FEM for the 2D cases. For FS-FEM, this method can be straightly applied to tetrahedral elements because smoothing domain boundaries are composed by triangular faces only. However, for non-simplex elements, further work is required, as explained in the following section.

5.7.1 | Construction of FS-FEM shape functions for non-simplex elements

FS-FEM smoothing domains are built on element faces connecting face vertices with the element centroid (Figure 2). For a triangular face, the smoothing domain is given by the union of two tetrahedrons (tetrahedrons 1-2-3-G and 1-2-3-G^1 in Figure 4). For a quadrangular face, the smoothing domain is given by the union of two pyramids (pyramids 1-2-3-4-G and 1-2-3-4-G^1 in Figure 5). In the element-based formulation, the smoothing domain is processed element by element by considering only the part of the domain contained in one element. Therefore for a triangular face the two tetrahedrons composing the smoothing domain (tetrahedrons 1-2-3-G and 1-2-3-G^1 in Figure 4) are processed separately. Similarly, for a quadrangular face processing is carried out on the two pyramids separately (pyramids 1-2-3-4-G and 1-2-3-4-G^1 in Figure 5). In the following part of the section, the face generating the smoothing domain will be referred as the base face of the smoothing domain, while the other faces of the smoothing domain will be referred as the lateral faces of the domain.

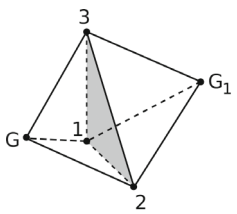

FIGURE 4 Example of smoothing domain generated by a triangular face (face 1-2-3). Points G and G^1 are the centroids of the two elements sharing face 1-2-3. The smoothing domain is composed by the two tetrahedra 1-2-3-G and 1-2-3-G^1.

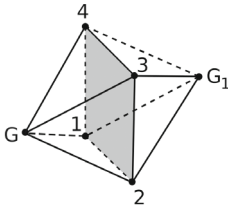

FIGURE 5 Example of smoothing domain generated by a quadrangular face (face 1-2-3-4). Points G and G^1 are the centroids of the two elements sharing face 1-2-3. The smoothing domain is composed by the two pyramids 1-2-3-4-G and 1-2-3-4-G^1.

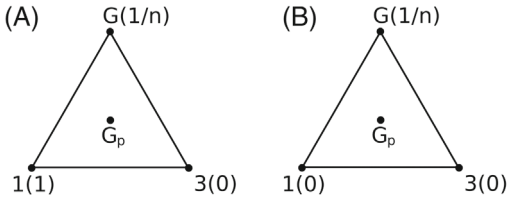

FIGURE 6 Value of the shape functions at the nodes of a triangular lateral face of a smoothing domain (example of face 1-3-G in Figure 4). The numbers in parenthesis near the node/point label are the value of the shape function at the node/point. G_p is the centroid of the face. (A) Example of the values when the shape function is associated to a node of the face (N_1 ($\boldsymbol{x}$) in the figure). (B) Example of the values when the shape function is not associated to a node of the face.

In order to apply the simple averaging point interpolation method, the shape function value must be known at each vertex of the smoothing domain. The values at the vertices of the base face are known since each vertex is coincident with a mesh node and delta function property 1 above applies. Shape function at element centroid G (or G_1) (Figures 4 and 5), can be evaluated as follows:[6-8]

$$N_i\left(\boldsymbol{x}_G\right) = \frac{1}{n}. \tag{59}$$

Here n is the number of nodes of the element.

In order to satisfy the linear compatibility property (property 3), linear variation of the shape function is assumed on smoothing domain faces where integration is done. Integration requires one Gauss point only and shape functions must be evaluated at the centroid of the smoothing domain face only (Equation 31). Their values can be calculated by the simple averaging point interpolation method. For a lateral face of the smoothing domain, shape function $N_i\left(\boldsymbol{x}_{\mathrm{Gp}}\right)$ at face centroid $\boldsymbol{x}_{\mathrm{Gp}}$ is given by (Figure 6):

$$N_i\left(\boldsymbol{x}_{G_p}\right) = \frac{1+\frac{1}{n}+0}{3} = \frac{n+1}{3n}, \quad \text{if node } i \text{ is a vertex of the face,} \tag{60a}$$

$$N_i\left(\boldsymbol{x}_{G_p}\right) = \frac{\frac{1}{n}+0+0}{3} = \frac{1}{3n}, \quad \text{if node } i \text{ is not a vertex of the face.} \tag{60b}$$

Similarly, for a triangular base face (Figure 7):

$$N_i\left(\boldsymbol{x}_{G_p}\right) = \frac{1+0+0}{3} = \frac{1}{3}, \quad \text{if node } i \text{ is a vertex of the face,} \tag{61a}$$

$$N_i\left(\boldsymbol{x}_{G_p}\right) = \frac{0+0+0}{3} = 0, \quad \text{if node } i \text{ is not a vertex of the face.} \tag{61b}$$

For a quadrangular base face, piecewise linear shape functions are constructed by decomposing the face into four triangles. Each triangle is obtained by connecting the face centroid G_f to the vertices of each face edge, as shown in Figure 8. Integration of Equation (30) is done on each triangle by means of Equation (31). Shape function value at the

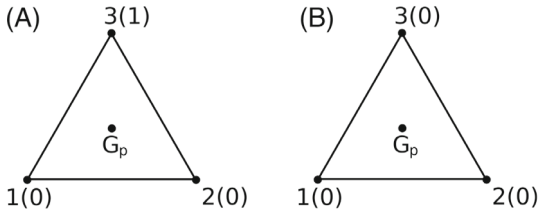

FIGURE 7 Value of the shape functions at the nodes of a triangular base face of a smoothing domain (example of face 1-2-3 in Figure 4). The numbers in parenthesis near the node label are the value of the shape function at the node. G_p is the centroid of the face. (A) Example of the values when the shape function is associated to a node of the face (N_3 ($\boldsymbol{x}$) in the figure). (B) Example of the values when the shape function is not associated to a node of the face.

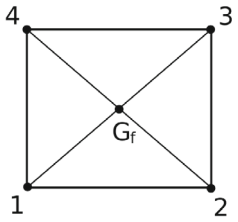

FIGURE 8 Subdivision of a quadrangular base face of a smoothing domain in four triangles. G_f is the centroid of the face.

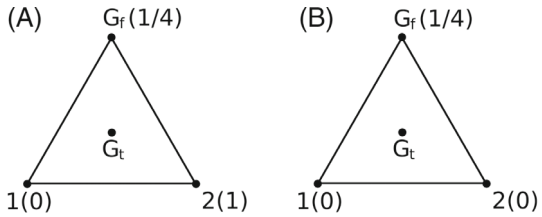

FIGURE 9 Value of the shape functions at the one of the four triangles used to split the quadrangular face in Figure 8. The numbers in parenthesis near the node label are the value of the shape function at the node/point. G_f is the centroid of the quadrangular face. (A) Example of the values when the shape function is associated to a node of the face (N_2 ($\boldsymbol{x}$) in the figure). (B) Example of the values when the shape function is not associated to a node of the face.

quadrangle face centroid G_f is calculated as in the case of element centroid G (Equation 59):

$$N_i\left(\boldsymbol{x}_{G_f}\right) = \frac{1}{4} \text{ if node } i \text{ is a vertex of the face,} \tag{62a}$$

$$N_i\left(\boldsymbol{x}_{G_f}\right) = 0 \text{ if node } i \text{ is not a vertex of the face.} \tag{62b}$$

In Equations (62a) and (62b), it is assumed that the shape function value of the nodes not coincident with one of the vertices of the face is zero on the face. Therefore only the four nodes defining the face contributes to the value of the shape function at G_f.

Simple averaging point interpolation method is then applied to each of the four triangles constituting the base face in order to calculate the shape function value at each triangle centroid G_t (Figure 9) used in Equation (31):

$$N_i\left(\boldsymbol{x}_{G_t}\right) = \frac{0+1+\frac{1}{4}}{3} = \frac{5}{12} \text{ if node } i \text{ is a vertex of the triangle,} \tag{63a}$$

$$N_i\left(\boldsymbol{x}_{G_t}\right) = \frac{0+0+\frac{1}{4}}{3} = \frac{1}{12} \text{ if node } i \text{ is not a vertex of the triangle.} \tag{63b}$$

5.7.2 | Standard patch test

The solid patch test proposed by Macneal and Harder[15] is used to test the non-simplex formulation of FS-FEM. Further analysis of the sensitivity to mesh distortion will be done later in Section 6.

In the solid patch test, the following prescribed boundary conditions are applied on the external faces of a unit cube (Figure 10):

$$\begin{aligned} u(\boldsymbol{x}) &= 10^{-3}(2x + y + z)/2, \\ v(\boldsymbol{x}) &= 10^{-3}(x + 2y + z)/2, \\ w(\boldsymbol{x}) &= 10^{-3}(x + y + 2z)/2, \end{aligned} \tag{64}$$

where $u(\boldsymbol{x})$, $v(\boldsymbol{x})$, and $w(\boldsymbol{x})$ are the displacements in x, y, and z directions, respectively, at a point of coordinates $\boldsymbol{x} = [x, y, z]^T$.

The length of the edges of the unit cube is taken equal to 1 m. The cube material is assumed linear elastic and isotropic. Its Young's modulus and Poisson's ratio are $E = 1$ MPa and $\nu = 0.25$, respectively.

The theoretical solution for stress and strain fields is given by:[15]

$$\varepsilon_x = \varepsilon_y = \varepsilon_Z = \gamma_{xy} = \gamma_{yz} = \gamma_{zx} = 10^{-3}, \tag{65a}$$

$$\begin{aligned} \sigma_x = \sigma_y = \sigma_z &= 2000 \text{ Pa}, \\ \tau_{xy} = \tau_{yz} = \tau_{zx} &= 400 \text{ Pa}. \end{aligned} \tag{65b}$$

The hexahedral mesh used in the solid patch test is shown in Figure 10. The coordinates of the internal nodes are given in Table 1.

The comparison between the theoretical solution and the FS-FEM solution is done at the internal nodes given in Table 1 for the displacements and at each smoothing cell for the stress and strain fields. This choice is based on the fact that the stress and strain fields given by the theoretical solution are constant in space (Equations 65a and 65b) and the stress and strain fields calculated by FS-FEM are constant over a smoothing domain.

The comparison between the theoretical and numerical values of the displacement is done by evaluating the relative error for each displacement component:

$$\begin{aligned} e_u(\boldsymbol{x}_i) &= \frac{|u(\boldsymbol{x}_i) - u_{\text{SFEM}}(\boldsymbol{x}_i)|}{u(\boldsymbol{x}_i)}, \\ e_v(\boldsymbol{x}_i) &= \frac{|v(\boldsymbol{x}_i) - v_{\text{SFEM}}(\boldsymbol{x}_i)|}{v(\boldsymbol{x}_i)}, \\ e_w(\boldsymbol{x}_i) &= \frac{|w(\boldsymbol{x}_i) - w_{\text{SFEM}}(\boldsymbol{x}_i)|}{w(\boldsymbol{x}_i)}, \end{aligned} \tag{66}$$

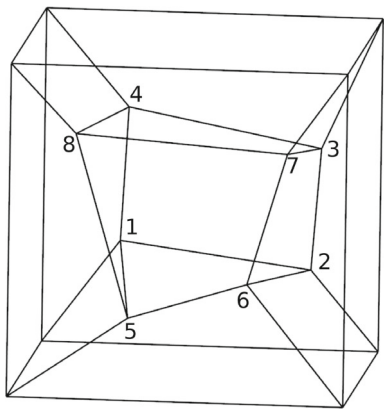

FIGURE 10 Unit cube patch test. Internal node number is shown.

TABLE 1 Coordinates of the internal nodes of the unit cube patch test[15]

Node number	x	y	z
1	0.249	0.342	0.192
2	0.826	0.288	0.288
3	0.850	0.649	0.263
4	0.273	0.750	0.230
5	0.320	0.186	0.643
6	0.677	0.305	0.683
7	0.788	0.693	0.644
8	0.165	0.745	0.702

TABLE 2 Maximum value of the errors for the unit cube patch test

	$e_u(x_i)$	$e_v(x_i)$	$e_w(x_i)$	$e_\sigma^{\Omega_k}$	$e_\varepsilon^{\Omega_k}$
Maximum value	0.0	0.0	0.0	2.56e−15	1.08e−15

Note: Errors on displacements components $(e_u(x_i), e_v(x_i), e_w(x_i))$ are evaluated at mesh nodes. Errors on stress and strain tensor components $(e_\sigma^{\Omega_k}, e_\varepsilon^{\Omega_k})$ are evaluated at each smoothing domain.

where $e_u(\boldsymbol{x}_i)$, $e_v(\boldsymbol{x}_i)$, and $e_w(\boldsymbol{x}_i)$ are the relative errors at node i (with coordinates $\boldsymbol{x}_i$ given in Table 1) on x, y, and z displacements, respectively, $u(\boldsymbol{x}_i)$, $v(\boldsymbol{x}_i)$, and $w(\boldsymbol{x}_i)$ are the theoretical displacements at node i given by Equation (64) and $u_{\text{SFEM}}(\boldsymbol{x}_i)$, $v_{\text{SFEM}}(\boldsymbol{x}_i)$, and $w_{\text{SFEM}}(\boldsymbol{x}_i)$ are the corresponding S-FEM numerical values.

The error for the stress and strain fields is given by the maximum value of the relative error calculated on each component of the stress and strain tensors at each smoothing domain Ω_k. For the stress field, the error reads:

$$e_\sigma^{\Omega_k} = \max_{\sigma_{\text{cmp}} \in A} \left(\frac{\left| \sigma_{\text{cmp}} - \sigma_{\text{cmp}}^{\Omega_k} \right|}{\sigma_{\text{cmp}}} \right), \quad \text{for } A = \{\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{yz}, \tau_{xz}\}, \tag{67}$$

where $e_\sigma^{\Omega_k}$ is the maximum relative error on the stress tensor components at smoothing domain Ω_k, σ_{cmp} is one component of the stress tensor of the theoretical solution given by Equation (65b) and $\sigma_{\text{cmp}}^{\Omega_k}$ is the corresponding S-FEM numerical value.

For the strain field, the error reads:

$$e_\varepsilon^{\Omega_k} = \max_{\varepsilon_{\text{cmp}} \in A} \left(\frac{\left| \varepsilon_{\text{cmp}} - \varepsilon_{\text{cmp}}^{\Omega_k} \right|}{\varepsilon_{\text{cmp}}} \right), \quad \text{for } A = \{\varepsilon_x, \varepsilon_y, \varepsilon_z, \gamma_{xy}, \gamma_{yz}, \gamma_{xz}\}, \tag{68}$$

where $e_\sigma^{\Omega_k}$ is the maximum relative error on the stress tensor components at smoothing domain Ω_k, ε_{cmp} is one component of the stress tensor of the theoretical solution given by Equation (65a) and $\varepsilon_{\text{cmp}}^{\Omega_k}$ is the corresponding S-FEM numerical value.

The errors are given in Table 2. They are of the same order of magnitude of the machine precision. It can be concluded that the FS-FEM formulation used for non-simplex elements passes the patch test and is able to represent a constant stress and strain state.

6 | NUMERICAL EXPERIMENTS

In order to verify that the proposed element-based formulation for ES-FEM and FS-FEM methods works, numerical experiments have been carried out for 2D and 3D cases. All the elements commonly used in finite element analysis have

been tested and mesh convergence has been verified by comparing numerical solution obtained by the proposed approach with finite element and reference solutions available in the literature.

Convergence tests have been carried out first on regular meshes with element aspect ratios equal or close to 1. As done in Reference 10, in order to check the effect of element distortion on the stability and accuracy of the proposed formulation, further convergence tests have been done on distorted meshes obtained from the regular one by moving the nodes as follows:

$$\begin{aligned} x &= x_0 + \alpha_{\text{ir}} \cdot r_x \cdot \Delta h_x, \\ y &= y_0 + \alpha_{\text{ir}} \cdot r_y \cdot \Delta h_y, \\ z &= z_0 + \alpha_{\text{ir}} \cdot r_z \cdot \Delta h_z, \end{aligned} \tag{69}$$

where $\boldsymbol{x}_0 = [x_0, y_0, z_0]^T$ and $\boldsymbol{x} = [x, y, z]^T$ are the coordinates of the node before and after the distortion, respectively, $\alpha_{\text{ir}} \in [0, 1]$ is an irregularity factor (element is not distorted for $\alpha_{\text{ir}} = 0$ and distortion increases with α_{ir}), r_x, r_y, and r_z are three computer generated random numbers included in the interval $[-0.5, 0.5]$ and Δh_x, Δh_y, and Δh_z are the mean edge size of the finite elements in the mesh in x, y, and z directions, respectively.

The element-based formulation for ES-FEM and FS-FEM methods have been implemented in Code_Aster, a free finite element software developed by EDF R&D for solid mechanics.[16] All the numerical results shown in this article have been obtained by means of this modified version of Code_Aster. Element matrix computation routines of existing finite elements have been replaced by a generic S-FEM routine. Auxiliary elements are created dynamically during the simulation as "late elements" ("éléments tardifs" in the original French documentation of the code). These elements are standard elements available in the code library. Any routine in the code can dynamically add them to an existing mesh provided by the user. They do not belong to the mesh and they are not shown in calculation results, but they participate in code calculations, like stiffness matrices calculations.

6.1 | 2D test case

For the 2D case, the elastic case of the cantilever beam shown in Figure 11 has been studied. The thickness of the beam is assumed to be extremely small with respect to its length L and its height D. The beam is therefore in a plane stress condition. The parabolic distributed shear force applied on the free edge on the right (see Figure 11) is given by:

$$t_y(y) = \frac{6P}{D^3} y^2 - \frac{3}{2}\frac{P}{D}, \tag{70}$$

where D is the height of the beam (see Figure 11) and P is the resultant shear force applied on the free edge.

The material is assumed linear elastic and isotropic. Its Young's modulus and Poisson's ratio are E and ν, respectively.

The analytical solution of the problem is available in Reference 6. The displacement field is given by the following equations:

$$\begin{aligned} u_x &= \frac{Py}{6\text{EI}} \left[(6L - 3x)x - (2 + \nu)\left(y^2 - \frac{D^2}{4} \right) \right], \\ u_y &= -\frac{P}{6\text{EI}} \left[3\nu y^2 (L - x) + (4 + 5\nu)\frac{D^2 x}{4} + (3L - x)x^2 \right], \end{aligned} \tag{71}$$

where $I = \frac{D^3}{12}$ is the area moment of inertia per unit width of the beam cross-section.

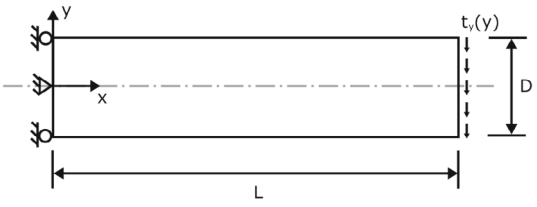

FIGURE 11 Cantilever beam subjected to a parabolic shear force on the free edge on the right

The plane stress field is given by the following equations:

$$\begin{aligned}\sigma_x &= \frac{P(L-x)y}{I},\\ \sigma_y &= 0,\\ \tau_{xy} &= -\frac{P}{2I}\left(\frac{D^2}{4}-y^2\right).\end{aligned} \tag{72}$$

For the numerical test, the problem parameters have been taken as $E = 2.1 \times 10^{11}$ Pa, $\nu = 0.3$, $D = 50$ m, $L = 500$ m, and $P = 10$ kN.[6] The thickness of the beam has been fixed to 1 m. First, two uniform meshes of four-node quadrilateral elements (Q4) and linear triangular elements (T3) have been used. The T3 mesh has been obtained from the Q4 mesh by splitting each quadrilateral element in two triangles using one of its diagonals (Figure 12). The aspect ratio of the Q4 elements has been fixed to 1 to ensure high precision of the finite element solution. Distorted meshes have been generated from these regular meshes by moving the mesh nodes by means of Equation (69). An examples of distorted Q4 mesh is shown in Figure 13 ($\alpha_{ir} = 0.7$). The highest irregularity factor used in the tests is $\alpha_{ir} = 0.7$, which corresponds to a very distorted mesh. For higher irregularity factors, overlapping elements appear in the mesh (Figure 14), invalidating the hypothesis at the base of S-FEM theory (nonoverlapping smoothing domains, see section 4).

For the regular mesh case ($\alpha_{ir} = 0$), mesh convergence is shown in Figure 15 for the FEM solution and for the element-based S-FEM formulation presented in this article. Solutions are compared in terms of elastic strain energy. In Figure 16, the comparison is done in terms of the relative error in energy norm, calculated as follows:

$$\text{Error} = \sqrt{\frac{\int (\boldsymbol{\sigma}_{FE} - \boldsymbol{\sigma}_{ref}) : \boldsymbol{C}^{-1} : (\boldsymbol{\sigma}_{FE} - \boldsymbol{\sigma}_{ref})}{\int \boldsymbol{\sigma}_{ref} : \boldsymbol{C}^{-1} : \boldsymbol{\sigma}_{ref}}}, \tag{73}$$

where $\boldsymbol{C}$ is the stiffness tensor of the generalized Hooke's law (Equation 9) and $\boldsymbol{\sigma}_{FE}$ and $\boldsymbol{\sigma}_{ref}$ are the finite element (FEM or S-FEM) and reference (Equation 72) stress tensors, respectively. Integration in Equation (73) is carried out over the whole mesh.

The comparison between FEM and ES-FEM solutions for the distorted mesh case ($\alpha_{ir} = 0.7$) is shown in Figures 17 and 18 for the elastic strain energy and relative error in energy norm, respectively.

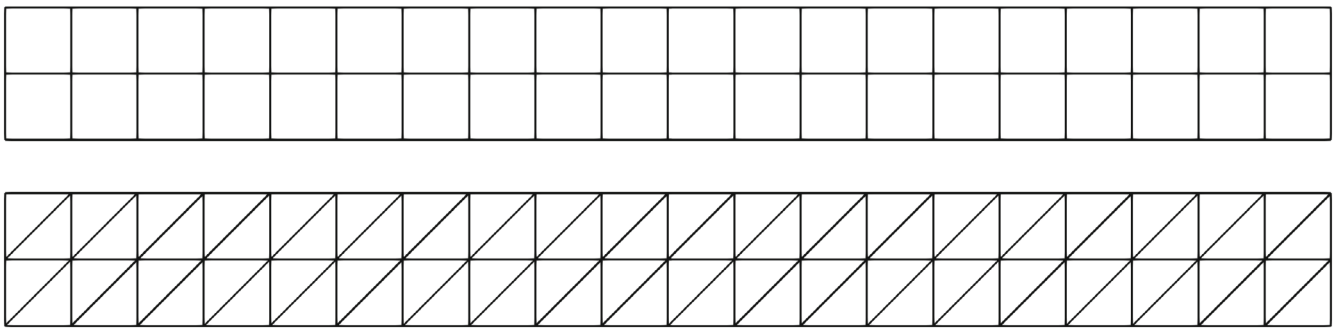

FIGURE 12 Examples of Q4 and T3 meshes used for the cantilever beam of Figure 11. T3 mesh is obtained from Q4 mesh by splitting each quadrilateral element in two triangular elements.

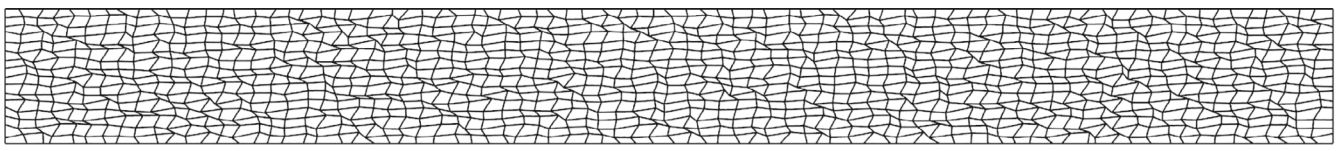

FIGURE 13 Example of a distorted Q4 mesh ($\alpha_{ir} = 0.7$)

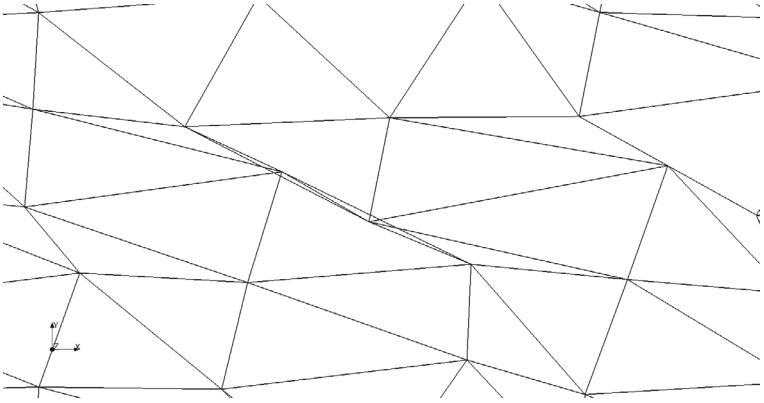

FIGURE 14 Example of overlapping T3 elements generated by high irregularity factors

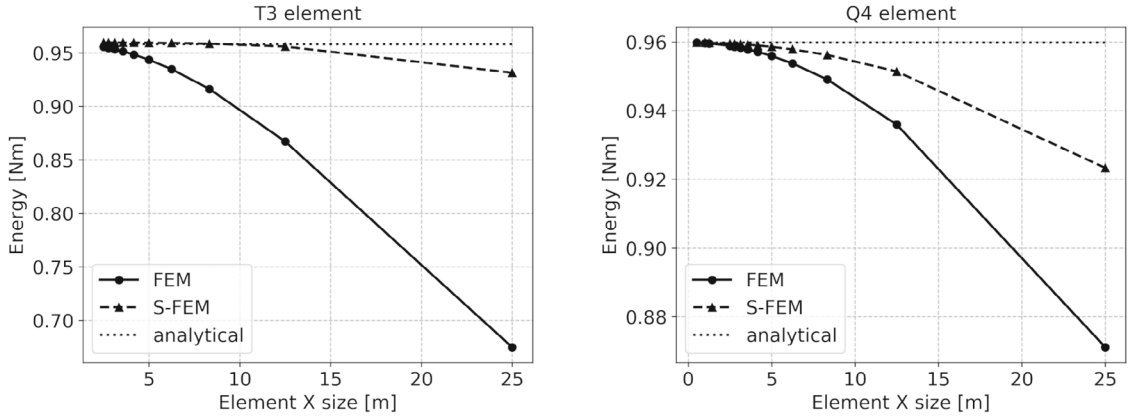

FIGURE 15 Results for test case of Figure 11. Comparison between the elastic strain energy obtained from S-FEM and FEM models for different element sizes and the value from the analytical solution for T3 and Q4 meshes ($\alpha_{ir} = 0$)

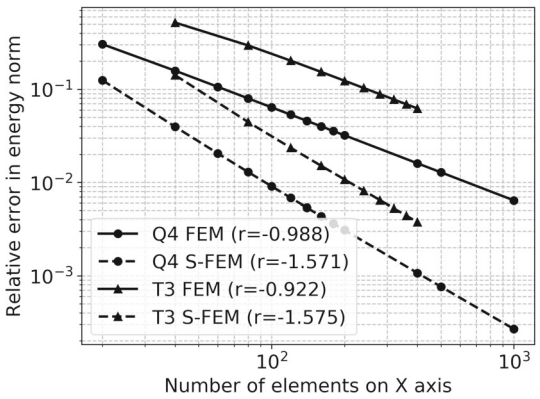

FIGURE 16 Results for test case of Figure 11. Relative error in energy norm as defined in Equation (73) for the S-FEM and FEM numerical solutions for different element sizes for T3 and Q4 meshes ($\alpha_{ir} = 0$)

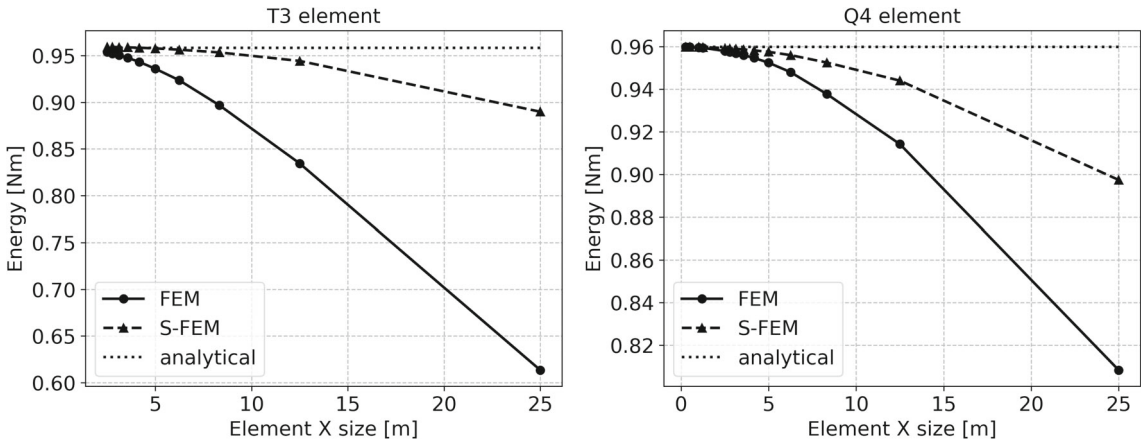

FIGURE 17 Results for test case of Figure 11. Comparison between the elastic strain energy obtained from S-FEM and FEM models for different element sizes and the value from the analytical solution for T3 and Q4 meshes ($\alpha_{ir} = 0.7$)

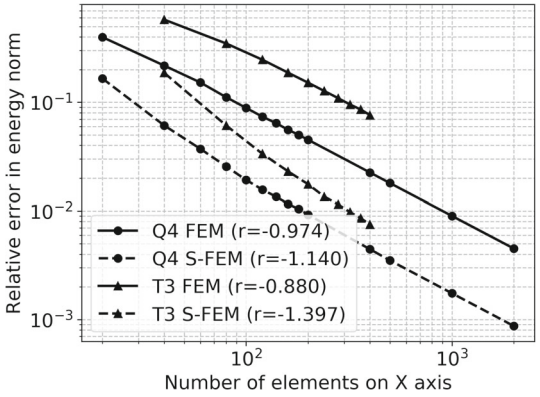

FIGURE 18 Results for test case of Figure 11. Relative error in energy norm as defined in Equation (73) for the S-FEM and FEM numerical solutions for different element sizes for T3 and Q4 meshes ($\alpha_{ir} = 0.7$)

As expected,[6] the element-based ES-FEM method converges toward the reference solution with mesh refinement and the accuracy and mesh convergence rate are always higher than for the FEM, independently of the distortion of the mesh. These results show that ES-FEM is less sensitive to mesh distortion than FEM.

6.2 | 3D test case

The cubic cantilever shown in Figure 19 is subjected to a uniform pressure of 1 Pa on the upper face.[10] The face lying in the y-z plane ($x = 0$) is fixed. The material is linear elastic (Young's modulus $E = 1$ Pa, Poisson's ratio $\nu = 0.25$). A reference solution is given in Reference 10. It has been calculated by means of a FE model with a very refined mesh of 10-node tetrahedral elements. The reference displacement at point A at $(1.0, 1.0, -0.5)$ (Figure 19) is 3.3912 m, the reference strain energy is 0.9486 N m. The theoretical solution is not available. Therefore, in order to calculate the relative error in energy norm of the element-based FS-FEM solution by means of Equation (73), a reference stress field is calculated by means of a FE model using a refined regular mesh. The mesh is composed by 64,000 cubic 8-node hexahedral elements (74,080 nodes). The displacement at point A obtained by this refined FE model is 3.3910 m, the strain energy is 0.948633 N m. These results are in excellent agreement with the solution given in Reference 10. Relative error in strain energy norm (Equation 73) is evaluated on this refined mesh using the calculated reference stress field and the element-based FS-FEM stress field from the coarser mesh projected on this refined mesh.

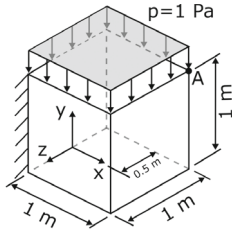

FIGURE 19 Cubic cantilever subjected to a uniform pressure of 1 Pa on the upper free face

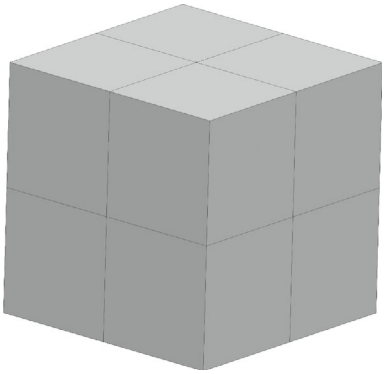

FIGURE 20 Example of uniform hexahedral mesh of cubic cantilever of Figure 19

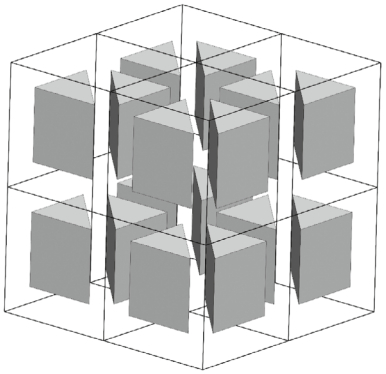

FIGURE 21 Example of the mesh of uniform wedge elements obtained by splitting the hexahedral mesh of Figure 20. Wireframe view of hexahedral mesh of Figure 20 is shown. Wedge elements are shrunk to show how they have been obtained from hexahedral elements.

Several simulations have been carried out in order to test mesh convergence of the element-based FS-FEM formulation and its applicability to any kind of standard element used in FE simulations. First, regular meshes have been tested. The eight-node hexahedral element (H8) mesh is shown in Figure 20. The element aspect ratio is 1. The mesh is refined by regular element subdivision preserving the element aspect ratio. Meshes using different element types have been built from the hexahedral element mesh by element subdivision: six-node wedge element (W6) mesh, four-node tetrahedral element (T4) mesh and five-node pyramidal element (P5) mesh are shown in Figures 21–23, respectively. Distorted meshes have been generated from these regular meshes by moving the mesh nodes by means of Equation (69). An example of distorted H8 mesh is shown in Figure 24. The highest irregularity factor used in the tests is $\alpha_{\text{ir}} = 0.7$, which corresponds to a very distorted mesh. For higher irregularity factors, the FE software is not able to calculate equivalent nodal forces for the distributed pressure load applied to the top surface of the cube due to the high distortion of the surface mesh. Moreover, for higher irregularity factors, some overlapping elements appear in the mesh, as in the 2D test case described in Section 6.1.

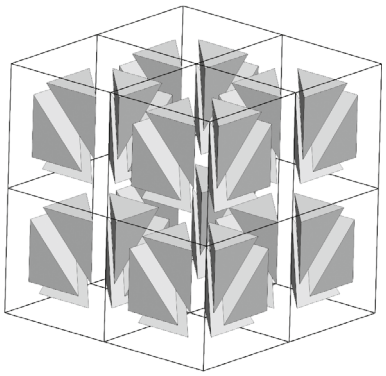

FIGURE 22 Example of the mesh of uniform tetrahedral elements obtained by splitting the hexahedral mesh of Figure 20. Wireframe view of hexahedral mesh of Figure 20 is shown. Tetrahedral elements are shrunk to show how they have been obtained from hexahedral elements.

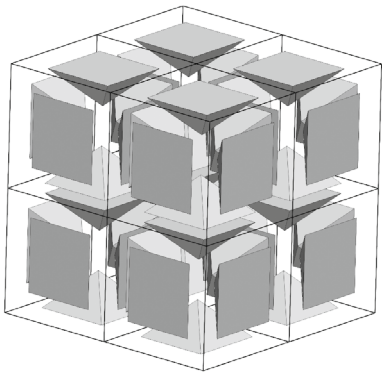

FIGURE 23 Example of the mesh of uniform pyramidal elements obtained by splitting the hexahedral mesh of Figure 20. Wireframe view of hexahedral mesh of Figure 20 is shown. Pyramidal elements are shrunk to show how they have been obtained from hexahedral elements.

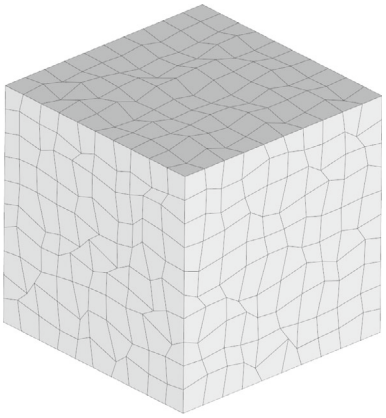

FIGURE 24 Example of a distorted H8 mesh ($\alpha_{ir} = 0.7$)

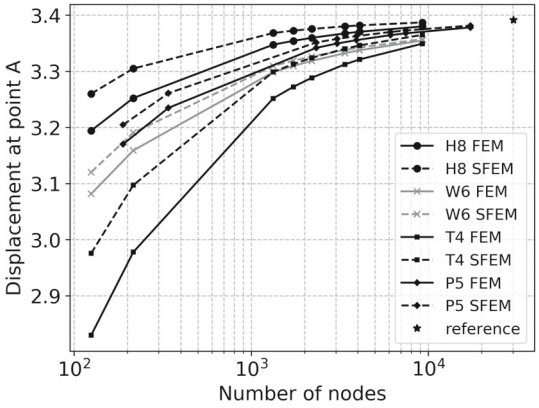

FIGURE 25 Results for test case of Figure 19. Comparison between the values of the displacement of point A in Figure 19 obtained from the FS-FEM and FEM models for meshes of different element type and different element size ($\alpha_{ir} = 0$). A reference solution from Reference 10 is shown as well.

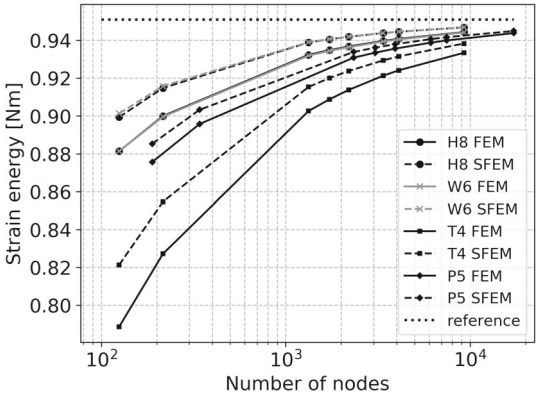

FIGURE 26 Results for test case of Figure 19. Comparison between the values of the strain energy obtained from the FS-FEM and FEM models for meshes of different element type and different element size ($\alpha_{ir} = 0$). A reference solution from Reference 10 is shown as well.

The comparison between reference, finite element and element-based FS-FEM solutions for the regular meshes is shown in Figures 25 and 26 for displacement of point A and strain energy, respectively. The relative error in energy norm is shown in Figures 27–30. As expected,[10] the element-based FS-FEM method converges toward the reference solution with mesh refinement and the mesh convergence rate is higher than for the FEM. The accuracy is always higher than for the FEM.

A similar comparison is done for the distorted meshes (irregularity factor $\alpha_{ir} = 0.7$) in Figures 31–36. As expected, the error is greater for the distorted meshes than for the regular meshes. The S-FEM accuracy and convergence rate with mesh refinement are always higher than for the FEM. It should be pointed out that for the distorted meshes the FE software complains about elements in the mesh with nonconstant sign of the Jacobian of the isoparametric transformation matrix, which is not the case for S-FEM since no isoparametric transformation is used (Section 4). Variation of Jacobian sign over an element is detrimental for the numerical solution accuracy, as shown for example in Figure 37 for a hexahedral mesh. These results show that S-FEM is less sensitive to mesh distortion than FEM.

6.3 | Computational cost of ES-FEM and FS-FEM methods

As shown in References 2, 7, 8, and 10, stiffness matrices generated by ES-FEM and FS-FEM methods are less coarse than the ones generated by FEM. Consequently, the computational cost for stiffness matrix inversion is much higher for ES-FEM and FS-FEM than for FEM. However, solution accuracy is higher for ES-FEM and FS-FEM than for FEM, as

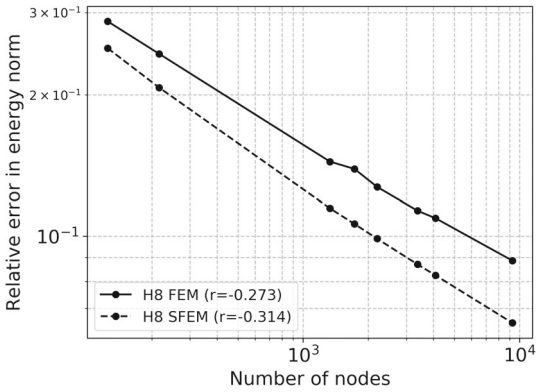

FIGURE 27 Results for test case of Figure 19. Relative error in energy norm as defined in Equation (73) for the S-FEM and FEM numerical solutions for different element sizes for H8 meshes ($\alpha_{ir} = 0.0$)

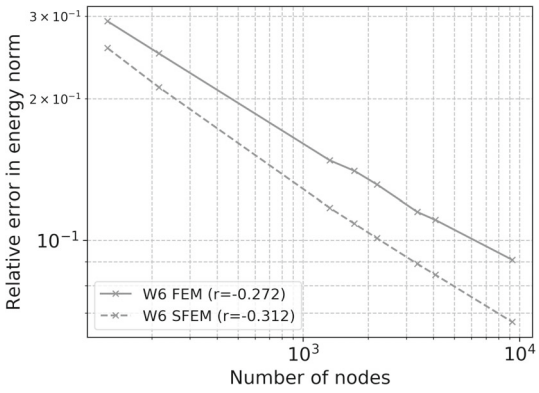

FIGURE 28 Results for test case of Figure 19. Relative error in energy norm as defined in Equation (73) for the S-FEM and FEM numerical solutions for different element sizes for W6 meshes ($\alpha_{ir} = 0.0$)

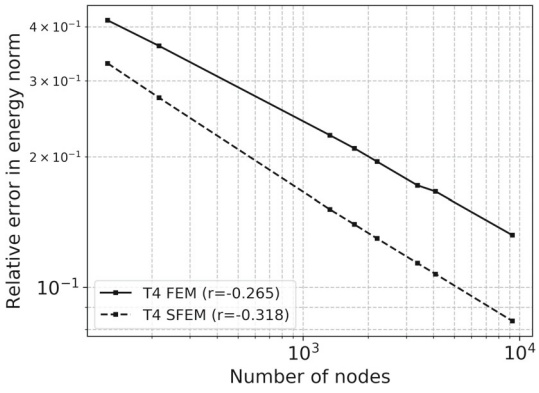

FIGURE 29 Results for test case of Figure 19. Relative error in energy norm as defined in Equation (73) for the S-FEM and FEM numerical solutions for different element sizes for T4 meshes ($\alpha_{ir} = 0.0$)

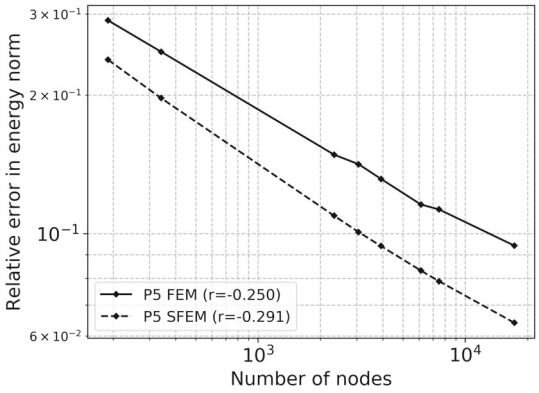

FIGURE 30 Results for test case of Figure 19. Relative error in energy norm as defined in Equation (73) for the S-FEM and FEM numerical solutions for different element sizes for P5 meshes ($\alpha_{\text{ir}} = 0.0$)

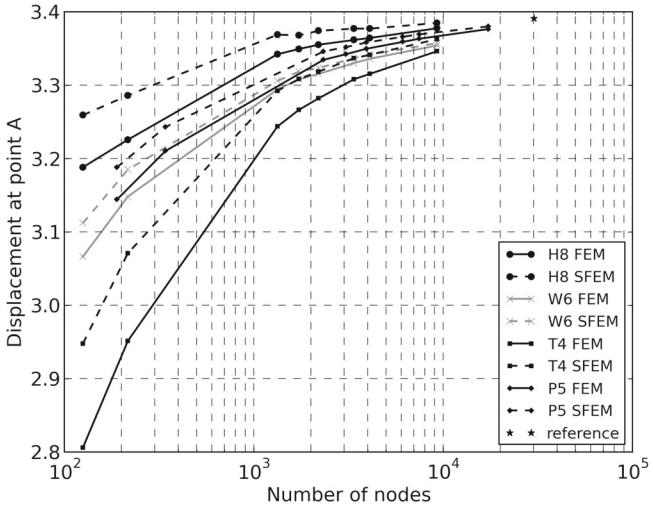

FIGURE 31 Results for test case of Figure 19. Comparison between the values of the displacement of point A in Figure 19 obtained from the FS-FEM and FEM models for meshes of different element type and different element size ($\alpha_{\text{ir}} = 0.7$). A reference solution from Reference 10 is shown as well.

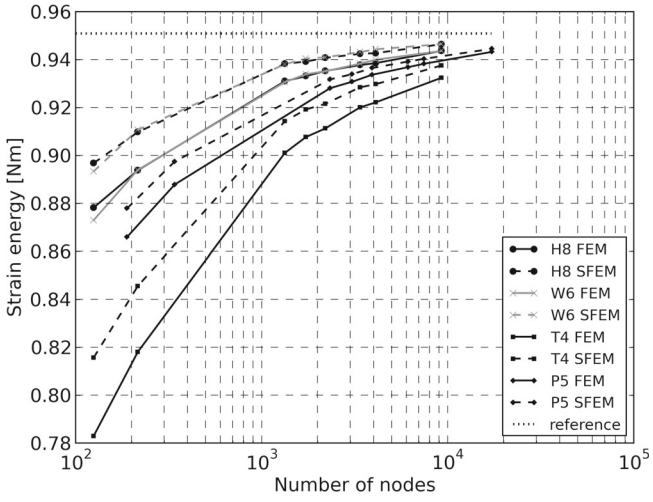

FIGURE 32 Results for test case of Figure 19. Comparison between the values of the strain energy obtained from the FS-FEM and FEM models for meshes of different element type and different element size ($\alpha_{\text{ir}} = 0.7$). A reference solution from Reference 10 is shown as well.

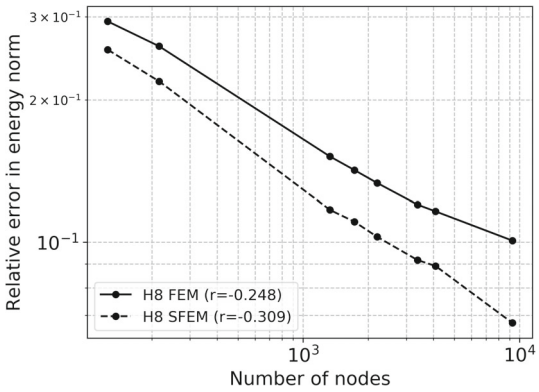

FIGURE 33 Results for test case of Figure 19. Relative error in energy norm as defined in Equation (73) for the S-FEM and FEM numerical solutions for different element sizes for H8 meshes ($\alpha_{\text{ir}} = 0.7$)

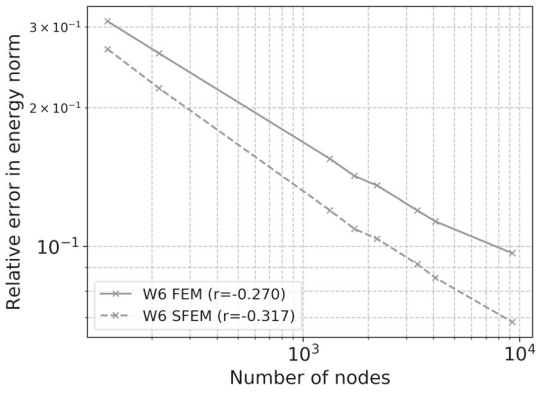

FIGURE 34 Results for test case of Figure 19. Relative error in energy norm as defined in Equation (73) for the S-FEM and FEM numerical solutions for different element sizes for W6 meshes ($\alpha_{\text{ir}} = 0.7$)

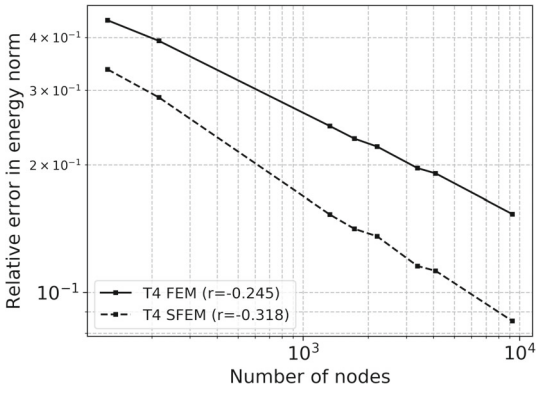

FIGURE 35 Results for test case of Figure 19. Relative error in energy norm as defined in Equation (73) for the S-FEM and FEM numerical solutions for different element sizes for T4 meshes ($\alpha_{\text{ir}} = 0.7$)

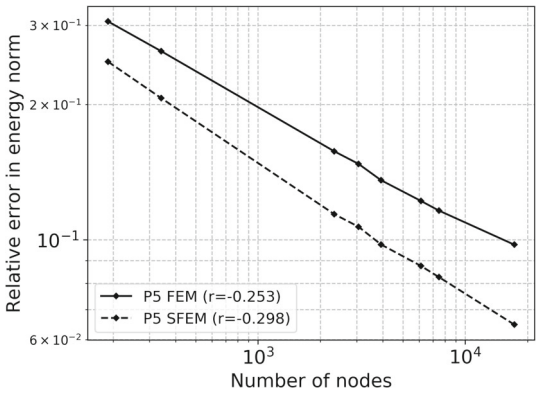

FIGURE 36 Results for test case of Figure 19. Relative error in energy norm as defined in Equation (73) for the S-FEM and FEM numerical solutions for different element sizes for P5 meshes ($\alpha_{ir} = 0.7$)

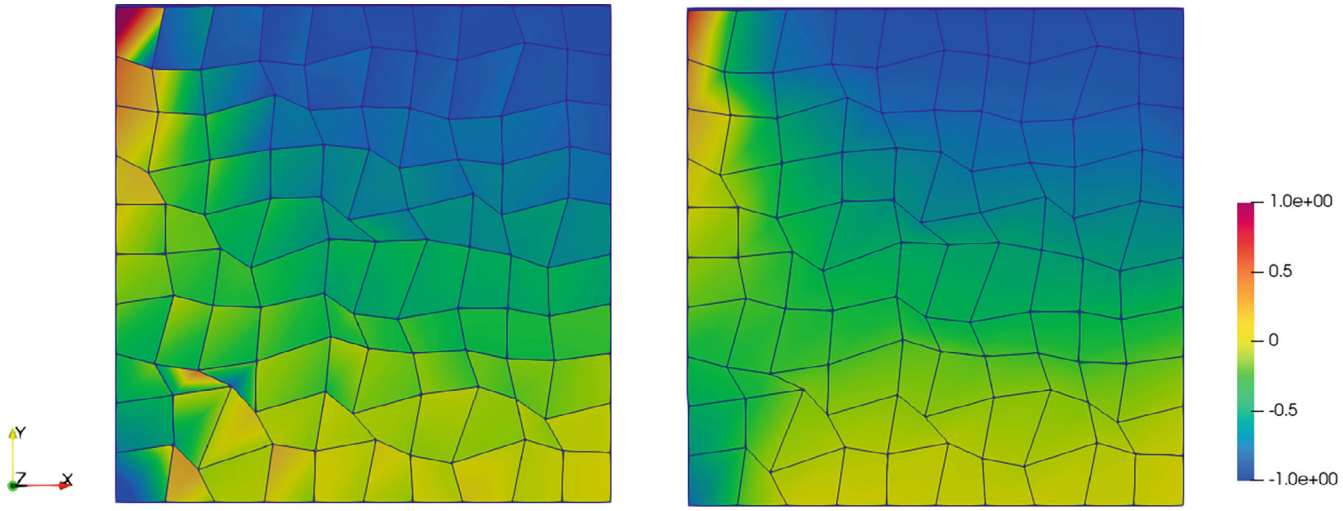

FIGURE 37 Effect of the change in the sign of the Jacobian on the FEM stress field, σ_y stress component (Pa) for the external face at $z = 0.5$ of the cube of Figure 19. The H8 mesh is distorted with an irregularity factor $\alpha_{ir} = 0.7$. On the left: FEM solution. On the right: FS-FEM solution

shown in Sections 6.1 and 6.2. Therefore, a more relevant parameter for the performance comparison of the methods is the computational efficiency, as done in References 2 and 7-10: computation time should be compared at a fixed accuracy rather than a fixed mesh. Since FEM accuracy is lower than ES-FEM and FS-FEM accuracy, in order to get the same accuracy, a finer mesh is needed for FEM compared to ES-FEM and FS-FEM, which increase the computation time.

The element-based formulation presented in this article adds an overhead to the standard S-FEM formulation because some time is needed to create and manage auxiliary elements. In order to evaluate this overhead, an example for the hexahedral mesh from the 3D case presented in Section 6.2 will be used. In Figure 27, the relative error in energy norm is shown for this case. For FEM, the biggest case is the one with 9281 nodes (the last point on the right on the FEM curve in the figure) for which the error is close to 9×10^{-2}. The same error is obtained by FS-FEM with a coarser mesh composed by 3375 nodes. Computation times for these two models are shown in Table 3, where the model preparation time and the solver time are shown. The model preparation time includes everything is needed to prepare the model for the computation, whereas the solver time is the time needed to solve the system of linear algebraic equations produced by the numerical discretization. For the FS-FEM method, the model preparation time includes also the time needed to create and manage the auxiliary elements necessary to the element-based formulation.

As clearly shown in Table 3, for a given mesh, solver time for FEM is lower than for FS-FEM. However, for a given accuracy, solver time for FEM (9281 nodes mesh) is higher than for FS-FEM (3375 nodes mesh). Model preparation time

TABLE 3 Computation time for FEM and FS-FEM for two different hexahedral meshes (element aspect ratio equal to 1)

	FEM		FS-FEM	
Mesh nodes	**Preparation (s)**	**Solver (s)**	**Preparation (s)**	**Solver (s)**
3375	0.07	1.10	0.17	2.94
9281	0.11	3.95	0.38	14.59

Note: "Preparation" means the preparation computation time needed to prepare the model for the simulation, including the time needed to create and manage auxiliary elements for FS-FEM. "Solver" means the time needed by the linear solver to solve the discretized model.

is much lower than solver time both for FEM and FS-FEM. An estimation of the overhead of element-based FS-FEM formulation can be obtained by comparing the model preparation time for FEM and FS-FEM for a given mesh: for the two meshes shown in Table 3, model preparation time for FS-FEM is about 3 times the model preparation time for FEM. However, this overhead is much smaller than the solver time. Therefore it does not impact the total computation time significantly.

7 | CONCLUSIONS

An element-based formulation for ES-FEM and FS-FEM models has been presented in detail. It has been shown that S-FEM smoothing domain-based equations can be equivalently rewritten on an element basis if an auxiliary element is introduced in the mesh for each smoothing domain. This auxiliary element is used for stiffness matrix calculations only. The element-based formulation allows the implementation of ES-FEM and FS-FEM models in an existing finite element code for any type of finite element because it fits into the existing element-based architecture of FE codes.

The effectiveness of the element-based formulation has been shown by numerical tests and comparison with existing solutions in literature. Finally, the FS-FEM model has been extended to non-tetrahedral elements by formulating shape functions for non-simplex smoothing domains. The ability of the element-based formulation to manage any type of solid element has been shown by numerical examples.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Daniele Colombo https://orcid.org/0000-0001-7249-9486
Ralf Frotscher https://orcid.org/0000-0002-7313-7353
Manfred Staat https://orcid.org/0000-0003-4363-6570

REFERENCES

1. Zeng W, Liu GR. Smoothed finite element methods (S-FEM): an overview and recent developments. *Arch Comput Methods Eng*. 2018;25:397-435. doi:10.1007/s11831-016-9202-3
2. Liu GR, Thoi TN. *Smoothed Finite Element Method*. CRC Press; 2010.
3. Liu GR, Nguyen-Xuan H, Nguyen-Thoi T. A theoretical study on the smoothed FEM (S-FEM) models: properties, accuracy and convergence rates. *Int J Numer Methods Eng*. 2010;84:1222-1256. doi:10.1002/nme.2941
4. Liu GR, Dai KY, Nguyen TT. A smoothed finite element method for mechanics problems. *Comput Mech*. 2007;39:859-877. doi:10.1007/s00466-006-0075-4
5. Liu GR, Nguyen-Thoi T, Nguyen-Xuan H, Lam KY. A node-based smoothed finite element method (NS-FEM) for upper bound solutions to solid mechanics problems. *Comput Struct*. 2009;87:14-26. doi:10.1016/j.compstruc.2008.09.003
6. Liu GR, Nguyen-Thoi T, Lam KY. An edge-based smoothed finite element method (ES-FEM) for static, free and forced vibration analyses of solids. *J Sound Vib*. 2009;320:1100-1130. doi:10.1016/j.jsv.2008.08.027
7. Dai KY, Liu GR, Nguyen TT. An n-sided polygonal smoothed finite element method (nSFEM) for solid mechanics. *Finite Elem Anal Des*. 2007;43:847-860. doi:10.1016/j.finel.2007.05.009
8. Nguyen-Thoi T, Liu GR, Nguyen-Xuan H. An n-sided polygonal edge-based smoothed finite element method (nES-FEM) for solid mechanics. *Int J Numer Methods Biomed Eng*. 2010;27(9):1446-1472. doi:10.1002/cnm.1375
9. Cui X, Liu GR, Li G, Zhang G, Zheng G. Analysis of plates and shells using an edge-based smoothed finite element method. *Comput Mech*. 2010;45:141-156. doi:10.1007/s00466-009-0429-9

10. Nguyen-Thoi T, Liu GR, Lam KY, Zhang GY. A face-based smoothed finite element method (FS-FEM) for 3D linear and geometrically non-linear solid mechanics problems using 4-node tetrahedral elements. *Int J Numer Methods Eng*. 2009;78:324-353. doi:10.1002/nme.2491
11. Bathe KJ. *Finite Element Procedures*. Prentice-Hall; 1996.
12. Zienkiewicz OC, Taylor RL. *The Finite Element Method*. Vol 1. 5th ed. Butterworth-Heinemann; 2000.
13. Cook RD. *Finite Element Modeling for Stress Analysis*. John Wiley & Sons, Inc.; 1995.
14. Liu GR. A G space theory and a weakened weak (W^2) form for a unified formulation of compatible and incompatible methods: Part II applications to solid mechanics problems. *Int J Numer Methods Eng*. 2009;81(9):1127-1156. doi:10.1002/nme.2720
15. Macneal RH, Harder RL. A proposed standard set of problems to test finite element accuracy. *Finite Elem Anal Des*. 1985;1:3-20.
16. EDF R&D, code_aster finite element code. Accessed December 23, 2021. http://www.code-aster.org

How to cite this article: Colombo D, Drira S, Frotscher R, Staat M. An element-based formulation for ES-FEM and FS-FEM models for implementation in standard solid mechanics finite element codes for 2D and 3D static analysis. *Int J Numer Methods Eng*. 2023;124(2):402-433. doi: 10.1002/nme.7126