A computational framework for a two-scale generalized/extended finite element method: generic imposition of boundary conditions

Mohammad Malekan\textsuperscript{a}, Felício B. Barros\textsuperscript{a}, Roque L. S. Pitangueira\textsuperscript{a}, Phillipe D. Alves\textsuperscript{b}, Samuel S. Penna\textsuperscript{a}

\textsuperscript{a}Graduate Program in Structural Engineering (PROPEEs), School of Engineering, Federal University of Minas Gerais (UFMG), Av. Antônio Carlos 6627, Pampulha, 31270-901, Belo Horizonte, MG, Brazil
\textsuperscript{b}Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Newmark Laboratory, 205 North Mathews Avenue, Urbana, IL 61801, USA

Abstract

This paper presents a computational framework to generate numeric enrichment functions for two-dimensional problems dealing with single/multiple local phenomenon. The two-scale generalized/extended finite element method (G/XFEM) approach used here is based on the solution decomposition, having a global and local scale components. This strategy allows the use of a coarse mesh even when the problem produces complex local phenomena. For this purpose, local problems can be defined where these local phenomena are observed and are solved separately using fine meshes. The results of the local problems are used to enrich the global one improving the approximate solution. The implementation of the two-scale G/XFEM formulation follows the object-oriented approach presented by the authors in a previous work, where it is possible to combine different kinds of elements and analysis models with the partition of unity enrichment scheme. Beside the extension of the G/XFEM implementation to enclose the global-local strategy, the imposition of different boundary conditions is also generalized. This generalization is very important since the global-local approach relies on the boundary information transferring process between the two scales of the analysis. The flexibility for the numerical analysis of the proposed framework is illustrated by several examples. Different analysis models, element formulations and enrichment functions are employed and the accuracy, robustness and computational efficiency are demonstrated.

Keywords: Generalized FEM; eXtended FEM; Object-oriented design; Two-scale analysis; Boundary condition

1. Introduction

Nowadays, it is well accepted to solve a wider range of practical problems using numerical approaches. Finite Element Method (FEM) is the most widely used numerical method to
solve various physical problems and it is available in a wide range of commercial packages. Fracture mechanics problems present non-smooth solutions that can demand a sequence of remeshing that may become a costly operation, especially when the discontinuity is located within a complex geometry. Generalized and eXtended Finite Element Method (G/XFEM), (Melenk and Babuska, 1996, Strouboulis et al., 2000b, Duarte et al., 2000, Belytschko and Black, 1999), is a good alternative to deal with these problems. The generalized and extended finite element methods are basically identical methods (Belytschko et al., 2009), so hereafter the term G/XFEM will be used to refer to these methods. In the G/XFEM, similar to FEM, the approximation is built over a mesh of elements using interpolation functions. However, the approximation is associated with nodal points as in meshless methods, and it is enriched in the same fashion as in \(hp\)-Cloud method (Duarte and Oden, 1995, Duarte, 1996, Liszka et al., 1996, Oden et al., 1998). Special functions multiply the original FEM functions and smooth and non-smooth solutions can be modeled independently of the mesh. This functions can be analytically known or numerically generated, as observed by Strouboulis et al. (2001).

Global-local FEM was proposed by Noor (1986) in order to solve non-linear problems. A local problem is defined where a local phenomenon exists. The global-local FEM approach has two steps. The first step is done with a coarse FEM mesh that ignores the effect of the local phenomena. This is followed by the second step which includes an analysis of the local region using refined finite element meshes. In global-local G/XFEM (Duarte et al., 2007) G/XFEM\textsuperscript{gl}, enrichment functions are constructed numerically from the solution of a local problem and three steps are required. The first and second steps are similar to the one from the global-local FEM. Similar to the FEM analysis, the key parameter for the local analyses is the application of field state variables from the first step as boundary conditions on the local boundaries. In the third step, the results of the local problem are used to enrich the partition of unity functions used in the global problem which improves the approximate solution. The great advantage is providing a well-refined description of the local problem without computationally overburden the analysis of the global problem.

Numerical experiments performed by Duarte and Kim (2008), Kim et al. (2009) and Kim et al. (2010) have already shown computational efficiency and accuracy of the G/XFEM\textsuperscript{gl} strategy, but implementation is not the main focus of these works. Kim et al. (2010) applied Dirichlet, Neumann and spring boundary conditions to the local problem in contrast to applying just Dirichlet boundary condition in Duarte et al. (2007), Duarte and Kim (2008). Comparison of the effectiveness of global-local FEM with the G/XFEM\textsuperscript{gl} was done by Kim et al. (2008) in which their numerical experiments demonstrate that the G/XFEM\textsuperscript{gl} is much more robust than the global-local FEM. Kim et al. (2011) proposed a parallel G/XFEM approach that uses customized enrichment functions for applications where a limited priori knowledge about the solution is available. The methodology involves the parallel solution of local boundary value problems using boundary conditions from a coarse global problem. The global-local strategy
based on G/XFEM approach is applied to high-cycle fatigue crack growth in 3D bodies by Pereira et al. (2011). Their coarse-scale mesh in the G/XFEM\textsuperscript{gl} doesn’t need to model the crack surface explicitly. Instead, the cracks are modeled through global-local enrichment functions. Gupta et al. (2012) presented a priori error estimate accounting for the effect of inexact boundary conditions applied to local problems using G/XFEM\textsuperscript{gl} approach in order to solve crack propagation problems. Application of the G/XFEM with global-local enrichment function in order to analyze nonlinear fracture mechanics problems have been shown by Kim et al. (2012). Gupta et al. (2013) presented extensions of the 2D G/XFEM\textsuperscript{gl} proposed by Duarte and Kim (2008), Kim et al. (2010) to numerically generate enrichment functions for 3D fracture problems with confined plasticity. A two-scale approach using the G/XFEM\textsuperscript{gl} applied to multi-site cracking problems was presented by Evangelista et al. (2013) where realistic boundary conditions are applied and multiple cracks with different geometries in a three-dimensional airfield slab are considered. They concluded that the proposed G/XFEM\textsuperscript{gl} is capable to tackle multi-site cracking problems by avoiding refined crack front meshes in the global part as well as numerical round-off errors. Plews and Duarte (2014) used an interdependent solution of global and local problems in order to resolve multi-scale effects due to fine-scale heterogeneities under G/XFEM strategy. Also, they have highlighted the efficiency of the G/XFEM\textsuperscript{gl} method.

This paper presents a generic implementation of the G/XFEM\textsuperscript{gl} based on object-oriented concept that follows the literature presented here. The application of object-oriented programming for FEM implementation has been receiving great attention over the last decade. As a result, a bunch of G/XFEM codes used object-oriented concept as their implementation strategy (Strouboulis et al., 2001, 2000a, Bordas et al., 2007, Dunant et al., 2007, Nistor et al., 2008, Chamrová and Patzák, 2010, Neto et al., 2013). In Alves et al. (2013a), the FEM programming environment proposed by Fonseca and Pitangueira (2007) is expanded to enclose the standard version of G/XFEM. This computational framework, so called INSANE (Interactive Structural Analysis Environment) is an open source software available at http://www.insane.dees.ufmg.br. In the present paper, a new expansion that includes the G/XFEM\textsuperscript{gl} is presented, extending the ideas firstly discussed in (Alves et al., 2013b, Malekan et al., 2016). The implementation, conducted through the development of comprehensive object-oriented design, allows generalization of the global-local approach in such way that any types of partition of unity methods, analysis model and enrichment strategy can be combined. Also, imposition of the boundary conditions is generalized in order to have different boundary condition types to be transferred from the global problem and applied on the local problem boundaries. This generalization allows that stress and strain field can be applied as boundary conditions for local problems in addition to the capability of applying displacement boundary condition. Details of the implementation are discussed and important aspects of using these strategies are highlighted in the numerical examples. The outline of the paper is as follows. A brief explanation of G/XFEM and G/XFEM\textsuperscript{gl} are presented in section 2. The INSANE class organization
and its expansion to include G/XFEM\textsuperscript{gl} method are discussed in section 3. In the section 4, numerical examples are presented, and the final section is devoted to concluding remarks and discussion.

2. G/XFEM and G/XFEM\textsuperscript{gl}, a brief explanation

The G/XFEM was developed for modeling structural problems with discontinuities (Melenk and Babuska, 1996, Duarte et al., 2000, Belytschko and Black, 1999). Furthermore, it can be considered an instance of the Partition of Unity (PU) method, in the sense that it employs a set of partition of unity functions to guarantee interelement continuity. Such strategy creates conforming approximations which are improved by a nodal enrichment scheme. The G/XFEM covers a wide range of scientific and engineering problems which include fracture, dislocations, inclusions, and multiscale problems.

G/XFEM approximation trial spaces consist of patches of elements or clouds, a partition of unity, and the patches of elements or cloud approximation spaces. A conventional finite elements mesh can be considered for which \( \{K_e\}_{e=1}^{NE} \) is a set of \( NE \) elements, defined by \( N \) nodes, \( \{x_j\}_{j=1}^N \). A generic patch of elements or cloud \( \omega_j \in \Omega \) is obtained by the union of finite elements sharing the vertex node \( x_j \). The assemblage of the interpolation functions, built at each element \( K_e \subset \omega_j \) and associated with node \( x_j \), composes the function \( N_j(x) \) defined over the support cloud \( \omega_j \). As \( \sum_{j=1}^N N_j(x) = 1 \) at every point \( x \) in the domain \( \bar{\Omega} \), the set of functions \( \{N_j(x)\}_{j=1}^N \) constitutes a PU. A set of \( q \) linearly independent functions is defined at each cloud \( \omega_j \) as:

\[
I_j \overset{\text{def}}{=} \{L_{j1}(x), L_{j2}(x), \ldots, L_{jq}(x)\} = \{L_{ji}(x)\}_{i=1}^q \quad L_{j1}(x) = 1
\] (1)

The generalized FE 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 \( I_j \) at the generic cloud \( \omega_j \):

\[
\{\phi_{ji}\}_{i=1}^q = N_j(x) \times \{L_{ji}(x)\}_{i=1}^q
\] (2)

The enrichment function is obtained by multiplying a PU function of \( C^0 \) type with compact support \( \omega_j \) by the function \( L_{ji}(x) \). The resulting shape function \( \phi_{ji}(x) \), inherits characteristics of both functions, i.e., the compact support and continuity of the PU and the approximate character of the local function. Further details can be found in (Duarte et al., 2000, Oden et al., 1998, Strouboulis et al., 2001, Melenk and Babuška, 1995, Babuska and Melenk, 1997).

For the global-local analysis, the G/XFEM\textsuperscript{gl} method combines the standard G/XFEM with the global-local strategy proposed by Noor (1986) to numerically build functions that can accurately describe localized behavior and use them to enrich the global solution space. The analysis is divided in three steps:
– **Coarse-scale problem (step 1):**

A coarse FEM mesh is used throughout the whole domain. Figure 1 exemplifies this step for a problem with several cracks. The position of these cracks can be either between the element edges or inside of the element boundaries, over their areas (Alves et al., 2013b). Consider a domain $\bar{\Omega}_G = \Omega_G \cup \partial \Omega_G$ of an elastic problem in $\mathbb{R}^n$. The boundary is decomposed in $\partial \Omega_G = \partial \Omega^u_G \cup \partial \Omega^\sigma_G$ with $\partial \Omega^u_G \cap \partial \Omega^\sigma_G = \emptyset$, where indices $u$ and $\sigma$ refer to the Dirichlet and Neumann boundary conditions. $u^0_G \in X^0_G(\Omega_G)$ represents the solution of the approximate space $X^0_G(\Omega_G)$ (built by FEM or G/XFEM functions) for the initial global problem in its weak form, shown in:

$$
\int_{\Omega_G} \sigma(u^0_G) : \varepsilon(v^0_G) \, dx = \int_{\partial \Omega^\sigma_G} \tilde{t} \cdot v^0_G \, ds \quad (3)
$$

where $\sigma$, $\varepsilon$, $v^0_G \in X^0_G(\Omega_G)$, and $\tilde{t}$ are stress tensor, strain tensor, test functions, and prescribed traction vector, respectively.

![Figure 1: Typical problem with several local domains](image)

– **Fine-scale problem (step 2):**

A refined mesh is used in a small part of the initial global problem. Figure 2 shows an example where each crack composes a local problem. $\Omega_L$ is a sub-domain from $\Omega_G$. This sub-domain may contain cracks, holes or other special features, as suggested by Duarte and Kim (2008). The corresponding local solution $u_L \in X^E_L(\Omega_L)$ is obtained from:

$$
\int_{\Omega_L} \sigma(u_L) : \varepsilon(v_L) \, dx + \int_{\partial \Omega_L \cap \partial \Omega^u_G} u_L \cdot v_L \, ds + \int_{\partial \Omega_L \setminus (\partial \Omega_L \cap \partial \Omega_G)} u_L \cdot v_L \, ds \\
= \int_{\partial \Omega_L \cap \alpha \Omega^\sigma_G} \tilde{t} v_L \, ds + \int_{\partial \Omega_L \cap \partial \Omega^u_G} \tilde{u} \cdot v_L \, ds + \int_{\partial \Omega_L \setminus (\partial \Omega_L \cap \partial \Omega_G)} (t(u^0_G) + \kappa u^0_G) \cdot v_L \, ds \quad (4)
$$

where $v_L \in X^E_L(\Omega_G)$ represents the test functions, $X^E_L(\Omega_G)$ is the space generated by FEM or G/XFEM functions (the super index E comes from the enrichment space designation),
η is the penalty parameter and κ is the stiffness parameter to consider Cauchy boundary condition. This space can be defined as:

\[
X^E_L(\Omega_L) = \left\{ \tilde{u}(x) = \sum_{j=1}^{N_L} N_j(x) [\hat{u}^E_j(x) + \mathcal{H}\tilde{u}^E_j(x) + \tilde{u}^E_j(x)] \right\}
\]

functions \( \hat{u}^E_j(x) \), \( \mathcal{H}\tilde{u}^E_j(x) \) and \( \tilde{u}^E_j(x) \) are continuous, discontinuous and singular components of the approximate solution and \( N_j \) is the partition of unity function used in global problem. The number of nodes of the local domain is given by \( N_L \).

**Enriched global problem (step 3):**

Some of the nodes from initial global problem are enriched using numerical functions calculated in step 2. The global problem is enriched by solution \( u_L \) (from step 2). The new solution \( u^E_G \in X^E_G(\Omega_G) \) is obtained from:

\[
\int_{\Omega_G} \sigma(u^E_G) : \varepsilon(v^E_G) dx = \int_{\partial\Omega_G} \bar{t} \cdot v^E_G ds
\]

where \( v^E_G \) represents the test functions and \( X^E_G(\Omega_G) \) is the initial space increased by \( u^gl_k(x) \) from local problem \( u_L \):

\[
X^E_G(\Omega_G) = \left\{ \tilde{u}(x) = \sum_{j=1}^{N} N_j(x) [\hat{u}^E_j(x) + \sum_{k \in I^{gl}} N_k(x) u^gl_k(x)] \right\}
\]

and \( k \in J^{gl} \) represent the set of nodes enriched by the local solution and \( u^gl_k(x) \) is the cloud-wise function obtained from the local solution \( u_L \) from Eq. (4).

Figure 2: Several local problems
In G/XFEM<sup>gl</sup>, numerical solution produced by local analysis can be easily introduced into the global problem. The local problem can be strongly refined and it does not affect the computational performance of the problem. Therefore, the problem is solved using less memory and processing machine than standard G/XFEM, as observed by Duarte and Kim (2008) and Kim et al. (2010).

There are two different approaches to represent the discontinuity in this kind of problems: putting the discontinuity in both global and local problem (either geometrically or using a discontinuous enrichment function to represent the discontinuity), or putting the discontinuity only in the local problem.

3. INSANE Computational Environment

The INSANE environment (Fonseca and Pitangueira, 2007, Alves et al., 2013a), is an open source software. The INSANE computational environment is composed by three great applications: pre-processor, processor and post-processor. Here, a summary of several modules of the numerical core application (the processor) are presented aiming to introduce the INSANE system, corresponding to the standard FEM/GFEM approach, and also to show the generalization performed here to enclose the new implementation on the G/XFEM<sup>gl</sup> formulation and also the generalization of the imposition of boundary conditions. More information on G/XFEM implementation and general framework of the G/XFEM<sup>gl</sup> implemented in INSANE can be found in Alves et al. (2013a) and Malekan et al. (2016), respectively.

3.1. Object Oriented Design

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 from numerical core of INSANE. Here, the focus is on the main modifications in order to expand the computational system to comprise new capabilities into the G/XFEM<sup>gl</sup>.

Persistence interface treats the input data and persists the output data. For G/XFEM<sup>gl</sup>, this class was extended to deal with more than one Model. In that case, the data is separated in a global model and several local models, corresponding to the global and the several local problems respectively. Assembler interface is responsible for assembling the linear equation system provided by the discretization of the initial value problem. This class is implemented following the generic representation:

\[
A \ddot{X} + B \dot{X} + C X = D
\]  

(8)

where \(X\) is the solution vector; the single dot represent its first time derivative and the double dots its second time derivative; \(A, B\) and \(C\) are matrices with the properties of the problem and \(D\) is a vector that represents the system excitation. In static analysis, Eq. (8) is simplified by eliminating the two first terms. The resulting matrix system is:
where Eq. (9) is used to represent each step of the global-local solution given by Eqs. (3), (4), and (6). In these equations, the matrix $C$ is the model stiffness matrix, $X$ is the vector of nodal displacements, $D$ is the vector of forces. The vector $D$ is composed of two parts: the vector $N$ and the vector $E$ which are the forces vector applied directly to the nodes and the forces/displacements prescribed by equivalent nodal vector, respectively. The sub-indices $u$ and $p$ informs if the vector is unknown or prescribed.

Solution abstract class starts the solution process and has the necessary resources for solving the matrix system. G/XFEM\textsuperscript{gl} process is implemented by GlobalLocal class. Since G/XFEM\textsuperscript{gl} is composed by more than one model (one global model and several local models), the problem has more than one assembler, each one responsible to build the corresponding equation in the form of (9). This is the main difference from standard G/XFEM solution class. Figure 4 shows in details the UML diagram for this class.

The Model interface contains the data of the discrete model and provides to Assembler informations to assemble the final matrix system. Both Model and Solution communicate with the Persistence interface, which treats the input data and persists the output data to the other applications, whenever it observes a modification of the discrete model state. This interface consists of the Node, Element, EnrichmentType, ProblemDriver classes. The Node class is designed to manage the geometric representation of a node entity as well as the information from the discrete model. The special characteristics of GFEM\textsuperscript{gl}, such as the relationship between...
the global and the local elements, are handled with a new extension of the *Element* class, called *GFemElement*. Figure 5 shows the new parameters added in *GFemElement* class in order to add the ability to solve problems using the global-local scheme.

The fields of this class are: if the element belongs to a local problem, *LOCAL_NAME* indicates in which local model it is inserted, *GLOBAL_ELEMENT* indicates which element from the global model contains the current element. If the element belongs to a global problem, *LOCAL_ELEMENTS* indicates the elements of the local problem that discretize the domain of this global element, *BOUNDARY_INFORMATION* informs which kind of boundary conditions is transferred from step 1 to step 2, *STEP_GL* informs in which step of the problem the solution is being processed, and if the element belongs to a local problem, *LOCAL_TWINS* indicates which element, if there is, from another local problem coincides with it.

Additionally, *GlobalLocalEnrichment* extends *EnrichmentType*, an abstract class from which derived classes enable different types of enrichment strategies. *GlobalLocalEnrichment* is one of these classes and provides specific methods to build the functions used to enrich the PU of the global problem in the third step of the analysis. Those function are built from the numerical
solution of the local problem (second step). *ProblemDriver* class informs the *Assembler* all the necessary data for assembling the final system of the model. For GFEM\textsuperscript{gl} case, an additional loop is performed using local elements of each global element, since they play the role of integration cells. Actually, information of each local element will be used to enrich corresponding global element based on relationship between second and third steps of the global-local strategy. Further information on preliminary global-local implementation can be found in (Malekan et al., 2016).

### 3.2. Generic boundary condition

Figure 6 shows the relationship between the types of objects found in the package *Value*, which is used to define a generic boundary condition of the problem (e.g., conditions of Dirichlet, Neumann or Cauchy required by Eq. (4)). The definition of these boundary conditions in different geometric entities is performed through a *ElementValue* object, which is derived for the specific types *ElementVolumeValue* (boundary conditions applied in a volume), *ElementAreaValue* (boundary conditions applied to an area) and *ElementLineValue* (boundary conditions imposed on a line), as can be seen in Fig. 7. *ElementValue* consists of the following attributes:

- An array of objects of type *PointValue*, responsible for storing the coordinates of a point and information that defines the boundary condition at this point.

- An object of type *Shape*, responsible for the interpolation of the generic boundary condition.

![UML diagram of the Value package](image)

**Figure 6: UML diagram of the Value package**

The class *Shape* is responsible for providing the approximation function that interpolates the different *PointValue* applied in a region of the element that can be a line, area or a volume, depending on the element type. The responsible class for combining this information and providing the equivalent nodal value is *EquivalentNodalValue* class, whose attributes are objects of *Element* and *ElementValue* classes.
3.2.1. Generic imposition of boundary conditions

The imposition of the boundary conditions is divided into three main steps: in the first one, the information is collected directly from nodes. The second one is the calculation of equivalent nodal value and in the third one, the penalized equivalent nodal value is performed.

For the first step, the process of data acquisition is nodal (since the input information is also provided at the nodal level), and can be described through the sequence diagram shown in Fig. 8. The loop is performed directly on the model nodes with the goal of building the matrix $N_p$, which will be sent to Assembler.

The second step is responsible for assembling the portion of vector $E$ of Eq. (9) that is not under penalty process. In Fig. 9, it can be seen that a loop occurs in the elements, verifying the existence of different dimensional aspects of boundary condition domain (point, line, area or volume).

The third step is responsible for assembling portions of the penalized problem, whether concerning the portion of stiffness matrix $C$ or on the corresponding portion of vector $E$ from Eq. (9) which are obtained by method $getCpm$ and method $getEpm$, respectively. The index ‘pm’ refers to penalty method that is used here to calculate $C_{pm}$ and $E_{pm}$ matrices. It can be seen in Fig. 10 that the communication with the FEM model is employed only to identify the penalty factor in the solution process. Then, similar process occurs during the second step, identifying the various types of boundary conditions that may be described in the element.

Figure 11 shows the generalization process, which is common for both steps two and three. The “Actor” (Assembler) requests from ProblemDriver class the portion of the matrix associated with a particular element. The ProblemDriver is, in fact, an interface that contains generic methods used to inform Assembler all the necessary data for assembling the final sys-
tem of the model, Eq. (8). The information of Element and ElementValue are provided by ElementNodalValue. The ElementNodalValue class is responsible to undertake appropriate numerical integrations and return the equivalent nodal values in matrix form. These information are already in the class ProblemDriver that will send it back to the Assembler.

3.3. Information transferring from the global to the local problem

One of the actions performed during the process of generalization of the imposition of boundary conditions on INSANE environment is to add the ability to obtain boundary conditions of an element of the local problem from an element of the global problem. This is required in the global-local analysis process in which, it is necessary to impose boundary conditions for the local problem provided by the global mesh solution. There are basically three types of boundary conditions that can be used:

- Dirichlet boundary conditions: discussed in section 3.3.1.
- Neumann boundary conditions: discussed in section 3.3.2.
- Cauchy or Spring boundary conditions: discussed in section 3.3.3.

The class responsible for boundary conditions transferring between domains is the EquivalentNodalValue class that is part of the generalValue package. The EquivalentNodalValue class has methods that fall into those capable of modifying the matrix $C$ (stiffness) and those able to modify the vector $D$ (force) of the system (8), as can be seen in Fig. 12.

The local problem of the Eq. (4) can be divided into the following parts:
Figure 9: Equivalent nodal force calculation

\[ \int_{\Omega_L} \sigma(u_L) : \varepsilon(v_L) \, dx \]  
\[ \eta \int_{\partial \Omega_L \cap \partial \Omega_G} u \cdot v_L \, ds + \kappa \int_{\partial \Omega_L \backslash (\partial \Omega_L \cap \partial \Omega_G)} u \cdot v_L \, ds \]  
\[ \int_{\partial \Omega_L \cap \partial \Omega_G^c} t \cdot v_L \, ds \]  
\[ \eta \int_{\partial \Omega_L \cap \partial \Omega_G^c} u \cdot v_L \, ds + \int_{\partial \Omega_L \backslash (\partial \Omega_L \cap \partial \Omega_G)} (t(u_G^0) + \kappa u_G^0) \cdot v_L \, ds \]

Equations (11) and (12) correspond to the original matrix \( C \) and its modification by the application of penalty method, respectively. The same occurs with Eqs. (13) and (14) with respect to the vector \( D \) and its modification.

Figure 13 shows the imposition of the boundary conditions to local problem, using the numerical solution of the global problem. In Fig. 13(a), the global domain \( \Omega_G \) is presented and the local domain is highlighted. After analysis of the global domain is performed, the resulting numerical solution should be transferred to the elements of the local domain as boundary conditions.

Figure 13(b) depicts the new mesh for the local domain \( \Omega_L \) and the solution computed on global problem providing boundary conditions for the extracted local problem. Such solution can be a field of displacements, tractions or both of them, resulting in Dirichlet, Neumann or Cauchy boundary conditions, respectively.
Figure 10: Generalization of the penalty calculation

Figure 13(c) shows the domain $\Omega_L$ and details of an element of the local mesh. For the highlighted finite element, the four faces are presented as well as their Gauss points, chosen here to be in number of three (PG1, PG2, PG3). The imposition of the boundary conditions is made on the faces of boundary $\partial\Omega_L \setminus (\partial\Omega_L \cap \partial\Omega_G)$, $\Omega_L \cap \partial\Omega_G^e$ or $\Omega_L \cap \partial\Omega_G^o$. The results are transmitted directly to the Gauss point of the requested face and used in the integration of the Eq. (4).

3.3.1. Dirichlet boundary condition

Once the solution of the global problem $u_G^0$ is obtained, the results are transferred to the Gauss points of the boundary $\partial\Omega_L \setminus (\partial\Omega_L \cap \partial\Omega_G)$ or $\Omega_L \cap \partial\Omega_G^u$. Figure 14 shows the local finite element depicted in Fig. 13(c) and its corresponding boundary $\partial\Omega_L \setminus (\partial\Omega_L \cap \partial\Omega_G)$ in which the displacement $u_G^0$ provided by the global element is imposed as boundary conditions. The generic Eq. (4), is, in this case, simplified as:

$$\int_{\Omega_L} \sigma(u_L) : \varepsilon(v_L) dx + \eta \int_{\partial\Omega_L \cap \partial\Omega_G^u} u_L \cdot v_L ds = \int_{\partial\Omega_L \cap \partial\Omega_G^u} \bar{t} v_L ds + \eta \int_{\partial\Omega_L \cap \partial\Omega_G^o} \bar{u} \cdot v_L ds + \int_{\partial\Omega_L \setminus (\partial\Omega_L \cap \partial\Omega_G)} t u_G^0 \cdot v_L ds$$

Equation (15) is used to assemble the system (9), following the modifications of the matrix $C$ and $D$ according to EquivalentNodalValue class (Fig. 12).

The $\eta$ parameter: The penalty coefficient $\eta$ used in Eqs. (4) and (15) is calculated in this work, following relation proposed in Duarte and Kim (2008) and shown in Eq. (16):
Figure 11: Detail of the typical equivalent Cpp

\[ \eta = 10^8 \times E \times J \]  \hspace{1cm} (16)

where \( E \) is the modulus of elasticity of the material and \( J \) is the Jacobian of global element originating local elements.

3.3.2. Neumann boundary condition

Similarly to the Dirichlet boundary conditions, the numerical results of the global problem in terms of stress are transferred as prescribe tractions to the Gauss points of the boundary \( \partial \Omega_L \setminus (\partial \Omega_L \cap \partial \Omega_G) \) or \( \partial \Omega_L \cap \partial \Omega_G \) of local elements. Figure 15 shows the local finite element (extracted from Fig. 13(c)) being subject to the Neumann boundary conditions of \( t \) on the boundary \( \partial \Omega_L \setminus (\partial \Omega_L \cap \partial \Omega_G) \). Similarly to section 3.3.1, the Eq. (4) can be, in the present case, simplified as:

\[
\int_{\Omega_L} \mathbf{\sigma}(\mathbf{u}_L) : \mathbf{\varepsilon}(\mathbf{v}_L) \, d\mathbf{x} = \int_{\partial \Omega_L \cap \partial \Omega_G} \mathbf{t} \mathbf{v}_L \, d\mathbf{s} + \int_{\partial \Omega_L \setminus (\partial \Omega_L \cap \partial \Omega_G)} \mathbf{t}(\mathbf{u}^0_G) \cdot \mathbf{v}_L \, d\mathbf{s} \]  \hspace{1cm} (17)

\[
\mathbf{u} = \bar{\mathbf{u}} \text{ in } \partial \Omega_G^e \]  \hspace{1cm} (18)

This is the new equation represented in INSANE by the system (9). It is assembled following the modifications of the matrix \( C \) and \( D \) determined by the \texttt{EquivalentNodalValue} class (Fig. 12).
3.3.3. Cauchy boundary condition

Cauchy boundary conditions can be imposed, by transferring the numerical solution in terms of displacements and stress to the Gauss points of the boundary $\partial \Omega_L \setminus (\partial \Omega_L \cap \partial \Omega_G)$. Figure 16 shows the local finite element extracted from the local problem shown in Fig. 13(c) and its corresponding global boundary conditions on $\partial \Omega_L \setminus (\partial \Omega_L \cap \partial \Omega_G)$.

The transfer of Cauchy boundary conditions suggests that there is a spring of stiffness $'\kappa'$ in the contour of the local element, conveying an approximate combined results of force and displacement, according to the works of Kim et al. (2010).

$$\int_{\Omega_L} \sigma(u_L) : \varepsilon(v_L) \, dx + \eta \int_{\partial \Omega_L \cap \partial \Omega_G} u_L \cdot v_L \, ds + \kappa \int_{\partial \Omega_L \setminus (\partial \Omega_L \cap \partial \Omega_G)} u_L \cdot v_L \, ds$$

$$= \int_{\partial \Omega_L \cap \partial \Omega_G} t \cdot v_L \, ds + \eta \int_{\partial \Omega_L \cap \partial \Omega_G} \bar{u} \cdot v_L \, ds + \int_{\partial \Omega_L \setminus (\partial \Omega_L \cap \partial \Omega_G)} \left( t(u_G^0) + \kappa u_G^0 \right) \cdot v_L \, ds$$

(19)

Similarly to Dirichlet and Neumann conditions, Eq. (19) is implemented in INSANE by the system (9). It is assembled following the modifications of the matrix $C$ and $D$ determined by the `EquivalentNodalValue` class (Fig. 12).

4. Numerical examples

This section presents two linear-elastic problems in $\mathbb{R}^2$. Section 4.1 presents a wedge with concentrated force problem with one and more than one local domain, section 4.2 presents a fracture mechanics problem, and section 4.3 presents a plate problem under distributed load orthogonal to its midsurface. These problems are brought in order to illustrate the generalization of the current global-local G/XFEM approach for various mathematical models and element
Figure 13: Imposition of the BCs to local problem, using the numerical solution of the global problem
formulations. The geometry and boundary conditions are very simple, since the goal is not to demonstrate the capabilities of the methods, but the general approach proposed to enclose the G/XFEM formulation in INSANE environment. As already discussed in section 3.3, a generic strategy is implemented in order to transfer different boundary condition types from global to local boundaries. Among the three aforementioned boundary conditions, according to Kim et al. (2010), the Dirichlet boundary conditions (a limited case of Cauchy boundary condition) lead to worse results than Cauchy boundary conditions. Thus, for all three examples, the Dirichlet boundary conditions are applied on the local problem boundaries in order to demonstrate the robustness of the methodology in the worst case scenario. One exception is the fracture mechanics problem, in which the Cauchy boundary condition is also evaluated. In the last example, related to the plate, not only displacement but also the rotation are transferred to the local boundary, illustrating another aspect of the generalization of this implementation.

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 the 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 $n^L$ local elements.
and number of Gauss points for each local element be equal to \( GP \). Thus, the number of integration points for this global element is obtained by: \( \sum_{i=1}^{n_{Le}} GP_i \). In other words, the global numerical integration is done over Gauss points of each local element that represent a part of the global element. It should be noted that all numerical values for the numerical examples in section 4.1 and 4.2 are presented in consistent units.

4.1. Wedge with concentrated force problem

This example shows a wedge with unit thickness subjected to a concentrated force in one of its corners as shown in Fig. 17. The material characteristics of the wedge are as follows:

- Elastic Modulus \( E = 1.0 \);
- Poisson ratio \( \nu = 0.3 \);

The exact solution for the radial stress component \( \sigma_r \) of this problem is provided by the theory of elasticity (Timoshenko and Goodier, 1951) and shown in Eq. (20). The analysis model used is the state of plane deformations and the element type is triangular.

$$\sigma_r = \frac{20 \cos \theta}{r \left[ \alpha + 0.5 \sin(2\alpha) \right]} \tag{20}$$

Due to symmetry of the problem, only half of the wedge is analyzed. Here, two strategies to solve the G/XFEM\(^d\) problem are used. In the first one (Fig. 18), a single local domain have been selected in order to provide the required information to enrich the global domain. In the second strategy (Fig. 19), the corresponding clouds of the global domain will be enriched with the global-local enrichment function obtained from independent local domains composed
by elements that share the global node of the cloud that should be enriched. Number of Gauss points for global and local elements are equal to 13 and 7, respectively.

4.1.1. One local domain

Figure 18 shows the solution strategy using G/XFEMgl.

![Figure 18: Solution strategy using global-local enrichment with one local problem](image)

The penalty parameter $\eta$ of Dirichlet boundary condition for this analysis type is chosen equal to $1 \times 10^8$. The local domain in Fig. 18 covers four finite elements from global domain. The local domain with four finite elements allows that three nodes from global problem are enriched with global-local enrichment function.

4.1.2. Several superimposed local domains

In this second strategy, the hatched region of global problem from Fig. 19 is used to define three local regions. Each local region will be responsible for the global-local enrichment of a global domain node. This figure illustrates how this strategy is performed.

The penalty parameter $\eta$ of Dirichlet boundary condition for problem with several local domains is chosen equal to $1 \times 10^{15}$. Figure 20 shows the three local domains used in the solution of the problem with their respective global node which will be enriched by global-local enrichment. It is observed that each local area consists of the cloud of the global node which will be further enriched with the global-local enrichment functions.

Figure 21 compares the results of the stress component $\sigma_r$ for each position $y$ along the symmetric axis, considering six results:

- **reference**: analytical result of tension $\sigma_r$, from equation (20);
- **GFEM-P0**: result of stress component $\sigma_r$ for G/XFEM method with polynomial enrichment of order zero;
Figure 19: Solution strategy using global-local enrichment with several local problems

(a) Local 1  
(b) Local 2  
(c) Local 3

Figure 20: Cloud for each node of the global domain

- oneLocal: result of stress component $\sigma_r$ for G/XFEM$^{gl}$ strategy with only one local domain. The local domain with four elements is chosen for this part (Fig. 18);

- sevLocal: result of stress component $\sigma_r$ for G/XFEM$^{gl}$ strategy with the combination of three local domains.

It can be concluded from Fig. 21 that the G/XFEM methodology, GFEM-P0 case, is not able to capture the stress concentration while the G/XFEM$^{gl}$ method produced equivalent results to the reference analysis. Also, the two solution strategies of G/XFEM$^{gl}$ produced similar results. However, the strategy of several local domains may be attractive to simulations performed using parallel processing. Once the solution of the initial global problem is obtained, the local problems can be solved independent.

4.2 Fracture mechanics problem

This example considers a rectangular plate with an edge crack submitted to an axial stress, as shown in Fig. 22. The cracked zone produces singular stress field near the crack tip. The crack surface is geometrically represented in the global and local models using double nodes.
This problem analyzed under plane stress state, has the following parameters:

- Modulus of elasticity $E = 1.0$;
- Poisson ratio $\nu = 0.3$;
- Traction $\sigma = 1.0$.

The reference solution of this problem is obtained using a mesh of 33000 quadrilateral elements (CPS4, a 4-node bilinear plane stress quadrilateral element) in ABAQUS®. For G/XFEM\textsuperscript{ed} analysis, however, a smaller number of finite elements as well as of degrees of freedom (DOFs) is used. As it can be seen in Fig. 23, only 50 quadrilateral elements are used for the global mesh. The reason for using a smaller number of DOFs is explained by the use of global-local enrichment function, which is suitable for high stress concentration. This strategy is used to describe the singular stress near the crack tip. Following subsections are described some relevant aspects of the global-local strategy through this example. The integration order of G/XFEM\textsuperscript{ed} for both global and local problems is considered equal to $12 \times 12$. The penalty parameter, $\eta$, of Dirichlet boundary condition is chosen equal to $1 \times 10^8$ and the Spring stiffness, $\kappa$, for Cauchy boundary condition is chosen equal to $1 \times 10^2$. 

![Figure 21: Stress result $\sigma_r$ for the global domain](image)
4.2.1. Mesh refinement

This section studies regular and geometric $h$-refinements for local mesh. The local domain is composed by four elements, as shown in Fig. 23. The approximate function of the global problem is enriched by the known analytic solution around a crack in mode I (Eq. (21)).

\[
\begin{align*}
    u_I^x(r, \theta) &= \frac{K_I}{2G\sqrt{2\pi}} r^{\lambda}\left\{[\kappa - Q(\lambda + 1)\cos \lambda \theta] - \lambda \cos(\lambda - 2)\theta\right\} \\
    u_I^y(r, \theta) &= \frac{K_I}{2G\sqrt{2\pi}} r^{\lambda}\left\{[\kappa + Q(\lambda + 1)\sin \lambda \theta] + \lambda \sin(\lambda - 2)\theta\right\}
\end{align*}
\]

(21)

\[
\kappa = \frac{3 - \nu}{1 + \nu}
\]

(22)

where $K_I$ is the stress intensity factor for mode I. Also, for current problem $\lambda$ and $Q$ are equal 0.5 and 0.333, respectively. The enrichment of the solution of the first step with the analytical function (21) aims to provide an accurate boundary condition to the local problem. The transference of the numerical solution from initial global domain (step 1) to local domain (step 2) is done using Dirichlet boundary condition. For the global-local problem (step 3), only the crack tip node is enriched with global-local function and analytical enrichment is excluded from this step because we want to show the performance of the numerical enrichment. If
the analytical enrichment kept available for the enriched node with global-local strategy, the numerical enrichment can be useless and no global-local strategy needs to be performed in order to capture a better result.

**Figure 23:** Global and local domain of the problem

**Regular mesh:** four different meshes for the local problem are shown in Fig. 24. Results of the corresponding analyses are presented in Fig. 25 and Table 1.

![Regular meshes analyzed](image)

**Geometric mesh:** Four different meshes with geometric refining for the local problem and given by a factor $f = 10\%$, as shown in Fig. 26, and following Szabó and Babuska (1991). Results are shown in Fig. 27. Also, Table 2 presents a comparison between the strain energy values for each mesh configuration of the global domain.

As it is expected, results show that regular meshes provide worst solutions when compared to geometric meshes, since the proposed problem presents a high singularity of the stress field.
Table 1: DOFs and strain energy for regular refinement

<table>
<thead>
<tr>
<th>Analysis</th>
<th>DOFs</th>
<th>Strain energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference result</td>
<td>199270</td>
<td>10.98314</td>
</tr>
<tr>
<td>2x2</td>
<td>134</td>
<td>10.71143</td>
</tr>
<tr>
<td>4x4</td>
<td>134</td>
<td>10.75792</td>
</tr>
<tr>
<td>8x8</td>
<td>134</td>
<td>10.78458</td>
</tr>
<tr>
<td>16x16</td>
<td>134</td>
<td>10.79264</td>
</tr>
</tbody>
</table>

Figure 25: Results for $\sigma_x$ in different positions given by the coordinate $x$ and the same coordinate $y$ of the crack, for global domain refined with regular mesh next to the crack tip.

The next experiment varies the level of refinement (geometric refinement with factor of $f = 10\%$, $15\%$ and $25\%$) for $L_2$ and $L_3$ (level $L_1$ is removed since it presented bad results, and level $L_4$ is removed since the results were similar when compared to level $L_3$). Six mesh configurations are investigated, and results for stress component $\sigma_x$ are shown in Fig. 28. Definition for level of refinement for $L_2$ and $L_3$ are as follows:

- $f = 10\%$: level $L_2$ (10L2) and level $L_3$ (10L3);
- $f = 15\%$: level $L_2$ (15L2) and level $L_3$ (15L3);
- $f = 25\%$: level $L_2$ (25L2) and level $L_3$ (25L3);

Table 3 presents a comparative between DOFs and strain energy.

It can be concluded from the results shown in Fig. 28 and Table 3 that the global-local enrichment provided a more significant improvement in approximation when the geometric
mesh reduction rate is equal to 10.

4.2.2. Quality of the boundary conditions applied to local domains

This section evaluates how the quality of the boundary conditions changes the results for the problem. For this analysis, local problems (step 2) are composed by 4 finite elements with geometric refinement L3 (without enrichment). Global final domain (step 3) presents only the crack tip node enriched with global-local enrichment functions (enrichment applied in step 1 is removed). Four cases are presented:

- $\text{1enr}$: initial global problem with crack tip node enriched by analytic singular functions (Eq. (21));

- $\text{0enr}$: initial global problem without enrichment and crack physically represented by the mesh;

- $\text{0enrWC}$: initial global problem without enrichment and without crack physically represented by the mesh (crack is represented only at the local domain);
Figure 27: Results for $\sigma_x$ (step 3) in different positions given by the coordinate $x$ and the same coordinate $y$ of the crack, for global domain refined with geometric mesh.

Figure 28: Results for $\sigma_x$ (step 3) in different positions given by the coordinate $x$ and the same coordinate $y$ of the crack, for global domain refined with geometric mesh (for $f = 10\%, 15\%$ and $25\%$).
Table 3: DOFs and strain energy for level reduction

<table>
<thead>
<tr>
<th>Analysis</th>
<th>DOFs</th>
<th>Strain energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference result</td>
<td>199270</td>
<td>10.98314</td>
</tr>
<tr>
<td>10L2</td>
<td>134</td>
<td>10.0684</td>
</tr>
<tr>
<td>15L2</td>
<td>134</td>
<td>10.80585</td>
</tr>
<tr>
<td>25L2</td>
<td>134</td>
<td>10.80042</td>
</tr>
<tr>
<td>10L3</td>
<td>134</td>
<td>10.80928</td>
</tr>
<tr>
<td>15L3</td>
<td>134</td>
<td>10.80909</td>
</tr>
<tr>
<td>25L3</td>
<td>134</td>
<td>10.80519</td>
</tr>
</tbody>
</table>

- all: initial global problem with all nodes enriched by analytic singular functions (Eq. (21)).

Results for stress component $\sigma_x$ are presented in Fig. 29. Table 4 presents a comparison between DOFs and strain energy. As already presented in Duarte et al. (2007), except when the crack is not physically represented in the step 1, this analysis shows that final results are not highly affected by the quality of the solution of the initial global problem.

Table 4: DOFs and strain energy for the cases analyzed

<table>
<thead>
<tr>
<th>Analysis</th>
<th>DOFs</th>
<th>Strain energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference result</td>
<td>199270</td>
<td>10.98314</td>
</tr>
<tr>
<td>1enr</td>
<td>134</td>
<td>10.80684</td>
</tr>
<tr>
<td>0enrWC</td>
<td>134</td>
<td>10.80223</td>
</tr>
<tr>
<td>0enr</td>
<td>134</td>
<td>10.10835</td>
</tr>
<tr>
<td>all</td>
<td>134</td>
<td>10.80928</td>
</tr>
</tbody>
</table>

4.2.3. Local domain size and number of enriched nodes

The final answer for a problem solved using global-local strategy depends on the size of the local domain, 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. This section investigates the influence of the local domains size. The initial global problem from section 4.2.2 is used here and it has no enrichment ($0enr$ case). For local problems a geometric mesh level L3 is used. Three cases are here considered:

- $4elemlnode$: local domain composed by 4 elements and crack tip node enriched with global-local enrichment function (Fig. 30).

- $12elemlnode$: local domain composed by 12 elements and crack tip node enriched with global-local enrichment function (Fig. 31(a)).
Figure 29: Results for $\sigma_x$ (step 3) in different positions given by the coordinate $x$ and the same coordinate $y$ of the crack, for quality assessment of boundary conditions applied to local domains

- **12elem10nodes**: local domain composed by 12 elements and 10 nodes (step 3) enriched with global-local enrichment function (Fig. 31(b)).

Table 5 presents DOFs and strain energy for aforementioned cases analyzed here. Results show that local size does not change significantly the answers of the final problem (“4elem1node” and “12elem1node” have almost the same answer). This was expected because according to Fig. 31 the local mesh of the additional elements is not refined. So the local solution is not improved out of the central four elements. Here, the advantage of using a broader area for the local problem is allowing the enrichment of a greater number of PUs. Finally, “12elem10nodes” presents best results to the problem, since a great number of nodes are enriched with the solution from the local problem.

Table 5: DOFs and strain energy for the global-local strategy with different local domains

<table>
<thead>
<tr>
<th>Analysis</th>
<th>DOFs</th>
<th>Strain energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference result</td>
<td>199270</td>
<td>10.98314</td>
</tr>
<tr>
<td>4elem1node</td>
<td>134</td>
<td>10.80223</td>
</tr>
<tr>
<td>12elem1node</td>
<td>134</td>
<td>10.80390</td>
</tr>
<tr>
<td>12elem10nodes</td>
<td>154</td>
<td>10.86523</td>
</tr>
</tbody>
</table>

4.2.4. Effect of the imposition different boundary condition on the local domain

This section presents the analysis considering the transfer of two types of boundary condition between global and local elements: Dirichlet and Cauchy boundary conditions. The initial
local domain composed by 4 elements.

global domain used here does not have any kind of enrichment, only geometrically represented the crack. In the local domain geometric mesh of L3 level is used. The numerically enrichment function is used to enrich the crack tip node.

Figure 32 presents the distribution results of stress component $\sigma_x$ for current problem considering the information transferring of Dirichlet and Cauchy boundary conditions between global and local domains. The stress distribution of Fig. 32 shows that the results are very similar for both boundary condition types.

In addition, Table 6 shows the results for the strain energy of the final global problem for different boundary conditions transferred from global problem over the local problem.

<table>
<thead>
<tr>
<th>Analysis</th>
<th>DOFs</th>
<th>Strain energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference result</td>
<td>199270</td>
<td>10.98314</td>
</tr>
<tr>
<td>Dirichlet</td>
<td>136</td>
<td>10.80223</td>
</tr>
<tr>
<td>Cauchy</td>
<td>136</td>
<td>10.80229</td>
</tr>
</tbody>
</table>

4.3. Clamped square plate

A clamped square plate, as shown in Fig. 33, is subjected to a distributed load over its area and orthogonal to its mid-surface. This problem examines the behavior of G/XFEM$^{\text{cl}}$ using the Reissner-Mindlin formulation and performing the boundary condition transfer between the global and local problem with different degrees of freedom. Here, the displacement field is described by the in-plane or transversal displacement, $\omega$, and the rotations, $\theta_x$ and $\theta_y$, of the
normal to the plate middle plane. In this case, only $C^0$ is required continuity for the displacement field.

The transversal displacement at the center of the plate is normalized with respect to the Kirchoff-Love thin-plate solution (Timoshenko and Woinowsky-Krieger, 1959), which is given by:

$$w_{central} = 0.00126 \frac{qL^4}{D}$$

where $D$ is the bending stiffness defined by:

$$D = \frac{Eh^3}{12(1-\nu^2)}$$

Four types of analysis are considered here:

- G/XFEM with polynomial enrichment of order zero, which means a first degree approximation ($\text{GFEM}_p = 0$),
- G/XFEM with first order of polynomial enrichment, which means a second degree approximation ($\text{GFEM}_p = 1$),
- G/XFEM$_{gl}$ with polynomial enrichment of order zero for both global and local problems, which means a first degree approximation combined with the numerical enrichment ($\text{G/XFEM}_{gl}^p = 0$),
- G/XFEM$_{gl}$ with first order of polynomial enrichment for both global and local problems, which means a second degree approximation combined with the numerical enrichment ($\text{G/XFEM}_{gl}^p = 1$).
It should be mentioned here that a combination of polynomial enrichment and global-local enrichment are performed for the two last types of analysis. For each analysis a sequence of uniform meshes of $N \times N$ with $N = 2, 4, 8$ is employed, Fig. 34 (global problem in the case of the G/XFEM$^{gl}$ method). The local domain is the central region of the plate and it is fixed for all three global meshes which provides the geometrical enrichment. The geometrical enrichment consists in enriching all nodes that are located in a fixed predefined region. The local domain consists of 64 elements, as shown in Fig. 34. The effect of local domain size already been shown in the section 4.2.3. The dimension of local domain considered here is equal to $2.5 \times 2.5$. This domain contains 1, 4, and 16 global elements of $h = 2.5, 1.25, \text{and } 0.625$, respectively, as it is shown in Fig. 34. The parameter $h$ is the global element size.

The integration order for G/XFEM and G/XFEM$^{gl}$ (initial and final global problems and local problem) models is considered equal to $2 \times 2$ and $2 \times 2 \times 1$, respectively. The penalty parameter $\eta$ of Dirichlet boundary condition is chosen equal to $1 \times 10^{10}$.

As can be seen in Fig. 34, the number of enriched nodes using global-local approach are 1, 4, and 16 nodes for element size of $h = 2.5, 1.25, \text{and } 0.625$, respectively. Due to the over estimation of the shear strain when the problem tends to be under pure bending conditions. In order to overcome the locking problem, one can increase the polynomial degree of the approximation functions as proposed by (Mendonça et al., 2011, Garcia et al., 2000).

Figure 35 shows the normalized maximum central displacement against the inverse of global element size, $1/h$. It can seen from this figure that the results obtained from G/XFEM$^{gl}$ better converge to the analytical solution than the standard G/XFEM results. Also, the results with first order of polynomial enrichment show a better behavior since it decrease the effect of shear
Figure 33: Geometry of the clamped square plate. With: $L = 10$, Young modulus $E = 1092000$, Poisson’s ratio $\nu = 0.3$, thickness $= 0.1$, and $q = 1$. Due to symmetry, only one-quarter of the plate (shaded area) is considered for the numerical analysis.

locking.

5. Conclusions

In this paper it is presented a computation framework to generate numerically enrichment functions for two-dimensional problems dealing with single/multiple local phenomenon. Firstly an overview of the G/XFEM as well as the enrichment global-local strategy is given. Nonetheless, the main focus here is the programming environment and its extension to enclose new enrichment strategy. This environment was conceived under a segmented conception in Fonseca and Pitangueira (2007) for FEM analysis and allows a code expansion whose complexity may be increased in a gradual manner. In Alves et al. (2013a) such characteristic is exploited to implement a generic enrichment strategy for any PU given by the FEM formulation. Here, this generic enrichment strategy is extended to embrace the numerical functions obtained from the solution of local problems under the global-local approach.

Details of the design of classes are discussed aiming to demonstrate the generic implementation that allows the combination of different analysis approaches. The numerical examples illustrate such flexibility. Different kinds of enrichments are combined with the numerical functions provided by the global-local strategy. The boundary information that needs to be transferred
Figure 34: Global and local domains for three global meshes. Blue elements represent the local domain and black markers represent the nodes that are enriched by the global-local enrichment function.

from the global to the local problem are imposed by Dirichlet and Cauchy strategy, as well for different analysis models. Additionally, some characteristics of the global-local enrichment method are also evaluated.

References


Figure 35: Normalized maximum central displacement vs. global element size


