23rd ABCM International Congress of Mechanical Engineering
December 6-11, 2015, Rio de Janeiro, RJ, Brazil

INTEGRATED GRAPHICAL SYSTEM TO PHYSICALLY NONLINEAR ANALYSIS BASED ON THE GENERALIZED FINITE ELEMENT METHOD

Leandro Lopes da Silva
Roque Luiz da Silva Pitangueira
Samuel Silva Penna
Felício Bruzzi Barros
Gabriel Neves Alves Ferreira
Federal University of Minas Gerais - Antônio Carlos Avenue 6627, Pampulha, CEP. 31270-901, Belo Horizonte, MG, Brazil
leandro.silva_engenhariacivil@yahoo.com.br, roque@dees.ufmg.br, spenna@dees.ufmg.br, feliciobbarros@gmail.com, gabriel.gnaf@gmail.com.

Abstract. *This paper refers to the expansion of INSANE (INteractive Structural ANalysis Environment), an open source project developed by the Structural Engineering Department of Federal University of Minas Gerais. It presents the implementation details of an interactive graphical application that integrates pre-processor, constitutive models, nonlinear equation solvers and post-processor, in such a way that provides all the tools to perform two-dimensional physically nonlinear analysis using Generalized Finite Element Method (GFEM). To overcome the limitations of the data input format of the INSANE, that is based on XML (eXtensible Markup Language) text files, a pre-processor was developed. Such a pre-processor has three modules: Geometry, Mesh and Attributes where the user, respectively, creates a planar subdivision representing the model's geometry; generates a mesh using boundary mapping or domain decomposition; and defines the model attributes, such as nodal restraints, loadings, constitutive model, as well as nodal enrichment parameters for GFEM. Constitutive models based on elastic degradation are employed, allowing adopting fixed or rotating smeared crack models, or the isotropic damage models proposed by Mazars, de Vree, Lemaitre and Chaboche, Mazars and Lemaitre, Simo and Ju or Ju. After the definition of the GFEM's model, the user may choose one of the nonlinear equation solvers available and inform the related parameters. Load control method, displacement control method, generalized displacement control method and arch length control method are available. Once the analysis has been finished, the results can be depicted using the post-processor module tools.*

Keywords: *Interactive Graphical Software, Physically Nonlinear Analysis, Generalized Finite Element Method, Smeared Crack Models, Isotropic Damage Models.*

1. INTRODUCTION

This paper presents a graphical interactive environment for physically nonlinear analysis of structures. Considering the theoretical framework of damage mechanics, such analysis can be processed evaluating the medium degradation in a smeared way.

The degradation phenomenon can be analyzed through constitutive methods. These methods consider that the medium, even though degraded, remains continuous, but that the properties of the material are modified in the regions where the load reaches a critical strength parameter. This approach allows modeling the medium deterioration, without considering the existence of discrete cracks.

The analysis is made using the Finite Element Method (FEM), but the Generalized Finite Element Method (GFEM), a method that allows the enrichment of the partition of unity functions to obtain shape functions, can be employed to optimize the accuracy of the approximating function and to improve the analysis.

The INSANE (*INteractive Structural ANalysis Environment*) computational system is a free software project, implemented in Java language according to the Object-Oriented Programming paradigm, developed at the Structural Engineering Department of the Federal University of Minas Gerais (UFMG) - `http://www.insane.dees.ufmg.br`. The system is a segmented computational environment, friendly to changes and scalable in complexity. This paper refers to the expansion of this system, which has several resources for the physically nonlinear analysis and was improved to allow such kind of analysis in an integrated graphical environment.

For such improvement, aiming for a more friendly interaction with the user, a graphic interactive pre-processor was developed, with suitable resources for physically nonlinear analysis. Also, the communication between this application, the processor and the post-processor was established.

2. INTERACTIVE GRAPHICAL APPLICATIONS OF THE INSANE SYSTEM

The object-oriented software development process, employed in the INSANE system, comprehends three main steps. Initially, the analysis step is aimed to emphasize the identification and description of the objects - or concepts - related to the scope of the problem. Next, the project step aims to define the logical software elements, its attributes and methods. Finally, during the construction step, the so-called components of the project are implemented using Java programming language and its APIs (Application Program Interface). For the interactive graphical applications of the INSANE, the features discussed in the following were used.

2.1 Software design patterns

Software design patterns can be defined as the basic structure of a well-succeeded software, capable of offering a scheme for the subsystems or components of a software system that will be designed. This scheme should be specific to solve the problem at issue and generic enough to deal with future problems and requirements. The utilization of software design patterns improves code readability and allows the reuse of the computational system.

According to Penna (2007), the INSANE system presents an architecture formed by layers based on the combination of three software design patterns: the *Model-View-Controller* (*MVC*) pattern, which allows the independence between the model and view layers; the *Observer* pattern which, through a mechanism of change propagation, allows the synchronization of the model with pairs of view-controllers; and the *Command* pattern, which encapsulates different actions of the program in separate classes in an organized way. This segmentation favors the program expansion and the creation of new commands. Figure 1 illustrates the combination of these patterns.

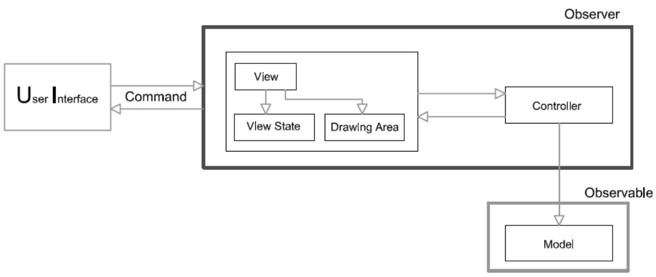

Figure 1. INSANE architecture (Penna, 2007)

2.2 Data structure

The development of geometric models requires data storage, which should enable construction manipulation and geometrical operations, in a structured and accessible form. In view of this, adjacency structures are used in the treatment of computational models.

The model object shown previously (Fig. 1) consists in a data structure capable of storing the topological entities of the analyzed model in an organized way, enabling a rapid access to these data later.

The data structure for planar subdivisions employed by the graphical applications of the INSANE system is the half-edge structure. It is an improvement of the “winged-edge” structure, differing from it by its topological organization. The main alteration is the transformation of one edge into two half-edges.

According to Mäntylä (1987), this data structure is developed inside a hierarchy, made of five different levels, which are the **Solid**, the **Face**, the **Loop**, the **Half-Edge** and the **Vertex**, whose access to one of these levels follows this hierarchy order.

This five levels adjacency structure does not make a connection between the faces in order to compose the geometric model. To do so, Del Savio *et al.* (2004) added an intermediate level. This level is the **Edge**, which possesses a reference to **Half-Edge**, since the edge consists of two half-edges, one left and one right, incident on the same vertices. With this change, **Solid** now has a reference to **Edge** and **Vertex**, and **Half-Edge** has a reference to its respective **Edge**. This hierarchic structure is illustrated in Fig. 2.

2.3 Mesh generation

Mesh generation in two-dimensional domains may be classified as direct or algebraic and indirect or based on differential equations (Fonseca, 1989). Direct or algebraic methods are so called because they generate a mesh over the domain based on some defined algebraic algorithm. Direct methods can be subdivided into mesh generation in elementary domains, meshing by coordinate transformation, conformal mappings, isoparametric mappings, transfinite mappings and domain decomposition.

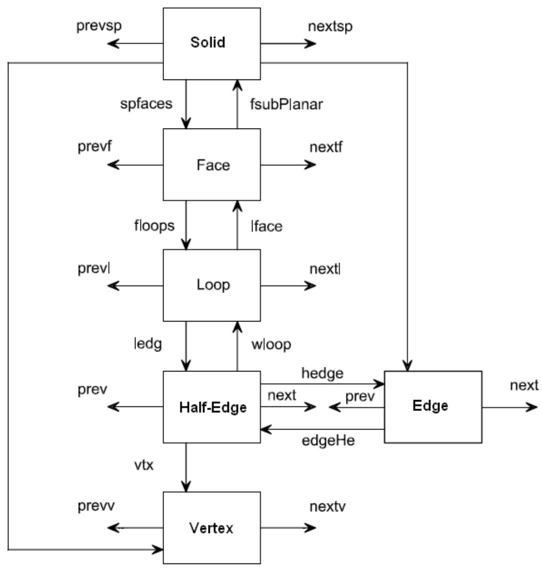

Figure 2. Half-Edge data structure (Del Savio *et al.*, 2004)

The transfinite mapping establishes curvilinear coordinate systems defined by the boundary of arbitrary domains. This method describes an approximate surface that matches the real or idealized surface in a non-countable number of points, property that is responsible for the transfinite mapping name. It is capable of modeling the contour of surfaces with no errors in their geometry (Gonçalves, 2004).

Domain decomposition methods are based on space decomposition techniques, such as quaternary tree (*quadtree*), boundary advancement and the *Delaunay* method, consisting of a relevant alternative in cases of domains with very complex shapes and in cases where it is desired drastically vary the density of elements in the domain.

The current version of the INSANE mesh generator allows Transfinite Mapping (Fonseca, 1989), Random Node Generation (Ho-Le, 1988), Concentrated Asymmetric Mesh and Grid Based Mesh (Ho-Le, 1988).

3. INSANE RESOURCES FOR PHYSICALLY NONLINEAR ANALYSIS

3.1 Incremental-iterative processes

In a nonlinear analysis, it comes up the problem of solving the system with $N+1$ unknowns, being N incremental displacements and one load factor increment, and $N+1$ equations, being N balance equations and one constraint equation. For the solution of this system it is necessary to use an incremental-iterative process (Fuina, 2009).

The Standard Newton-Raphson method, one of the incremental-iterative processes implemented in the INSANE system according to generic algorithm proposed by Yang and Shieh (1990), assumes that the tangent stiffness matrix is recalculated at each iteration. In this case, the incremental equilibrium equation (Eq. (1)) corresponding to the iteration j of step i can be written as:

$$[K]^i_{j-1} \cdot \{\delta U\}^i_j = \delta\lambda^i_j \cdot \{P\} + \{Q\}^i_{j-1} \tag{1}$$

where,
$[K]^i_{j-1}$ is the tangent stiffness matrix at iteration $j-1$ of step i, function of the displacement field $\{U\}^i_{j-1}$;
$\{\delta U\}^i_j$ is the incremental displacement vector of iteration j of step i;
$\delta\lambda^i_j$ is the load factor increment of iteration j of step i;
$\{P\}$ is the reference load vector;
$\{Q\}^i_{j-1}$ is the residual force vector of iteration $j-1$ of step i.

A value for the load factor increment, $\delta\lambda_j$, is established in function of some control parameter, being able to obtain $\{\delta U\}_j$ in function of the its portion associated with the reference load $\{\delta U\}^P_j$ and the residual load $\{\delta U\}^Q_j$.

This formulation is quite general and applies to several control methods, provided they reset the constraint equation. In Fuina (2009) can be found formulations, as implemented in the INSANE system, for obtaining the load parameters for: load, direct displacement, arc length, generalized displacement, work and orthogonal residue control methods.

3.2 Constitutive methods

In the constitutive methods, according to Ingraffea and Wawrzynek (2004), the stiffness of the medium suffers degradation in order to simulate the discontinuity in the displacement field caused by cracking.

There are several models proposed to consider smeared deterioration that are applicable to quasi-brittle materials, among them stand out the Smeared Cracking and Damage Models. In these models, the material is idealized as a continuous medium with degradation due to the cracking process.

Smeared Cracking Models consider that the cracked region is formed by a set of small cracks parallel to each other. Thus, a cracked area would be represented by a set of finite elements with orthotropic behavior (Pitangueira, 1998). These models can be classified into two categories, according to Rots (1988), due consideration as the direction of the cracks: fixed or rotating cracking direction models.

The damage models, according to Murakami (2012), in the case of random distribution of microvoids or isotropic distribution of spherical voids, the damage state is considered to be isotropic and can be represented by a damage scalar variable. On the other hand, when the microvoids have a oriented geometry or configuration, the damage state is anisotropic and damage scalar variable cannot be applied accurately, employing tensor damage variables, unless the void density is small (Lemaitre, 1992).

It is observed that the degradation is not a directly measurable physical quantity, however, it can be determined by gradually reducing a mechanical property such as, for example, elastic modulus, and the degraded medium can be considered isotropic, orthotropic, or even anisotropic.

A theoretical and computational framework for constitutive models based on elastic degradation was developed by Penna (2011) in the INSANE. Several classical models to deal with the material degradation were discussed in the context of this structure and implemented in this framework. Among them, some damage models and smeared cracking model.

3.3 Generalized Finite Element Method

The Generalized Finite Element Method (GFEM) (Strouboulis *et al.*, 2000) can be understood as a variation of the conventional Finite Element Method (FEM). The strategy used in GFEM is to employ Partition of Unity (PU) functions that, enriched, define the shape functions. The choice of the PU functions depends on the type of problem to be analyzed (Barros, 2002; Alves, 2012).

The use of conventional FEM functions (e g., the Lagrangian functions) as PU functions, besides facilitating the application of the method, ensures stability to the analysis, directly verifying the boundary conditions (Alves, 2012). A generic $\tilde{\mathbf{u}}(\mathbf{x})$ is obtained by the following linear combination of the shape functions (Eq. (2)):

$$\tilde{\mathbf{u}}(\mathbf{x}) = \sum_{j=1}^{N} N_j(\mathbf{x}) \left\{ \mathbf{u}_j + \sum_{i=2}^{q} L_{ji}(\mathbf{x}) \mathbf{b}_{ji} \right\} \tag{2}$$

where,
$N_j(\mathbf{x})$ are the conventional FEM functions as Partition of Unity functions;
$L_{ji}(\mathbf{x})$ are the components of the set of enrichment functions;
$\mathbf{u}_j$ and $\mathbf{b}_{ji}$ are nodal parameters associated to each component $N_j(\mathbf{x})$ and $N_j(\mathbf{x}) \cdot L_{ji}(\mathbf{x})$, respectively.

As a final result of the process, it is obtained the product function, which presents the approximating characteristics of the local approximate function, while inheriting the PU compact support. The utilization of this method provides employment of FEM functions for the PU as well as improves the analysis of problems in which the enrichment with special functions (with singularities, for example) presents significant advantage (Barros, 2002).

A GFEM implementation has been performed into INSANE, following the object-oriented approach presented in Alves *et al.* (2013).

4. METHODOLOGY FOR INTEGRATING

4.1 Pre-processor

Consistently with the basic architecture of INSANE (Fig. 1), the pre-processor also uses a combination of *MVC*, *Command* and *Observer* design patterns. Such a combination is shown in Fig. 3, which is a breakdown of Fig. 1. As can be seen in Fig. 3, several view-controller pairs observe the model of the pre-processor (*Model*). Noteworthy is the numerical core that supplies and specializes the interface in function of the analysis type, through a class responsible for mediating the communication of the graphical application with the numerical core, denoted in the abstract form by *Combinable*.

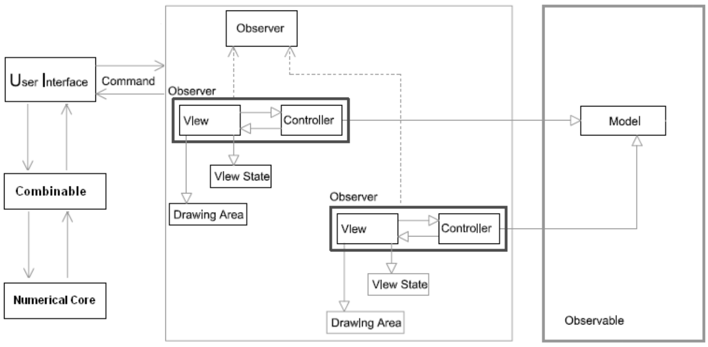

Figure 3. MVC-Observer (Adapted from (Penna, 2007))

4.2 Pre-processor, processor and post-processor

The data input in the INSANE system is provided by a text file written in eXtensible Markup Language (XML), following previously defined rules. The improvement of the pre-processor presented here provided a graphical environment integrated with the other two major system applications: processor, the application that represents the numerical core of the system and is responsible for obtaining the results; and the post-processor, an interactive application that provides tools for viewing results.

Therefore, it was necessary to establish communication between these applications. The data structure, section 2.2, is responsible for storing the topological entities, which enables communication between the pre-processor modules. It is also used by the *Combinable* class for creation of the object *Model*, concurrently with the creation of the object *Solution*, adapting the graphical inputs to objects that can be understood by the numerical core and making the pre-processor communication with the processor. The *Combinable* class is also responsible for the storage of other analysis information, such as constitutive model, material law and its parameters, the degenerate geometric parameters and loadings.

The integration of the post-processor to this environment is possible by writing XML file with the results of all the processing steps, having in mind that this application loads the informations from these text files to shows the results.

4.3 Constitutive models

As in section 3.2, several constitutive models to deal with the smeared degradation of the material medium were implemented in the INSANE system.

A *Combinable* class is responsible for mediating the communication of the graphical application with the numerical core, being responsible for managing the interations among the constitutive models to the analysis types, providing the constitutive model choice dialog in the *Attributes* module.

5. THE INTEGRATED GRAPHICAL SYSTEM

Following the methodology presented above, the interactive graphical pre-processor was improved. Firstly some corrections and adjustments was done to allow the analysis of two-dimensional static linear problems by Finite Element Method (FEM). Next, the pre-processor was adapted with the additions of features and commands needed to the physically static nonlinear analysis by FEM or GFEM of two-dimensional problems, using constitutive methods.

The graphical pre-processor has three modules, namely: *Geometry* Module, where the geometry of the model is drawn; *Mesh* Module, where the model is discretized according to the discrete method employed and through the boundary mapping, making use of the work by Gonçalves (2004), or by the domain decomposition technique; and *Attributes* Module, in which are defined the constitutive model, as well as material and damage evolution laws, if any, material and geometrical properties, boundary conditions and, optionally, nodal enrichment parameters. Using *Attributes* Module, by activating *Process* command, also are defined the parameters to be adopted in the problem solution as discussed in section 3.1. Finally, the *Model* and *Solution* objects created implicitly are sent to the numerical core for processing the analysis, whose results can be uploaded to the post-processor.

The pre-processor division into modules allows, for example, setting several different meshes to the same geometric model. For itself, the definition of the constitutive model in *Attributes* Module allows the geometric model, already discretized and with established geometric properties and boundary conditions, to be evaluated several times by simply changing the constitutive model and material and damage evolution laws, if any, and its parameters, making analysis more convenient. At each processing a new folder *Processing-XX* is created for storing input and output files of that processing in another folder defined at the program start, enabling later information retrieval.

6. AN ILLUSTRATIVE EXAMPLE

In order to illustrate the use of the integrated graphical environment, it is reproduced an example presented in Penna (2011). The geometric configuration, the mesh, the loading and the boundary conditions are shown in Fig. 4.

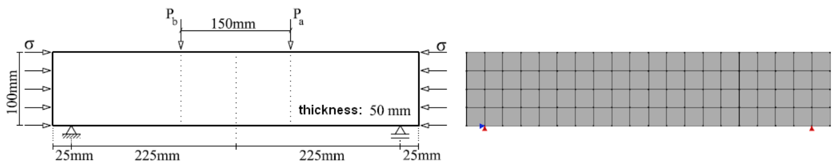

Figure 4. Geometry and finite element model (Penna, 2011)

It was adopted smeared cracking model with bilinear laws for tension and compression described by the parameters: initial elastic modulus $E_0 = 32000.0N/mm^2$, elastic modulus of descending branch $E_2 = 3000.0N/mm^2$, Poisson ratio $\nu = 0.2$, compressive strength $f_c = 30.0N/mm^2$, tensile strength $f_t = 3.0N/mm^2$, characteristic length $h = 50mm$, fracture energy per crack length $G_f = 0.06N/mm$ and shear retention factor $\beta_r = 0.0$. To obtain the equilibrium paths, it was used the generalized displacement control method with initial load increase factor of 10.0, tolerance for convergence 1×10^{-4} and reference load $P_a = P_b = 0.5N$, as shown in Fig. 4. The distributed pre-stress load was held constant, $\sigma = 1.0N/mm^2$.

In the INSANE system, after defining the folder for model information storage, the *Geometry* Module is automatically opened. It consists of resources for drawing, selecting and viewing plane geometric entities. For the example reproduced here, a rectangle $(500 \times 100)mm^2$ is drawn, consisting of vertices, edges, and face, as shown in Fig. 5(a).

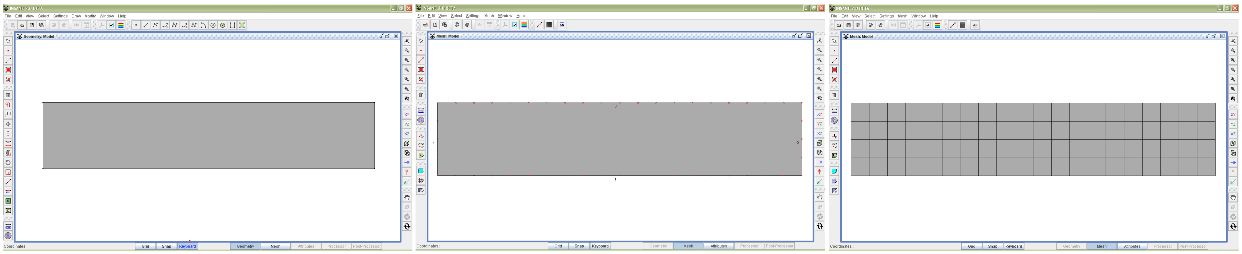

Figure 5. (a) *Geometry* Module (b) *Mesh* Module: Edge division (c) Discretization

Once the geometry is defined, it is possible to pass to the next module, the *Mesh* Module. This flow can occur for the number of meshes that the analyst want to process, accumulating windows of this module whose data structures, that will be modified when generating the mesh, are independent of each other. When changing to the *Mesh* Module, it is opened a dialog for specialization of the analysis type. At this stage, it is necessary to choose between the static analysis with loads in the plane or normal to the plane (plates). From implementation point of view, at this moment it is defined which class will manage the ongoing analysis, previously presented as *Combinable*.

After this, another dialog shows up, in which the user must choose the mesh generation technique. In the current stage of the software, it allows Transfinite Mapping, Random Node Generation, Concentrated Asymmetric Mesh and Grid Based Mesh. The first is boundary mapping technique, while others are domain decomposition techniques.

The Mesh Module has, in addition to resources for selection and visualization, mesh generation commands. In case of selecting the Transfinite Mapping, it is performed by setting, initially, the number of divisions of certain edges. Then the program automatically takes care of propagating the divisions on the other edges. For the example, it is necessary to set the edges 1 and 2 to be divided into twenty and four segments, respectively. Automatically divisions propagate to the edges 3 and 4 (Fig. 5(b)).

In this step, the discretization has not yet occurred. However, the split points of the edges are highlighted (Fig. 5(b)). This allows a previous evaluation of the discretization, when it is even possible to decide to modify the divisions. Then by choosing the discretization command, the mesh will be generated (Fig. 5(c)).

In this module it is also defined the *Analysis Model* from a list, defined when choosing the *Combinable*. For the example, it was adopted the analysis in *Plane Stress*.

Once defined the discretization and the *Analysis Model*, the user now has to go to *Attributes* Module. This module has, in addition to resources for selection and visualization, commands to define the Constitutive Model, material law and its parameters, degenerated geometrical parameters, damage evolution law and its parameters, if any, the boundary conditions of forces and displacements and, optionally, nodal enrichment parameters. As for the loading types, nodal loads, concentrated loads and linearly distributed loads in any position of edges and constant loads in the face of the elements are allowed. Fig. 6(a) illustrates the discretization with the boundary conditions according to Fig. 4.

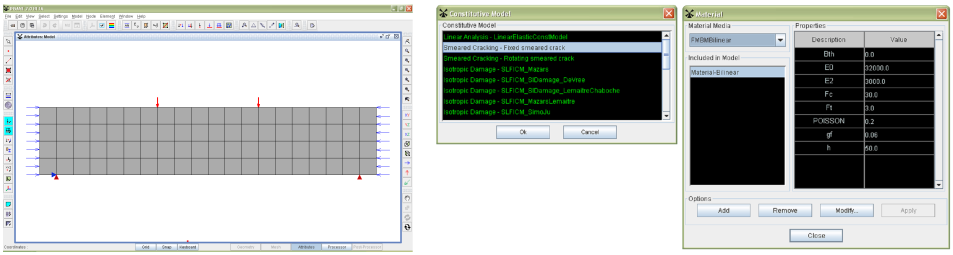

Figure 6. *Attributes* Module: (a) Boundary conditions (b) Definition of constitutive model and materials parameters

It is emphasized the advantage of setting the Constitutive Model in the *Attributes* Module, which enables the model, already discretized, with the boundary conditions and established geometric parameters, to be analyzed several times by simply changing the Constitutive Model. For the example, Fig. 6 (b) illustrates the process of choosing the Constitutive Model and material law and setting its parameters.

Then, the created materials laws are associated with their respective sections (degenerations), which in turn are associated to the elements, allowing adopting elements with different sections and materials.

It is also allowed the assignment of nodal enrichment with monomials and/or special (singular) function, through the Generalized Finite Element Method (Eq. (2)). Therefore, the user creates the enrichments defining, for the case of monomials, the amount of monomials and exponents of its components, while, in case of singular function, defines the crack tip and the angle between direction of the crack and Cartesian coordinate x, among other parameters. These enrichments can be assigned to nodes of interest. In this example, two analyzes were performed: with no enrichment and with polynomial enrichment through the monomials x and y, created and assigned to all nodes as illustrated in Fig. 7. Through this enrichment the shape function of the four nodes quadrilateral element is improved to a quadratic function.

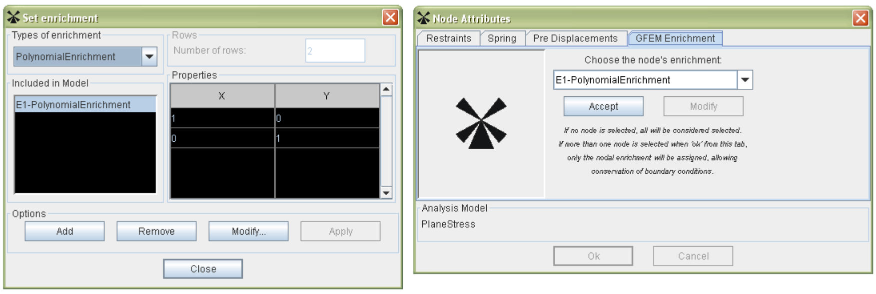

Figure 7. *Attributes* Module: Creating and assigning nodal enrichment

The user interaction goes ahead by defining parameters of the solution. When activating the *Process* command, firstly, an analysis of the data consistency is performed, through the *Combinable*, and, in case of no inconsistency, it opens the dialog where the user defines the incremental-iterative process and its parameters.

After processing, the *Post-Processor* command is released for loading the results, allowing its visualization. A lot of resources can be used to visualize the results, as can be seen in Fig. 8. For the above example, Fig. 8(a) shows the variation of horizontal normal stress over the domain, together with the deformed shape, for a specific analysis step for the case with no enrichment, while the Fig. 8(b) and 8(c) show the equilibrium paths regarding vertical displacement of the node where the load P_b is applied, for the cases with no enrichment and with polynomial enrichment, respectively.

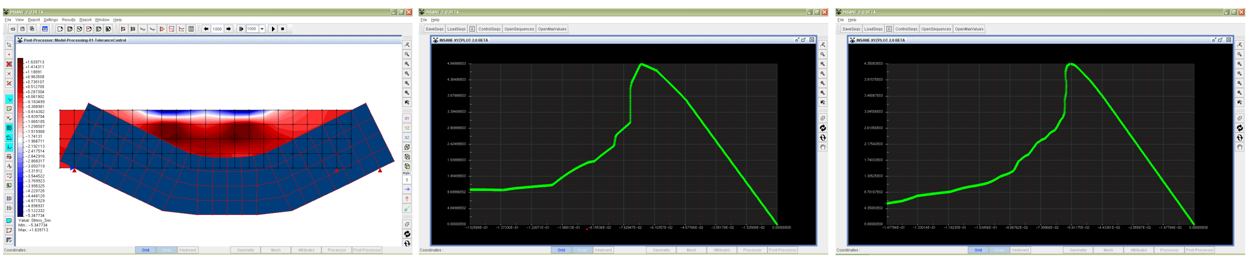

Figure 8. *Post-Processor* Module: (a) Stress state and deformed shape: no enrichment (b) Equilibrium paths: no enrichment (c) polynomial enrichment

7. FINAL REMARKS

An overview of the computational platform INSANE is given with emphasis on development of an integrated graphical system. The system integrates all the resources of the numerical core with a pre and a post-processor.

The pre-processor allows to generate a FEM mesh, based on transfinite mapping or domain decomposition, from a two-dimensional geometric model previously defined. Besides the standard attributes of a FEM model (materials, loading, restraints), the pre-processor also allows to combine any types of analysis model, incremental-iterative process and constitutive model in order to generate a physically nonlinear model. After defining the nonlinear model, the INSANE numerical core is called to perform the analysis. Then, the results of the analysis can be seen using the resources of the post-processor application. The integrated graphical system presented in this paper aims to combine the features of FEM and GFEM methods, as well as methods for the evaluation of the medium degradation.

8. ACKNOWLEDGEMENTS

The authors gratefully acknowledge the important support of the Brazilian research agencies CNPq (in Portuguese “Conselho Nacional de Desenvolvimento Científico e Tecnológico” - Grants 486959/2013-9, 309005/2013-2 and 308785 /2014-2) and FAPEMIG (in Portuguese “Fundação de Amparo à Pesquisa do Estado de Minas Gerais” - Grant PPM-00310-13).

9. REFERENCES

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

Alves, P.D., 2012. *Estratégia Global-Local Aplicada ao Método dos Elementos Finitos Generalizados*. Master’s thesis, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.

Barros, F.B., 2002. *Métodos Sem Malha e Métodos dos Elementos Finitos Generalizados em Análise Não-Linear de Estruturas*. Ph.D. thesis, EESC - USP, São Carlos, SP, Brasil.

Del Savio, A.A., Santi, M.R. and Martha, L.F., 2004. “Traçado de curvas offset para auxilio na geração de malhas”. *XXV Congresso Ibero Latino Americano de Mecânica Computacional - CILAMCE, Recife, Pernambuco.*

Fonseca, G.L., 1989. *Geração de malhas através de mapeamentos transfinitos*. Master’s thesis, PUC - Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ, Brasil.

Fuina, J.S., 2009. *Formulações de Modelos Constitutivos de Microplanos para Contínuos Generalizados*. Ph.D. thesis, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.

Gonçalves, M.A.B., 2004. *Geração de Malhas Bidimensionais de Elementos Finitos Baseada em Mapeamentos Transfinitos*. Master’s thesis, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.

Ho-Le, K., 1988. “Finite element mesh generation methods: a review and classification”. *Computer Aided Design*, Vol. 20 (1), pp. 27–38.

Ingraffea, A.R. and Wawrzynek, P.A., 2004. “Computational fracture mechanics: A survey of the field”. *European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS 2004*, pp. 1–21.

Lemaitre, J., 1992. *A Course on Damage Mechanics*. A Course on Damage Mechanics. Springer-Verlag.

Mäntylä, M., 1987. *An Introduction to Solid Modeling*. Computer Science Press, Inc., New York, NY, USA.

Murakami, S., 2012. *Continuum Damage Mechanics: A Continuum Mechanics Approach to the Analysis of Damage and Fracture*. Solid Mechanics and Its Applications. Springer.

Penna, S.S., 2007. *Pós-Processador para Modelos Bidimensionais Não-Lineares do Método dos Elementos Finitos*. Master’s thesis, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.

Penna, S.S., 2011. *Formulação Multipotencial para Modelos de Degradação Elástica - Unificação Teórica, Proposta de Novo Modelo, Implementação Computacional e Modelagem de Estruturas de Concreto*. Ph.D. thesis, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.

Pitangueira, R.L.S., 1998. *Mecânica de Estruturas de Concreto com Inclusão de Efeitos de Tamanho e Heterogeneidade*. Ph.D. thesis, PUC - Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro.

Rots, J.G., 1988. *Computational modeling of concrete fracture.* Ph.D. thesis, Delft University of Technology, Delft, Holanda.

Strouboulis, T., Babuška, I. and Copps, K., 2000. “The design and analysis of the generalized finite element method”. *Computer Methods in Applied Mechanics and Engineering*, Vol. 181, No. 1-3, pp. 43–69. ISSN 0045-7825.

Yang, Y.B. and Shieh, M.S., 1990. “Solution method for nonlinear problems with multiple critical points”. *AIAA Journal*, Vol. 28, No. 12, pp. 2110–2116.

10. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.