REVISTA INTERDISCIPLINAR DE PESQUISA EM ENGENHARIA

CILAMCE 2016

XXXVII IBERIAN LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGINEERING
BRASÍLIA - DF - BRAZIL

Development and Implementation of a Well-conditioning Approach Toward Generalized/Extended Finite Element Method into an Object-oriented Platform

Mohammad Malekan

Felicio B. Barros

Ramon P. Silva

malekan@dees.ufmg.br

felicio@dees.ufmg.br

ramon@ufmg.br

Graduate Program in Structural Engineering (PROPEEs), School of Engineering, Federal University of Minas Gerais (UFMG), Av. Antonio Carlos, 6627, Pampulha, 31270-901, Belo Horizonte, MG, Brazil

Abstract. *This paper shows and discusses a generic implementation of the well-conditioning approach toward generalized/extended finite element method. This implementation, performed into an academic computational platform, follows the object-oriented approach presented before for the standard version of GFEM in which the shape functions of finite elements are hierarchically enriched by analytical functions, according to the problem behavior. The stable version of GFEM is employed here to avoid the bad effects of blending elements on the approximate solution convergence rate. Beside this, a* ***sparse*** *matrix data structure is used to accelerate the solution procedure. This approach uses the advantage of sparse matrix to solve the matrix system of equations. The implementations are explained in detail and different aspects of this approach are discussed through numerical examples.*

Keywords: *Generalized/eXtended FEM, Object-oriented programming, Two-scale Analysis, Stable GFEM, Matrix sparsity*

1 INTRODUCTION

During the last three decades, finite element method (FEM) became a powerful and robust method to solve various engineering and applied science problems. However, there are phenomena which the conventional form of the FEM cannot satisfactorily describe. Problems subject to large deformation and crack propagation, which require several level of remeshing are among those that lead to arise the new approaches. One of the method that can easily deal with the discontinuity problem is generalized/extended finite element method (G/XFEM). In G/XFEM (Melenk and Babuska, 1996; Oden et al., 1998; Belytschko and Black, 1999), similar to FEM, the approximation is built over a mesh of elements using interpolation functions. However, the approximation is associated with nodal points. Special functions multiply the original FEM functions and smooth and non-smooth solutions can be modeled independently of the mesh.

In addition, the local enrichments of the G/XFEM approximations lead to arising the blending elements (an element containing both enriched and non-enriched nodes) into the problem domain (Chessa et al., 2003; Gracie et al., 2008; Tarancón et al., 2009; Shibanuma and Utsunomiya, 2009). The presence of these elements results in an arbitrarily ill-conditioned matrix and penalizes the convergence rate of the approximate solution. To address those problems, Menk and Bordas (2011) proposed a method based on pre-conditioners, Laborde et al. (2005); Béchet et al. (2005) used enrichment functions at a fixed region around the problem domain, called geometrical enrichment, which lead to optimal convergence rates in the G/XFEM.

Recently, Babuska and Banerjee (2011, 2012) presented a new approach for one-dimensional domains, so-called Stable GFEM (SG/XFEM here, considering the equivalence between GFEM and XFEM) which involves simple modification of the enrichment functions in order to create an enrichment space that is near-orthogonal to the finite element approximation space, while preserving all the attractive features of the G/XFEM. The SG/XFEM is aiming to improve the conditioning property of the G/XFEM. Another advantage is that the SG/XFEM does not use ramp-functions in the transitory elements between regions of different kinds of enriched functions as proposed by Fries (2008). Babuska and Banerjee (2011, 2012) show that the SG/XFEM is optimally convergent and it has no issues with the blending elements. Gupta et al. (Gupta et al., 2013, 2015) extended one-dimensional SG/XFEM to two- and three-dimensional fracture mechanics.

In Alves et al. (2013), the available FEM programming environment is expanded to enclose the standard version of G/XFEM. This environment, so called INSANE (Interactive Structural Analysis Environment) is an open source software available at http://www.insane.dees.ufmg.br and written in *Java* language. More after, the G/XFEM method is extended to have numerically-built enrichment function within the so-called method global-local G/XFEM (G/XFEM^{gl}) in (Malekan et al., 2016a,b; Malekan and Barros, 2016).

To minimize memory consumption and at the same time speed up the solution of linear equations, one can use existing numerical linear algebra libraries such as *SuiteSparse* (Davis, 2004). It is highly-efficient in terms of memory usage and speed. These libraries are written in `C` language and a *Java* application is performed through the Java Native Interface (JNI), which provides mechanisms for performing data transfer to/from the *Java* and provides rules for adapting name and signature of native methods called from the *Java* application.

In the present paper, a new expansion that includes the Stable GFEM is presented. The implementation, conducted through the development of comprehensive object-oriented design, allows generalization of the approach in such way that any types of partition of unity methods, analysis model and enrichment strategy can be combined. Also, the *SuiteSparse* library is used through the JNI in Andrade and Silva (2015) to accelerate the solution procedure.

2 PROBLEM STATEMENT

The problem considered here consists of a linear elastic cracked domain Ω bounded by the boundary $\partial\Omega$ such that:

$$\partial\Omega = \partial\Omega_t \cup \partial\Omega_u \cup \partial\Omega_c \tag{1}$$

where $\partial\Omega_t$ is the part of the boundary where surface tractions $\bar{t}$ are applied as Neumann conditions, $\partial\Omega_u$ is the part of the boundary where displacements $\bar{u}$ are imposed (as Dirichlet boundary conditions), and $\partial\Omega_c$ is the crack surface. The strong form of the equilibrium equations and boundary conditions can be written as:

$$\nabla \cdot \boldsymbol{\sigma} + \boldsymbol{b} = \mathbf{0} \quad \text{in } \Omega \tag{2}$$

$$\boldsymbol{\sigma} \cdot \hat{\boldsymbol{n}} = \bar{\boldsymbol{t}} \quad \text{on } \partial\Omega_t : \text{ external traction} \tag{3}$$

$$\boldsymbol{\sigma} \cdot \hat{\boldsymbol{n}} = \mathbf{0} \quad \text{on } \partial\Omega_c : \text{ traction free crack} \tag{4}$$

$$\boldsymbol{u} = \bar{\boldsymbol{u}} \quad \text{on } \partial\Omega_u : \text{ prescribed displacement} \tag{5}$$

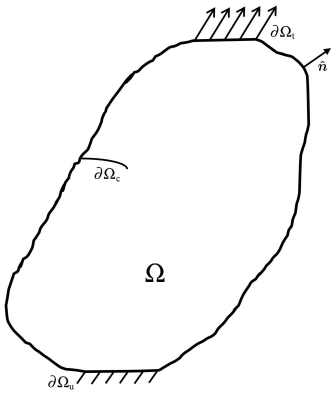


Figure 1: Boundary condition definition on the cracked domain

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor, $\boldsymbol{u}$ is the displacement field vector, $\hat{\boldsymbol{n}}$ is the unit outward normal and $\boldsymbol{b}$ and $\bar{\boldsymbol{t}}$ are the body force and external traction vector. For small strains and displacements, strain-displacement relation can be written as:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}(\boldsymbol{u}) = \nabla_{\boldsymbol{s}} \boldsymbol{u} \quad \text{in } \Omega \tag{6}$$

In the above equation $\nabla_{\boldsymbol{s}}$ is the symmetric part of the gradient operator and $\boldsymbol{\varepsilon}$ is the linear strain tensor. The constitutive relation for linear elastic material is given by Hook's law:

$$\boldsymbol{\sigma} = \boldsymbol{D}\,\boldsymbol{\varepsilon} \tag{7}$$

where $\boldsymbol{D}$ is the Hook's tensor. The variational form of the equilibrium equation can be written as:

$$\int_{\Omega} \boldsymbol{\sigma}(\boldsymbol{u}) : \varepsilon(\boldsymbol{v}) d\Omega = \int_{\Omega} \bar{\boldsymbol{b}} \cdot \boldsymbol{v} d\Omega + \int_{\Gamma_t} \bar{\boldsymbol{t}} \cdot \boldsymbol{v} d\Gamma \tag{8}$$

3 GENERALIZED and STABLE GENERALIZED FEM

The GFEM was developed for modeling structural problems with discontinuities (Melenk and Babuska, 1996; Duarte et al., 2000). Furthermore, it can be considered an instance of the Partition of Unity Method, PUM (Babuska and Melenk, 1997), in the sense that it employs a set of Partition of Unity, PU, functions to guarantee interelement continuity. Such strategy creates conforming approximations which are improved by a nodal enrichment scheme. This basic idea shares the same characteristics from XFEM proposed by Belytschko and Black (1999).

A conventional finite elements mesh can be considered for which $\{\mathcal{K}_e\}_{e=1}^{NE}$ is a set of NE elements, defined by N nodes, $\{\boldsymbol{x}_j\}_{j=1}^{N}$. A generic patch of elements or cloud $\omega_j \in \bar{\Omega}$ is obtained by the union of finite elements sharing the vertex node $\boldsymbol{x}_j$. The assemblage of the interpolation functions, built at each element $\mathcal{K}_e \subset \omega_j$ and associated with node $\boldsymbol{x}_j$, composes the function $\mathcal{N}_j(\boldsymbol{x})$ defined over the support cloud ω_j. As $\sum_{j=1}^{N} \mathcal{N}_j(\boldsymbol{x}) = 1$ at every point $\boldsymbol{x}$ in the domain $\bar{\Omega}$, the set of functions $\{\mathcal{N}_j(\boldsymbol{x})\}_{j=1}^{N}$ constitutes a partition of unity (PU). The generalized finite element shape functions are determined by the enrichment of the PU functions, which is obtained by the product of such functions by each one of the components of the set $\mathcal{I}_j$ at the generic cloud ω_j:

$$\{\phi_{ji}\}_{i=1}^{q} = \mathcal{N}_j(\boldsymbol{x}) \times \{L_{ji}(\boldsymbol{x})\}_{i=1}^{q} \tag{9}$$

The enrichment scheme is obtained by multiplying a PU function of C^0 type with compact support ω_j by the function $L_{ji}(\boldsymbol{x})$, named as a local approximation (also called enrichment function). The resulting shape function $\phi_{ji}(\boldsymbol{x})$ inherits characteristics of both functions, i.e., the compact support and continuity of the PU and the approximate character of the local function.

As a consequence, the generalized global approximation, denoted by $\tilde{\boldsymbol{u}}(\boldsymbol{x})$, can be described as a linear combination of the shape functions associated with each node:

$$\tilde{\boldsymbol{u}}(\boldsymbol{x}) = \sum_{j=1}^{N} \mathcal{N}_j(\boldsymbol{x}) \left\{ \boldsymbol{u}_j + \sum_{i=2}^{q} L_{ji}^{p}(\boldsymbol{x}) \boldsymbol{b}_{ji} + \sum_{\alpha=2} L_{j\alpha}^{s}(\boldsymbol{x}) \boldsymbol{c}_{j\alpha} \right\} \tag{10}$$

where $\boldsymbol{u}_j$ is nodal parameters associated with standard FE shape function, $\mathcal{N}_j(\boldsymbol{x})$, $\boldsymbol{b}_{ji}$ and $\boldsymbol{c}_{j\alpha}$ are nodal parameters associated with GFEM shape functions, $\mathcal{N}_j(\boldsymbol{x}) \cdot L_{ji}(\boldsymbol{x})$, respectively. $L_{ji}^{p}(\boldsymbol{x})$ and $L_{j\alpha}^{s}(\boldsymbol{x})$ are polynomial and singular enrichment functions, respectively. An example of the enrichment function, L_{ji}^{p}, based on polynomials is:

$$L_{ji}^{p}(\boldsymbol{x}) = \left(\frac{x - x_j}{h_j} \right)^m \times \left(\frac{y - y_j}{h_j} \right)^p \tag{11}$$

where (x_j, y_j) are the coordinates of node $\boldsymbol{x}_j$, m and p are degree of polynomials in x and y directions, respectively, and h_j is an scaling factor. Another example for the enrichment function by considering the singularities can be defined as (Barros et al., 2013):

$$\{{}^{x}L_{j\alpha}^{s}(\boldsymbol{x})\}_{\alpha=1} = \frac{A_1}{2G} r^{\lambda_1} \{[\kappa - Q_1(\lambda_1 + 1)] cos\lambda_1\theta - \lambda_1 cos(\lambda_1 - 2)\theta\} \tag{12}$$

$$\{{}^{y}L_{j\alpha}^{s}(\boldsymbol{x})\}_{\alpha=1} = \frac{A_1}{2G} r^{\lambda_1} \{[\kappa + Q_1(\lambda_1 + 1)] sin\lambda_1\theta + \lambda_1 sin(\lambda_1 - 2)\theta\} \tag{13}$$

where λ_1, Q, and A_1 are some coefficients related to the crack and they are a function of relative position of crack to global coordinate system. Also, $\kappa = 3 - 4\nu$ for plane stress analysis and $G = \dfrac{E}{2(1+\nu)}$.

The G/XFEM has many interesting features. However, the patch approximation spaces χ_α, which give rise to the excellent approximation property of the G/XFEM, can also adversely affect its performance. The conditioning of the stiffness matrix $\boldsymbol{K}_{G/XFEM}$ of G/XFEM can be much worse than that of the standard FEM (Gupta et al., 2013). Thus, the stable G/XFEM is considered here in order to keep approximation capability of the G/XFEM and control the conditioning of the system matrices. As already mentioned in the introduction, the SG/XFEM initially proposed in (Babuska and Banerjee, 2011, 2012), and extended to 2D by Gupta et al. (2013) and 3D fracture mechanics by Gupta et al. (2015), provides a robust and simple solution to the problem of ill-conditioning of the G/XFEM. Following (Babuska and Banerjee, 2011, 2012; Gupta et al., 2013, 2015), the SG/XFEM enrichment functions are constructed based on a local modification of the G/XFEM enrichments functions, as following:

$$L_{ji}^{S}(\boldsymbol{x}) = L_{ji}(\boldsymbol{x}) - I_{\omega_j}(L_{ji})(\boldsymbol{x}) \tag{14}$$

with:

$$I_{\omega_j}(L_{ji})(\xi) = \sum_{\alpha \in \mathcal{I}(\tau)} L_{ji}(\boldsymbol{x}_\alpha)\mathcal{N}_\alpha(\xi) \tag{15}$$

where $I_{\omega j}(L_{ji})$ is the piecewise bi-linear FE interpolant of the G/XFEM enrichment function L_{ji} and L_{ji}^{S} is the modified SG/XFEM enrichment function, vector $\boldsymbol{x}_\alpha$ has the coordinates of node α of element τ and $\mathcal{N}_\alpha$ is the piecewise linear FE shape function for node α. Then, similar to G/XFEM formulation (9), the shape function of SG/XFEM can be calculated as:

$$\phi_{ji}^{S}(\boldsymbol{x}) = \mathcal{N}_j(\boldsymbol{x}) \times L_{ji}^{S}(\boldsymbol{x}) \tag{16}$$

By referring to Eqs. (14) and (15), the main additional part for the SG/XFEM method is the evaluation of the finite element interpolant, $I_{\omega j}(L_{ji})$, which its computational evaluation is almost straightforward. In other words, we only need to compute the FE shape functions values at integration points. Further details on the numerical aspects of SG/XFEM and its detailed explanations can be found in (Gupta et al., 2013). So far, the SG/XFEM method has been applied to 2D and 3D dimensional fracture mechanics problems using singular and Heaviside enrichment functions in (Gupta et al., 2013, 2015). Another concept that will be used here is the global-local or two-scale strategy. The global-local G/XFEM originally proposed by Duarte and Babuška (2005), combines the standard G/XFEM with the global-local strategy proposed by Noor (1986). G/XFEM^{gl} is suitable for problems with local phenomena, such as stress field next to the crack tip. The analysis is divided in three steps: *Initial global problem (step 1)* that uses a coarse FEM mesh, *Local problem (step 2)* which uses a refined mesh in a small part of the initial global problem, and the *Final global problem (step 3)* that some of the nodes from initial global problem are enriched using numerical functions calculated in step 2.

4 SuiteSparse MATRIX APPROACH

In order to optimize computational time, the *sparse matrix* can be held in some form of compact data structure that avoids storing the numerically zero entries in the matrix. The two most common

formats for sparse direct methods are the triplet matrix and the compressed-column matrix (or the compressed-row matrix) (Davis, 2004). The simplest sparse matrix data structure is a list of the non-zero entries in arbitrary order, also called the triplet form. However, in the compressed-column format, which is used in INSANE computational framework, each column is represented as a list of values and their corresponding row indices for each column. To create this data structure, the first step counts the number of entries in each column of the matrix, and the column pointer array, as well as its corresponding value, is constructed as the cumulative sum of the column counts. The entries are placed in their appropriate columns in a second step.

Popular choice to solve static finite element problems ($A\boldsymbol{x} = b$) are direct methods that rely on numerical factorizations. Among various direct methods, one can use the unsymmetric multifrontal method in particular (Davis and Duff, 2004; Pais et al., 2012; Zheng and Luo, 2014; Davis et al., 2016). The UMFPACK (Davis, 2009) code is written based on the multifrontal method. This code is chosen to accelerate the processor part mainly because it is well-designed and its source codes are easily available. It already has been used by different commercial and in-house codes, such as ANSYS (2015); NASTRAN (2015); Hecht (2012).

A typical multifrontal algorithm consists of the symbolic and the numerical factorization. In the symbolic factorization stage, an elimination tree is formed. The numerical factorization includes pivoting, assembling, and updating is then performed on each of the frontal matrices. For a frontal matrix, the following transformation is performed (Yu et al., 2011).

$$\begin{bmatrix} P & Q \\ R & C \end{bmatrix} = \begin{bmatrix} L_1 & 0 \\ L_2 & C - L_2 U_2 \end{bmatrix} \begin{bmatrix} U_1 & U_2 \\ 0 & 1 \end{bmatrix} \tag{17}$$

where block P contains all of the pivots in the current supernode, blocks Q and R contain the corresponding nonzero rows and columns of the pivots, $P = L_1 U_1$, $L_2 = R U_1^{-1}$, and $U_2 = L_1^{-1} Q$. Figure 2 shows a schema of the matrices notations for the multifrontal method used in UMFPACK. Also, algorithm 1 is a simplified version of the unsymmetric multifrontal method used by UMFPACK.

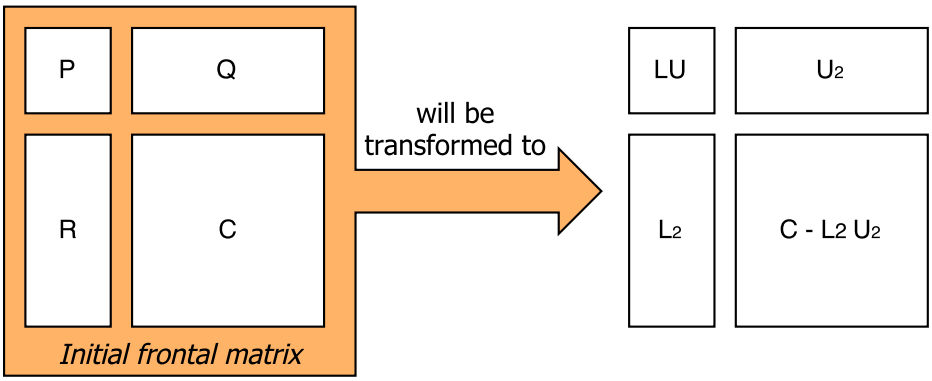


Figure 2: Matrices notations for the multifrontal method

Algorithm 1 General multifrontal numerical factorization approach used in UMFPACK (Davis, 2004; Yu et al., 2011)

```
1:  procedure NUMERICAL FACTORIZATION
2:      Initialization
3:      Symbolic factorization and form the elimination tree
4:      for all frontal matrices do
5:          while factorize the frontal do
6:              Assemble using update matrices and original matrix elements
7:              Perform row pivoting
8:              Factor the pivot to get a part of L1 and U1
9:              Update part of L2
10:         end while
11:         Update U2 and C
12:     end for
13: end procedure
```

5 OOP ENVIRONMENT

The INSANE environment (Fonseca and Pitangueira, 2007; Alves et al., 2013; Malekan et al., 2016a), is an open source software implemented in Java, an OOP language. The INSANE computational environment is composed by three great applications: pre-processor, processor and post-processor. The INSANE numerical core is composed by the interfaces *Assembler*, *Model* and *Persistence* and the abstract class *Solution*. Figure 3 shows the unified modeling language (UML) diagram of the INSANE numerical core.

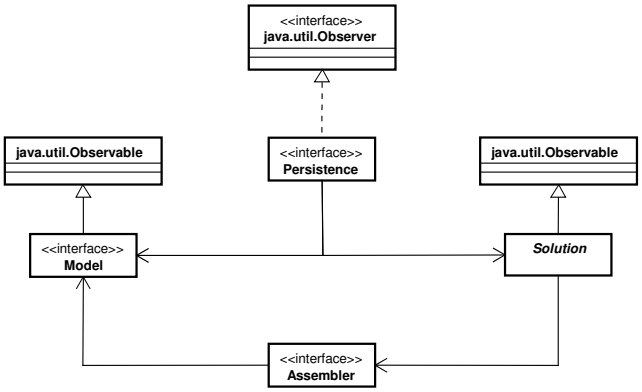


Figure 3: Organization of INSANE numerical core

Different parts of the INSANE numerical core are written and linked to each other to solve the following generic representation of an initial value problem:

$$\boldsymbol{A}\,\ddot{\boldsymbol{X}} + \boldsymbol{B}\,\dot{\boldsymbol{X}} + \boldsymbol{C}\,\boldsymbol{X} = \boldsymbol{D} \tag{18}$$

where $\boldsymbol{X}$ is the solution vector; the single dot represent its first time derivative and the double dots its second time derivative; $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are matrices with the properties of the problem and $\boldsymbol{D}$ is a vector that represents the system excitation. *Solution* abstract class starts the solution process and has the necessary resources for solving the matrix system.

The *EnrichmentType* class is a part of the *Model* interface which provides required information for the enrichment strategy of the G/XFEM method. The *GlobalLocalEnrichment*, *PolynomailEnrichment*, and *CrackEnrichment* (singularity enrichment) extend *EnrichmentType* class to have G/XFEM enrichment functions for various problems.

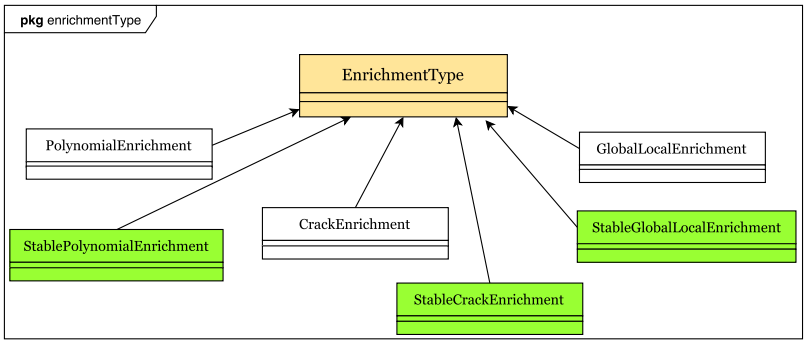


Figure 4: UML diagram of the *EnrichmentType* package

The class *PolynomailEnrichment* holds some arrays to represent the monomial functions used in Eq. (9). On the other hand, the class *CrackEnrichment* holds some necessary parameters to simulate functions with high singularity. *EnrichmentType* class has some generic methods which are responsible to calculate enrichment functions and their derivatives in a point. Each node, instance of an object *Node*, can have a list of the objects of class *EnrichmentType*. This relationship allows the existence of multiple types of enrichment functions for the mesh as a whole and also for each node.

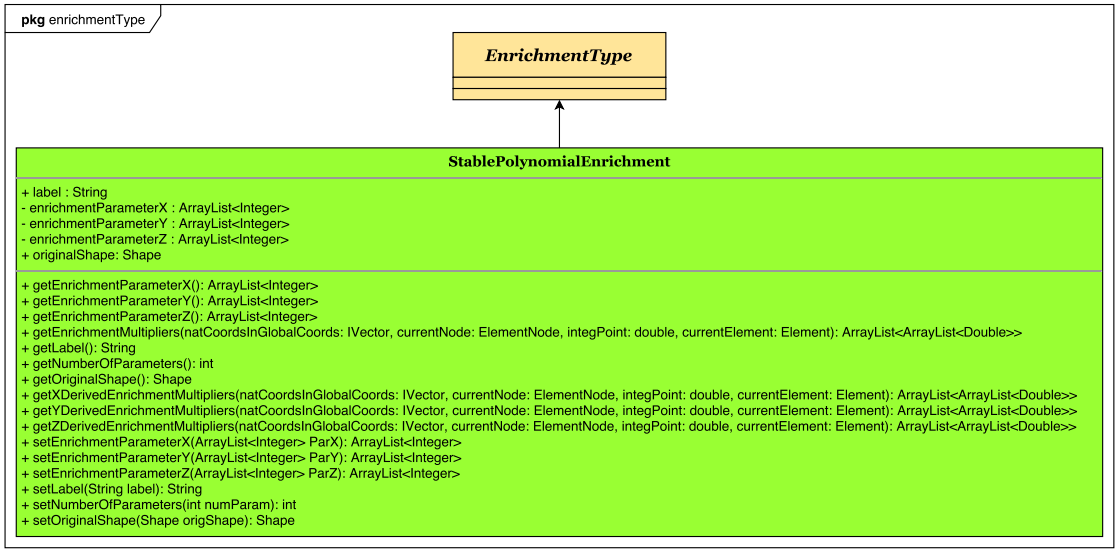


Figure 5: UML diagram of the *StablePolynomialEnrichment* class

Furthermore, to implement the SG/XFEM approach, the *StableGlobalLocalEnrichment*, *Sta-*

blePolynomailEnrichment, and *StableCrackEnrichment* are added under *EnrichmentType* class. The *StableGlobalLocalEnrichment* provides specific methods to build the SGFEM enriched functions from the solution of the local problem and applied in the third step of the global-local problem. Figure 4 shows the *EnrichmentType* class UML diagram. *EnrichmentType* is an abstract class and its methods are abstract. Thus, both G/XFEM and SG/XFEM related enrichment classes contain the same method, as the *EnrichmentType* class. Figures 5, 6, and 7 show the UML diagrams of these new classes. In all UML diagrams, white, cream, and green colors represent unchanged, modified, and new classes, respectively.

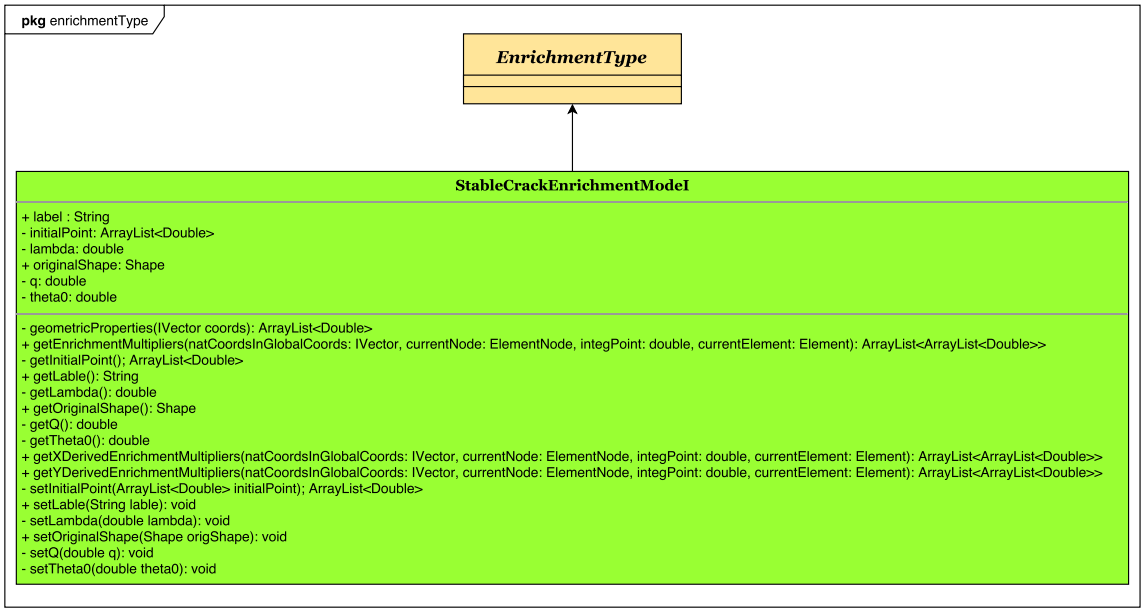


Figure 6: UML diagram of the *StableCrackEnrichment* class

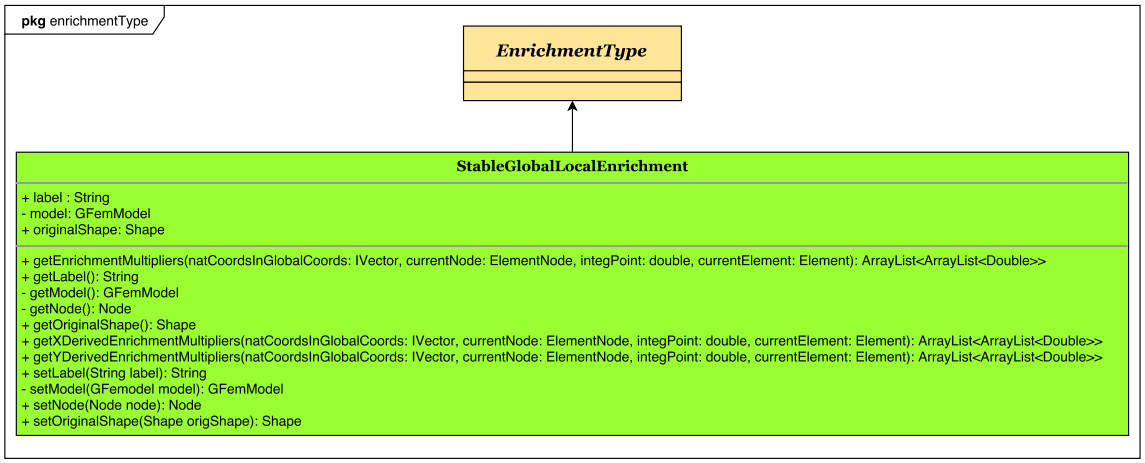


Figure 7: UML diagram of the *StableGloblLocalEnrichment* class

Figure 8 shows the structure of the *SparseMatrix* class that extends the *IMatrix* class (Andrade and Silva, 2015). The main implementation for conducting and utilizing the UMFPACK capabilities

is written in this class. After filling the required parts of this class, the corresponding libraries related to the UMFPACK will be called in order to solve the system of equations.

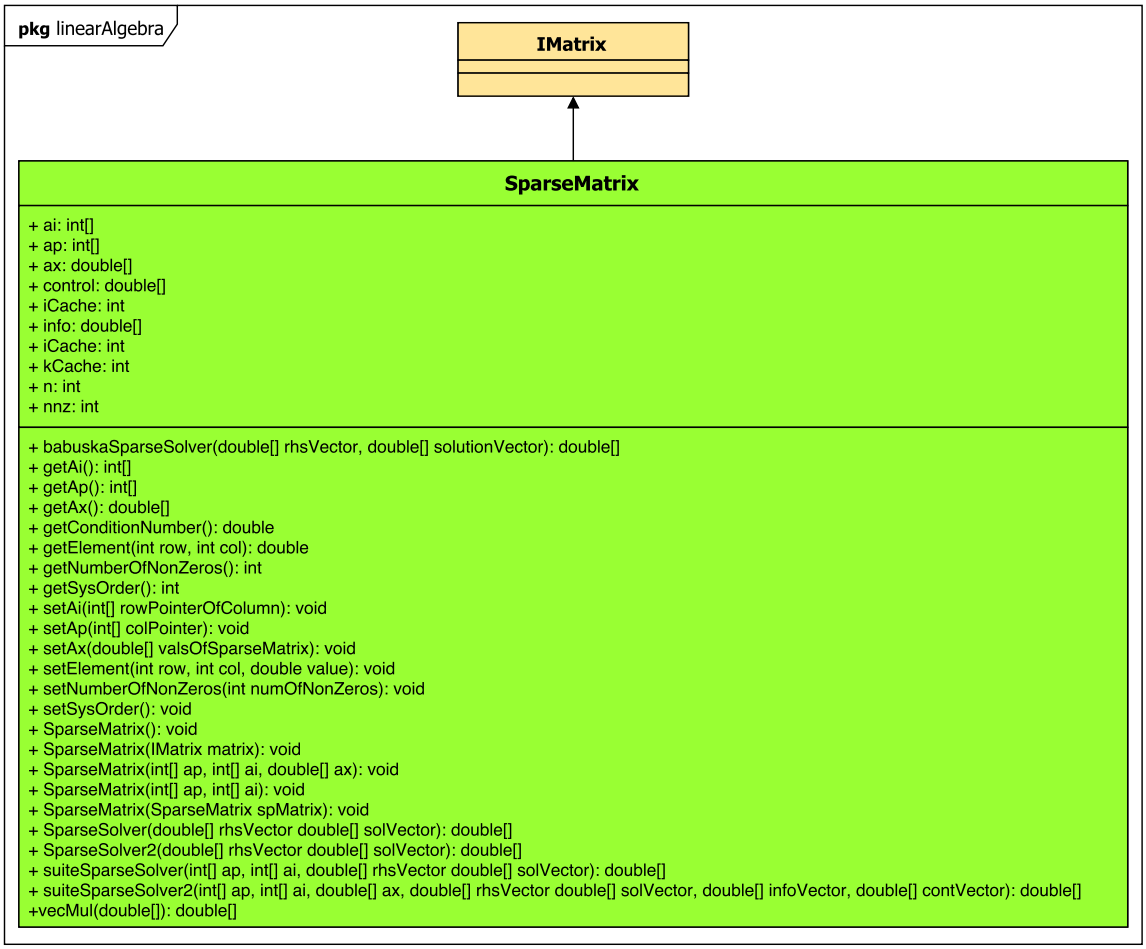


Figure 8: UML diagram of the *SparseMatrix* class

6 NUMERICAL EXAMPLE

This section presents two linear-elastic problems in $\Re^2$.Section 6.1 presents a double-edge cracked plate and section 6.2 presents a plate with an edge crack. The geometry and boundary conditions are very simple and the goal of choosing them is to demonstrate the capabilities of the SG/XFEM method as well as the SG/XFEM^{gl} method. Both problems are analyzed under plane stress state, have the following parameters (in consistent units): modulus of elasticity $E = 1.0$, Poisson ratio $\nu = 0.3$, and the shear stress $\tau = 1.0$. The integration order for double-edge and edge cracked (initial and final global problems and local problem) plates are considered equal to 8×8 and 6×6, respectively.

6.1 Double-edge cracked plate

This example considers a double-edge cracked plate submitted to a shear stress, as shown in Fig. 9. The cracked zone produces singular stress field near the crack tips. The objective of this example is

to illustrate the use of SGFEM with singular enrichment (shown in Eqs. (12) and (13)) for fracture mechanic problems.

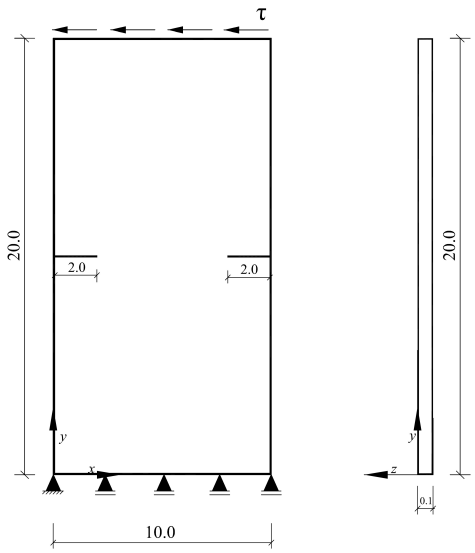
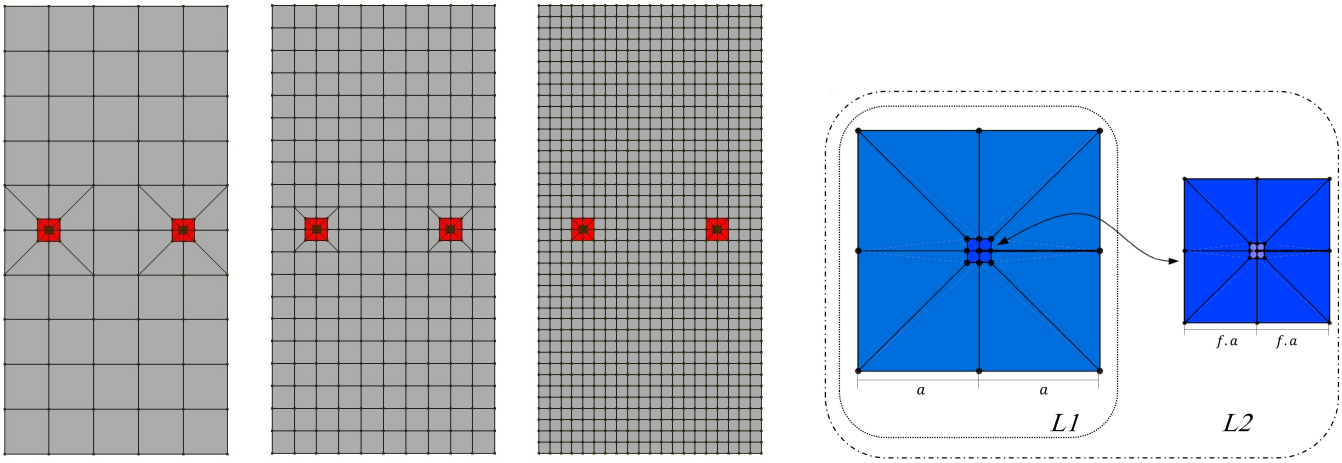


Figure 9: ***Geometry and loading of the double-edge cracked plate***



(a) Three meshes (b) Geometric mesh with reduction rate of f

Figure 10: Double-edge cracked problem meshing strategy.

The reference solution of this problem is obtained using a mesh of 13944 quadrilateral elements (CPS4, a 4-node bilinear plane stress quadrilateral element) in $ABAQUS^{®}$.

Using INSANE, there are three different average element sizes (h) in this study, h = 2.0, 1.0, and 0.5. The total number of elements are 98, 248, and 832 elements with a combination of regular and geometrical mesh distribution considering these three element sizes, as shown in Fig. 10(a). As it can be seen from this figure, only four elements from the cloud associated with the crack tip is discretized with the geometric mesh. The geometric refining approach in this study is based on four refinement level (L4) with f = 10% reduction rate, as shown in Fig. 10(b), according to Szabó and

Babuska (1991). Only nodes **inside** the red area (38 nodes) will be enriched with the polynomial and singular enrichment functions. This is because having an equivalent enrichment zone for all three meshed. Of course, there is no need to geometric mesh when the problem is enriching with singular enrichment, because this enrichment function is able to capture the singularity around the crack tip. But, we decided to have the same mesh for all enrichment cases. The P2 (quadratic) enrichment function is considered here to enrich the problem with the polynomial enrichment. The formulation of this function for x direction is as follows:

$$\phi_j^T(\boldsymbol{x}) = \begin{bmatrix} \mathcal{N}_j(\boldsymbol{x}) & 0 & \left(\frac{x-x_j}{h_j}\right)^2 \mathcal{N}_j(\boldsymbol{x}) & 0 \\ 0 & \mathcal{N}_j(\boldsymbol{x}) & 0 & \left(\frac{x-x_j}{h_j}\right)^2 \mathcal{N}_j(\boldsymbol{x}) \\ & & \left(\frac{y-y_j}{h_j}\right)^2 \mathcal{N}_j(\boldsymbol{x}) & 0 \\ & & 0 & \left(\frac{y-y_j}{h_j}\right)^2 \mathcal{N}_j(\boldsymbol{x}) \end{bmatrix} \tag{19}$$

Similar expression goes for y direction. Although the analytical solution is quadratic, but the enrichment function is cubic and the is no linear polynomials. The reason behind using only quadratic polynomials is that the linear polynomials in the case of SG/XFEM returns zero for enrichment functions. The polynomial enrichment of order one provides null enrichment in Eq. (14), because the FE interpolant I_{ω_j}, used in the stable strategy is also linear. It was highlighted that the FE interpolant is not considered in the SG/XFEM for the linear terms of the polynomial enrichment, so the interpolation is performed only for the polynomials of second degree.

Figure 11 shows the convergence rate results against the inverse of the element size for having different enrichment types. As it can be seen from this figure, both error values and rate of convergences for SG/XFEM shown an improvement over the G/XFEM and FEM approaches. Specially, for having both polynomial and singular enrichment together (poly_crackMixed), the difference between the convergence rate is meaningful, $0.3 - 0.4$ for both FEM and G/XFEM, while $0.4 - 0.9$ for SG/XFEM.

The condition numbers for three methods are illustrated in Fig. 12. Rate of growth in condition numbers for all three methods are almost similar. This is because we only enrich a small part of the problem and with limited number of nodes, so the total system of equations between three method cannot change heavily.

As mentioned earlier, using the *SuiteSparse* library is a powerful tool to accelerate solving procedure of matrix system of equations. Table 1 clearly shows the advantage of using the LU factorization algorithm provided by the *SuiteSparse* library over a so-called conventional approaches (Babuska iterative approach). In INSANE this last one also corresponds to the LU factorization, but it is totally written in *Java* and the stiffness matrix is fully stored. Only corresponding data to the enriching problem with the singular enrichment are brought in this Table.

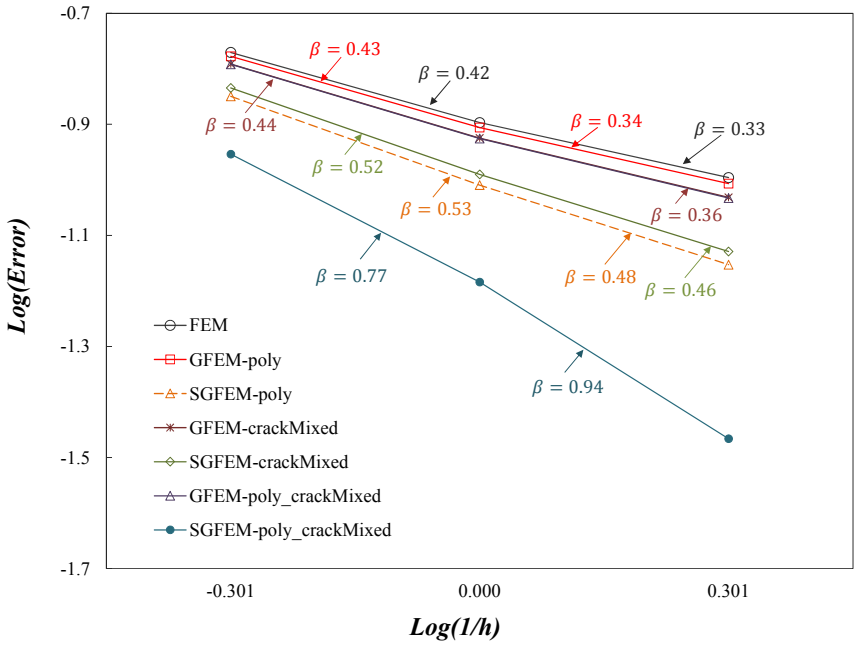


Figure 11: Relative error in strain energy against the inverse of average element size $(1/h)$ for FEM, G/XFEM, and SG/XFEM methods with different enrichment strategies. β refers to the convergence rate.

Table 1: Comparing CPU time between conventional and using the *SuiteSparse* library

	G/XFEM			***SGFEM***		
h size	2	1	0.5	2	1	0.5
DOF	389	718	1940	389	718	1940
CPU time (msec) Babuska iterative method	145	722	6369	139	667	6081
CPU time (msec) SuiteSparse method	8	25	48	8	23	44

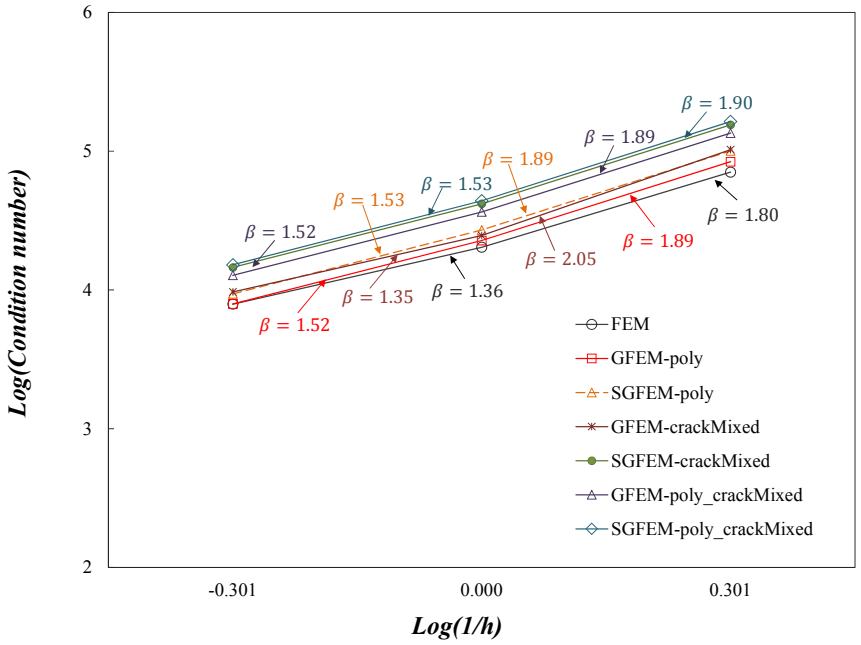


Figure 12: Condition number against the inverse of average element size $(1/h)$ for FEM, GFEM, and SGFEM methods with different enrichment strategies. β is the growth rate of the condition number.

6.2 Plate with an edge crack

This example is completely similar to 6.1, but having only one edge crack. The objective of this problem is to illustrate the use of SG/XFEM as well as global-local Stable G/XFEM for fracture mechanic problems. The reference solution of this problem is obtained using a mesh of 89711 quadrilateral elements (CPS4, a 4-node bilinear plane stress quadrilateral element) in $ABAQUS^{®}$. This problem will be solved for both SG/XFEM and G/XFEM approaches using global-local enrichment function. The Dirichlet boundary condition (a limiting case of Cauchy boundary condition) will be applied on the local problem boundaries. Numerical integration for the first and second steps of the global-local analysis is done based on standard Gaussian quadrature procedure. In the third step, the numerical integration for those global elements that contain local elements is done over the Gauss points of local elements, as proposed by Kim et al. (2010). Consider that a global element contains n^{L_e} local elements and the number of Gauss points for each local element is equal to GP. Thus, the number of integration points for this global element is obtained by: $\sum_{i=1}^{n^{L_e}} GP_i$. The penalty parameter, η, of Dirichlet boundary condition is chosen equal to 1×10^8.

There are three different average element sizes (h) in this study, h = 2.0, 1.0, and 0.5. The total number of elements are 50, 200, and 800 elements with regular distribution considering these three element sizes. The final answer for a problem solved using global-local strategy depends on the size of the local domain as well as the local problem mesh, as shown by Duarte and Kim (2008). Large local domains are preferable, since they give a better numerical solution to the global problem. However, large local domains also increase the computational cost in solving problems. Figure 13 shows the global-local steps and also the local domain discretization for the case of $h = 2.0$.

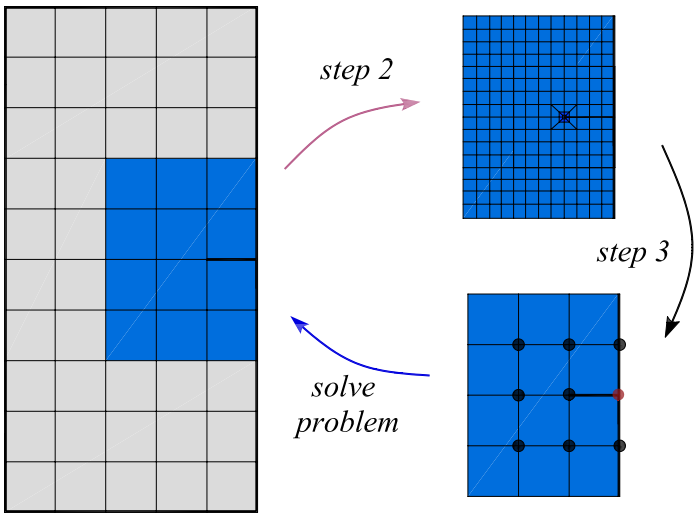


Figure 13: Global-local strategy sequences and local domains discretization for global element size of $h = 2.0$. Black markers represent the nodes will be enriched with global-local enrichment function. Brown marker represents two overlapped nodes.

For the three discretizations, the local problem has exactly the same description with a combination of regular and geometric mesh, similar to the one of section 6.1. Only the four elements from the cloud associated with the crack tip in local problem is discretized with geometric mesh. The geometric refining approach in this example is based on four refinement levels (L4) with f = 10% reduction rate.

Béchet et al. (2005); Laborde et al. (2005) showed that for a optimal convergence rate, the crack tip enrichment must be fixed within a predefined geometry. This approach with a fixed enrichment domain which is independent of the mesh size is called the *geometrical enrichment*. The geometrical enrichment can be used to achieve the optimal convergence rate by enriching the elements located in a predefined area. Figure 14 shows location of enriched nodes for different mesh sizes using geometrical enrichment strategy. The number of enriched nodes for global element size of $h = 2.0$, 1.0, and 0.5 are equal to 10, 27, and 85, respectively.

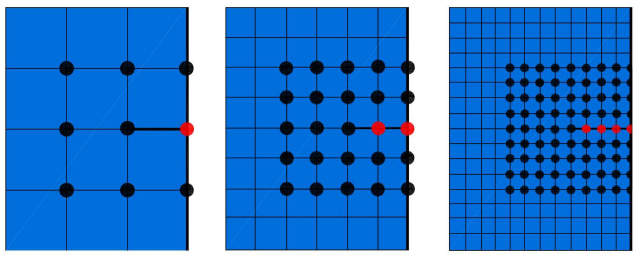


Figure 14: Enriched nodes in the local region from global domain using geometrical enrichment strategy for element size of: $h = 2.0$ (left picture), $h = 1.0$ (middle picture), and $h = 0.5$ (right picture). Again, the red marker indicates that there are two overlapped nodes there.

According to (Babuska and Melenk, 1997), the optimal convergence rate for linear finite element problems with non-smooth solution, like a problem with a crack, the convergence rate is around $\mathcal{O}(h^{1.0})$. As it can seen in Fig. 15, the convergence rates for FEM and G/XFEM^{gl} method are about 0.3 while for SG/XFEM^{gl} is between $0.5 - 1.2$. This shows the capability of the stable G/XFEM to deliver a better convergence rate compared to classical G/XFEM.

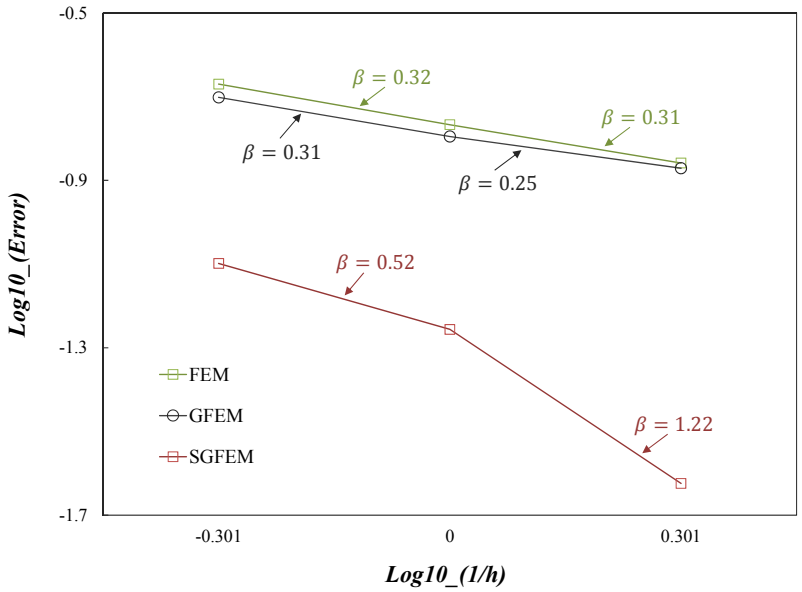


Figure 15: Strain energy error vs. the inverse of element size $(1/h)$ for global problem (third step) in the case of geometrical enrichment. β refers to the convergence rate.

The condition numbers for global-local GFEM/SGFEM methods for geometrical enrichment are illustrated in Fig. 16. When geometrical enrichment is used, condition number corresponding to the G/XFEM^{gl} presents a growth rate of about 3.6, while for SG/XFEM^{gl} is about 2.8. The SG/XFEM result is closer to the FEM one, which is around 2.0.

Again, to show the superiority of the *SuiteSparse* approach over the conventional solver implemented in INSANE, the CPU time of the solution part is brought for this example for both G/XFEM and SG/XFEM with global-local enrichment. Similar to the section 6.1, the *SuiteSparse* approach delivers the solution very fast.

7 CONCLUSIONS

The aim of this work was to present the implementation of Stable G/XFEM for the INSANE computational framework, a FEM object-oriented programming environment. The validation of this implementation and some additional conclusions about the SG/XFEM for different enrichment strategies was presented by numerical examples for solid Mechanics. The SG/XFEM method is implemented and tested for both standard and global-local approaches. In all enrichment cases, the SG/XFEM delivers better convergence rates in strain energy. The current global-local implementation is done only for single iteration. So, if it is modified to include several iterations, a more

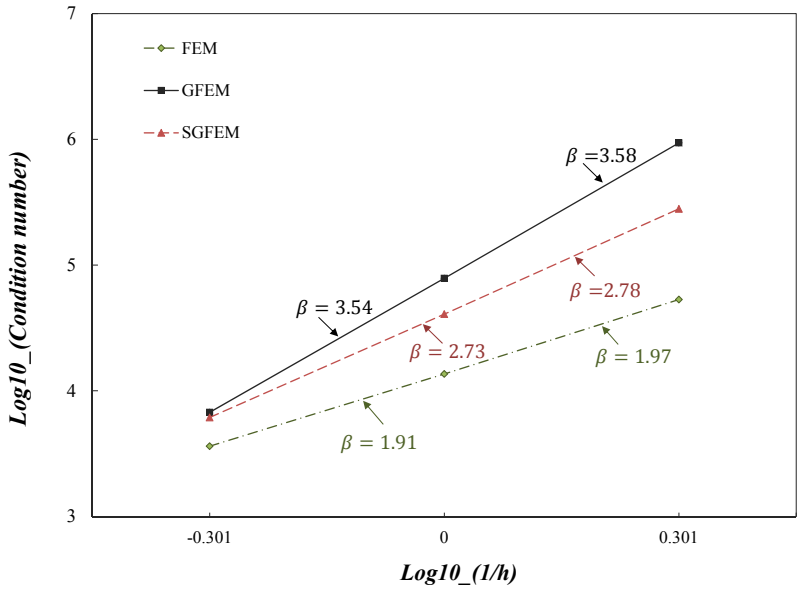


Figure 16: Condition number against the inverse of element size $(1/h)$ for global problem (third step) in the case of geometrical enrichment. β is the growth rate of the condition number.

Table 2: Comparing CPU time between conventional and *SuiteSparse* method

	G/XFEM^{gl}			***SGFEM^{gl}***		
h size	2	1	0.5	2	1	0.5
DOF	154	522	1900	154	522	1900
CPU time (msec) Babuska iterative method	18	976	64587	11	549	43014
CPU time (msec) SuiteSparse method	5	15	41	4	12	37

reliable results than G/XFEM method can be obtained. Also, a *SuiteSparse* library is used to show the advantage of this approach over conventional methods of solving system of equations.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the important support of the Brazilian research agencies CNPq (National Council for Scientific and Technological Developments - Grants 486959/2013-9 and 309005/2013-2) and CAPES (Coordination for the Improvement of Higher Education Personnel).

References

Alves P.D., Barros F.B., and Pitangueira R.L.S. An object oriented approach to the generalized finite element method. *Advances in Engineering Software*, 59:1–18, 2013.

Andrade M.P. and Silva R.P. Implementation of numerical linear algebra libraries via native interface in the insane platform (in Portuguese). In *XXXVI Latin American Congress on Computational Methods, CILAMCE2015*. Rio de Janeiro, Brazil, 2015.

ANSYS. Ansys user's manual, ansys inc. PA, USA. 2015.

Babuska I. and Banerjee U. Stable generalized finite element method (sgfem). Technical Report, Technical Report ICES REPORT 11-07, The Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, Texas, USA, 2011.

Babuska I. and Banerjee U. Stable generalized finite element method (sgfem). *Computer methods in applied mechanics and engineering*, 201-204:91–111, 2012.

Babuska I. and Melenk J.M. The partition of unity method. *International Journal for Numerical Methods in Engineering*, 40:727–758, 1997.

Barros F.B., de Barcellos C.S., Duarte C.A., and Torres D.F. Subdomain-based error techniques for generalized finite element approximations of problems with singular stress fields. *Computational Mechanics*, 52:1395–1415, 2013.

Béchet E., Minnebo H., Moës N., and Burgardt B. Improved implementation and robustness study of the x-fem for stress analysis around cracks. *International Journal for Numerical Methods in Engineering*, 64:1033–1056, 2005.

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

Chessa J., Wang H., and Belytschko T. On the construction of blending elements for local partition of unity enriched finite elements. *International Journal for Numerical Methods in Engineering*, (57):1015–1038, 2003.

Davis T.A. A column pre-ordering strategy for the unsymmetric-pattern multifrontal method. *ACM Transations on Mathematical Software*, 30(2):165–195, 2004.

Davis T.A. Umfpack version 5.4.3 user guide. 2009.

Davis T.A. and Duff I. Algorithm 832: UMFPACK V4.3-an unsymmetric-pattern multifrontal method. *ACM Transations on Mathematical Software*, 30(2):196–199, 2004. doi:10.1145/992200.992206.

Davis T.A., Rajamanickam S., and Sid-Lakhdar W.M. A survey of direct methods for sparse linear systems. *Acta Numerica*, 25:383–566, 2016.

Duarte C.A., Babuska I., and Oden J.T. Generalized finite element methods for three-dimensional structural mechanics problems. *Computers & Structures*, 77(2):215–232, 2000.

Duarte C.A. and Babuška I.M. A global-local approach for the construction of enrich. func. for the gfem and its application to propagating three-dimensional cracks. Technical Report, ECCOMAS Thematic Conference on Meshless Methods, 2005.

Duarte C.A. and Kim D.J. Analysis and applications of a generalized finite element method with global-local enrichment functions. *Computer Methods in Applied Mechanics and Engineering*, 197:487–504, 2008.

Fonseca F.T. and Pitangueira R.L.S. An object oriented class organization for dynamic geometrically non-linear. In *CMNE/CILAMCE (Congress on Numerical Methods in Engineering)*. Porto, Portugal, 2007.

Fries T.P. A corrected xfem approximation without problems in blending elements. *International Journal for Numerical Methods in Engineering*, 75(5):503–532, 2008.

Gracie R., Wang H., and Belytschko T. Blending in the extended finite element method by discontinuous galerkin and assumed strain methods. *International Journal for Numerical Methods in Engineering*, 74:1645–1669, 2008.

Gupta V., Duarte C.A., Babuska I., and Banerjee U. A stable and optimally convergent generalized fem (sgfem) for linear elastic fracture mechanics. *Computer methods in applied mechanics and engineering*, 266:23–39, 2013.

Gupta V., Duarte C.A., Babuska I., and Banerjee U. Stable gfem (sgfem): Improved conditioning and accuracy of gfem/xfem for three-dimensional fracture mechanics. *Computer methods in applied mechanics and engineering*, 289:355–386, 2015.

Hecht F. New development in freefem++. *Journal of Numerical Mathematics*, 20(3-4):251–265, 2012.

Kim D.J., Pereira J.P., and Duarte C.A. Analysis of three-dimensional fracture mechanics problems: A two-scale approach using coarse-generalized fem meshes. *International Journal for Numerical Methods in Engineering*, 81:335–365, 2010.

Laborde P., Renard J.P.Y., and Salaün M. High-order extended finite element method for cracked domains. *International Journal for Numerical Methods in Engineering*, 64:354–381, 2005.

Malekan M. and Barros F.B. Well-conditioning global-local analysis using stable generalized/extended finite element method for linear elastic fracture mechanics. *Computational Mechanics*, pages 1–13, 2016.

Malekan M., Barros F.B., Pitangueira R.L.S., and Alves P.D. An object-oriented class organization for global-local generalized finite element method. *Latin American Journal of Solids and Structures*, 13(13):2529–2551, 2016a.

Malekan M., Barros F.B., Pitangueira R.L.S., Alves P.D., and Penna S.S. A computational framework for a two-scale generalized/extended finite element method: generic imposition of boundary conditions. *Engineering Computations*, pages 1–63, 2016b.

Melenk J.M. and Babuska I. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 39:289–314, 1996.

Menk A. and Bordas S.P.A. A robust preconditioning technique for the extended finite element method. *International Journal for Numerical Methods in Engineering*, 85:1609–1632, 2011.

NASTRAN. Msc nastran reference manual, msc software inc, USA. 2015.

Noor A.K. Global-local methodologies and their application to nonlinear analysis. *Finite Elements in Analysis and Design*, 2:333–346, 1986.

Oden J.T., Duarte C.A., and Zienkiewicz O.C. A new cloud-based hp finite element method. *Computer Methods in Applied Mechanics and Engineering*, 153:117–126, 1998.

Pais M.J., Yeralan S.N., Davis T.A., and Kim N.H. An exact reanalysis algorithm using incremental cholesky factorization and its application to crack growth modeling. *International Journal for Numerical Methods in Engineering*, 91:1358–1364, 2012. doi:10.1002/nme.4333.

Shibanuma K. and Utsunomiya T. Reformulation of xfem based on pufem for solving problem caused by blending elements. *Finite Element in Analysis and Design*, 45(11):806–816, 2009.

Szabó B. and Babuska I. *Finite Element Analysis*. John Wiley & Sons, Inc., 1991.

Tarancón J.E., Vercher A., Giner E., and Fuenmayor F.J. Enhanced blending elements for xfem applied to linear elastic fracture mechanics. *International Journal for Numerical Methods in Engineering*, 77:126–148, 2009.

Yu C.D., Wang W., and Pierce D. A CPU-GPU hybrid approach for the unsymmetric multifrontal method. *Parallel Computing*, 37:759–770, 2011.

Zheng A.X. and Luo X.Q. Numerical study of quasi-static crack growth problems based on extended finite element method. *Journal of Shanghai Jiaotong University*, 19(6):736–746, 2014. doi: 10.1007/s12204-014-1557-8.