UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA DE ESTRUTURAS

On the Generalized Finite Element Method in Nonlinear Solid Mechanics Analyses

Dorival Piedade Neto

CORRECTED VERSION

(Original version is available at EESC-USP)

Text presented to the São Carlos School of Engineering of the University of São Paulo as one of the requisites for obtaining the degree of Doctor in Science (Structural Engineering).

Advisor: Sergio Persival Baroncini Proença

São Carlos
November 2013

Piedade Neto, Dorival

P613o On the Generalized finite element method in nonlinear Solid mechanics analyses / Dorival Piedade Neto; orientador Sergio Persival Baroncini Proença. São Carlos, 2013.

Tese (Doutorado) - Programa de Pós-Graduação em Engenharia de Estruturas -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2013.

1. Generalized finite element method. 2. Solid mechanics. 3. Nonlinear analysis. 4. Contact problem. 5. Object-oriented programming. 6. Python programming language. I. Título.

FOLHA DE JULGAMENTO

Candidato: Engenheiro **DORIVAL PIEDADE NETO**.

Título da tese: "On the generalized finite element method in nonlinear solid mechanics analyses".

Data da defesa: 29/11/2013

Comissão Julgadora:	**Resultado:**
Prof. Titular **Sergio Persival Baroncini Proença (Orientador)** (Escola de Engenharia de São Carlos/EESC)	APROVADO
Prof. Dr. **Ney Augusto Dumont** (Pontifícia Universidade Católica do Rio de Janeiro/PUC-RJ)	APROVADO
Prof. Dr. **Marco Lúcio Bittencourt** (Universidade Estadual de Campinas/UNICAMP)	Aprovado
Profa. Dra. **Larissa Driemeier** (Escola Politécnica/USP)	Aprovado
Prof. Titular **Walter Savassi** (Escola de Engenharia de São Carlos/EESC)	APROVADO

Coordenadora do Programa de Pós-Graduação em Engenharia Civil (Engenharia de Estruturas):
Profa. Associada **Ana Lucia Homce de Cresce El Debs**

Presidente da Comissão de Pós-Graduação:
Prof. Titular **Denis Vinicius Coury**

Acknowledgment

First, I would like to demonstrate my gratitude to God. The past four years were the most fascinating and, at the same time, challenging years in my life. A doctoral research is a pleasant and difficult task, which only can be performed with support of the family and friends. Words cannot express the gratitude I feel for such a large amount of people who helped me in the last four years.

Intentionally, I am not citing a single person name here. I refuse to take the risk of omitting a name, which in my opinion would be an unforgivable mistake. Besides, the most important is the gratitude I keep in my mind.

First of all, I would like to thank the most important persons in my life: my parents, my sister and my nephew. Also, I would like to thank my girlfriend and her family.

Of course, I would like to thank my advisor and my friends and colleagues from SET, especially the ones working under the supervision of my advisor.

I also want to thank all the professors I have worked with in the past years, as a graduate student and as an employee allocated in SET. Thank you for all I have learned from you. Of course, an especial acknowledgment goes to the referees evaluating the present thesis.

Also, it is important to thank the persons who helped me to correct the English mistake I have made in the present text. Certainly, without their help, many errors would remain in this final release. Unfortunately, I suppose some mistakes insist to remain here. I apologize and assume the responsibility for not noticing them.

Of course, a special ‘thank you’ goes to my colleagues who also works at SET, especially the ones allocated in the Informatics and Computational Mechanics Laboratory.

I also would like to thanks CAPES for the financial support in the first years of the research, during which I was not employed yet.

Resumo

PIEDADE NETO, D. (2013). Sobre o Método dos Elementos Finitos Generalizados em análises da Mecânica dos Sólidos não-linear. São Carlos. 212p. Tese (Doutorado). Escola de Engenharia de São Carlos, Universidade de São Paulo.

O Método dos Elementos Finitos Generalizados (MEFG) é um método numérico baseado no conceito de partição da unidade (PU) e inspirado no Método da Partição da Unidade (MPU) e o método das Nuvens-hp. De acordo com o MEFG, a PU é obtida por meio de funções de interpolação Lagragianas de primeiro grau, definidas sobre uma rede de elementos similar àquela do Método dos Elementos Finitos (MEF). De fato, o MEFG pode ser considerado uma extensão do MEF para a qual se pode aplicar enriquecimentos em regiões específicas do domínio, buscando melhorias na solução. Esta técnica já foi aplicada com sucesso em problemas com descontinuidades e singularidades, como os originários da Mecânica da Fratura. Apesar disso, a maioria das publicações sobre o método está relacionada a análises lineares. A presente tese é uma contribuição aos poucos estudos relacionados a análises não-lineares de Mecânica dos Sólidos por meio do MEFG. Um de seus principais tópicos é o desenvolvimento de um elemento de contato generalizado do tipo ‘segmento a segmento’ baseado no método *mortar*. Fenômenos não lineares devidos ao material e à cinemática também são considerados nos modelos numéricos. Um projeto de orientação a objetos para a implementação de uma plataforma de análises não-lineares foi desenvolvido, escrito em linguagem de programação Python. Os resultados validam a formulação e demonstram os ganhos e possíveis desvantagens da abordagem a problemas não lineares por meio do MEFG.

Palavras-chave: Método dos Elementos Finitos Generalizados; Mecânica dos Sólidos; análise não-linear; problemas de contato; Programação Orientada a Objetos; linguagem de programação Python.

Abstract

PIEDADE NETO, D. (2013). *On the Generalized Finite Element Method in nonlinear Solid Mechanics analyses*. São Carlos. 212p. Thesis (Doctoral) - São Carlos School of Engineering, University of São Paulo.

The Generalized Finite Element Method (GFEM) is a numerical method based on the Partition of Unity (PU) concept and inspired on both the Partition of Unity Method (PUM) and the hp-Cloud method. According to the GFEM, the PU is provided by first-degree Lagragian interpolation functions, defined over a mesh of elements similar to the Finite Element Method (FEM) meshes. In fact, the GFEM can be considered an extension of the FEM to which enrichment functions can be applied in specific regions of the problem domain to improve the solution. This technique has been successfully employed to solve problems presenting discontinuities and singularities, like those that arise in Fracture Mechanics. However, most publications on the method are related to linear analyses. The present thesis is a contribution to the few studies of nonlinear analyses of Solid Mechanics by means of the GFEM. One of its main topics is the derivation of a segment-to-segment generalized contact element based on the mortar method. Material and kinematic nonlinear phenomena are also considered in the numerical models. An Object-Oriented design was developed for the implementation of a GFEM nonlinear analyses framework written in Python programming language. The results validated the formulation and demonstrate the gains and possible drawbacks observed for the GFEM nonlinear approach.

Keywords: Generalized Finite Element Method; Solid Mechanics; nonlinear analysis; contact problem; Object-Oriented Programming; Python programming language.

Abbreviation's list

2D – Two-dimensional

3D – Three-dimensional

BEM – Boundary Element Method

CAPES – Coordination for the Improvement of Higher Level Education Personnel

CNPq – National Council for Scientific and Technological Development

DOF – degree of freedom

FE – Finite Element

FEM – Finite Element Method

GDOF – generalized degree of freedom

GFEM – Generalized Finite Element Method

IBVP – Initial Boundary Value Problem

OOP – Object-Oriented Programming

OO – Object-Oriented

PDE – Partial Differential Equation

PU – Partition of Unity

PUM – Partition of Unity Method

PVW – Principle of Virtual Work

SCIEnCE – São Carlos Integrated Environment for Computational Engineering

XFEM – eXtended Finite Element Method

Summary

1 -Introduction

The quantitative evaluation of the structural behavior of solids is based on the concepts of Solid Mechanics, which is a part of Continuum Mechanics. Among its fundamental concepts, both solids and fluids are idealized as continuous media, referred to as continuum, enabling the application of mathematical modeling tools as the ones arising from the differential and integral calculus. By employing such framework, the physical problem can be described by a set of differential equations, from which result function fields describing the idealized continuous medium behavior.

The classical solution procedure of the above mentioned systems of Partial Differential Equations (PDE) consists in obtaining analytical expressions accounting for displacement, stress and strain fields over the problem domain. In general, the solution is restricted to some predefined solid shape and to specific boundary conditions, as well to other simplifying hypotheses. For instance, a limited magnitude to the displacement and strain values and a simplified material response can be assumed at once, aiming to achieve a mathematical model describing a linear behavior. Moreover, in some cases, one can assume that the loading is applied in a quasi-static procedure, such that the inertial effects can be neglected, resulting in a static analysis.

Although a broad class of problems can be solved by means of such idealized conditions, several engineering applications exhibit features clearly distinguished from those assumed in the linear static theory. Actually, in real world applications, under certain conditions, some structures may exhibit large displacements and strains, such that the changes in their configuration influence their structural response. Furthermore, their material can present nonlinear behavior, resulting in immediate or time-dependent irreversible strain (plasticity or visco-plasticity). Such effects give rise to several nonlinear theories for effectively modeling these real world problems.

Advancing further in the complexity of modeling, the problem can be composed by not only a single solid, but by a set of distinct solids, which, during the mechanical process, may occupy the same region in the space at the same time. If this situation occurs, a contact problem is configured. According to Belystchko, Liu and Moran (2000), due to several reasons, contact problems are among the most difficult nonlinear problems and demand appropriate methodologies and algorithms for their successful treatment.

In fact, contact problems are intrinsically nonlinear, even though a linear behavior is assumed for the material and kinematic relations. The nonlinear character of contact problems arises from the fact that, in mathematical models, the phenomenon is defined by means of changes in the boundary conditions, which can be determined only during the solution process. Moreover, several additional mathematical conditions must be verified such that a stable solution framework can be achieved.

Regarding the contact among solids, the first scientist that correctly described (quantitatively) the problem was Heinrich Hertz, in 1881, in the article *"Über die Berührung fester Elasticher Körper"*. This contribution is nowadays regarded as the origin of the Contact Mechanics. A complete report on the historical aspects of the subject can be found in Kikuchi and Oden (1988).

According to Johnson (2003), after Hertz, new contributions on the subject have been made only in the twentieth century, with *Signorini's* article *"Sopra alcune questioni di elastostatica"*, presented in 1933, and *"Questioni di elasticita nonlinearizzata e semi linearizzata"*, published in 1959. Both articles are cited by Kikuchi and Oden (1988) as the works resuming the subject in the last century. Presently, frictionless contact problems among deformable solids and rigid obstacles are often referred to as *"Signorini problems"*.

Johnson (2003) also cites the books *"Contact Problems in Theory of Elasticity"*, published by *L. A. Galin* in 1953, and *"Contact Problems in the Classical Theory of Elasticity"*, published by *G.M.L. Gladwell*, in 1980. In both cases, the linear elasticity theoretical hypotheses are considered. The book by Kikuchi and Oden (1988) can be itself considered as an essential reference on the mathematical aspects of Contact Mechanics.

From the 1950's, general mechanical problems represented by Partial Differential Equations (PDE) have started to be solved numerically and computationally by means of the Finite Element Method (FEM). Initially applied only to linear problems, the method proved to be very powerful, such that the engineering and scientific community started to develop complementary techniques for solving also nonlinear problems. In the late 1960's nonlinear FEM applications quickly started to be published.

Regarding contact problems, the first FEM approaches date from the late 1970's and the first half of the 1980's. Some authors have included the subject as part of their general nonlinear solid mechanics books, like Bathe (1996) and Belystchko, Liu and Moran (2000), mainly presenting the general aspects of the numerical approach for solving contact problems. Currently, one finds entire books dedicated to the Computational Contact Mechanics, such as Laursen (2002) and Wriggers (2006).

One of the greatest difficulties faced in deriving contact element formulation is that the FEM treats the continuum as a set of discrete points. Accordingly, the unknowns are nodal displacements, from which stress and strain fields can be obtained. For the rest of the domain, the displacements, stresses and strains are obtained by means of shape functions associated with each finite element. Also, pressures over the elements surfaces are also treated as discrete nodal equivalent forces.

Such discrete characteristic represents a challenge for adequately treating the contact among solids. Actually, in the general case, the nodes distribution on the contact surfaces is not coincident, requiring additional strategies to enforce the contact forces and to avoid the penetration in the contact region.

Several studies on numerical techniques to enforce the contact conditions over contacting solid surfaces can be found in the literature. In general, they are based on the control of the so-called *Condition of Impenetrability*. Specifically for the case of two-dimensional (2D) problems, the target surface is represented by means of a line segment. Since the restrictions due to the contact are applied at solid nodes, these contact elements are referred to as *node-to-segment* contact elements.

A simple node-to-segment contact element for solving 2D Signorini problems is derived in the Master of Science dissertation by Piedade Neto (2009), providing accurate results for the tested problems. Even for the case of contact among solids and curved rigid obstacles, the penetration observed in the results is almost negligible, contrarily to the expectations based on the discrete nature of the displacement enforcement in the solid.

Laursen (1992) relates that such a discrete approach can lead to errors, especially for the case of frictional contact, and cites the work of Pires and Oden (1983), discussing the difficulties in proving the existence of solution for a Coulomb model frictional Signorini problem treated by a discrete contact strategy. The same authors advocate on the ineffectiveness of such a discrete approach for treating frictional contact, by resorting to physical arguments and citing the works of Oden and Martins (1985) and Kikuchi and Oden (1988).

Although node-to-segment contact elements lead to good results in some specific conditions, taking into account the above mentioned researches, the continuous enforcement of the contact conditions over the contacting region seems to be mathematically consistent and physically plausible. For a 2D model, this approach gives rise to contact elements represented by a line segment, associated with the finite elements' sides. Hence, such contact elements are referred to as *segment-to-segment* contact element.

A segment-to-segment contact element for the Signorini problem is derived in Piedade Neto (2009), employing both Lagrange Multipliers and penalty based formulations. The used contact detection techniques are based on Fisher and Wriggers (2005), who describe the derivation of the *mortar* contact elements.

The mortar contact elements are derived on the basis of the work of Bernardi, Maday and Patera (2001), which introduced the mortar technique aiming another purpose: the domain decomposition for discretization schemes with non-coincident meshes. Due to its efficiency, the technique was adapted to be used in contact problems, giving rise to a new class of contact elements. The subject is also presented in both computational contact mechanics books already cited, i.e., Laursen (2002) and Wriggers (2006).

The mortar method formulation results from the enforcement of the contact constraints over the solids' surfaces, by performing a numerical integration over the contact element domain (a segment in the 2D space). The contact element activation is also detected by means of the mean gap value, computed from the gaps observed at the integration points along the contact element. Like any other contact element, the element is activated if such gap value indicates a violation of the Condition of Impenetrability. Similarly, the traction in the contact element domain, which must obey the Traction Condition, is computed by a numerical integration, and so represents the mean traction value over the contact element.

Regarding the advantages of the continuous mortar approach, Laursen (2002) explains that, although good convergence results can be achieved by using node-to-segment contact elements, optimal convergence rates are obtained only when an integral enforcement of the contact is employed.

The same convergence gains are pointed out by Wriggers (2006) as one of the main advantages of the mortar contact element. In fact, the decrease in the number of iterations is especially notable for the quadratic interpolation finite element. The same author points out that this gain is almost negligible for the linear interpolation finite element. Such fact was also observed by Piedade Neto (2009) for the segment-to-segment contact element applied to the Signorini problem.

In spite of this advantage, Piedade Neto (2009) found that the segment-to-segment contact elements can present solutions violating the traction condition. According to this condition, tensile tractions are not allowed in the contacting surfaces, since the model does not consider adherence among the solids' surface.

Although physically not coherent with the adopted hypotheses, the mortar mathematical formulation in fact allows the occurrence of tensile tractions in part of the contact element. This occurs due to the fact that the method uses the mean value of the tractions as the reference for verifying the traction condition in a contact element. Therefore, even if tensile tractions arise in part of the contact element, if the mean value still indicates a compression tensile, the element remains active. This phenomenon is generally observed in contact elements located at the edges of the contact region.

On the other hand, the global structural behavior is clearly not affected by such phenomenon. Even the displacement errors near such regions are small. In fact, the major concern is related to errors in the stress and strain fields in the neighboring region this part of the contact interface.

It is worth to mention that the technical literature does not address explicitly references of this particular behavior of the mortar contact elements. In this respect, Wriggers (2006) reports that if more accurate stress results in the contact surface are necessary, one should employ an adaptative strategy in order to improve such results. Clearly, a more careful analysis of the theme is important to find out details about such numerical effect.

The results and the conclusions obtained in the author's Master of Science dissertation have motivated the inclusion of the contact subject as one of the main themes of the present thesis. Furthermore, studies on scientific databases have pointed out that the use of other nonconventional numerical methods for solving contact problems, as the Generalized Finite Element Method (GFEM), has not been sufficiently investigated.

The Generalized Finite Element Method (GFEM) is a numerical method, based on the traditional Finite Element Method (FEM), presenting as its main characteristic the possibility of improving the solution in specific regions of the problem's domain without demanding mesh refinement.

The GFEM has one of its bases in the Partition of Unity Method (PUM), derived by Melenk and Babuška (1996), in which the concept of enriching Partial Differential Equations (PDE) numerical solutions is introduced. The referred paper also includes the definition of the mathematical bases of the method, as well as its convergence proof. The other basis for the GFEM is associated to HP-Clouds Method, derived by Duarte and Oden (1996a, 1996b), in which the nodal enrichment strategy is presented.

One of the most notable situations for which the GFEM has shown to be clearly powerful is related to Fracture Mechanics problems. For these applications, the traditional FEM approach demands a hierarchical strategy, both for updating the crack topology and for improving the crack tip solution, in which the stress concentration demands an excessive mesh refinement. Clearly this approach requires the availability of robust generation algorithms and leads to a costly numerical simulation.

On the other hand, a more efficient approach for these problems is achieved by using the local enrichment GFEM strategy both to introduce the crack's discontinuities and to improve the stress and strain results in the crack tip region. The discontinuities can be inserted in the solid's domain by means of Heaviside function enrichments, while the stress concentration effects in the crack tips can be obtained by special function enrichments based on the asymptotic William solution for the two-dimensional crack problem (see Mohammadi (2008)).

In spite of this fact, the GFEM application to nonlinear solid analysis remains poorly investigated. For instance, the first publication in periodicals, regarding material nonlinear analysis, date from the last decade (Barros (2002) and Torres (2003)). Fewer publications on the GFEM application for contact problems are available, and, in general, these papers address very specific contact conditions. In fact, the nonlinear solid mechanics analysis by means of the GFEM seems to have been little studied and contributions regarding the method's performance in such conditions are still of great interest.

The effects of the contact on the internal crack surfaces modeled by using the GFEM Heaviside enrichment are presented in Dolbow (1999). Although frictional effects over the crack's surfaces is considered, the derived formulation is limited to the contact between a given crack's surfaces, and cannot be directly extended to the general case of contact, in which large relative displacements among distinct solids can occur.

A similar strategy was adopted by Khoei and Nikbakht (2006), who applied the formulation to model the contact between sliding surfaces. It is important to mention that distancing between the surfaces is not supported by the proposed formulation. Yet, Phadke (2005) presents an application of the GFEM to a very simple frictional contact in a truss element.

Taking all the above mentioned facts into account, it is important to underline that the derivation of GFEM enriched contact elements for the general contact problems, with no restrictions to the relative surface displacements, is a subject not found in the bibliographic research. Thus, it represents an original contribution for the contact mechanics and the Generalized Finite Element Method. This is true not only for the case of deformable solid contact, but also for the Signorini problem. Actually, the derivation of such generalized contact elements is the guideline of the contribution hereby presented.

Moreover, if one takes into account the reported state of the art on the nonlinear analysis using GFEM, a critical analysis of the numerical aspects of the method in such conditions seems to be an additional important contribution to emphasize, compatible with a doctoral research. Still in the field of nonlinearities, cases of elastoplasticity, dynamic analysis and hyperelastic contact solved by means of the GFEM constitute topics also studied in the present work.

Finally, it is also important to highlight a contribution resulting from the present research: the GFEM computational implementation and the conceiving of an efficient Object-Oriented design for the method.

Even though several research works on the GFEM were performed at the Structural Engineering Department of the São Carlos School of Engineering in the past years, the codes developed in such works were not conceived aiming expansions, and therefore, did not support the implementation of a general nonlinear analysis framework. Besides, since the development of a new computational code has shown to be necessary, a natural design choice is to conceive a data structure that enables several developers to work collaboratively in the same code. In this context, the Object Oriented Programming (OOP) arises as the programming paradigm to be employed in the code's development.

These OOP technical advantages were perceived by the FEM code developers in the 1990's. One of the first references pointing towards this direction is Alves Filho and Devloo (1991). In the following year, a more detailed description of the OOP application for the FEM was provided by Zimmermann et al (1992) and Dubois-Pélerin et al (1993). Since then, hundreds of papers have been published on the application of the OOP to solve PDE, regarding both the FEM and the Boundary Element Method (BEM) (Mackerle (2000)). The subject worthiness has been established by the development of PhD Theses about the FEM OOP, as Hedelal's (1994) and Archer's (1996).

The OOP benefits in the FEM influenced some researchers to employ such programming paradigm to implement GFEM codes. However, in contrast to the hundreds of papers on the traditional FEM OO programming, the theme has been little published for the GFEM. Actually, Pereira's (2004) dissertation is one of the few texts on the OOP GFEM.

Regarding publications in periodicals, a description of the eXtended Finite Element Method (XFEM) is presented by Bordas et al. (2007). It is important to notice that the XFEM is similar to the GFEM, and both can almost be considered the same method, according to Belytschko, Gracie and Ventura (2009).

Considering those facts, a last (but not less important) contribution of the present work is the development of an Object Oriented (OO) class framework suitable for the GFEM, designed to achieve the pointed goals for the resulting code. The existing knowledge presented in the technical literature was not neglected in the code design, but some original design options have been chosen to better support the method's characteristics. Documenting the developed OO framework is also a contribution of the present thesis, evidenced by the publication of the proposed OO design for the GFEM in an international journal (Piedade Neto, Ferreira and Proença (2013)).

In what follows, the theoretical foundations of the mechanical problems hereby considered are addressed in Chapter 2. First, one presents the Initial Boundary Value Problem (IBVP) definition for the kinematically linear model, considering either a linear elastic constitutive model or a von Mises based isotropic model. Next, the kinematically nonlinear formulation of the IBVP is introduced, defining again two distinct constitutive models: the Saint Venant-Kirchhoff and the Neo-Hookean material models. In the following, the weak form of the IBVP is briefly presented, based on the Principle of Virtual Work (PVW). Once the weak form is presented for the single solid problem, the strong form of the contact IBVP is presented, defining the conditions of the frictionless contact problem. Then, the contact IBVP weak form is presented.

In Chapter 3 the numerical approach of the previously described physical problems is presented. Following the formulation of the traditional Finite Element Method (FEM), the Generalized Finite Element Method (GFEM) is presented as an extension of the previous one. The fundamental concepts of Partition of Unity (PU) and nodal enrichment are briefly discussed and the enrichment functions used in the present work are defined. Once the GFEM is presented, the techniques necessary to solve the nonlinear problems arising from the theory presented in Chapter 2 are defined. Special focus is given to the contact numerical approach, presenting the definition of the generalized contact elements derived in the present work. Finally, the resulting framework for solving generic nonlinear problems is presented, encompassing functionalities for solving the entire set of nonlinear models previously discussed.

Chapter 4 is reserved for documenting the Object Oriented Programming adopted design. The chapter starts with a brief description of the programming paradigm, pointing out its characteristics, benefits and possible drawbacks. The programming language adopted to develop the code is also briefly described. The symbols and conventions used in the GFEM

OO design, which composes most of the chapter, are also displayed. Frequent references to the previous chapter are addressed in order to clarify the overall perspective.

In Chapter 5 a set of simple numerical examples is provided, in order to assess the correctness of both the developed computational framework and the derived formulation. Since the use of the GFEM for solving some of the nonlinear problems is original, Chapter 5 represents more than a simple testing of the computational framework and indeed aims the validation of the GFEM formulation applied to nonlinear solid mechanics analysis.

In Chapter 6 examples of higher complexity are provided. The results achieved by means of refined FEM analyses are used as the reference solution, since in general such examples do not present analytical solutions, in opposite to what happens for most of the examples presented in Chapter 5. These FEM models were also solved using the developed computational framework.

Chapter 7 is dedicated to the discussion of all the aspects treated in the thesis, regarding in particular the developed generalized contact element formulation and the numerical results associated to them. The numerical results of the other implemented nonlinear analysis features are also commented, especially in the cases for which they represent original contributions for the GFEM. The chapter also includes a brief general conclusion on the conceived OO design for the GFEM.

2 -Theoretical foundations

The present chapter is devoted to the mathematical modeling of the mechanical problems hereby considered. Initially, a brief exposition on the theoretical concepts of Continuum Mechanics is presented. The general guidelines for such reviewing are the books of Laursen (2002), Wriggers (2006) and Bonet and Wood (2008). Following the strong form of the Initial Boundary Value Problem (IBVP), its weak form based on the Principle of Virtual Works (PVW) is derived, aiming to the numerical approximation to be introduced in Chapter 3.

The first problem to be considered is related to a single solid model under linear elastic behavior. Afterwards, by keeping a linear kinematic approach, the small strains elastoplastic material model is introduced. The elastoplastic framework is restricted to the von Mises stress based model with nonlinear isotropic hardening law.

Advancing to a more general model, the previous restrictions to the small displacement hypothesis are circumvented by adopting a nonlinear kinematical description. Accordingly, nonlinear strain and associated stress tensors are presented, and related by the Saint Venant-Kirchhoff constitutive model. The occurrence of finite strains is also accounted for the work conjugate stress tensor by means of a Neo-Hookean hyperelastic model.

Next, the hypotheses and mathematical conditions for representing the interaction between solids are presented, basing the formulation of the contact problem.

The option for defining the problem in specific conditions (two-dimensional space and specific material models, for instance) is mostly related to the objective of addressing the theoretical foundations in a straight and concise fashion. In spite of that, one underlines that such specific strategy does not exclude the possibility to extend the numerical approach to more general cases, for which an extended modeling follows a similar path.

2.1. The Initial Boundary Value Problem (IBVP) - linear kinematics

Let $\boldsymbol{S}$ be a solid idealized as a continuous media, referred to as continuum. To each of the continuum's point corresponds one point defined in an Euclidean space, giving rise to an entity referred to as material point. The study of the solid's behavior relies in the response of each of its material points due to the action of mechanical forces. A fixed orthogonal basis of unit vectors $\boldsymbol{e}_1$, $\boldsymbol{e}_2$ and $\boldsymbol{e}_3$ is attached to the Cartesian adopted referential. Although the present study is focused on two-dimensional idealizations, one assumes temporarily that the solid configuration is positioned in a three-dimensional Euclidean space. This general conception is indicated in Figure 2.1.

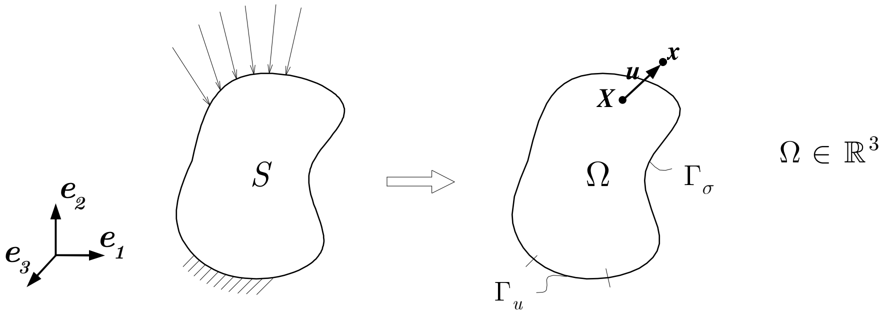


Figure 2.1. Solid idealized in the three-dimensional Euclidean space.

By definition, the material points representing the particles of the solid belong to an open configuration set named Ω. The boundary of the configuration set is named $\partial\Omega$. The union of Ω and $\partial\Omega$ gives rise to the closed set $\overline{\Omega}$. Each of the points X belonging to $\overline{\Omega}$ can be referenced by its initial position vector $\boldsymbol{X}$, resulting in the following Lagrangian description framework:

$$\boldsymbol{X} = X\boldsymbol{e}_1 + Y\boldsymbol{e}_2 + Z\boldsymbol{e}_3 = \left(X,Y,Z\right)^T . \tag{2.1}$$

During the mechanical process, the continuum points can move from their initial position, occupying a new configuration at each time instant *t* belonging to a time domain $\mathrm{I} = \{t \in \mathbb{R} \big| 0 \leq t \leq T\}$. Such position $\boldsymbol{x} = \varphi\left(\boldsymbol{X},t\right)$ is here referenced as the current position of the point $\boldsymbol{X}$ at a given time *t*. The current position $\boldsymbol{x}$ is also represented by means of a position vector and Cartesian coordinates such that

$$\boldsymbol{x} = x\boldsymbol{e}_1 + y\boldsymbol{e}_2 + z\boldsymbol{e}_3 = \left(x,y,z\right)^T . \tag{2.2}$$

The difference between the current and the initial position of a given point defines a vector field $\boldsymbol{u}$ representing the point displacement at a given time *t*, such that

$$\boldsymbol{u}\left(\boldsymbol{X},t\right) = \boldsymbol{x}(\boldsymbol{X},t) - \boldsymbol{X} . \tag{2.3}$$

The displacement vector components can be directly computed from the initial and current position components, resulting in

$$\boldsymbol{u} = (x - X)\boldsymbol{e}_1 + (y - Y)\boldsymbol{e}_2 + (z - Z)\boldsymbol{e}_3 = (u,v,w)^T . \tag{2.4}$$

The displacement field encompasses rigid body motions, composed by linear displacements of the whole body or its rotation in relation to a given point, and also possible changes in the solid shape, referred to as deformation. The definition of a measure for the strain, valid along the whole solid domain, is fundamental to the study of the solid's deformation.

An important hypothesis for defining a tensorial quantity representing the strain at a given point relies on the magnitude of the displacement field values observed in the continuum during the mechanical process. If $\boldsymbol{du} << \boldsymbol{dX}$, a linear relation between the displacement and the strain can be stated, given rise to the so-called Cauchy strain or linear strain tensor, defined as

$$\boldsymbol{\varepsilon} := \nabla^s \boldsymbol{u} = \tfrac{1}{2}\left(\nabla \boldsymbol{u}^T + \nabla \boldsymbol{u}\right), \tag{2.5}$$

in which ∇^s operator represents the symmetric part of the gradient applied over the displacement field. Considering the adopted system of reference, a matrix presentation of such second order strain tensor is

$$\varepsilon = \begin{bmatrix} \varepsilon_{xx} & \gamma_{xy} & \gamma_{xz} \\ \gamma_{yx} & \varepsilon_{yy} & \gamma_{yz} \\ \gamma_{zx} & \gamma_{zy} & \varepsilon_{zz} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial u}{\partial X} & \dfrac{1}{2}\left(\dfrac{\partial u}{\partial Y} + \dfrac{\partial v}{\partial X}\right) & \dfrac{1}{2}\left(\dfrac{\partial u}{\partial Z} + \dfrac{\partial w}{\partial X}\right) \\ \dfrac{1}{2}\left(\dfrac{\partial u}{\partial Y} + \dfrac{\partial v}{\partial X}\right) & \dfrac{\partial v}{\partial Y} & \dfrac{1}{2}\left(\dfrac{\partial w}{\partial Y} + \dfrac{\partial v}{\partial Z}\right) \\ \dfrac{1}{2}\left(\dfrac{\partial u}{\partial Z} + \dfrac{\partial w}{\partial X}\right) & \dfrac{1}{2}\left(\dfrac{\partial w}{\partial Y} + \dfrac{\partial v}{\partial Z}\right) & \dfrac{\partial w}{\partial Z} \end{bmatrix}. \tag{2.6}$$

It is worth of mentioning that the linear strain tensor presented in (2.6) does not fulfill the frame invariance requirements, see Spencer (2004). The frame indifference issue is also referred to in the technical literature as objectivity. Despite of this fact, for situations in accordance with the infinitesimal displacement hypothesis, it represents a suitable strain measurement entity.

Nonzero strain values evidence that the solid's particles present relative movements among each other, which give rise to internal forces. The distribution of such internal forces can be represented by defining a stress tensor.

Moreover, aiming to hereafter focus on a linear relation between the stress and the linear strain tensor, giving rise to a linear elastic model, one adopts the Cauchy stress tensor $\boldsymbol{\sigma}$, which also can be represented by means of a symmetric matrix, defined as

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \sigma_{zz} \end{bmatrix}. \tag{2.7}$$

Further information on the Cauchy stress tensor can be found in Timoshenko and Goodier (1970).

It is important to notice that, on the other hand, the emergence of internal forces is caused by external mechanical loads applied over the solid. Part of these loads is acting directly in the solid's domain, being mathematically represented by a vector $\boldsymbol{f}$ of body forces. Another group of external loads is applied in specific regions of the solid's boundary, therefore being considered in the model as static boundary conditions. Depending on the way one considers the boundary conditions, the set $\partial\Omega$ can be divided into two distinct and complementary parts referred to as Γ_u and Γ_σ, such that

$$\begin{aligned} &\Gamma_\sigma \bigcup \Gamma_u = \partial\Omega, \\ &\Gamma_\sigma \bigcap \Gamma_u = \varnothing. \end{aligned} \tag{2.8}$$

The portion Γ_u is associated with the so-called Dirichlet boundary conditions, in which prescribed displacement values are enforced in such region. Dirichlet boundary conditions can be summarized as

$$\boldsymbol{u} = \overline{\boldsymbol{u}}, \text{for all } \boldsymbol{X} \in \Gamma_u, t \in \mathbf{I}\,. \tag{2.9}$$

The portion Γ_σ is associated with the so-called Neumann boundary conditions. Prescribed traction values are enforced in such region, and can be summarized as

$$\boldsymbol{\sigma n} = \overline{\boldsymbol{t}}, \text{for all } \boldsymbol{X} \in \Gamma_\sigma, t \in \mathbf{I}\,, \tag{2.10}$$

in which $\boldsymbol{n}$ is the vector representing the outward unit normal to Γ_σ.

Within the classical mechanics framework, the equilibrium of the solid is described by the Newton's law of motion. Accordingly, the inertial forces can play an important role in the phenomenon, depending on the velocities and acceleration observed in the body's particles. In this context, the displacement and vector field values over the whole solid's domain at the initial reference constitute important information to mathematical modeling. Hence, the so-called initial conditions are defined as

$$\begin{aligned} &\dot{\boldsymbol{u}}\Big|_{t=0} = \boldsymbol{v}_0, \text{for all } \boldsymbol{X} \in \overline{\Omega}, \\ &\boldsymbol{u}\Big|_{t=0} = \boldsymbol{u}_0, \text{for all } \boldsymbol{X} \in \overline{\Omega}. \end{aligned} \tag{2.11}$$

In the last relation the dot over the displacement $\boldsymbol{u}$ denotes the first time derivative of the displacement field, whereas $\boldsymbol{v}_0$and $\boldsymbol{u}_0$ are initial values prescribed for the velocity and displacement at the initial time. Likewise, two dots over $\boldsymbol{u}$ represent the second time derivative, resulting in the acceleration field.

In order to complete the definition of the Initial Boundary Value Problem (IBVP), the linear momentum balance in accordance with the Newton's Law must be verified at each particle of the solid for all time $t \in \mathbf{I}$ such that

$$\nabla \cdot \boldsymbol{\sigma} + \boldsymbol{f} = \rho \ddot{\boldsymbol{u}} \,. \tag{2.12}$$

Above, ρ is the mass density of the solid at the given material point. Regarding the notation employed in equation (2.12), the dot product (contraction) of the gradient operator ∇ applied to the Cauchy stress field results in the divergent of such stress tensor.

In spite of the fact that the real world problems are related to a three-dimensional space, some practical problems may be represented only by means of a two-dimensional idealization. In the present work one focuses on such applications, so that the presented theory is attached to an Euclidean space $\mathbb{R}^2$.

The idealized two dimensional Euclidean space is placed in the plane defined by the unit vectors $\boldsymbol{e_1}$ and $\boldsymbol{e_2}$ indicated in Figure 2.1 (XY plane). The real solid thickness, in the plane stress case, and the member axis, in the plane strain idealization, are associated to the Z coordinate, which is orthogonal to the *XY* plane.

In such system of reference, for the plane stress model, it is possible to conclude that

$$\tau_{zy} = \tau_{yz} = \sigma_z = 0. \tag{2.13}$$

The correspondent conditions for the plane strain state are

$$\gamma_{zy} = \gamma_{yz} = \varepsilon_z = 0. \tag{2.14}$$

Finally, the IBVP definition is completed by the relation between the stress field, represented by the second order tensor $\boldsymbol{\sigma}$, and the strain field, represented by the second order tensor $\boldsymbol{\varepsilon}$. The relationship between the two tensors, expressed in a general form by means of the fourth order constitutive tensor $\mathbb{C}$, is given by

$$\boldsymbol{\sigma} = \mathbb{C} : \boldsymbol{\varepsilon} \,. \tag{2.15}$$

Several different components are defined to the $\mathbb{C}$ tensor, if the general case is considered. On the other hand, if simpler conditions are considered, as material isotropy and a linear elastic behavior, the number of independent components decreases significantly. Further simplifications can be achieved if plane stress and plane strain hypotheses are adopted.

The above presented formulation defines the so-called strong form of the IBVP.

In what follows, two different constitutive models exploring the linear kinematic formulation are addressed.

2.1.1. Linear elastic constitutive model

Considering an isotropic linear elastic material, the stress strain relation presented in (2.15) for the plane strain hypotheses is written using matrix notation as

$$\underbrace{\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}}_{\boldsymbol{\sigma}} = \underbrace{\frac{E}{\left(1+\upsilon\right)\left(1-2\upsilon\right)}\begin{bmatrix} 1-\upsilon & \upsilon & 0 \\ \upsilon & 1-\upsilon & 0 \\ 0 & 0 & \left(1-2\upsilon\right)/2 \end{bmatrix}}_{\mathbb{C}}\underbrace{\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix}}_{\boldsymbol{\varepsilon}}, \qquad (2.16)$$

in which *E* is the Young modulus and υ is the Poisson's ratio.

For the plane stress, the relation is written as

$$\underbrace{\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}}_{\boldsymbol{\sigma}} = \underbrace{\frac{E}{\left(1-\upsilon^2\right)}\begin{bmatrix} 1 & \upsilon & 0 \\ \upsilon & 1 & 0 \\ 0 & 0 & \left(1-\upsilon\right)/2 \end{bmatrix}}_{\mathbb{C}}\underbrace{\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix}}_{\boldsymbol{\varepsilon}}. \qquad (2.17)$$

2.1.2. Small strain elastoplastic constitutive model

Plasticity is a nonlinear phenomenon associated to the material response, for which a non unique relation between stress and strain is observed upon loading and unloading above certain level of stress state.

Hence, plasticity appears when the stress state in a given point of the continuum exceeds a limit, characteristic of the material. In general, below such limit, a linear elastic response of the material prevails. For a stress-based elastoplastic model, the reference initial limit is often taken as the one-dimensional (1D) yield stress σ_Y. A yield criterion furnishes a scalar value equivalent to the local stress state intensity to be compared to the yield stress limit.

When the yield stress is exceeded at a given point, part of the strain components observed at that point turns to be irreversible. Therefore, if the load is removed, a residual strain state remains in the solid. Consistent to the small strain hypothesis, an additive strain relation can be assumed, such that

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p, \qquad (2.18)$$

in which ε is the total strain, ε^e is the elastic (reversible) strain and ε^p is the plastic (irreversible) strain. The stress tensor is associated to the elastic counterpart of the strain, such that

$$\boldsymbol{\sigma} = \mathbb{C} : \varepsilon^e = \mathbb{C} : \left(\varepsilon - \varepsilon^p\right). \tag{2.19}$$

The relation shown in (2.19) is also assumed as valid for the stress and strain tensor rates. Moreover, in the present work, one considers that the plastic behavior is described by a rate-independent plastic model, as shown in Souza Neto, Perić and Owen (2008).

Besides the occurrence of irreversible strain, the plastic behavior can also incorporate a phenomenon denominated hardening. On the contrary to the perfectly plastic case, for which no stress increment is admissible beyond the yield stress, in the hardening regime, the material is able to sustain stress increments in correspondence to the accumulated plastic strain.

Mathematically, the hardening is described by defining a hardening law. The perfectly plastic model can be understood as a particular case for which the hardening is void. For the present study, a negative hardening, also referred to as softening, is not considered.

The yield criterion of the elastoplastic model to be implemented in the computational framework is based on the von Mises criterion, according to which the stress state level is determinated by the deviatoric part of the stress tensor. Therefore, the yield criterion is defined as

$$f(\boldsymbol{\sigma}, \alpha) := \sqrt{2J_2\left(\boldsymbol{\sigma}\right)} - \sqrt{\tfrac{2}{3}}K(\alpha) = \left\|\boldsymbol{s}\left(\boldsymbol{\sigma}\right)\right\| - \sqrt{\tfrac{2}{3}}K(\alpha). \tag{2.20}$$

In (2.20) J_2 is the second invariant of the deviatoric stress tensor $\boldsymbol{s}$ and $K(\alpha)$ is the adopted isotropic hardening law. In our study only an isotropic hardening model is considered. For details on isotropic, kinematic or even mixed hardening laws, we refer to Proença (2010), Souza Neto, Perić and Owen (2008) and Simo and Hughes (1998).

The adopted isotropic hardening law is based on the one presented in Simo and Hughes (1998), being defined as

$$K(\alpha) := \sigma_Y + \alpha\, k + \left(\sigma_\infty - \sigma_Y\right)\left(1 - e^{-\alpha\omega}\right), \tag{2.21}$$

in which α is a parameter associated with the local hardening 'level' at a given material point, whereas k and ω are the material parameters respectively associated with the linear and exponential components of the hardening law. The yield and 'infinity' stress values, σ_Y and

σ_∞, are material parameters. It is important to notice that the hardening law presented in (2.21) reduces to the classical bi-linear hardening law if σ_∞ is equal to σ_Y.

Relation (2.20) is written in such a way that admissible stress states result in non-positive values for the yield criterion, which is mathematically equivalent to the expression

$$f(\boldsymbol{\sigma},\alpha) \leq 0\,. \tag{2.22}$$

It is important to notice that the inequality (2.22) defines a region in the stress space. The stress states for which the yield criterion is negative define an open set referred to here as elastic domain. Stress states positioned inside the elastic domain are associated with a linear elastic response, resulting in reversible strain increments and a linear stress strain relation, according to the constitutive relations in (2.16) and (2.17).

On the other hand, a zero value for the yield criterion defines geometrically a surface in the stress space. Such surface represents the boundaries of the elastic domain. Stress states positioned over the surface are related to a plastic material response, and are defined here as the plastic domain. For stress states in the plastic domain, three different possible conditions can be observed.

If the load increment is such that the stress state at a given point of the solid tends to advance in the elastic domain, one observes the behavior predicted by the linear elastic model, with no plastic strain increments, according to the linear elastic constitutive relation. In this case, one observes the so-called elastic unloading.

On the other hand, if the stress state tends to advance beyond the admissible space, defined by the yield criterion, one observes a plastic response. Since positive values for *f* are not admissible, the plastic surface changes its configuration, according to the defined hardening law, finding an equilibrium configuration different from the one that would be predicted by means of the linear elastic model. Irreversible plastic strains are observed, and the stress tensor increment is computed according to an elastoplastic constitutive relation. In this case, one observes the so-called plastic loading.

The third possible situation is related to the less trivial case for which the observed stress and strain increments lead to a state that lies exactly on the same previous plastic domain surface. This case is also referred to as neutral loading.

Taking all these aspects into account, it becomes clear that the elastoplastic model also demands a criterion for the evolution of the plastic strain. For instance, for the adopted von Mises stress based yield criterion, the plastic strain increment is associated with the deviatoric stress tensor, and is defined as

$$\dot{\boldsymbol{\varepsilon}}_p = \dot{\lambda}\frac{\boldsymbol{s}}{\left\|\boldsymbol{s}\right\|} = \dot{\lambda}\boldsymbol{N}\,. \tag{2.23}$$

In equation (2.23) $\dot{\lambda}$ is a scalar value defining the magnitude of the plastic strain rate at a given point of the solid domain. The reasons for adopting $\boldsymbol{N}$ in (2.23) are based on further mathematical issues which are not discussed in the present text, giving rise to a so-called associative plasticity model. For the ones interested in the justifications on this theme, we reference Simo and Hughes (1998).

Taking into account that the plastic strain is not reversible, the $\dot{\lambda}$ scalar must always yield non negative values. On the other hand, it presents a zero value if no plastic strain rate is observed. So, the following condition is always observed in the elastoplastic model:

$$\dot{\lambda} \geq 0\,. \tag{2.24}$$

The phenomenological behavior already discussed and defined by means of the conditions stated in (2.22) and (2.24) can be encompassed in the so-called plasticity complementarity condition, stated as

$$f\dot{\lambda} = 0\,. \tag{2.25}$$

Altogether, the conditions stated in (2.22), (2.24) and (2.25) define the so-called Karush-Kuhn-Tucker conditions, which arise from the nonlinear programming, see Luenberger (2005). Such conditions appear also in the definition of the contact problem, to be discussed later in the present text.

A fourth condition can be defined by considering the variation rate of the yield criterion if one takes into account the expected elastoplastic model behavior. Such a condition, referred to as consistency condition, is defined as

$$\dot{f}\dot{\lambda} = 0\,. \tag{2.26}$$

In order to completely define the mathematical formulation describing the elastoplastic constitutive model, one must also define a rule for the evolution of the hardening parameters appearing in the hardening law. For models considering both isotropic and kinematic hardening, the involved parameters can be arranged as vector components.

For the particular case in which only isotropic hardening is considered, as hereby assumed, by relation (2.21) the hardening involves only the α parameter. The current value of this parameter depends on the stress strain history of a given point of the solid, and defines the elastic and plastic domains already mentioned earlier. For the specific case of the adopted model (a ‘strain hardening’ approach), the evolution law is described by means of the following relation:

$$\dot{\alpha} = \sqrt{\frac{2}{3}}\dot{\lambda}\,. \tag{2.27}$$

Regarding the formulation for computing $\dot{\lambda}$, it is derived from the consistency condition, according to which $\dot{f}$ is void for nonzero values of $\dot{\lambda}$, which is expressed by the equation

$$\dot{f} = \frac{\partial f(\boldsymbol{\sigma},\alpha)}{\partial t} = \frac{\partial f}{\partial \boldsymbol{\sigma}}\frac{\partial \boldsymbol{\sigma}}{\partial t} + \frac{\partial f}{\partial \alpha}\frac{\partial \alpha}{\partial t} = f_{\boldsymbol{\sigma}}\dot{\boldsymbol{\sigma}} + f_{\alpha}\dot{\alpha} = 0\,. \tag{2.28}$$

In (2.28) $f_{\boldsymbol{\sigma}}$ and f_{α} represent the derivatives of the yield criterion with respect to the stress tensor and the isotropic hardening parameter, respectively. By substituting relation (2.19) in its time rate ‘version’ and (2.27) in equation (2.28), one obtains

$$\dot{f} = f_{\boldsymbol{\sigma}}\mathbb{C}\left(\dot{\boldsymbol{\varepsilon}} - \dot{\boldsymbol{\varepsilon}}^{p}\right) + f_{\alpha}\dot{\alpha} = f_{\boldsymbol{\sigma}} : \mathbb{C}\dot{\boldsymbol{\varepsilon}} - \dot{\lambda}\left(f_{\boldsymbol{\sigma}} : \mathbb{C}\boldsymbol{N} + \frac{2}{3}\frac{dK(\alpha)}{d\alpha}\right) = 0\,. \tag{2.29}$$

Considering that in the adopted model $f_{\sigma} = \boldsymbol{N}$, and using the hardening law stated in (2.21), the above condition results in

$$\dot{\lambda} = \frac{\boldsymbol{N} : \mathbb{C}\dot{\boldsymbol{\varepsilon}}}{\boldsymbol{N} : \mathbb{C}\boldsymbol{N} + \frac{2}{3}\left[k + \omega\left(\sigma_{\infty} - \sigma_{Y}\right)e^{-\alpha\omega}\right]}\,. \tag{2.30}$$

Finally, it is possible to observe that when $\dot{\lambda}$ is not equal to zero, i.e., when plastic behavior occurs, the general form of the constitutive tensor can be computed by means of the relation

$$\mathbb{C}^{ep} = \mathbb{C} - \frac{\boldsymbol{N} : \mathbb{C} \otimes \boldsymbol{N} : \mathbb{C}}{\boldsymbol{N} : \mathbb{C}\boldsymbol{N} + \frac{2}{3}\left[k + \omega\left(\sigma_{\infty} - \sigma_{Y}\right)e^{-\alpha\omega}\right]}\,. \tag{2.31}$$

The relation (2.31) is the general form of the fourth-order elastoplastic constitutive tensor, also referred to as tangent constitutive tensor. It represents the exact tangent stiffness at a given point of the solid, considering the current stress strain rates and the accumulated local plastic strain.

The numerical strategy for solving the elastoplastic problem is presented in Chapter 3. Accordingly, the nonlinear constitutive model is considered in an incremental form and convergence difficulties can emerge if one adopts directly the tangent constitutive tensor presented in (2.31).

The consistent computation of the constitutive tensor, to be used in the incremental constitutive model, gives rise to the so-called algorithmic tangent constitutive tensor. Such tangent constitutive tensor is presented in Chapter 3, both for the plane stress and the plane strain model.

The presented elastoplastic formulation was derived for the three dimensional Euclidean space, considering the whole set of stress and strain components. Departing from this formulation the particular issues related to the plane stress and plane strain hypotheses can be derived.

Similarly to what happens for the linear elastic model, the plane strain and plane stress model relies on the hypotheses that the shear stress and strain components related to the third dimension ($\tau_{xz}, \tau_{yz}, \varepsilon_{xz}$ and ε_{yz}) are equal to zero. In addition to it, for the plane strain model the normal strain component ε_z is neglected, being set to zero, whereas for the plane stress, such assumption is made for the normal stress component σ_z.

In spite of this similarity, the consequences in the elastoplastic model are not trivial in either cases, as it happens for the linear elastic model. Specifically for the plane strain model, the resulting model may be derived in a quite simple straightforward manner, by considering the hypotheses commented in the last paragraph.

For the plane stress model, the adopted hypotheses result in major changes in the formulation. In fact, the straightforward application of the formulation to plane stress results in elastic and plastic strain components for which the zero normal stress component hypotheses of the plane stress model is not guaranteed.

Regarding the plane stress formulation for the elastoplastic problem, Souza Neto, Perić and Owen (2008) present several techniques for solving the elastoplastic problem consistently accounting for its hypotheses. In the present work, the adopted approach to treat the plane stress problems is based on Simo and Hughes (1998).

Accordingly, the mathematical strategy proposed is based on the definition of constrained stress spaces both for the stress field $\boldsymbol{\sigma}$ and its deviatoric part $\boldsymbol{s}$, for which one considers the zero components hypothesis of the plane stress model. Departing from those constrained spaces, one defines a mapping $\overline{\boldsymbol{P}}$ relating the constrained $\boldsymbol{\sigma}$ and $\boldsymbol{s}$ fields. A mapping $\boldsymbol{P}$ relating the constrained deviatoric stress and strain fields is also defined. Making use of such mappings, the general formulation presented is then particularized for the plane stress model.

The numerical formulation and the algorithms for solving plane stress and plane strain rate independent elastoplastic problems, considering a von Mises isotropic hardening model, are presented in Chapter 3.

2.2. The Initial Boundary Value Problem - nonlinear kinematics

The fundamental hypothesis assumed for the derivations presented in the previous section refers to the small magnitude of the values observed for the deformation fields. Although many practical applications can be modeled according to this restriction, many others are related to deformations for which the strains cannot be considered infinitesimal anymore. For these applications, a finite displacement formulation is necessary.

Differently from the linear kinematic description, for the finite displacement model, the changes in the solid configuration cannot be neglected for accounting the equilibrium. By adopting a Lagrangian description framework, the material particles are referenced by their position at the initial configuration.

Aiming to describe these changes, one defines an infinitesimal vector $\boldsymbol{dX}$ representing the distance between two neighboring points of the continuum at initial time. At time t, the distance between these points may have changed, resulting in a different vector $\boldsymbol{dx}\left(\boldsymbol{X},t\right)$. The relation between these vectors can be defined by means of a second order tensor $\boldsymbol{F}$, resulting that

$$\boldsymbol{dx}(\boldsymbol{X},t) = \boldsymbol{F}(\boldsymbol{X},t)\,\boldsymbol{dX}\,. \tag{2.32}$$

The tensor $\boldsymbol{F}$ represents a mapping between the reference and the current deformed configuration. It is a key entity in the finite displacement model, being referred to as the deformation gradient. Another expression for the deformation gradient is given by

$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}\left(\boldsymbol{X},t\right)}{\partial \boldsymbol{X}} = \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{X}} + \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{X}} = \boldsymbol{I} + \nabla \boldsymbol{u}\,, \tag{2.33}$$

in which $\boldsymbol{I}$ is the second-order identity tensor and $\nabla \boldsymbol{u}$ is the displacement gradient.

It is important to notice that for this definition no assumption is made in the magnitude of the displacement values. This does not mean at all that no restrictions must be stated on the displacement fields. For instance, in order to achieve a consistent mapping, the relation stated in (2.32) must be inversible, i.e, there must be a tensor $\boldsymbol{F}^{-1}$ such that

$$\boldsymbol{dX} = \boldsymbol{F}^{-1}\boldsymbol{dx}\,. \tag{2.34}$$

For the reference system adopted (Figure 2.1), the deformation gradient is represented by the matrix computed as

$$\boldsymbol{F}=\begin{bmatrix}\dfrac{\partial x}{\partial X} & \dfrac{\partial x}{\partial Y} & \dfrac{\partial x}{\partial Z}\\ \dfrac{\partial y}{\partial X} & \dfrac{\partial y}{\partial Y} & \dfrac{\partial y}{\partial Z}\\ \dfrac{\partial z}{\partial X} & \dfrac{\partial z}{\partial Y} & \dfrac{\partial z}{\partial Z}\end{bmatrix}=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}+\begin{bmatrix}\dfrac{\partial u}{\partial X} & \dfrac{\partial u}{\partial Y} & \dfrac{\partial u}{\partial Z}\\ \dfrac{\partial v}{\partial X} & \dfrac{\partial v}{\partial Y} & \dfrac{\partial v}{\partial Z}\\ \dfrac{\partial w}{\partial X} & \dfrac{\partial w}{\partial Y} & \dfrac{\partial w}{\partial Z}\end{bmatrix}. \tag{2.35}$$

The mathematical condition for the existence and uniqueness of the $\boldsymbol{F}$ mapping is that the determinant of the matrix presented in (2.35) is nonzero. Such determinant is an important entity, being referred to as the Jacobian matrix determinant, and represented as J. In fact, J can be used for relating the infinitesimal volume in the initial and current configuration at a given point, and must be positive in order to exclude the self penetrability of the material. In mathematical terms, this condition is expressed as

$$J>0. \tag{2.36}$$

Another important property of tensor $\boldsymbol{F}$ is that it can be decomposed into two different components, as

$$\boldsymbol{F}=\boldsymbol{R}\boldsymbol{U}, \tag{2.37}$$

in which $\boldsymbol{R}$ is a tensor related to the rigid body rotation component of the deformation gradient, and $\boldsymbol{U}$ represents its stretch. Such fact indicates that $\boldsymbol{F}$ can be used as a mathematical entity to measure the deformation.

However, according to the polar decomposition presented in (2.37), a rigid body rotation would result in changes in the value of $\boldsymbol{F}$ even if no stretch is observed in the given point. Moreover, it is clear from relation (2.35) that $\boldsymbol{F}$ is nonsymmetric, in general. These are clearly unfavorable properties for a strain tensor.

On the other hand, an important property of the rotation tensor $\boldsymbol{R}$ is that it is orthogonal, which means that $\boldsymbol{R}^T\boldsymbol{R}=\boldsymbol{I}$, which can be explored in order to defined a second order tensor $\boldsymbol{C}$ such that

$$\boldsymbol{C}=\boldsymbol{F}^T\boldsymbol{F}=\boldsymbol{U}^T\boldsymbol{R}^T\boldsymbol{R}\boldsymbol{U}=\boldsymbol{U}^T\boldsymbol{U}. \tag{2.38}$$

The tensor $\boldsymbol{C}$ is referred to as the right Cauchy-Green tensor. From (2.38) it is possible to conclude that it encompasses only terms related to the stretch, such that it can be efficiently employed to evaluate the strain at a given point. Despite this fact, if no deformation is observed, it is reduced to the second order identity tensor, which, likewise for the deformation gradient, results in a nonzero value for no strain.

An effective tensor for the finite deformation is the Green strain tensor, which can be computed by means of the relation

$$\boldsymbol{E} = \tfrac{1}{2}\left(\boldsymbol{C} - \boldsymbol{I}\right) = \tfrac{1}{2}\left(\boldsymbol{F}^T\boldsymbol{F} - \boldsymbol{I}\right). \tag{2.39}$$

In terms of the displacement field, the Green strain tensor is computed by means of the relation

$$\boldsymbol{E} = \tfrac{1}{2}\left[\left(\nabla\boldsymbol{u} + \boldsymbol{I}\right)^T\left(\nabla\boldsymbol{u} + \boldsymbol{I}\right) - \boldsymbol{I}\right] = \tfrac{1}{2}\left[\nabla\boldsymbol{u}^T + \nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T\nabla\boldsymbol{u}\right]. \tag{2.40}$$

Equivalent tensors can be also derived for an Eulerian description framework, in which the points are referenced by its position at the current configuration. For this description framework, one attains the so-called left Cauchy strain tensor and the Almansi strain tensor. The rate of the variation of such entities can also be computed for both Lagrangian and Eulerian frameworks by performing time derivatives of such tensors. These derivations are found in several Continuum Mechanics text books, as Spencer (2004) and Bonet and Wood (2008), being fundamental in order to provide mathematical tools for the definition of the equilibrium for the finite displacement problem.

Likewise for the small displacement model, for the finite motion problem, the equilibrium is defined in terms of the force per unit area at the current configuration, resulting in the Cauchy stress tensor (see equation (2.12)). Obviously, differently from the infinitesimal motion problem, the current configuration cannot be considered essentially similar to the initial one. In this context, the objectivity of the Cauchy stress tensor turns fundamental, a fact that is demonstrated in Bonet and Wood (2008). In spite of this, the same reference also shows that its rate of change within a Lagrangian description does not result in an objective tensor.

At this point, the difference between the complexity found for the infinitesimal and the finite deformation problem formulations turns remarkable. The referred complexity relies not only in the derivations, but also in other important aspects like the objectivity of the resulting tensors, for instance.

Once a detailed description of the several steps for deriving such formulation is out of the objectives of the present text, in what follows, a brief comment on the principal issues regarding such theme are presented, focusing the important relations to be used in the numerical implementation aspects.

One also finds in Bonet and Wood (2008) a complete description of the equilibrium condition for the finite motion formulation, resulting, in particular, in the definition of the Piola-Kirchhoff stress tensors.

The first Piola-Kirchhoff stress tensor $\boldsymbol{P}$ is an unsymmetrical tensor which physically allows one to describe the local equilibrium with the force stated in the current configuration, and its area of application defined in the initial configuration. Therefore, $\boldsymbol{P}$ provides a Lagrangian description of the equilibrium.

The first Piola-Kirchhoff tensor relates to the Cauchy stress tensor by

$$\boldsymbol{P} = J\boldsymbol{\sigma}\boldsymbol{F}^{-T}. \tag{2.41}$$

The second Piola-Kirchhoff stress tensor $\boldsymbol{S}$ can be derived by performing a complete *pull back* operation in the Cauchy stress tensor such that it results in

$$\boldsymbol{S} = J\boldsymbol{F}^{-1}\boldsymbol{\sigma}\boldsymbol{F}^{-T}. \tag{2.42}$$

The second Piola-Kirchhoff stress tensor cannot be related to a physical meaning like the Cauchy and the first Piola-Kirchhoff stress tensors. However, according to Bonet and Wood (2008), '… the second Piola-Kirchhoff tensor is intrinsically independent of a possible rigid body motion'.

On the other hand it can be shown that the first Piola-Kirchhoff stress tensor $\boldsymbol{P}$ is work conjugate to the rate of the deformation gradient $\dot{\boldsymbol{F}}$, while the second Piola-Kirchhff stress tensor $\boldsymbol{S}$ is work conjugate with the rate of the Green strain tensor $\dot{\boldsymbol{E}}$. In the present work, $\boldsymbol{E}$ and $\boldsymbol{S}$ are the conjugate pair of tensors adopted for modeling the finite displacement problem.

It is important to reinforce the fact that due to the nonlinearity of the kinematics involved in the formulation, essentially different from the infinitesimal strain model, the constitutive relation between the stress and strain tensors must be stated in terms of their rates.

The formulation hereby adopted considers only materials for which the mechanical process can be described by considering the initial and current configuration, regardless of the intermediary configuration, given rise to the term 'path independent' constitutive relation. According to Bonet and Wood (2008), materials for which this behavior is observed can be termed as hyperelastic materials.

The definition of a general hyperlastic material model relies on the definition of a function Ψ representing the stored internal energy. In the total Lagrangian approach hereby adopted, the strain energy function is defined in terms of the Green strain tensor, therefore expressed as $\Psi(\boldsymbol{E})$, which must obey several mathematical requirements in order to result in a consistent model.

According to this model, the conjugate stress tensor can be derived from the internal specific energy as

$$\boldsymbol{S} = \frac{\partial \Psi}{\partial \boldsymbol{E}}. \tag{2.43}$$

The resulting fourth order constitutive tensor is then derived as

$$\mathcal{C} = \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{E}}. \tag{2.44}$$

2.2.1. The Saint Venant-Kirchhoff material constitutive model

The Saint Venant-Kirchhoff material model derives from the strain energy function defined as

$$\Psi(\boldsymbol{E}) := \frac{1}{2}\Lambda\left[tr(\boldsymbol{E})\right]^2 + \mu\boldsymbol{E} : \boldsymbol{E}, \tag{2.45}$$

in which $tr(\bullet)$ is the trace operator and Λ and μ are Lamé material constants to the isotropic elastic material, given by the relations

$$\begin{aligned} \Lambda &= \frac{E}{(1+v)(1-2v)}, \\ \mu &= \frac{E}{2(1+v)}. \end{aligned} \tag{2.46}$$

By applying (2.43) to (2.45), one achieves the second Piola-Kirchhoff stress tensor given as

$$\boldsymbol{S} = \Lambda tr(\boldsymbol{E})\boldsymbol{I} + 2\mu\boldsymbol{E}. \tag{2.47}$$

Finally applying (2.44) in (2.47), the constitutive model for the Saint Venant-Kirchhoff material results as

$$\mathcal{C} = \Lambda\boldsymbol{I} \otimes \boldsymbol{I} + 2\mu\mathbb{I}. \tag{2.48}$$

In relation (2.48) $\mathbb{I}$ is the forth order identity tensor. It is easy to verify that the resulting constitutive tensor for the Saint Venant-Kirchhoff material model is equivalent to the constitutive tensor of the kinematically linear elastic model. Moreover, the derivation of the particular relations for the plane stress and plane strain states is similar to the ones made for that model.

Although no restrictions on the magnitude of the displacement gradients are required in the definition of the Saint Venant-Kirchhoff material model, in practice, if large strains are observed, the nonlinear analysis procedure may fail, as the condition stated in (2.36) does not hold anymore.

2.2.2. The Neo-Hookean material constitutive model

The Neo-Hookean material model derives from the strain energy function defined as

$$\Psi := \frac{1}{2}\mu\left[tr\left(\boldsymbol{C}\right)-3\right]+\frac{\Lambda}{4}\left(J^{2}-1\right)-\left(\frac{\Lambda}{2}+\mu\right)\ln\left(J\right). \tag{2.49}$$

It is easy to verify that the strain energy value determined from the function presented in (2.49) tends to infinity if the Jacobian matrix determinant tends to zero or to infinity, physically meaning that an infinity amount of energy is necessary to reduce the volume of a given point to zero or to infinity.

By applying condition (2.43) to (2.49), one achieves the relation defining the second Piola-Kirchhoff stress tensor consistent with the Neo-Hookean material behavior as

$$\boldsymbol{S} = \frac{\Lambda}{2}\left(J^{2}-1\right)\boldsymbol{C}^{-1}+\mu\left(\boldsymbol{I}-\boldsymbol{C}^{-1}\right) \tag{2.50}$$

Finally applying (2.44) to (2.50), the constitutive model for the Neo-Hookean material results in

$$\mathbb{C} = \Lambda J^{2}\boldsymbol{C}^{-1}\otimes\boldsymbol{C}^{-1}+\left[2\mu-\Lambda\left(J^{2}-1\right)\right]\mathrm{I}_{\boldsymbol{C}^{-1}}. \tag{2.51}$$

In (2.51) $\mathrm{I}_{\boldsymbol{C}^{-1}}$ is a forth order identity tensor, which is presented in Wriggers (2006) as

$$\mathrm{I}_{\boldsymbol{C}^{-1}ABCD} = \frac{1}{2}\left(\boldsymbol{C}^{-1}_{AC}\boldsymbol{C}^{-1}_{BD}+\boldsymbol{C}^{-1}_{AD}\boldsymbol{C}^{-1}_{BC}\right). \tag{2.52}$$

It is worth to observe that for null deformation, the constitutive tensor $\mathbb{C}$ from relation (2.51) coincides exactly to the tensor predicted by relation (2.48).

2.3. The weak form of the IBVP

Aiming to develop a numerical approach for the Initial Boundary Value Problem (IBVP), the derivation of its weak form is necessary. In order to do so, it is worth to mathematically define the involved vector spaces. The definitions hereby presented are based on Laursen (2002).

The solution of the IBVP is given by the vector field $\boldsymbol{u}_t$ representing the displacement at any time $t \in \mathrm{I}$, and belonging to a space $\mathscr{U}_t$ defined as

$$\mathscr{U}_t = \left\{ \boldsymbol{u}_t : \bar{\Omega} \to \mathbb{R}^2 \middle| \boldsymbol{u}_t \in \boldsymbol{H}^1(\Omega), \boldsymbol{u}_t = \bar{\boldsymbol{u}}(\boldsymbol{X}, t) \,\text{for}\, \boldsymbol{X} \in \Gamma_u \right\}. \tag{2.53}$$

The space of weighting functions $\mathscr{V}$ is defined as

$$\mathscr{V} = \left\{ \boldsymbol{w} : \bar{\Omega} \to \mathbb{R}^2 \middle| \boldsymbol{w} \in \boldsymbol{H}^1(\Omega), \boldsymbol{w} = 0 \,\text{for}\, \boldsymbol{X} \in \Gamma_u \right\}. \tag{2.54}$$

In both (2.53) and (2.54), $\boldsymbol{H}^1(\Omega)$ is the Sobolev space, representing the space containing all vectorial functions defined in the solid domain Ω, such that its values and first derivatives are square integrable.

In the present study, the Principle of Virtual Work (PVW) is adopted as the weak form. Then, the Galerkin approach is assumed to define the weighting functions $\boldsymbol{w}$, representing a field of virtual displacements $\boldsymbol{\delta u}$. The virtual displacement field $\boldsymbol{\delta u}$ is fully in accordance with the definition stated in (2.54). The choice for the PVW as a weak form is due to the fact that such methodology allows one to naturally consider any constitutive relation, even nonlinear, as the elastoplastic model previously presented.

Taking into account the Dirichlet boundary conditions of the virtual displacement fields and the enforcement of the Neumann boundary conditions, according to the PVW, the general weak form of the IBVP can be expressed as

$$\int_\Omega \left[\boldsymbol{\sigma} : \boldsymbol{\delta\varepsilon} + \rho \ddot{\boldsymbol{u}} \, \boldsymbol{\delta u} \right] d\Omega = \int_\Omega \boldsymbol{f} \boldsymbol{\delta u} \, d\Omega + \int_{\Gamma_\sigma} \bar{\boldsymbol{t}} \, \boldsymbol{\delta u} \, d\Gamma. \tag{2.55}$$

An alternative presentation of equation (2.55) is given by

$$\begin{aligned}
&\delta W^{\text{internal}} + \delta W^{\text{external}} + \delta W^{\text{dynamical}} = 0, \\
&\delta W^{\text{internal}} = \int_\Omega \boldsymbol{\sigma} : \boldsymbol{\delta\varepsilon} \;\; d\Omega, \\
&\delta W^{\text{external}} = -\int_\Omega \boldsymbol{f} \boldsymbol{\delta u} \, d\Omega - \int_{\Gamma_\sigma} \bar{\boldsymbol{t}} \, \boldsymbol{\delta u} \, d\Gamma, \\
&\delta W^{\text{dynamical}} = \int_\Omega \rho \ddot{\boldsymbol{u}} \, \boldsymbol{\delta u} \, d\Omega.
\end{aligned} \tag{2.56}$$

Equation (2.56) obviously states exactly the same variational principle presented in equation (2.55), except for the fact that it presents the internal, external and dynamical work terms separately. Even though it might seem useless for the moment, such notation proves to be useful in the multiple bodies contact problem, in which a virtual work counterpart due to contact arises in the resulting variational equation.

For the case of the nonlinear kinematics, the internal virtual work, defined in terms of the Green strain tensor and its conjugate second Piola-Kirchhoff stress tensor (Lagrangian description framework), results in

$$\delta W^{\text{internal}} = \int_{\Omega} \boldsymbol{S} : \delta\boldsymbol{E} \, d\Omega \,. \tag{2.57}$$

2.4. Contact problem definition

In all the previously discussed models, conditions (2.9) and (2.10) are used to define the boundary conditions by explicitly enforcing displacement and traction values over the whole boundary, for all $t \in \mathbf{I}$. Therefore, this approach assumes the exact knowledge of the boundary conditions for the whole mechanical process. However, in the contact problems, the conditions in some parts of the boundaries are not known beforehand and depend on the structural behavior of the solids encompassed in the mechanical problem. In this case, regardless of the adopted kinematic and constitutive model, the IBVP automatically becomes nonlinear.

The contact physical phenomenon is characterized by the interaction between different parts of the solids, which, during the mechanical process, may tend to occupy the same position in the space at the same time. Since this is not physically admissible, interaction forces between the solid's parts arise and the solids must deform so that no penetration is observed.

In general, the interaction is considered between different solids, however this is not a necessary condition, since the contact can also occurs between different parts of the same solid, giving rise to the so-called auto-contact. In spite of this fact, the framework for defining the contact problem can consider the contact between two different solids without loss of generality. The general scheme is depicted in Figure 2.2.

Similarly to the previously defined single solid problem, parts of the solid's boundary are associated to Dirichlet and Neumann boundary conditions, which, as already discussed, are known beforehand. The only difference from the previous case is that such portions are defined for each solid separately.

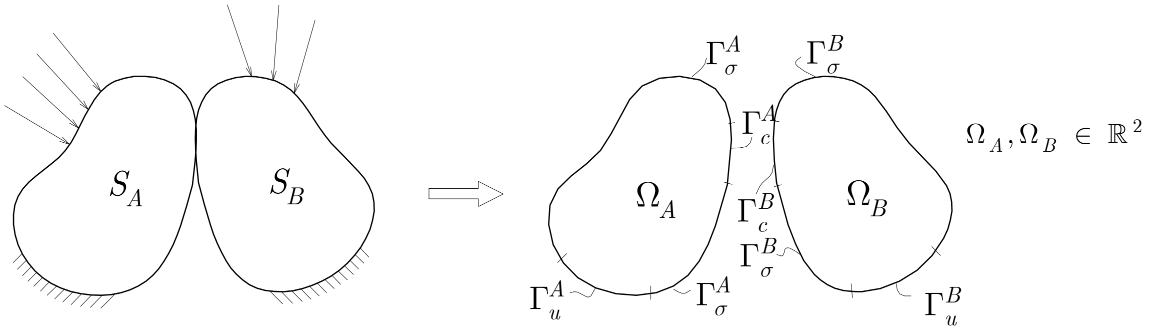


Figure 2.2. General contact scheme between solids idealized in a two dimensional Euclidian space.

On the other hand, a part of the solid's boundary can contact another one during the mechanical process, in a region here referred to as Γ_c. Conceptually, this portion contains the potential contacting points of the boundary. When such boundary portion is considered, the conditions stated in (2.8) must be redefined. For a generic set of two solids, as the one presented in Figure 2.2, the boundary conditions are defined as

$$\begin{aligned}
&\Gamma_u^A \bigcup \Gamma_\sigma^A \bigcup \Gamma_c^A = \partial\Omega^A, \\
&\Gamma_u^B \bigcup \Gamma_\sigma^B \bigcup \Gamma_c^B = \partial\Omega^B, \\
&\Gamma_u^A \bigcap \Gamma_\sigma^A = \Gamma_\sigma^A \bigcap \Gamma_c^A = \Gamma_c^A \bigcap \Gamma_u^A = \varnothing, \\
&\Gamma_u^B \bigcap \Gamma_\sigma^B = \Gamma_\sigma^B \bigcap \Gamma_c^B = \Gamma_c^B \bigcap \Gamma_u^B = \varnothing.
\end{aligned} \tag{2.58}$$

Obviously, the generalization to include any number of solids can be naturally provided by the extension of the above conditions.

Differently from the Dirichlet and Neumann boundary conditions, which are explicitly defined for each time instant, the contact conditions are defined by a set of rules to be observed along the mechanical process. In what follows, these rules are mathematically defined considering the expected model's physical behavior.

As already discussed, the fundamental aspect regarding contact arises from the obvious fact that two different solid portions cannot occupy the same position in space at the same time. So, aiming the mathematical definition of the contact problem, an impenetrability condition must be defined.

In order to do so, a pair of contacting points must be identified. Let $\boldsymbol{X}^A$ be a point belonging to Γ_c^A. One considers that its contact pair is the point $\bar{\boldsymbol{X}}^B$ such that

$$\bar{\boldsymbol{X}}^B(\boldsymbol{X}^A,t) := \left\{ \boldsymbol{X}^B \in \Gamma_c^B \middle| \min \left\| \boldsymbol{x}(\boldsymbol{X}^A,t) - \boldsymbol{x}(\boldsymbol{X}^B,t) \right\|, t \in \mathrm{I} \right\}. \tag{2.59}$$

The contact pair is represented in the scheme depicted in Figure 2.3.

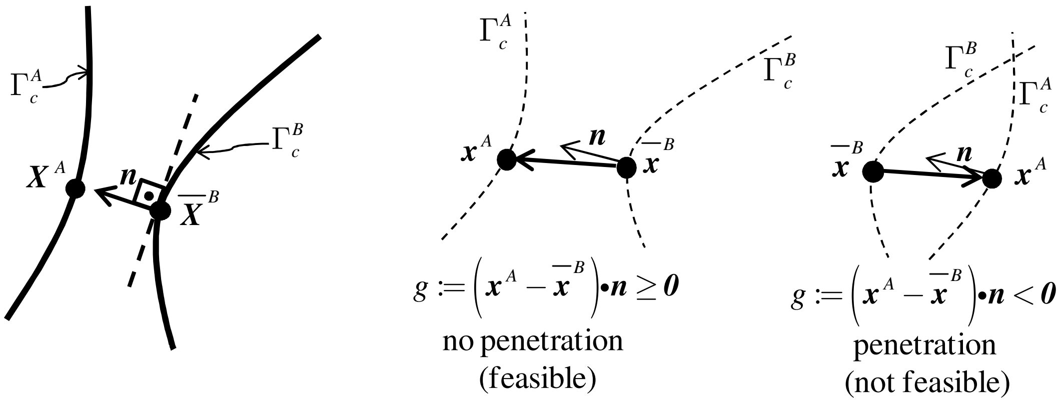


Figure 2.3. Contact pair scheme for defining the gap function.

The definition stated in (2.59) is based on the uniqueness of a minimum value for the distance between the points belonging to the contacting surfaces. This assumption cannot be assured for all situations, since it depends on the solids topology, as discussed in Laursen (2002). However, this issue can be easily circumvented in the numerical approach by defining a strategy for dealing with situations in which such uniqueness does not hold. A detailed discussion on the approach hereby adopted is presented in Wriggers (2006).

Once the contact pair has been identified, one defines the gap function $g(\boldsymbol{X}^A)$ as

$$g := \left(\boldsymbol{x}^A - \overline{\boldsymbol{x}}^B\right) \bullet \boldsymbol{n} . \tag{2.60}$$

The gap function results in a signed scalar value. Negative values indicate that one solid penetrates over the other, which, of course, is not feasible, as depicted in Figure 2.3.

It is worth noticing that the case of contact between a deformable solid and a rigid obstacle may be viewed simply as a particular case for which no displacement is observed in the second solid, since it is the rigid obstacle.

The impenetrability condition can be defined by means of the gap function as

$$g \geq 0 . \tag{2.61}$$

Considering the feasible values of the impenetrability condition, two distinct situations can be identified. When the gap function presents a positive value, no contact is observed between the pair of points. Accordingly, the interaction forces of both points are zero.

On the other hand, when the gap function is zero, the pair of points is in contact. Interaction forces arise in order to prevent interpenetration. However, the traction values depend on several characteristics of the whole structural set, such that they can only be achieved during the nonlinear solution process.

Regarding the interaction forces, obviously they are submitted to the behavior predicted by Newton’s third law. Hence, by making use of the Cauchy traction definition, one obtains that

$$\boldsymbol{t}^{A} = \boldsymbol{\sigma}(\boldsymbol{X}^{A})\boldsymbol{n}(\boldsymbol{X}^{A}) = -\boldsymbol{t}^{B} = \boldsymbol{\sigma}(\boldsymbol{X}^{B})\boldsymbol{n}(\boldsymbol{X}^{B}), \tag{2.62}$$

in which $\boldsymbol{n}$ is the outward unit normal at the contact pair of the boundary.

At this point, an important assumption is made in order to simplify the contact model. It regards the friction between the contacting surfaces. Even though it might be a strong hypothesis, in the present work the frictional effects are neglected, such that the traction is normal to the contact interface. In this situation, a single scalar parameter t_N representing the contact pressure is defined as

$$t_N := \boldsymbol{t}^{A}\boldsymbol{n}^{A} = \boldsymbol{t}^{B}\boldsymbol{n}^{B}. \tag{2.63}$$

The pressure t_N defined in (2.63) presents positive values if tensile tractions occur, while negative values are associated to compressive tractions.

Excluding the occurrence of adhesion between the contacting surfaces, it is possible to define the following condition related to the traction observed between the contacting points:

$$t_N \leq 0. \tag{2.64}$$

Finally, taking into account the simultaneous values observed for the gap function and the contact pressure, a complementarity condition can be stated as

$$g t_N = 0. \tag{2.65}$$

As it happens for the elastoplastic problem, the conditions stated in (2.61), (2.64) and (2.65) are equivalent to the Karush-Kuhn-Tucker conditions, which are used for enforcing constraints in the nonlinear programming approach.

Due to this similarity, the techniques derived for the solution of constrained nonlinear optimization problems can be explored for enforcing the contact conditions in the IBVP, and also for solving them efficiently. Regarding the constrained nonlinear programming approach, a detailed discussion of the subject is presented by Laursen (2002). In the same reference, one also finds a discussion on the contact problem treated as variational inequalities, reinforcing the consequences of the non-smooth nature aspect of such class of problems in the convergence.

In the hereby presented work we skip from this nonlinear programming approach discussion, moving directly to the definition of the contact IBVP weak form, based on the Principle of Virtual Work. In particular, the contact constraints are enforced by means of two commonly used nonlinear programming techniques: the Lagrange Multipliers and the penalty method.

2.5. Weak form of the contact problem

For defining the weak form of the contact IBVP, the schematic two bodies problem depicted in Figure 2.2 is considered. The generalization for any number of solids in the structural set follows forward from this schematic situation. For any of these cases, a virtual work counterpart associated to the contact forces arise, such that the Principle of Virtual Work, for the multiple body contact problem, is restated in a general sense as

$$\delta W^{\text{internal}} + \delta W^{\text{dynamical}} + \delta W^{\text{external}} + \delta W^{\text{contact}} = 0\,. \tag{2.66}$$

The main objective of the current section is to present such contact virtual work term.

Regarding the vector spaces defined in (2.53) and (2.54), it is clear that similar spaces must be defined for both solids. Moreover, such spaces can be used to define single spaces for the solutions and weighting functions for the whole structural set, encompassing all solids. By employing such approach, the internal, external and dynamical virtual work counterparts, for the schematic two body problem, are stated, respectively, as

$$\begin{aligned}
\delta W^{\text{internal}} &= \int_{\Omega^A} \boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon}\, d\Omega + \int_{\Omega^B} \boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon}\ d\Omega, \\
\delta W^{\text{dynamical}} &= \int_{\Omega^A} \rho\ddot{\boldsymbol{u}}\,\delta\boldsymbol{u}\, d\Omega + \int_{\Omega^B} \rho\ddot{\boldsymbol{u}}\,\delta\boldsymbol{u}\, d\Omega, \\
\delta W^{\text{external}} &= -\int_{\Omega^A} \boldsymbol{f}\delta\boldsymbol{u}\, d\,\Omega - \int_{\Gamma_\sigma^A} \bar{\boldsymbol{t}}\,\delta\boldsymbol{u}\, d\,\Gamma - \int_{\Omega^B} \boldsymbol{f}\delta\boldsymbol{u}\, d\,\Omega - \int_{\Gamma_\sigma^B} \bar{\boldsymbol{t}}\,\delta\boldsymbol{u}\, d\,\Gamma.
\end{aligned} \tag{2.67}$$

Obviously, the internal virtual work can be redefined in accordance with (2.57) for the finite displacement case. Also, as already mentioned, one observes that the statement for any number of solids is a straightforward adaptation, such that no loss of generality results from employing the specific case of the schematic two bodies contact problem.

The contact virtual work is associated to the virtual displacements in the Γ_c part of the solid boundary, and results nonzero when the contact pressures arise for avoiding the solids interpenetration. In spite of the fact that such work seems similar to the external virtual work counterpart associated to the Neumann boundary conditions, it is worth noticing that it is considerably more difficult to be stated, since the contact forces are not known *a priori*, and their characteristics depend on the whole structural set behavior.

Taking this fact into account, the contact tractions must be defined as unknowns, given rise to a mixed field contact IBVP, for which both displacement and contact pressure constitute fields to be found by solving the contact problem. By employing this strategy, a Lagrange multipliers constraint enforcement arises, such that the Lagrange multipliers field

λ_N is associated to the contact pressure, and is used to enforce the gap condition stated in (2.61). Within this scheme, the contact virtual work results as

$$\delta W^{\text{contact}} = \int_{\Gamma_c} \left[\lambda_N \delta g + \delta\lambda_N g\right] d\Gamma \,. \tag{2.68}$$

One observes that the gap function in (2.60) comes from the displacement fields of both contacting surfaces, for which the contact pressure is equal, due to the reasons already discussed in the previous section. Due to this fact, the weak enforcement of the contact conditions is performed by integrating (2.68) in one of the contact surfaces. In the numerical approach, this surface is associated to a mesh of contact elements, to be derived in Chapter 3.

The mixed displacement and contact pressure fields approach present some advantages. One of them is related to the fact that the Lagrange multipliers field allow one to directly achieve the value of the contact pressure along Γ_c. It is important to remember that the definition of the contact problem is conditioned to the value of the contact pressure, according to (2.64), such that the knowledge of its value is fundamental in the contact problem solution context.

In spite of this fact, such mixed approach also presents disadvantages. The first of them is the obvious fact that the number of unknowns is greater than the ones in the displacement based equations. Also, its computational implementation is considerably more difficult to conceive.

However, probably the most remarkable disadvantage of the Lagrange multipliers based formulation is related to the stability of the resulting numerical scheme. For the mixed field equation, the constrained weak form stability is conditioned to the fulfillment of the Babuška-Brezzi condition, as discussed in Wriggers (2006).

In fact, other techniques coming from the constrained nonlinear programming can be used for developing an alternative to the mixed field formulation derived using Lagrange multipliers. For instance, a penalty based scheme can also be used to represent the contact virtual work.

According to this approach, the contact virtual works results in

$$\delta W^{\text{contact}} = \int_{\Gamma_c} \varepsilon_N g^* \delta g^* \, d\Gamma, \; g^* := \begin{cases} g, \text{ if } g \leq 0 \\ 0, \text{ if } g > 0 \end{cases}. \tag{2.69}$$

In (2.69) ε_N is an adopted scalar value, referred to as penalty parameter. The physical meaning of such term is attached to the stiffness of an idealized spring that enforces the gap value, according to the impenetrability condition stated in (2.61).

It is important to notice that for the penalty based approach the enforcement is approximated, i.e., the accuracy of the enforcement depends on the idealized spring stiffness. In this context, small negative gap values can be observed in such approximated numerical approach. Therefore, the definition of the modified gap g^* does not constitute a violation of the theory (condition of impenetrability). In fact, this definition is made aiming to take into account such numerical issue.

Advancing in this idealization, the term $\varepsilon_N g^*$ can be understood as a 'simulation' of the contact force. In this case, the modified gap function g^*, defined in (2.69), completely fulfill the other frictionless contact problem conditions stated in (2.64) and (2.65).

Even though such approach does not result in a mixed field problem, and that no additional degrees of freedom are added to the problem, the penalty formulation also present its disadvantages. Perhaps the most obvious is that the contact pressure values are not available directly, as it happens for the Lagrange multipliers formulation. As already discussed, such pressure value may constitute an important information in the contact problem solution scheme.

Probably one of the most unfavorable issues regarding the penalty formulation is the necessity of defining an arbitrary value for the penalty parameter ε_N. Moreover, in theory, an exact enforcement is achieved only when the penalty parameter tends to infinity. However, adopting exceedingly high values for ε_N can yield to ill conditioning of the resulting system of equation. On the other hand, if it is not sufficiently high, the enforcement is not accurately observed.

Regardless of the employed method, it is known from the nonlinear programming that every technique for enforcing constrains presents advantages and disadvantages. The same is true for the contact problems. In fact, depending on the problem's characteristics, one formulation may be more efficient and robust than the other one. Taking this fact into account, the availability of more than one technique in the same framework proves to be the best strategy for efficiently dealing with such class of problems.

Such as the internal virtual work, the contact virtual work can be nonlinear depending on the hypotheses adopted for the model. The numerical solution of the resulting nonlinear problem in the present work relies on the linearization of the nonlinear terms by means of the Newton-Raphson’s procedure. Such theme is discussed in the next chapter.

3 - The numerical approach

The present chapter is devoted to detailing the numerical treatment of both linear and nonlinear problems of Solid Mechanics in accordance with the theory presented in Chapter 2.

In section 3.1, the Finite Element Method essentials are briefly introduced. Even though the method is not addressed directly in the present work, its issues constitute the basis for defining the Generalized Finite Element Method. In Section 3.2, the GFEM formulation is presented, highlighting the enrichment feature as its main characteristic. In the same section, the enrichment functions used in the present work are also presented.

In section 3.3, the use of both methods for solving dynamical problems is described. The time integration scheme adopted is based on the Newmark Method. Even though only linear behavior is considered at this point, one notices that the numerical framework can be extended with no major changes to nonlinear problems.

The general nonlinear problem solution based in the incremental iterative Newton-Raphson's method is then introduced. Following a general discussion, the application of the method to specific nonlinear problems, such as the small strain elastoplastic and the finite strain elastic problem, is then presented in the subsequent sections.

Next, the constrained nonlinear problem arising from the contact phenomena is presented. The solution of such problems by the FEM/GFEM approaches in described, with emphasis on the formulation of 'generalized' contact elements, formulated to correctly enforce the contact constraints in the GFEM model. The generalized segment-to-segment contact element, based on the mortar method, probably constitutes one of the main contributions of the present work.

At last, a general nonlinear solution algorithm, putting together all the previously discussed nonlinear issues, is presented. Such algorithmic structure constitutes the main core of the implemented nonlinear analysis framework.

3.1. The Finite Element Method (FEM)

The traditional approach for solving the IBVP is based on finding analytical expressions representing the field functions that constitute the solution of its strong form. The displacement based Finite Element Method is a numerical technique used to find approximated solutions for problems formulated in a weak form. Basically, the method is conceived to provide a strategy for finding the value of the unknown fields in specific points of the continuum, called nodes.

The process of representing the continuum by means of a finite number of nodes and elements is referred to as discretization. This strategy is schematically represented in Figure 3.1.

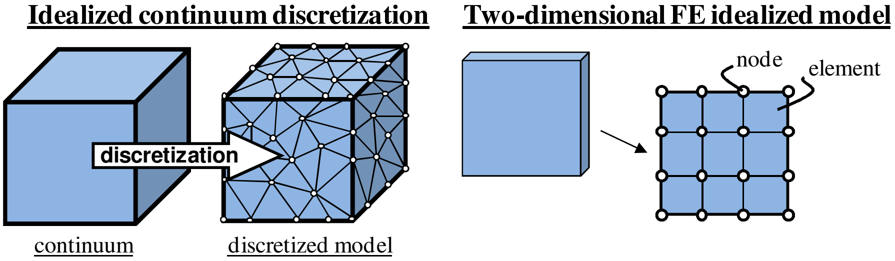

Idealized continuum discretization
Two-dimensional FE idealized model
discretization
continuum
discretized model
node
element

Figure 3.1. The continuum discretization in the FEM scheme (2D and 3D models).

Variables called degrees of freedom (DOF) are attached to the nodes, constituting the unknowns of the FEM model. A commonly employed formulation is based on displacement fields.

The regions of the continuum among nodes define finite elements, to which one attaches interpolation functions, also referred to as shape functions. These functions are used to interpolate the field value in the element's domain, by performing a linear combination of the shape functions and the respective nodal DOF.

In the present work, the element's initial configuration geometry, which is in general distorted, is described using the same strategy. In this case, the interpolation functions are defined using an auxiliary domain called natural coordinate reference domain. Figure 3.2 illustrates a one-dimensional, a triangular and a quadrilateral domain in the natural coordinate system.

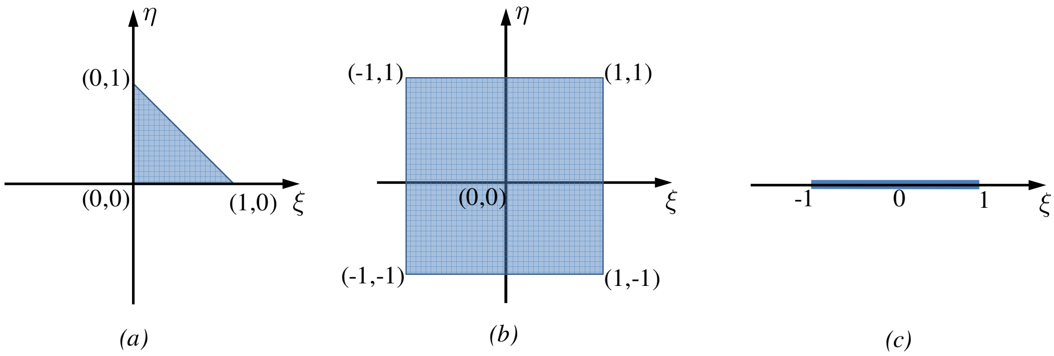


Figure 3.2. Triagular (a) and quadrilateral (b) domains in the natural coordinates system; one-dimensional (c) domain in the coordinate system.

For the two-dimensional elements, the geometry and displacement interpolations are given by the expressions

$$\begin{aligned} X(\xi,\eta) &= \sum_{i=1}^{n_{nd}} X_i N_i(\xi,\eta), Y(\xi,\eta) = \sum_{i=1}^{n_{nd}} Y_i N_i(\xi,\eta), \\ u(\xi,\eta) &= \sum_{i=1}^{n_{nd}} u_i N_i(\xi,\eta), v(\xi,\eta) = \sum_{i=1}^{n_{nd}} v_i N_i(\xi,\eta). \end{aligned} \tag{3.1}$$

In (3.1) n_{nd} is the number of nodes in the element, X_i and Y_i are the coordinates of the nodes defining the element in the initial configuration, and u_i and v_i are the nodal displacement components in the current configuration. Equivalent relations are stated for one-dimensional shape function, which can be applied, for instance, over the two-dimensional element's boundary lines.

As already discussed in Chapter 2, the weak form of the IBVP is based on the integration of field functions over the whole solid domain. In the FEM, this computation is performed in a piecewise approach, i.e., one integrates the functions restricted to the finite elements domain. The sum of all these piecewise integrations takes the value of the integration over the whole solid domain.

Lagrangian and Serendipity shape functions were implemented in the computational code developed in the present work. The nomenclature, shape and number of nodes of each of these elements are presented in Figure 3.3.

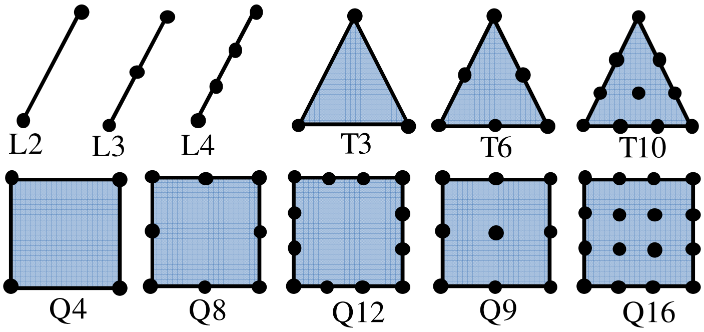


Figure 3.3. Nomenclature, shape and number of nodes of the elements implemented in the present work.

An important property of the set of shape functions N_i is that for any pair $(\overline{\xi},\overline{\eta})$ in the reference domain, the following condition must hold:

$$\sum_{i=1}^{n_{nd}} N_i\left(\overline{\xi},\overline{\eta}\right) = 1\,. \tag{3.2}$$

Due to this fact, this set of interpolation functions is commonly referred to as Partition of Unity (PU). In spite of this, one must have in mind that a general PU definition requires additional conditions to be verified. For instance, another condition in the PU definition is that

$$0 \leq N_i(\xi,\eta) \leq 1. \tag{3.3}$$

Consistent with this definition, only the linear Lagrangian shape functions defined over the triangular domain, and the bi-linear shape function for the case of quadrilaterals, can be considered as partitions of unity, in a strict sense. Even though it is an abuse of nomenclature, in the present work, all the previously presented shape functions are referred to as PU in the computational implementation.

Once the displacement field interpolation is defined, it can be used to compute an approximation for the strain field. According to the strain tensor definition stated in (2.6), one finds that

$$\varepsilon_x = \frac{\partial u}{\partial X} = \frac{\partial u}{\partial \xi}\frac{\partial \xi}{\partial X} + \frac{\partial u}{\partial \eta}\frac{\partial \eta}{\partial X}, \tag{3.4}$$

$$\varepsilon_y = \frac{\partial v}{\partial Y} = \frac{\partial v}{\partial \xi}\frac{\partial \xi}{\partial Y} + \frac{\partial v}{\partial \eta}\frac{\partial \eta}{\partial Y}, \tag{3.5}$$

$$\gamma_{xy} = \frac{\partial u}{\partial Y} + \frac{\partial v}{\partial X} = \frac{\partial u}{\partial \xi}\frac{\partial \xi}{\partial Y} + \frac{\partial u}{\partial \eta}\frac{\partial \eta}{\partial Y} + \frac{\partial v}{\partial \xi}\frac{\partial \xi}{\partial X} + \frac{\partial v}{\partial \eta}\frac{\partial \eta}{\partial X}. \tag{3.6}$$

In the FEM, the derivatives presented in relations (3.4) to (3.6) are performed over the interpolated values, resulting

$$\begin{gathered}\frac{\partial X}{\partial \xi} = \sum_{i=1}^{n_{nd}} X_i \frac{\partial N_i}{\partial \xi}, \frac{\partial Y}{\partial \xi} = \sum_{i=1}^{n_{nd}} Y_i \frac{\partial N_i}{\partial \xi}, \frac{\partial X}{\partial \eta} = \sum_{i=1}^{n_{nd}} X_i \frac{\partial N_i}{\partial \eta}, \frac{\partial Y}{\partial \eta} = \sum_{i=1}^{n_{nd}} Y_i \frac{\partial N_i}{\partial \eta}, \\ \frac{\partial u}{\partial \xi} = \sum_{i=1}^{n_{nd}} u_i \frac{\partial N_i}{\partial \xi}, \frac{\partial v}{\partial \xi} = \sum_{i=1}^{n_{nd}} v_i \frac{\partial N_i}{\partial \xi}, \frac{\partial u}{\partial \eta} = \sum_{i=1}^{n_{nd}} u_i \frac{\partial N_i}{\partial \eta}, \frac{\partial v}{\partial \eta} = \sum_{i=1}^{n_{nd}} v_i \frac{\partial N_i}{\partial \eta}.\end{gathered} \tag{3.7}$$

The derivatives relating the global and the natural coordinates are involved in the infinitesimal length definitions

$$dX = \frac{\partial X}{\partial \xi} d\xi + \frac{\partial X}{\partial \eta} d\eta, dY = \frac{\partial Y}{\partial \xi} d\xi + \frac{\partial Y}{\partial \eta} d\eta. \tag{3.8}$$

The relations stated in (3.8) can be arranged in matrix notation as

$$\begin{Bmatrix} dX \\ dY \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial X}{\partial \xi} & \dfrac{\partial X}{\partial \eta} \\ \dfrac{\partial Y}{\partial \xi} & \dfrac{\partial Y}{\partial \eta} \end{bmatrix} \begin{Bmatrix} d\xi \\ d\eta \end{Bmatrix} = \boldsymbol{J} \begin{Bmatrix} d\xi \\ d\eta \end{Bmatrix}. \tag{3.9}$$

In (3.9) $\boldsymbol{J}$ is the Jacobian matrix. The inverse relation is given by

$$\begin{Bmatrix} d\xi \\ d\eta \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial \xi}{\partial X} & \dfrac{\partial \xi}{\partial Y} \\ \dfrac{\partial \eta}{\partial X} & \dfrac{\partial \eta}{\partial Y} \end{bmatrix} \begin{Bmatrix} dX \\ dY \end{Bmatrix} = \boldsymbol{H} \begin{Bmatrix} dX \\ dY \end{Bmatrix}. \tag{3.10}$$

Assuming that $\boldsymbol{J}$ is invertible, one has that

$$\boldsymbol{H} = \boldsymbol{J}^{-1} = \frac{1}{J} \begin{bmatrix} \dfrac{\partial Y}{\partial \eta} & -\dfrac{\partial X}{\partial \eta} \\ -\dfrac{\partial Y}{\partial \xi} & \dfrac{\partial X}{\partial \xi} \end{bmatrix}, J = \det(\boldsymbol{J}). \tag{3.11}$$

The Jacobian matrix determinant allows one to perform the integration over a plane domain in the natural coordinates by means of the relation

$$\int_{\Omega} W \, d\Omega = \iint_{X\,Y} W\left(X,Y\right) dXdY = \iint_{\xi\,\eta} W\left(\xi,\eta\right) J \, d\xi d\eta \,, \tag{3.12}$$

in which W is a function defined in the two-dimensional domain.

By grouping the nodal displacements u_i and v_i of a given element in a vector d, one can state the relation between the nodal displacements and the strain components by means of a matrix of partial derivatives operation $\boldsymbol{B}$ defined such that

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \underbrace{\boldsymbol{B}}_{(3\text{x}2n_{nd})} \underbrace{\boldsymbol{d}}_{(2n_{nd}\text{x}1)} \,. \tag{3.13}$$

Following a Galerkin approach, the same relation is employed for the virtual displacements and strains

$$\boldsymbol{\delta\varepsilon} = \boldsymbol{B} \; \boldsymbol{\delta d}. \tag{3.14}$$

Assuming a linear elastic model, for instance, the relations stated in (3.13) and (3.14) can be used together with the constitutive relations stated in (2.16) or (2.17) to compute the internal virtual work presented in (2.56). The internal virtual work is then given as

$$\delta W^{\text{internal}} = \int_{\Omega} \boldsymbol{\sigma} : \boldsymbol{\delta\varepsilon} \, d\Omega = \int_{\Omega} \boldsymbol{\delta\varepsilon} : \mathbb{C} : \boldsymbol{\varepsilon} \, d\Omega \,. \tag{3.15}$$

Computing the above term considering a given plane element, the following relation results

$$\boldsymbol{\delta d} \iint_{X\,Y} t_{ck} \boldsymbol{B}^T \mathbb{C} \boldsymbol{B} \; dXdY \; \boldsymbol{d} = \underbrace{\boldsymbol{\delta d}}_{(1\text{x}2n_{nd})} \underbrace{\boldsymbol{K}_l}_{(2n_{nd}\text{x}2n_{nd})} \underbrace{\boldsymbol{d}}_{(2n_{nd}\text{x}1)} \,, \tag{3.16}$$

in which t_{ck} is the element thickness and $\boldsymbol{K}_l$ is the element stiffness matrix.

The double integral for computing $\boldsymbol{K}_l$ can be evaluated numerically. In general, one employs a Gaussian quadrature over the natural coordinate space. Accordingly, the stiffness matrix computation is performed as

$$\boldsymbol{K}_l = \sum_{i=1}^{n_{gp}} \boldsymbol{B}^T\left(\xi_i,\eta_i\right)\mathbb{C}\,\boldsymbol{B}\left(\xi_i,\eta_i\right)J\left(\xi_i,\eta_i\right)w_i\,. \tag{3.17}$$

In (3.17) the coordinates $\left(\xi_i,\eta_i\right)$ are given by the quadrature rule for n_{gp} integration points, and w_i are the weights also given by the same rule for these points. For the quadrilateral elements, the integration points and weights are easily obtained from the one-dimensional Gauss-Legendre quadrature rule. For triangular elements, one can use tables specially computed for this kind of domain, as for instance, by Cowper (1972).

Since each degree of freedom has a unique identification number within a global numbering system, it is possible to assemble all the element stiffness matrix in a global matrix, representing the stiffness of the whole structural set, hereby referred to as $\boldsymbol{K}_g$.

Similar computations are performed for the terms related to the external virtual work, resulting in a vector representing equivalent nodal forces $\boldsymbol{F}_g$. No further details on theses computation are shown here, since they are extensively discussed and presented in the technical literature, as Hughes (2000), Zienkiewicz and Taylor (2000), Assan (2003) and Savassi (2000), among many others.

Finally, from these indicated computations, one obtains the following linear system of equations

$$\underbrace{\boldsymbol{K}_g}_{(2n_{nd}\,\mathrm{x}\,2n_{nd})}\underbrace{\boldsymbol{d}_g}_{(2n_{nd}\,\mathrm{x}\;1)} = \underbrace{\boldsymbol{F}_g}_{(2n_{nd}\,\mathrm{x}\;1)}\;. \tag{3.18}$$

For static analyses, the system indicated in (3.18) demands the enforcement of the displacement boundary conditions. Otherwise, it results in a singular system of equation. The kinematic boundary conditions impositions can be performed using several techniques, like the Lagrange multiplier or the penalty method, for instance.

The resulting $\boldsymbol{d}_g$ represents the nodal displacement of the whole structure. Once this vector is known, one can compute the stress and strain fields using the relations previously presented.

3.2. The Generalized Finite Element Method (GFEM)

According to Duarte, Babuška and Oden (2000), the Generalized Finite Element Method (GFEM) has been independently developed by Babuška and others, being referred also as Special Finite Element Method, and Duarte and Oden, under the denomination hp-Clouds and Cloud-based hp Finite Element Method. According to Barros (2002), the first use of the denomination GFEM was made in Melenk (1995).

The most fundamental aspect regarding those methods is the enrichment of the partitions of unity, aiming to improve the solution. In the GFEM, the PU is given by FEM shape functions defined over a mesh of finite elements.

Specifically for the 2D models hereby presented, the shape functions adopted as PU are linear Lagragian functions defined over triangular domains and bi-linear Lagrangian shape functions defined over quadrilateral domain. These set of functions are in accordance with the conditions stated in (3.2) and (3.3).

The GFEM enrichment is performed by multiplying the PU functions by conveniently adopted functions, referred to as enrichment functions. In theory, any type of function can be adopted to enrich the PU. However its convenience is judged according to the expected solution.

The enrichment has a local character, being defined over selected nodes of the mesh and applied to all the PU functions associated to such nodes. The associated PU functions present nonzero values only at the elements surrounding the node, due to the well known compact support characteristic of the FEM. Therefore, the enrichment applied to a given node changes the interpolation only over the region defined by the elements sharing the node. This region is referred to in the GFEM as cloud. Figure 3.4 illustrates a cloud ω and the node representing its vertex, here referred to as α.

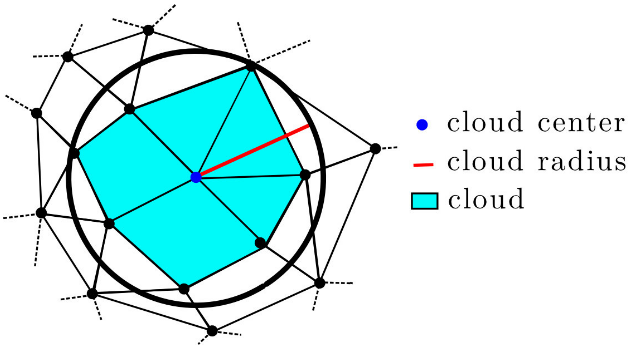


Figure 3.4. A cloud, its center, and its radius.

In Figure 3.4 one also finds the indication of an entity called cloud radius. This value is the radius of a circle circumscribing the cloud, and it is useful for defining enrichment functions, to be presented next.

Regarding the GFEM shape functions, initially all enrichment functions to be multiplied by the PU are arranged in an array $\boldsymbol{L} = \left\{L_1, L_2, ..., L_{n_l}\right\}$. Therefore, the nodal shape functions result as

$$N_{\alpha j}^{enr} = N_\alpha L_j . \tag{3.19}$$

The nodal approximation for a certain displacement component associated to the node α is then given by

$$u_\alpha = \sum_{j=1}^{n_l(\alpha)} N_{\alpha j}^{enr} u_{\alpha j}^{enr} . \tag{3.20}$$

It is important to notice that for the GFEM, the field component in a given node is the linear combination of generalized degrees of freedom (GDOF) $u_{\alpha j}^{enr}$. In general, the component L_1 is adopted as the unity, such that if it is the single component of the enrichment array, the regular FEM interpolation functions are recovered.

If one computes the global stiffness matrix $\boldsymbol{K}_g$ following the elemental assemblage commented in the previous section, the element stiffness matrix $\boldsymbol{K}_l$ presents a variable size from element to element, depending on the number of enrichment functions applied to its nodes. This fact is schematically represented in Figure 3.5.

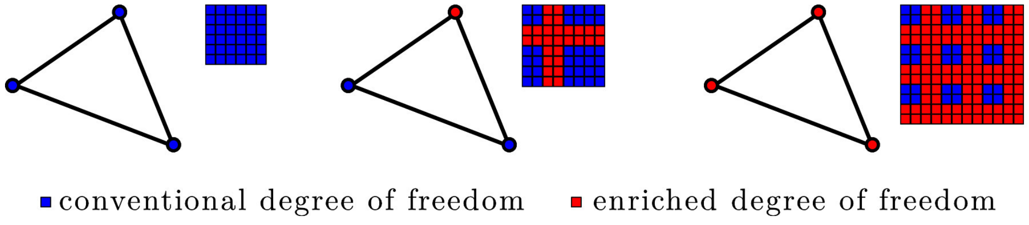

🟦 conventional degree of freedom
🟥 enriched degree of freedom

Figure 3.5. The GFEM local stiffness matrix size, depending on the number of enrichment functions applied to the element nodes.

In Figure 3.5, the leftmost scheme represents a triangular constant strain element with two DOF per node, resulting in a (6x6) size local stiffness matrix. Such non-enrichment condition is graphically represented by the set of blue nodes. The associated stiffness matrix terms are schematically represented by blue squares. If the same element has both DOF at one of its nodes enriched, additional lines and columns arise in the local stiffness matrix (red squares), resulting in a (8x8) matrix (middle scheme). If the whole set of original DOF are enriched by one additional enrichment function, $\boldsymbol{K}_l$ results (12x12).

In fact, any number of enrichment functions can be applied over each node, such that the element stiffness matrix can virtually present any size. Even though the enrichment represents a powerful feature of the method, dealing with the variable number of GDOF and different types of enrichment functions demands a considerably complex data structure to be considered in the computational code. An efficient approach to the GFEM coding is one of the main subjects discussed in Chapter 4.

Still with respect to the enrichment functions, it is worth commenting that a commonly employed enrichment scheme is performed using Heaviside functions. This approach allows one, for instance, to insert crack discontinuities in the solid's domain without demanding re-meshing. In fact, this discontinuous enrichment, coupled to enrichment functions able to enhance the stress gradient close to crack tips, makes the GFEM an efficient method to model crack mechanics problems. Despite the fact that these enrichment functions are of great interest for many simulations, they are not used in the present work, since they have already been extensively studied in other research works.

The major interest in the present research regards the polynomial enrichment. Its importance is clearly noticed if one considers the poor quality of the results generally obtained by means of the first degree interpolation used as PU in the GFEM.

The general form of the polynomial enrichment function is

$$L(m,n) = \frac{X^m Y^n}{h^{(m+n)}}, \tag{3.21}$$

in which h is the cloud radius (see Figure 3.4).

An alternative is the shifted enrichment function, defined as

$$L(m,n) = \frac{\left(X - X_\alpha\right)^m \left(Y - Y_\alpha\right)^n}{h^{(m+n)}}, \tag{3.22}$$

in which X_α and Y_α are the coordinates of the node α to which the enrichment is applied.

Generally, these enrichment functions are applied at each of the original displacement components of a given node. It is also common to employ an enrichment scheme aiming to achieve a given order of polynomial enrichment, which we refer to as first degree enrichment, second degree enrichment and so on. Figure 3.6 illustrates these sets of enrichments and the nomenclature used to refer to them in the present work.

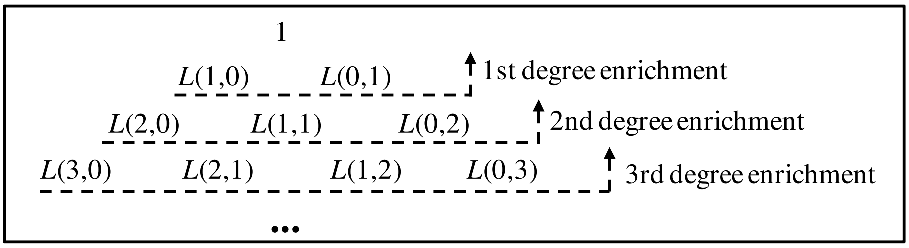


Figure 3.6. Levels of enrichment resulting from the polynomial functions enrichment.

One of the contributions of the present work is to assess the results achieved when polynomial and shifted enrichment functions are used for improving the linear interpolation PU ability for nonlinear problems.

One of the most remarkable advantages of the shifted enrichment functions is that they are zero in the node to which they are applied. Hence, the meaning of the original displacement DOF in such node is preserved, like in the FEM. This fact allows one to directly enforce the boundary displacement restriction in the same manner as used in the FEM.

Even though the nodal enforcement of displacement boundary conditions can be easily applied in these cases, this approach does not generate accurate results in most of the cases, where the element sides should also be restricted. The reason is that the enrichment changes the displacement field in the element sides between the nodes. In short, a nodal enforcement of the displacement boundary conditions does not guarantee its enforcement over the whole element's side. This numerical aspect is evaluated in one of the examples of Chapter 5.

Therefore, the correct enforcement of displacement boundary condition in the GFEM is a subject that introduces more complexity in the model. This enforcement is hereby performed either using Lagrange multipliers or the penalty method, integrating the condition over the element's side. Programming this segment wise enforcement is not a trivial task and demands a complex data structure.

It is important to comment that both polynomial and shifted enrichments can introduce linear dependencies in the resulting system of equation, which is an important additional difficulty to be circumvented. The usually employed solution method is based in the numerical strategy proposed by Babuška, which is detailed in Barros (2002). However, in the present work, this problem is automatically solved simply by using the sparse matrix routine from NumPy, to be introduced in the next Chapter, which naturally deals with this kind of numerical problem.

Finally, a last type of enrichment is also considered in the present work. In fact, this option is employed only in the linear dynamic example presented in Chapter 5, and it is referred to as double sine enrichment, being defined as

$$L(m,n) = \left[\sin\left(2\pi\frac{\left(X - X_0\right)}{l_x}\right)\right]^m \left[\sin\left(2\pi\frac{\left(Y - Y_0\right)}{l_y}\right)\right]^n . \tag{3.23}$$

In (3.23) X_0 and Y_0 represent the coordinates in which the sine functions associated to the horizontal and vertical directions start, while l_x and l_y allows one to adjust the frequency of variation of the sine function value in the horizontal and vertical directions, respectively.

3.3. The dynamical problem analysis

Recalling the Principle of Virtual Works stated in equation (2.56), one observes the existence of a dynamical term arising due to inertial forces. This term is proportional to the acceleration of the material points, such that it is common to neglect it if the observed values for this field result in forces much smaller than the other forces acting in the solid. The term quasi-static problem is used in this situation.

However, if the velocities and acceleration observed in the phenomenon induce forces that are not negligible, the respective dynamical term of the PVW must be considered. In this situation, the linear system of equation results in

$$\underbrace{\boldsymbol{M}_g}_{(2n_{nd}\,\mathrm{x}\,2n_{nd})} \underbrace{\ddot{\boldsymbol{d}}_g}_{(2n_{nd}\,\mathrm{x}\,1)} + \underbrace{\boldsymbol{K}_g}_{(2n_{nd}\,\mathrm{x}\,2n_{nd})} \underbrace{\boldsymbol{d}_g}_{(2n_{nd}\,\mathrm{x}\,1)} = \underbrace{\boldsymbol{F}_g}_{(2n_{nd}\,\mathrm{x}\,1)} . \tag{3.24}$$

In (3.24) $\ddot{\boldsymbol{d}}_g$ is the vector of the nodal accelerations, and $\boldsymbol{M}_g$ is the so-called mass matrix, which is assembled by contributions of element local mass matrices given by

$$\underbrace{\boldsymbol{M}_l}_{(n_{dof}\mathrm{x}n_{dof})} = \int_\Omega \rho \underbrace{\boldsymbol{N}}_{(n_{dof}\mathrm{x}1)} \underbrace{\boldsymbol{N}^T}_{(1\mathrm{x}n_{dof})} d\Omega, \tag{3.25}$$

in which $\boldsymbol{N}$ is a vector containing all the shape functions associated to the DOF of the element nodes. Note that these shape functions are the ones associated to the displacement fields, such that they encompass all the enriched interpolation functions N^{enr} for the GFEM.

This mass matrix is also known as the consistent mass matrix. A common procedure in dynamical analyses is to substitute the consistent mass matrix by another diagonal matrix called lumped mass matrix. Several strategies can be used to compute the lumped mass matrix. For instance, the lumped mass matrix terms $\bar{M}_{ij}$ can be computed by the following relation:

$$\bar{M}_{ij} = \begin{cases} 0 \text{ if } i \neq j \\ \dfrac{\sum_{i=1}^{n_{dof}} \sum_{j=1}^{n_{dof}} M_{ij}}{\sum_{i=1}^{n_{dof}} M_{ii}} M_{ij} \text{ if } i = j \end{cases}. \tag{3.26}$$

A numerical solution of (3.24) consists in finding the equilibrium configuration for discrete instants of time t_n, separated by time increments Δt such that $t_{n+1} = t_n + \Delta t$. In this context, equation (3.24) is rewritten as

$$\boldsymbol{M}\ddot{\boldsymbol{d}}_n + \boldsymbol{K}\boldsymbol{d}_n = \boldsymbol{F}_n,\ \boldsymbol{d}_n = \boldsymbol{d}\left(t_n\right),\ \ddot{\boldsymbol{d}}_n = \ddot{\boldsymbol{d}}\left(t_n\right),\ \boldsymbol{F}_n = \boldsymbol{F}\left(t_n\right). \tag{3.27}$$

Such incremental approach is called numerical time integration. According to Hughes (2000), the most widely used methods for solving this type of problem belong to the Newmark family of methods. This family of methods is defined by equation (3.27) and the following equations:

$$\boldsymbol{d}_{n+1} = \boldsymbol{d}_n + \Delta t\,\dot{\boldsymbol{d}}_n + \frac{\Delta t^2}{2}\left[\left(1-2\beta\right)\ddot{\boldsymbol{d}}_n + 2\beta\ddot{\boldsymbol{d}}_{n+1}\right], \tag{3.28}$$

$$\dot{\boldsymbol{d}}_{n+1} = \dot{\boldsymbol{d}}_n + \Delta t\left[\left(1-\gamma\right)\ddot{\boldsymbol{d}}_n + \gamma\ddot{\boldsymbol{d}}_{n+1}\right]. \tag{3.29}$$

In equations (3.27) to (3.29) the dot over the displacement vector denotes time derivative. Therefore, $\dot{\boldsymbol{d}}$ represents the vector of velocities and $\ddot{\boldsymbol{d}}$ represents the vector of accelerations. The subscripts in these vectors refer to the time instant (for instance, the current time t_n and the next time t_{n+1}).

The Newmark family varies depending on the values adopted for the parameters β and γ. According to Newmark (1959), in order to avoid spurious dumping, one must adopt $\gamma = 1/2$.

If $\beta = 0$, the displacement at the next time instant depends only on the velocity and the acceleration at the current time (which are known). An explicit time integration scheme is then configured. In this situation, the numerical time integration is conditionally stable, depending on the magnitude of the time increment.

Another commonly adopted value for β is $1/4$, resulting in a constant acceleration value (average value). An implicit time integration scheme is then achieved. A known property of this numerical time integration is its unconditional stability.

The displacement and velocity at initial time $\boldsymbol{t_0}$ constitute necessary information for the definition of the IBVP, as already stated in (2.11). The initial acceleration is then estimated from (3.27), as indicated

$$\ddot{\boldsymbol{d}}_0 = \boldsymbol{M}^{-1}\left[\boldsymbol{F}_0 - \boldsymbol{K}\,\boldsymbol{d}_0\right]. \tag{3.30}$$

For the subsequent time instants, these values are taken as the ones computed in the previous time step.

Since the acceleration in the next step is not known *a priori* for the implicit scheme, one departs from explicit predictions for the displacement and velocity at the current time, computed respectively as

$$\tilde{\boldsymbol{d}}_{n+1} = \boldsymbol{d}_n + \Delta t\,\dot{\boldsymbol{d}}_n + \frac{\Delta t^2}{4}\ddot{\boldsymbol{d}}_n, \tag{3.31}$$

$$\dot{\tilde{\boldsymbol{d}}}_{n+1} = \dot{\boldsymbol{d}}_n + \frac{\Delta t}{2}\ddot{\boldsymbol{d}}_n. \tag{3.32}$$

Using the predicted values, equations (3.28) and (3.29) can be rewritten, respectively, as

$$\boldsymbol{d}_{n+1} = \tilde{\boldsymbol{d}}_{n+1} + \frac{\Delta t^2}{4}\ddot{\boldsymbol{d}}_{n+1}, \tag{3.33}$$

$$\dot{\boldsymbol{d}}_{n+1} = \dot{\tilde{\boldsymbol{d}}}_{n+1} + \frac{\Delta t}{2}\ddot{\boldsymbol{d}}_{n+1}. \tag{3.34}$$

From (3.33) one can compute

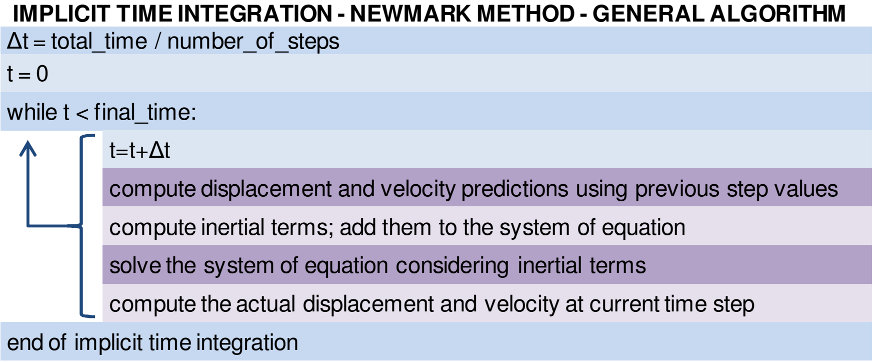
$$\ddot{\boldsymbol{d}}_{n+1} = \frac{4}{\Delta t^2}\left(\boldsymbol{d}_{n+1} - \tilde{\boldsymbol{d}}_{n+1}\right). \tag{3.35}$$

Finally, by substituting the acceleration from equation (3.35) in (3.27), and after rearranging terms, one obtains

$$\left[\frac{4\boldsymbol{M}}{\Delta t^2} + \boldsymbol{K}\right]\boldsymbol{d}_{n+1} = \boldsymbol{F}_{n+1} + \left(\frac{4\boldsymbol{M}}{\Delta t^2}\right)\tilde{\boldsymbol{d}}_{n+1}. \tag{3.36}$$

This is the system of equations to be solved at each time instant in dynamical analyses. Since no assumption is made regarding the stiffness matrix, this formulation can be employed also for nonlinear problems.

The general implicit time integration algorithm for performing dynamical analyses is summarized in Figure 3.7.



```
IMPLICIT TIME INTEGRATION - NEWMARK METHOD - GENERAL ALGORITHM
Δt = total_time / number_of_steps
t = 0
while t < final_time:
    t=t+Δt
    compute displacement and velocity predictions using previous step values
    compute inertial terms; add them to the system of equation
    solve the system of equation considering inertial terms
    compute the actual displacement and velocity at current time step
end of implicit time integration
```

Figure 3.7. General algorithm for the implicit time integration using the Newmark Method.

3.4. Nonlinear analysis

Within the numerical solution treated in the previous section, one of the specific procedures to be performed in each time step is the solution of the resulting system of equation. This means that the problem is solved in an incremental form.

As already discussed in Chapter 2, several factors can make the resulting system of equation nonlinear. The solution of the nonlinear systems of equations is in general based on its linearization at each time step of the incremental solution procedure.

The Newton's method has proven to be one of the most efficient methods for solving nonlinear FEM problems (Wriggers (2006)). Accordingly, the model linearization is derived by employing a Taylor series truncated in the first order derivative terms. Departing from this form, the method consists of an iterative strategy for successively computing approximate solution using this linear approximation.

For instance, taking an hypothetic scalar valued function $f(x)$ of a single scalar variable x, its Taylor series expansion at $\overline{x}$ is given by

$$f(\overline{x}+\Delta x) = f(\overline{x}) + \left.\frac{df(x)}{dx}\right|_{x=\overline{x}} \Delta x + \vartheta\,. \tag{3.37}$$

If $\overline{x}+\Delta x$ is sufficiently close to $\overline{x}$, the residual associated to the higher order terms ϑ tends to vanish, remaining the first order term as significant. Supposing that a root of $f\left(x\right)$ is given by the value $\overline{x}+\Delta x$, one achieves that

$$f(\overline{x}) + \left.\frac{df(x)}{dx}\right|_{x=\overline{x}} \Delta x = f(\overline{x}) + f'(\overline{x})\Delta x \cong 0\,. \tag{3.38}$$

The last relation provides a 'rough' estimate to the actual Δx as follows:

$$\Delta x = -\frac{f(\overline{x})}{f'(\overline{x})}. \tag{3.39}$$

Such estimative is more accurate as $\overline{x}$ is closer to the solution. According to the Newton's method strategy, one successively computes new values for $\overline{x}$ until it converges to the root of the nonlinear equation. The root search procedure basically consists of two steps:

a) Computing $\Delta x_n = \frac{f(x_n)}{f'(x_n)}$;

b) Computing the new value $x_{x+1} = x_n + \Delta x$.

The procedure is repeated iteratively until a previously established convergence criterion to $f\left(x\right) = 0$ is fulfilled. From the formulation above one can clearly notice that the method is not unconditionally stable. For instance, if $f'(x_n) = 0$, the method fails.

The method is also applicable for the case of nonlinear systems of equations, like the ones arising from the FEM and GFEM numerical approaches of the nonlinear problem. For this case, many other aspects regarding the method's stability arise (see, for instance, Marsden and Hughes (1994)). In general terms, if the initial guess is close enough to the solution and the coefficient matrix of the nonlinear system of equations is positive definite at $\overline{x}$, the method is robust and stable. Moreover, in this case, the method exhibits a quadratic convergence rate.

Aiming to illustrate the method applied for the FEM/GFEM, let's take the nonlinear system of equations written as

$$\boldsymbol{K}\left(\boldsymbol{d}\right)\boldsymbol{d} = \boldsymbol{F}, \tag{3.40}$$

in which $\boldsymbol{d}$ is a vector containing the nodal displacement parameters, for the FEM, or the GDOF for the GFEM. $\boldsymbol{F}$ is the equivalent nodal force vector for the FEM and a vector of equivalent nodal generalized forces for the GFEM. The matrix $\boldsymbol{K}\left(\boldsymbol{d}\right)$ represents the coefficient matrix. As one can notice, for the nonlinear problem the stiffness matrix $\boldsymbol{K}$ depends on the displacements (unknowns), which characterize the nonlinear feature of the model.

Supposing, without loss of generality, that the inertial effects are negligible, and that the loads do not depend on the deformed configuration, the nonlinear behavior arises essentially from the internal virtual work term.

Based in the Newton's method approach, let's suppose $\overline{\boldsymbol{d}}$ represents one model's deformed configuration, which is close to the equilibrium. In this situation, $\boldsymbol{K}(\overline{\boldsymbol{d}})\overline{\boldsymbol{d}} \neq \boldsymbol{F}$ and is possible to evaluate the residual vector

$$\boldsymbol{R}\left(\overline{\boldsymbol{d}}\right) = \boldsymbol{F} - \boldsymbol{K}(\overline{\boldsymbol{d}})\overline{\boldsymbol{d}}\,. \tag{3.41}$$

Taking into account that the current deformed configuration is close to the equilibrium, it is possible to evaluate a vector of increments $\Delta\boldsymbol{d}$ such that

$$\boldsymbol{R}(\overline{\boldsymbol{d}} + \Delta\boldsymbol{d}) \cong \boldsymbol{R}(\overline{\boldsymbol{d}}) + D\boldsymbol{R}(\overline{\boldsymbol{d}})\cdot\Delta\boldsymbol{d}\,. \tag{3.42}$$

Moreover, since the residual vector vanishes in the equilibrium condition, the linearization of the nonlinear system of equations yields to

$$D\boldsymbol{R}(\overline{\boldsymbol{d}})\cdot\Delta\boldsymbol{d} = \boldsymbol{K}_T(\overline{\boldsymbol{d}})\cdot\Delta\boldsymbol{d} = -\boldsymbol{R}(\overline{\boldsymbol{d}})\,. \tag{3.43}$$

In the FEM literature, the matrix $\boldsymbol{K}_T$ is referred to as the tangent stiffness matrix, and the vector $\boldsymbol{R}(\overline{\boldsymbol{d}})$ represents the residual force vector. Following the Newton's method approach one iteratively computes:

a) The vector $\Delta\boldsymbol{d}_n$, arising from the solution of $\boldsymbol{K}_T(\boldsymbol{d}_n)\cdot\Delta\boldsymbol{d}_n = -\boldsymbol{R}(\boldsymbol{d}_n)$;

b) The new vector $\boldsymbol{d}_{n+1} = \boldsymbol{d}_n + \Delta\boldsymbol{d}_n$.

The process is repeated iteratively until the solution convergence criterion is considered satisfied or a criterion for stopping, based on the maximum number of iterations, is exceeded. Regarding the first case, the definition of the convergence criteria is necessary. The second condition is important for avoiding an infinity loop in situations where the solution process diverges.

Probably the most straightforward convergence criterion to be adopted is based on the control of the magnitude of the residual force vector, which can be evaluated by computing its L2 norm, defined as

$$\left\|\boldsymbol{R}_n\right\|_2 = \sqrt{\boldsymbol{R}_n \cdot \boldsymbol{R}_n}\,. \tag{3.44}$$

This criterion is also referred to as nodal force L2 norm, and is stated as

$$\left\|\boldsymbol{R}_n\right\|_2 < \varepsilon_f\,, \tag{3.45}$$

in which ε_f is an adopted positive scalar tolerance to the nodal force L2 norm. Moreover, aiming to obtain a criterion less dependent of the scale of the force, one can express this criterion as

$$\frac{\left\|\boldsymbol{R}_n\right\|_2}{\left\|\boldsymbol{F}\right\|_2} < \varepsilon_f , \tag{3.46}$$

hereby referred to as the relative nodal force L2 norm.

Considering that the solution is computed in a step-by-step procedure by displacement increments, another criterion to be used regards controlling the magnitude of the increment displacement vector, as follows:

$$\left\|\Delta \boldsymbol{d}_n\right\|_2 < \varepsilon_d . \tag{3.47}$$

The criterion stated in (3.47) is also referred to as the displacement increments L2 norm. The relative displacement increments L2 norm can be stated as

$$\frac{\left\|\Delta \boldsymbol{d}_n\right\|_2}{\left\|\boldsymbol{d}_n\right\|_2} < \varepsilon_d . \tag{3.48}$$

One third criterion is presented in Bathe (1996), representing the internal energy increment during each iteration, being defined as

$$\Delta \boldsymbol{d}_n \cdot \boldsymbol{R}_n < \varepsilon_e . \tag{3.49}$$

In theory, all the previously presented convergence criteria could be applied to the GFEM. However, in such method, the unknowns are not only nodal displacements, as it happens for the FEM. In fact, the GDOF values compounding the vector d_n do not have a directly associated physical meaning in general, requiring a post-processing scheme in order to compute the displacements in the solid.

Due to this fact, in the present work, we propose a new convergence criterion, based in the increment of the post-processed displacements. According to it, these displacement increment values are computed in accordance with the GFEM methodology, at all the integration points in the mesh, and are grouped in a vector $\Delta \boldsymbol{d}_n^{pp}$.

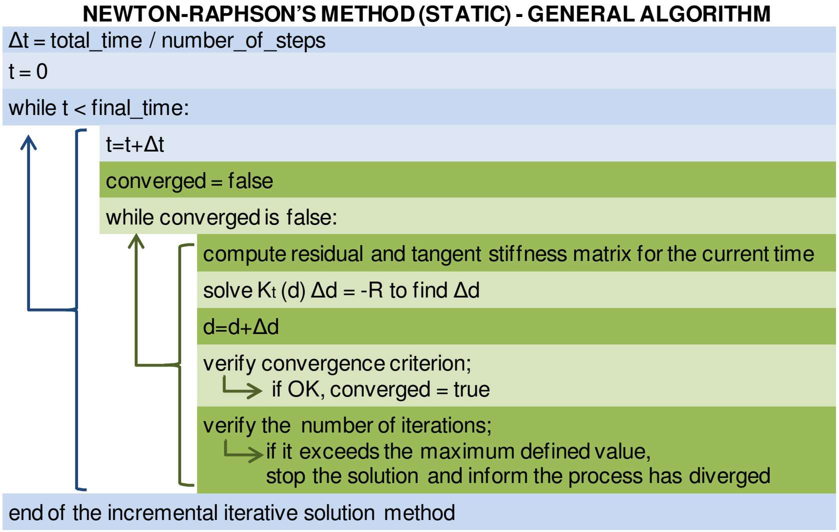
The criterion is then referred to as post-processed displacement increment L2 norm, being defined as

$$\left\|\boldsymbol{d}_n^{pp}\right\|_2 < \varepsilon_{pp} . \tag{3.50}$$

One must observe that there is no restriction on the usage of this norm also for the FEM. A brief evaluation of these norms both for the FEM and GFEM is presented in example 6.2 of the Chapter 6.

When the Newton's method is applied associated to an incremental analysis, as the one presented in Section 3.3, it is referred to as Newton-Raphson's method. Many nonlinear problems demand an incremental iterative approach in order to achieve convergence.

Finally, temporarily ignoring the inertial effect (aiming to simplify the scheme), the general algorithm for an incremental iterative Newton-Raphson's method solution is shown in Figure 3.8.



```
NEWTON-RAPHSON'S METHOD (STATIC) - GENERAL ALGORITHM
Δt = total_time / number_of_steps
t = 0
while t < final_time:
    t=t+Δt
    converged = false
    while converged is false:
        compute residual and tangent stiffness matrix for the current time
        solve Kt (d) Δd = -R to find Δd
        d=d+Δd
        verify convergence criterion;
            └> if OK, converged = true
        verify the number of iterations;
            └> if it exceeds the maximum defined value,
               stop the solution and inform the process has diverged
end of the incremental iterative solution method
```

Figure 3.8. General algorithm scheme for the Newton-Raphson's Method.

3.4.1. Small strain elastoplasticity

In the small strain elastoplastic modeling, the nonlinear behavior results from the response of the material, embedded by the constitutive relation. The kinematic relation employed in such model is identical to the one used for the linear elastic model.

In the numerical approach, one initially assumes that the model follows a linear elastic response. This prevision hypothesis is referred to as elastic trial state. This assumption is actually true for all points of the domain in which the stress state does not exceed the yield criterion, i.e., stress states for which the yield function (2.20) results negative.

On the other hand, it is possible that some points of the domain do not verify yield criterion, violating the condition stated in (2.22). In this case, the obtained stress state cannot be feasible, and must be corrected, to a level that corresponds to the nullity the yield function. Therefore, a new equilibrium configuration, for which the yield criterion is feasible, must be searched. In these points, the stress-strain response is given by the elastoplastic constitutive tensor.

As already commented in Chapter 2, for the Newton-Raphson's method, the elastoplastic constitutive tensor from (2.31) must be replaced by the so-called consistent elastoplastic tangent moduli. However, different formulation for this tensor arises whether a plane stress or plane strain hypotheses is assumed. For further details in the derivation of such tensor, Simo and Hughes (1998) is hereby referenced.

In what follows, one essentially reproduces the formulation both for the plane stress and plane strain hypotheses derived in Simo and Hughes (1998).

a) Plane strain hypothesis

After the initial solution for a given iteration is attained, assuming the elastic trial scheme, for each of the integration points, one computes the deviatoric strain and stress tensor:

$$\boldsymbol{e}_{n+1} = \varepsilon_{n+1} - \frac{1}{3} tr\left(\varepsilon_{n+1}\right)\boldsymbol{I} , \tag{3.51}$$

$$\boldsymbol{S}_{n+1}^{trial} = 2\mu\left(\boldsymbol{e}_{n+1} - \boldsymbol{e}_{n}^{p}\right). \tag{3.52}$$

For these equations, $\boldsymbol{e}_{n+1}$ is the deviatoric strain for the elastic trial, $tr(\bullet)$ is the trace operator, $\boldsymbol{S}_{n+1}^{trial}$ is the deviatoric stress for the elastic trial, $\boldsymbol{e}_{n}^{p}$ is the plastic strain for the given point, accumulated up to the last iteration and μ is the Lamé parameter, presented in (2.46).

Once $\boldsymbol{S}_{n+1}^{trial}$ is achieved, one computes the yield condition for the elastic trial

$$f_{n+1}^{trial} := \left\|\boldsymbol{S}_{n+1}^{trial}\right\| - \sqrt{\tfrac{2}{3}} K\left(\alpha_{n}\right), \tag{3.53}$$

in which $K\left(\alpha_{n}\right)$ is the isotropic hardening function presented in (2.21).

If $f_{n+1}^{trial} \leq 0$, the elastic trial hypothesis hold, and the values for the stress and strain tensors computed for the current iteration are in fact the trial values.

On the other hand, if $f_{n+1}^{trial} > 0$ one must compute the $\dot{\lambda}$ parameter presented in (2.23), such that the yield criterion is respected. For the finite time step numerical scheme, we refer this parameter as $\Delta\lambda$. It is computed iteratively using the Newton's method.

Within this iterative process, the value for $\Delta\lambda$ before the first iteration is zero ($\Delta\lambda^{(0)} = 0$), and the hardening parameter value is taken from the last iteration of the Newton-Raphson scheme, i.e., $\alpha_{n+1}^{(0)} = \alpha_{n}$. The nonlinear equation to be solved is

$$g\left(\Delta\lambda^{(k)}\right) := \left\|\boldsymbol{S}_{n+1}^{trial}\right\| - \sqrt{\tfrac{2}{3}} K\left(\alpha_{n+1}^{(k)}\right) - \left(2\mu\Delta\lambda^{(k)}\right). \tag{3.54}$$

For each iteration k the new value of $\Delta\lambda^{(k+1)}$ is computed as

$$\Delta\lambda^{(k+1)} = \Delta\lambda^{(k)} - \frac{g\left(\Delta\lambda^{(k)}\right)}{Dg\left(\Delta\lambda^{(k)}\right)}, \tag{3.55}$$

in which

$$Dg\left(\Delta\lambda^{(k)}\right) := -2\mu\left[1 + \frac{K'\left(\alpha_{n+1}^{(k)}\right)}{3\mu}\right]. \tag{3.56}$$

For each of the iterations, the isotropic hardening parameter value must also be updated by

$$\alpha_{n+1}^{(k+1)} = \alpha_n + \sqrt{\tfrac{2}{3}}\,\Delta\lambda^{(k+1)}. \tag{3.57}$$

The iterative process is repeated until $\left|g\left(\Delta\lambda^{(k)}\right)\right|$ is less than a previously defined tolerance value. After it has converged, one computes:

$$\boldsymbol{N}_{n+1} := \frac{\boldsymbol{S}_{n+1}^{trial}}{\left\|\boldsymbol{S}_{n+1}^{trial}\right\|}, \tag{3.58}$$

$$\alpha_{n+1} := \alpha_n + \sqrt{\tfrac{2}{3}}\,\Delta\lambda. \tag{3.59}$$

With these values one can compute the new value of the plastic strain tensor, and the actual stress tensor for the point by:

$$\boldsymbol{e}_{n+1}^{p} = \boldsymbol{e}_{n}^{p} + \Delta\lambda\,\boldsymbol{N}_{n+1}, \tag{3.60}$$

$$\boldsymbol{\sigma}_{n+1} = \kappa\, tr\left(\varepsilon_{n+1}\right)\boldsymbol{I} + \boldsymbol{S}^{trial} - 2\mu\Delta\lambda\,\boldsymbol{N}_{n+1}. \tag{3.61}$$

In (3.61) κ is the bulk modulus, which is computed by

$$\kappa = \frac{E}{3(1-2\upsilon)}. \tag{3.62}$$

Finally, the consistent elastoplastic tangent moduli is computed by

$$\mathbb{C}_{n+1}^{ep} = \kappa\boldsymbol{I}\otimes\boldsymbol{I} + 2\mu\theta_{n+1}\left[\mathbb{I} - \frac{1}{3}\boldsymbol{I}\otimes\boldsymbol{I}\right] - 2\mu\bar{\theta}_{n+1}\,\boldsymbol{N}_{n+1}\otimes\boldsymbol{N}_{n+1}, \tag{3.63}$$

$$\theta_{n+1} := 1 - \frac{2\mu\Delta\lambda}{\left\|\boldsymbol{S}_{n+1}^{trial}\right\|}, \tag{3.64}$$

$$\bar{\theta}_{n+1} := \frac{1}{1 + \frac{K'_{n+1}}{3\mu}} - \left(1 - \theta_{n+1}\right). \tag{3.65}$$

b) Plane stress hypothesis

Like for the plane strain, the first step of the analysis is the elastic trial, in which one computes the trial strain and stress states:

$$\boldsymbol{\varepsilon}_{n+1} = \boldsymbol{\varepsilon}_{n} + \nabla^{s}\boldsymbol{u}, \tag{3.66}$$

$$\boldsymbol{\sigma}_{n+1}^{trial} = \mathbb{C}\left(\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}^{p}\right) \tag{3.67}$$

From the elastic trial stress tensor one computes the deviatoric stress tensor and evaluates the yield criterion (equation (3.53)). Like it happens for the plane strain, if $f_{n+1}^{trial} \leq 0$, the elastic trial hypothesis hold, and the values for the stress and strain computed for the current iteration are in fact the trial values.

If $f_{n+1}^{trial} > 0$, one needs to compute $\Delta\lambda$. The nonlinear equation to be solved is

$$g\left(\Delta\lambda\right) = f^{2}\left(\Delta\lambda\right) := \frac{1}{2}\bar{f}^{2}\left(\Delta\lambda\right) - R^{2}\left(\Delta\lambda\right) = 0, \tag{3.68}$$

$$\bar{f}^{2}\left(\Delta\lambda\right) := \frac{1}{2}\frac{\frac{1}{3}\left(\sigma_{x}^{trial} + \sigma_{y}^{trial}\right)^{2}}{\left\{1 + \left[\frac{E}{3\left(1-v\right)}\right]\Delta\lambda\right\}^{2}} + \frac{\frac{1}{2}\left(\sigma_{x}^{trial} - \sigma_{y}^{trial}\right)^{2} + 2\left(\tau_{xy}^{trial}\right)^{2}}{\left[1 + 2\mu\,\Delta\lambda\right]^{2}}, \tag{3.69}$$

$$R^{2}\left(\Delta\lambda\right) := \frac{1}{3}K^{2}\left(\beta\right), \tag{3.70}$$

$$\beta = \alpha_{n} + \sqrt{\tfrac{2}{3}}\,\Delta\lambda\,\bar{f}\left(\Delta\lambda\right). \tag{3.71}$$

The same iterative algorithm presented in (3.55) is used to solve such nonlinear equation. For the plane stress, the term Dg is:

$$Dg\left(\Delta\lambda\right) = \frac{1}{2}D\bar{f}^{2}\left(\Delta\lambda\right) - DR^{2}\left(\Delta\lambda\right) = 0, \tag{3.72}$$

$$\begin{aligned} D\bar{f}^{2}\left(\Delta\lambda\right) := & \frac{-\frac{1}{3}\left(\sigma_{x}^{trial} + \sigma_{y}^{trial}\right)^{2}}{\left\{1 + \left[\frac{E}{3\left(1-v\right)}\right]\Delta\lambda\right\}^{3}}\left[\frac{E}{3\left(1-v\right)}\right] \\ & - 2\frac{\frac{1}{2}\left(\sigma_{x}^{trial} - \sigma_{y}^{trial}\right)^{2} + 2\left(\tau_{xy}^{trial}\right)^{2}}{\left[1 + 2\mu\,\Delta\lambda\right]^{3}}\left[2\mu\right] \end{aligned}, \tag{3.73}$$

$$DR^{2}\left(\Delta\lambda\right) := \frac{2}{3}K\left(\beta\right)\;K'\left(\beta\right)\frac{d\beta\left(\Delta\lambda\right)}{d\Delta\lambda}, \tag{3.74}$$

$$\frac{d\beta\left(\Delta\lambda\right)}{d\Delta\lambda} = \sqrt{\frac{2}{3}}\left(\bar{f}\left(\Delta\lambda\right) + \Delta\lambda\;\frac{d\bar{f}\left(\Delta\lambda\right)}{d\Delta\lambda}\right), \tag{3.75}$$

$$\frac{d\bar{f}\left(\Delta\lambda\right)}{d\Delta\lambda} = \frac{1}{2}\frac{1}{\bar{f}\left(\Delta\lambda\right)} D\bar{f}^{2}\left(\Delta\lambda\right). \tag{3.76}$$

When the nonlinear solution converges, one obtains $\Delta\lambda$ and updates the following:

$$\alpha_{n+1} = \alpha_{n} + \sqrt{\tfrac{2}{3}}\,\Delta\lambda\,\bar{f}\left(\Delta\lambda\right), \tag{3.77}$$

$$\boldsymbol{\varepsilon}^{p}_{n+1} = \boldsymbol{\varepsilon}^{p}_{n} + \Delta\lambda\,\boldsymbol{P}\,\boldsymbol{\sigma}^{trial}_{n+1} = \boldsymbol{\varepsilon}^{p}_{n} + \Delta\lambda\frac{1}{3}\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 6 \end{bmatrix}\begin{pmatrix} \sigma_{x} \\ \sigma_{y} \\ \tau_{xy} \end{pmatrix}^{trial}_{n+1}, \tag{3.78}$$

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{\Xi}\left(\Delta\lambda\right)\mathbb{C}^{-1}\boldsymbol{\sigma}^{trial}_{n+1}, \tag{3.79}$$

$$\boldsymbol{\Xi} := \left(\mathbb{C}^{-1} + \Delta\lambda\,\boldsymbol{P}\right)^{-1}, \tag{3.80}$$

$$\left(\varepsilon_{z}\right)_{n+1} = \frac{-\upsilon}{E}\left[\left(\sigma_{x}\right)_{n+1} + \left(\sigma_{y}\right)_{n+1}\right] - \left[\left(\varepsilon^{p}_{x}\right)_{n+1} + \left(\varepsilon^{p}_{y}\right)_{n+1}\right], \tag{3.81}$$

$$\left(\varepsilon^{p}_{z}\right)_{n+1} = -\left(\varepsilon^{p}_{x}\right)_{n+1} - \left(\varepsilon^{p}_{y}\right)_{n+1}. \tag{3.82}$$

The tensor $\boldsymbol{\Xi}$ is referred to as modified algorithmic elastic tangent moduli (see Simo and Hughes (1998)). The consistent elastoplastic tangent moduli is

$$\mathbb{C}^{ep}_{n+1} = \boldsymbol{\Xi} - \frac{\left[\boldsymbol{\Xi}\boldsymbol{P}\boldsymbol{\sigma}_{n+1}\right]\left[\boldsymbol{\Xi}\boldsymbol{P}\boldsymbol{\sigma}_{n+1}\right]^{T}}{\boldsymbol{\sigma}^{T}_{n+1}\,\boldsymbol{P}\boldsymbol{\Xi}\boldsymbol{P}\boldsymbol{\sigma}_{n+1} + \bar{\beta}_{n+1}}, \tag{3.83}$$

$$\bar{\beta}_{n+1} = \frac{2}{3}\frac{1}{1 - \frac{2}{3}K'_{n+1}\Delta\lambda}K'_{n+1}\boldsymbol{\sigma}^{T}_{n+1}\boldsymbol{P}\boldsymbol{\sigma}_{n+1}. \tag{3.84}$$

3.4.2. Finite displacement elastic analysis

For the solution of the nonlinear system of equation of the finite displacement elastic analysis, one needs to perform the linearization of the internal virtual work from (2.57).

Both for computing the internal force vector and the tangent matrix, one needs to compute the virtual Green strain tensor, which is given by

$$\boldsymbol{\delta E} = \frac{1}{2}\left[\boldsymbol{F}^{T}\,\nabla\boldsymbol{\delta u} + \nabla\boldsymbol{\delta u}^{T}\boldsymbol{F}\right]. \tag{3.85}$$

For its application in the FEM/GFEM, it can be expressed in matrix terms as

$$\boldsymbol{\delta E} = \begin{Bmatrix} \delta E_x \\ \delta E_y \\ \delta E_{xy} \\ \delta E_{yx} \end{Bmatrix} = \underbrace{\boldsymbol{B}^*}_{(4\text{x}n_{gdof})} \underbrace{\boldsymbol{\delta d}}_{(n_{gdof}\text{x}1)} , \tag{3.86}$$

in which $\boldsymbol{\delta d}$ is the vector containing the virtual displacement terms, and $\boldsymbol{B}^*$ is computed by

$$\boldsymbol{B}^* = \underbrace{\boldsymbol{G}}_{(4\text{x}4)} \underbrace{\boldsymbol{H}^*}_{(4\text{x}4)} \underbrace{\widehat{\boldsymbol{B}}}_{(4\text{x}n_{gdof})} , \tag{3.87}$$

$$\boldsymbol{G} = \begin{bmatrix} 1+\dfrac{\partial u}{\partial X} & 0 & \dfrac{\partial v}{\partial X} & 0 \\ 0 & \dfrac{\partial u}{\partial Y} & 0 & 1+\dfrac{\partial v}{\partial Y} \\ \dfrac{1}{2}\left(\dfrac{\partial u}{\partial Y}\right) & \dfrac{1}{2}\left(1+\dfrac{\partial u}{\partial X}\right) & \dfrac{1}{2}\left(1+\dfrac{\partial v}{\partial Y}\right) & \dfrac{1}{2}\left(\dfrac{\partial v}{\partial X}\right) \\ \dfrac{1}{2}\left(\dfrac{\partial u}{\partial Y}\right) & \dfrac{1}{2}\left(1+\dfrac{\partial u}{\partial X}\right) & \dfrac{1}{2}\left(1+\dfrac{\partial v}{\partial Y}\right) & \dfrac{1}{2}\left(\dfrac{\partial v}{\partial X}\right) \end{bmatrix}, \tag{3.88}$$

$$\boldsymbol{H}^* = \begin{bmatrix} \boldsymbol{H}^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H}^T \end{bmatrix}, \tag{3.89}$$

in which $\boldsymbol{H}^T$ is the transpose of the matrix presented in (3.11) and $\boldsymbol{0}$ is a (2x2) matrix full of zeros. The matrix $\widehat{\boldsymbol{B}}$ contains derivatives of the interpolation function with respect to ξ and η such that

$$\begin{Bmatrix} \dfrac{\partial u}{\partial \xi} & \dfrac{\partial u}{\partial \eta} & \dfrac{\partial v}{\partial \xi} & \dfrac{\partial v}{\partial \eta} \end{Bmatrix}^T = \widehat{\boldsymbol{B}}\ \boldsymbol{\delta d} . \tag{3.90}$$

The matrix $\boldsymbol{B}^*$ allows one to compute the internal forces contribution for a given element by the numerical integration

$$\underbrace{\boldsymbol{f}_{\text{int}}^{elem}}_{(n_{gdof}\text{x}1)} = \sum_{i=1}^{n_{gp}} \underbrace{\left(\boldsymbol{B}^*\left(\xi_i,\eta_i\right)\right)^T}_{(n_{gdof}\text{x}4)} \underbrace{\boldsymbol{S}\left(\xi_i,\eta_i\right)}_{(4\text{x}1)} J\left(\xi_i,\eta_i\right) w_i , \tag{3.91}$$

In (3.91), the points $\left(\xi_i,\eta_i\right)$ are given by the quadrature rule for n_{gp} integration points, w_i are the weights for these integration points and $\boldsymbol{S}$ is a vector containing the current value for the second Piolla-Kirchhoff stress components for the point. These stress components are computed using expression (2.47) for the Saint Venant-Kirchhoff material and expression (2.50) for the Neo-Hookean material.

The derivation of the tangent stiffness matrix for an element arise from the linearization internal virtual work presented in (2.57), resulting in

$$D\,\delta W_{elem}^{\text{internal}} = \int_{\Omega^e} \left[\boldsymbol{\delta E} : D\boldsymbol{S} + \boldsymbol{S} : D\boldsymbol{\delta E}\right] d\Omega \,. \tag{3.92}$$

The linearization of the second Piolla-Kirchhoff stress tensor results in

$$D\,\boldsymbol{S}\left(E\left(u\right)\right) = \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{E}} : D\boldsymbol{E}(u) = \mathbb{C}\left(u\right) : D\boldsymbol{E}(u) \,, \tag{3.93}$$

such as the linearization of the Green strain tensor results in

$$D\,\boldsymbol{E}\left(u\right) = \Delta\boldsymbol{E} = \frac{1}{2}\left[\boldsymbol{F}^T\,\nabla\left(\Delta\boldsymbol{u}\right) + \nabla^T\left(\Delta\boldsymbol{u}\right)\boldsymbol{F}\right]. \tag{3.94}$$

Aiming a FEM/GFEM, the term $\Delta\boldsymbol{E}$ can be expressed in matrix terms as

$$\Delta\boldsymbol{E} = \underbrace{\boldsymbol{B}^*}_{(4\text{x}n_{gdof})}\;\underbrace{\Delta\boldsymbol{d}}_{(n_{gdof}\text{x}1)} \,. \tag{3.95}$$

For further details on these derivations, we reference Holzapfel (2000).

Employing this notation, the first term of (3.92) results in

$$\int_{\Omega^e} \delta\boldsymbol{E} : \mathbb{C} : \Delta\boldsymbol{E}\,d\Omega = \boldsymbol{\delta d}^T\left[\int_{\Omega^e}\left(\boldsymbol{B}^*\right)^T \mathbb{C}\,\boldsymbol{B}^*\,d\Omega\right]\Delta\boldsymbol{d} = \underbrace{\boldsymbol{\delta d}^T}_{(1\text{x}n_{gdof})}\;\underbrace{\boldsymbol{K}_{T1}^{elem}}_{(n_{gdof}\text{x}n_{gdof})}\;\underbrace{\Delta\boldsymbol{d}}_{(n_{gdof}\text{x}1)} \,. \tag{3.96}$$

The linearization of the virtual Green strain results in

$$D\,\boldsymbol{\delta E} = \frac{1}{2}\left[\nabla^T\left(\Delta\boldsymbol{u}\right)\nabla\boldsymbol{\delta u} + \nabla^T\left(\boldsymbol{\delta u}\right)\nabla\left(\Delta\boldsymbol{u}\right)\right]. \tag{3.97}$$

For the plane stress and plane strain, the term $\boldsymbol{S} : D\,\boldsymbol{\delta E}$ can be expressed by

$$\widehat{\boldsymbol{\delta}}^T\,\widehat{\boldsymbol{S}} \,, \tag{3.98}$$

in which

$$\widehat{\boldsymbol{\delta}}^T = \left\{\frac{\partial\delta u}{\partial X} \quad \frac{\partial\delta v}{\partial X} \quad \frac{\partial\delta u}{\partial Y} \quad \frac{\partial\delta v}{\partial Y} \quad \frac{\partial\delta u}{\partial Y} \quad \frac{\partial\delta v}{\partial Y} \quad \frac{\partial\delta u}{\partial X} \quad \frac{\partial\delta v}{\partial X}\right\}, \tag{3.99}$$

$$\begin{aligned}\widehat{\boldsymbol{S}}^T = \Bigg\{ & S_{xx}\frac{\partial\Delta u}{\partial X} \quad S_{xx}\frac{\partial\Delta v}{\partial X} \quad S_{yy}\frac{\partial\Delta u}{\partial Y} \quad S_{yy}\frac{\partial\Delta v}{\partial Y} \\ & S_{xy}\frac{\partial\Delta u}{\partial X} \quad S_{xy}\frac{\partial\Delta v}{\partial X} \quad S_{xy}\frac{\partial\Delta u}{\partial Y} \quad S_{xy}\frac{\partial\Delta v}{\partial Y}\Bigg\}.\end{aligned} \tag{3.100}$$

These vectors can be assembled by

$$\widehat{\boldsymbol{\delta}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\boldsymbol{A}} \begin{Bmatrix} \dfrac{\partial \delta u}{\partial X} \\ \dfrac{\partial \delta u}{\partial Y} \\ \dfrac{\partial \delta v}{\partial X} \\ \dfrac{\partial \delta v}{\partial Y} \end{Bmatrix} = \boldsymbol{A}\boldsymbol{H}^{*}\widehat{\boldsymbol{B}}\,\boldsymbol{\delta d}, \tag{3.101}$$

$$\widehat{\boldsymbol{S}} = \underbrace{\begin{bmatrix} S_{xx} & 0 & 0 & 0 \\ 0 & 0 & S_{xx} & 0 \\ 0 & S_{yy} & 0 & 0 \\ 0 & 0 & 0 & S_{yy} \\ S_{xy} & 0 & 0 & 0 \\ 0 & 0 & S_{xy} & 0 \\ 0 & S_{xy} & 0 & 0 \\ 0 & 0 & 0 & S_{xy} \end{bmatrix}}_{\boldsymbol{A}^{*}} \begin{Bmatrix} \dfrac{\partial \Delta u}{\partial X} \\ \dfrac{\partial \Delta u}{\partial Y} \\ \dfrac{\partial \Delta v}{\partial X} \\ \dfrac{\partial \Delta v}{\partial Y} \end{Bmatrix} = \boldsymbol{A}^{*}\boldsymbol{H}^{*}\widehat{\boldsymbol{B}}\,\boldsymbol{\delta d}. \tag{3.102}$$

Employing this notation, the first term of (3.93) results in

$$\underbrace{\boldsymbol{\delta d}^{T}}_{(1\mathrm{x}n_{gdof})} \left\{ \int\limits_{\Omega^{e}} \left[\underbrace{\widehat{\boldsymbol{B}}^{T}}_{(n_{gdof}\mathrm{x}4)} \underbrace{\left(\boldsymbol{H}^{*}\right)^{T}}_{(4\mathrm{x}4)} \underbrace{\boldsymbol{A}^{T}}_{(4\mathrm{x}8)} \underbrace{\boldsymbol{A}^{*}}_{(8\mathrm{x}4)} \underbrace{\boldsymbol{H}^{*}}_{(4\mathrm{x}4)} \underbrace{\widehat{\boldsymbol{B}}}_{(4\mathrm{x}n_{gdof})} \right] d\Omega \right\} \underbrace{\Delta \boldsymbol{d}}_{(n_{gdof}\mathrm{x}1)} = \underbrace{\boldsymbol{\delta d}^{T}}_{(1\mathrm{x}n_{gdof})} \underbrace{\boldsymbol{K}_{T2}^{elem}}_{(n_{gdof}\mathrm{x}n_{gdof})} \underbrace{\Delta \boldsymbol{d}}_{(n_{gdof}\mathrm{x}1)} \,. \tag{3.103}$$

The integral over the element domain presented in (3.96) and (3.103) is performed numerically like it happens for the linear case, as already presented in (3.17).

The tangent stiffness matrix for the element is given by

$$\boldsymbol{K}_{T}^{elem} = \boldsymbol{K}_{T1}^{elem} + \boldsymbol{K}_{T2}^{elem}\,. \tag{3.104}$$

The computing of the global tangent matrix and the global residual force vector is obviously performed by the traditional assembly using the elemental terms of (3.104) and (3.91), respectively.

3.4.3. Contact problem

For problems related to contact phenomena, the numerical model results in a constrained nonlinear system of equations. The contact constraints arise when different parts of the solids tend to occupy the same position in the space at the same time. Therefore, the constraints must prevent the interpenetration among the solid. Since the contact conditions change during the mechanical process, in accordance with the deformation of the solids, the incremental iterative approach demands additional techniques to identify the occurrence of penetration and contact.

In two-dimensional modeling, the contact surface is a line segment. The contact of a point lying in a solid's boundary, at a given time instant, is performed by measuring the distance between the point and the other solids boundary lines. This distance is defined as the length of the straight segment connecting a point and its normal projection over the boundary.

Owing to the FEM/GFEM mesh discretization, the boundaries are defined by a set of line segments. For the Signorini contact problem, these segments are fixed, and are hereby referred to as obstacle. For the contact among deformable bodies, these segments move with deformation, and are referred to as targets. The targets are associated to the side of plane elements.

Let us initially consider the possible contact between a point **P** and a straight line segment 1-2, as depicted in Figure 3.9 (a).

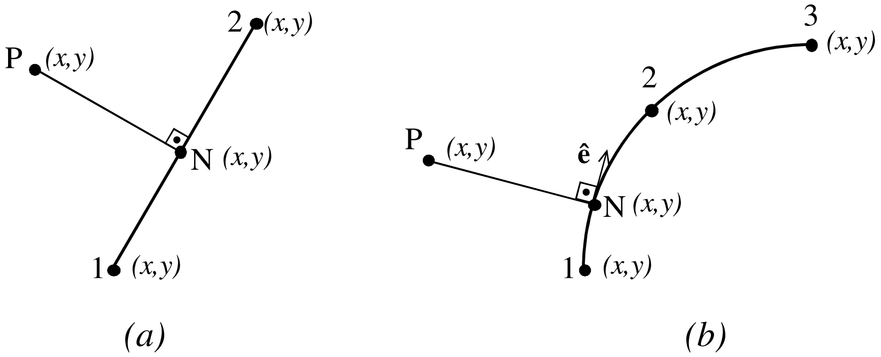


Figure 3.9. Computation of the distance between a point P and a line segment (a) defined by first degree interpolation (straight line); and (b) second degree interpolation (curved line segment).

For the straight line segment, the point **N** is determined by imposing orthogonality between the segment P-N and the segment 1-2. Since the tangent vector is constant in the segment 1-2, the condition results in a linear equation.

However, if the target segment is curved, like the one depicted in Figure 3.9 (b), the tangent vector changes along it. In this case, the position of N is achieved by finding its natural coordinate ξ_N attached to the curved segment such that

$$\left(x_P - x\left(\xi_N\right), y_P - y\left(\xi_N\right)\right) \bullet \left(x'\left(\xi_N\right), y'\left(\xi_N\right)\right) = 0\,. \tag{3.105}$$

However, this equation is nonlinear, demanding a numerical iterative solution method. The Newton's method is one possible approach, while a solution search procedure employing the 'Regula Falsi' method is a second one, see Piedade Neto (2009).

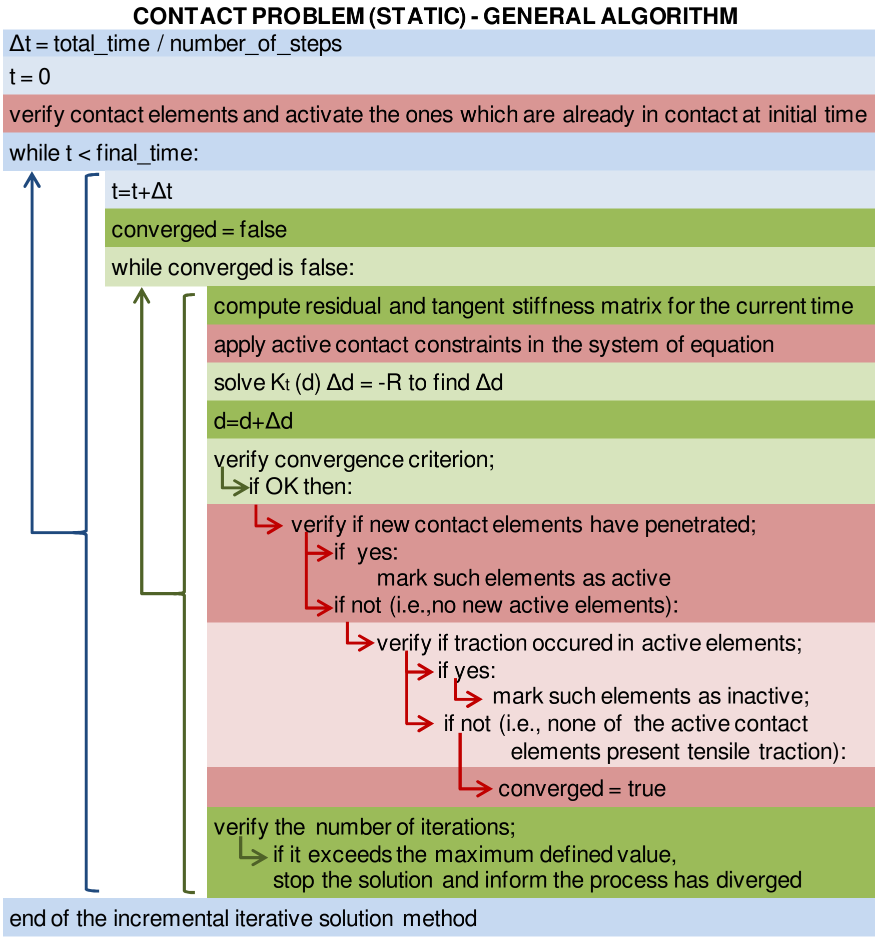
Once the contact pair is achieved at a given time instant, the occurrence of penetration is verified by computing the gap function, expressed in (2.60). If the gap value is positive, the points are not in contact. If the value is negative, penetration has occurred for these points.

If during the incremental iterative solution penetration is detected, constraints must be activated in the nonlinear system of equations. These constraints arise from an additional contribution to the virtual work expression, named contact virtual work. The additional term can be defined either using the Lagrange multipliers, as in (2.68), or the penalty methods, as in (2.69).

Regardless of the adopted approach, once a contact constraint is imposed, it enforces values for the displacement field such that the gap is zero. For subsequent time steps, if the contacting surfaces tend to interpenetrate, compression tractions are observed in the contacting interface. On the other hand, if the surfaces tend to distance one from another, tensile tractions arises in the contact surface, contradicting the condition expressed in (2.64). Therefore, the solution of the contact problem also demands the verification of the contact tractions in each of the iterations. If tensile tractions are observed, the respective contact constraint must be deactivated.

According to the conventional FEM numerical approach, the contact detection, the constraint terms computation and the traction verification are tasks assigned to contact elements. The general algorithm for the numerical solution of the contact problem is presented in Figure 3.10.

For 2D models, if contact is controlled at nodes in the solid's boundary, the contact element is referred to as node-to-segment contact element. Otherwise, if the contact control is performed at a line segment (plane element's side) the contact element is referred to as segment-to-segment contact element. The formulations of node-to-segment and segment-to-segment contact elements for the GFEM are discussed in the following sections.



```
CONTACT PROBLEM (STATIC) - GENERAL ALGORITHM
$\Delta t$ = total_time / number_of_steps
t = 0
verify contact elements and activate the ones which are already in contact at initial time
while t < final_time:
    t=t+$\Delta t$
    converged = false
    while converged is false:
        compute residual and tangent stiffness matrix for the current time
        apply active contact constraints in the system of equation
        solve $K_t$ (d) $\Delta d$ = -R to find $\Delta d$
        d=d+$\Delta d$
        verify convergence criterion;
            if OK then:
                verify if new contact elements have penetrated;
                    if  yes:
                        mark such elements as active
                    if not (i.e.,no new active elements):
                        verify if traction occured in active elements;
                            if yes:
                                mark such elements as inactive;
                            if not (i.e., none of  the active contact
                                    elements present tensile traction):
                                converged = true
        verify the  number of iterations;
            if it exceeds the maximum defined value,
            stop the solution  and inform the process has diverged
end of the incremental iterative solution  method
```

Figure 3.10. General algorithm for the contact problem numerical solution.

a) The node-to-segment contact element

As already commented, the node-to-segment contact elements are attached to nodes positioned in the boundary of the solid, being used to detect the contact, compute the constraints to avoid penetration and evaluate the traction condition.

Let one suppose that a given node-to-segment contact element present a normal projection over a curved target, as depicted in Figure 3.11.

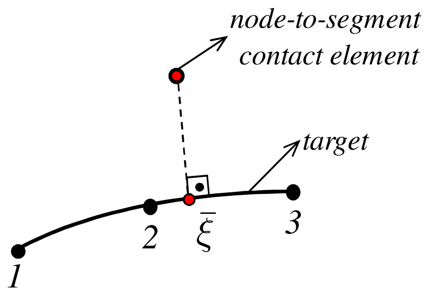


Figure 3.11. Node-to-segment contact element – normal projection over a target.

The position of the point corresponding to the contact pair is defined by solving the nonlinear equation arising from equation (3.105), in which

$$\begin{aligned} x\left(\xi\right) &= X\left(\xi\right)+u\left(\xi\right)=\sum_{i=1}^{n_{nd}} X_i N_i\left(\xi\right)+\sum_{i=1}^{n_{enr}} u_i^{enr} N_i^{enr}\left(\xi\right), \\ y\left(\xi\right) &= Y\left(\xi\right)+v\left(\xi\right)=\sum_{i=1}^{n_{nd}} Y_i N_i\left(\xi\right)+\sum_{i=1}^{n_{enr}} v_i^{enr} N_i^{enr}\left(\xi\right). \end{aligned} \tag{3.106}$$

In (3.106) n_{nd} is the number of nodes in the target segment whose partition of unity (L2, L3 or L4) contains the point ξ and n_{enr} is the number of nodal enriched shape functions containing that point. If no enrichment is applied to the nodes defining the target, the expression (3.106) fall within the FEM interpolation.

In general, the enrichment functions are defined in the global XY system of reference, such that the enrichment shape function is indicated as

$$N^{enr}(\xi)=N(\xi)\,L\left(X(\xi),Y(\xi)\right). \tag{3.107}$$

In this case, for computing the derivatives required by (3.105), one has to employ the chain rule, as follows

$$\frac{\partial N^{enr}}{\partial \xi}=\frac{\partial N}{\partial \xi}L+N\left[\frac{\partial L}{\partial X}\frac{\partial X}{\partial \xi}+\frac{\partial L}{\partial Y}\frac{\partial Y}{\partial \xi}\right]. \tag{3.108}$$

Once the solution of equation (3.105) is found, one can compute the gap value between these points, by means of the relation (2.60). First, the unit normal outward vector can be computed by the cross product between the unit tangent vector $\boldsymbol{a}_t$ and the unit vector $\boldsymbol{-e}_3$ (Figure 2.1), such that

$$\boldsymbol{n}=\boldsymbol{a}_t\times-\boldsymbol{e}_3=\left(x'\left(\bar{\xi}\right),y'\left(\bar{\xi}\right),0\right)\times\left(0,0,-1\right)=\left(-y'\left(\bar{\xi}\right),x'\left(\bar{\xi}\right),0\right). \tag{3.109}$$

With this relation, the gap can be computed as

$$g_{ns}=\left(\boldsymbol{x}_{nx}-\boldsymbol{x}\left(\bar{\xi}\right)\right)\bullet\boldsymbol{n}=\left(x_{ns}-x\left(\bar{\xi}\right)\right)\left(-y'\left(\bar{\xi}\right)\right)+\left(y_{ns}-y\left(\bar{\xi}\right)\right)x'\left(\bar{\xi}\right), \tag{3.110}$$

in which $\boldsymbol{x}_{ns}=\left(x_{ns},y_{ns}\right)$ is the current position of the node-to-segment contact element.

If the gap is evaluated as a negative value, the contact element must be activated to avoid penetration, providing constraint terms to be applied over the system of equations to enforce the contact impenetrability condition.

As already mentioned, the constraint terms derive from the contact virtual work, through the Lagrange multipliers method or the penalty method. In either of formulations, the expression of the virtual gap term (δg) is necessary, which is given by

$$\delta g = \left[\delta \boldsymbol{u}_{ns} - \delta \boldsymbol{u}\left(\bar{\xi}\right) - \left.\frac{\partial \boldsymbol{x}}{\partial \xi}\right|_{\bar{\xi}} \delta\xi\right] \bullet \boldsymbol{n} + \left[\boldsymbol{x}_{ns} - \boldsymbol{x}\left(\bar{\xi}\right)\right] \bullet \delta\boldsymbol{n}\,. \tag{3.111}$$

Wriggers (2006) discussed several aspects regarding the terms of (3.111), concluding that some of them vanishes, resulting in

$$\delta g = \left[\delta \boldsymbol{u}_{ns} - \delta \boldsymbol{u}\left(\bar{\xi}\right)\right] \bullet \boldsymbol{n}\,. \tag{3.112}$$

The node-to-segment contact element formulation is hereby applied only for small displacement problems. The key assumption in this case is that the changes in the normal vector are negligible (Laursen (2002)). Moreover, one considers that the contacting point at the target/obstacle remains unchanged.

The node-to-segment element contact condition can be enforced by a collocation method. Accordingly, the integral terms in (2.68) and (2.69) are substituted by the direct enforcement of the contact condition at the nodes where penetration is detected.

At the contact node, the PU function has unitary value. Due to this fact, the current position of this node is given by

$$\begin{aligned} x_{ns} &= X_{ns} + \sum_{i=1}^{n_{nsx}} u_i^{enr} N_i^{enr} = X_{ns} + \sum_{i=1}^{n_{nsx}} u_i^{enr} L_i\left(X_{ns}, Y_{ns}\right), \\ y_{ns} &= Y_{ns} + \sum_{i=1}^{n_{nsy}} v_i^{enr} N_i^{enr} = Y_{ns} + \sum_{i=1}^{n_{nsy}} v_i^{enr} L_i\left(X_{ns}, Y_{ns}\right). \end{aligned} \tag{3.113}$$

The Lagrange multiplier λ_{ns} constitutes a new variable associated to the node. In the GFEM, the nodal enrichment feature is applicable for this DOF as well, as indicated below

$$\lambda_{ns} = \sum_{i=1}^{n_{lm}} \lambda_i^{enr} L_i^{enr}\left(X_{ns}, Y_{ns}\right). \tag{3.114}$$

Aiming the FEM/GFEM approach, it is more convenient to adopt the following:

$$\boldsymbol{x}_{ns} = \begin{Bmatrix} x_{ns} \\ y_{ns} \end{Bmatrix} = \begin{Bmatrix} X_{ns} \\ Y_{ns} \end{Bmatrix} + \underbrace{\boldsymbol{B}_{\boldsymbol{ns}}}_{(2\text{x}n_{ns})} \underbrace{\boldsymbol{d}_{\boldsymbol{ns}}}_{(n_{ns}\text{x}1)} \;, \tag{3.115}$$

$$\boldsymbol{\delta u}_{ns} = \begin{Bmatrix} \delta u_{ns} \\ \delta v_{ns} \end{Bmatrix} = \underbrace{\boldsymbol{B}_{\boldsymbol{ns}}}_{(2\text{x}n_{ns})} \underbrace{\boldsymbol{\delta d}_{\boldsymbol{ns}}}_{(n_{ns}\text{x}1)} \;, \tag{3.116}$$

$$\bar{\boldsymbol{x}} = \begin{Bmatrix} \bar{x} \\ \bar{y} \end{Bmatrix} = \begin{Bmatrix} X\left(\bar{\xi}\right) \\ Y\left(\bar{\xi}\right) \end{Bmatrix} + \underbrace{\boldsymbol{B}_{\boldsymbol{tgt}}\left(\bar{\xi}\right)}_{(2\text{x}n_{tgt})} \underbrace{\boldsymbol{d}_{\boldsymbol{tgt}}}_{(n_{tgt}\text{x}1)} \;, \tag{3.117}$$

$$\boldsymbol{\delta\bar{u}} = \begin{Bmatrix} \delta u\left(\bar{\xi}\right) \\ \delta v\left(\bar{\xi}\right) \end{Bmatrix} = \underbrace{\boldsymbol{B}_{tgt}\left(\bar{\xi}\right)}_{(2\text{x}n_{tgt})} \underbrace{\boldsymbol{\delta d_{tgt}}}_{(n_{tgt}\text{x}1)} \ , \tag{3.118}$$

$$\bar{\boldsymbol{n}} = \begin{Bmatrix} n_x\left(\bar{\xi}\right) \\ n_y\left(\bar{\xi}\right) \end{Bmatrix}, \tag{3.119}$$

$$\lambda = \underbrace{\boldsymbol{M_{ns}}}_{(1\text{x}n_{lm})} \underbrace{\boldsymbol{\Lambda_{ns}}}_{(n_{ml}\text{x}1)} \ , \tag{3.120}$$

$$\delta\lambda = \underbrace{\boldsymbol{M_{ns}}}_{(1\text{x}n_{lm})} \underbrace{\boldsymbol{\delta\Lambda_{ns}}}_{(n_{lm}\text{x}1)} . \tag{3.121}$$

In the relations above, n_{ns} is the number of displacement GDOF considering both directions at the node-to-segment contact element. Likewise, n_{tgt} is the number of displacement GDOF associated to the nodes of the target. Finally, n_{ml} is the number of GDOF associated to the Lagrange multiplier attached to the node-to-segment contact element.

Both displacement and Lagrange multipliers GDOF are then encompassed in a single vector

$$\underbrace{\boldsymbol{d_{ns}^{Lag}}}_{(n_{ns}^{Lag}\text{x}1)} = \begin{Bmatrix} \boldsymbol{d_{ns}} \\ \boldsymbol{d_{tgt}} \\ \boldsymbol{\Lambda_{ns}} \end{Bmatrix}, \tag{3.122}$$

in which $n_{ns}^{Lag} = n_{ns} + n_{tgt} + n_{lm}$.

According to this notation, the gap value is given by

$$g = g_0 + \underbrace{\boldsymbol{n}^T}_{(1\text{x}2)} \underbrace{\boldsymbol{B_{ns}^{Lag}}}_{(2\text{x}n_{ns}^{Lag})} \underbrace{\boldsymbol{d_{ns}^{Lag}}}_{(n_{ns}^{Lag}\text{x}1)} \ , \tag{3.123}$$

$$g_0 = \left(X_{ns} - X\left(\bar{\xi}\right)\right)\bar{n}_x + \left(Y_{ns} - Y\left(\bar{\xi}\right)\right)\bar{n}_y, \tag{3.124}$$

$$\boldsymbol{B_{ns}^{Lag}} = \left[\boldsymbol{B}_{ns} \quad -\boldsymbol{B_{tgt}} \quad \boldsymbol{0_{Bns}}\right], \tag{3.125}$$

where $\boldsymbol{0_{Bns}}$ is a $\left(2\text{x}n_{lm}\right)$ matrix full of zeros.

The virtual gap is given by

$$\delta g = \underbrace{\boldsymbol{n}^T}_{(1\text{x}2)} \underbrace{\boldsymbol{B_{ns}^{Lag}}}_{(2\text{x}n_{ns}^{Lag})} \underbrace{\boldsymbol{\delta d_{ns}^{Lag}}}_{(n_{ns}^{Lag}\text{x}1)} \ . \tag{3.126}$$

Now, the Lagrange multiplier is written in terms of the vector $\boldsymbol{d_{ns}^{Lag}}$ as

$$\lambda = \underbrace{\boldsymbol{M}_{\boldsymbol{ns}}^{\boldsymbol{Lag}}}_{(1\mathrm{x}n_{ns}^{Lag})} \underbrace{\boldsymbol{d}_{\boldsymbol{ns}}^{\boldsymbol{Lag}}}_{(n_{ns}^{Lag}\mathrm{x}1)} = \left[\boldsymbol{0}_{\boldsymbol{Mns}}\ \boldsymbol{\Lambda}_{\boldsymbol{ns}}\right] \boldsymbol{d}_{\boldsymbol{ns}}^{\boldsymbol{Lag}}, \tag{3.127}$$

in which $\boldsymbol{0}_{\boldsymbol{Mns}}$ is a $\left(1\mathrm{x}\left(n_{ns} + n_{tgt}\right)\right)$ matrix full of zeros.

A similar relation is used to write the virtual Lagrange multiplier.

Employing the matrices just shown, the contact virtual work from (2.68) results as

$$\delta W^{\text{contact},Lag} = \boldsymbol{\delta d}_{\boldsymbol{ns}}^{\boldsymbol{Lag}}\, \boldsymbol{K}_{\boldsymbol{ns}}^{\boldsymbol{Lag}}\, \boldsymbol{d}_{\boldsymbol{ns}}^{\boldsymbol{Lag}} + \boldsymbol{f}_{\boldsymbol{ns}}^{\boldsymbol{Lag}}, \tag{3.128}$$

$$\underbrace{\boldsymbol{K}_{\boldsymbol{ns}}^{\boldsymbol{Lag}}}_{(n_{ns}^{Lag}\mathrm{x}n_{ns}^{Lag})} = \underbrace{\left(\boldsymbol{M}_{\boldsymbol{ns}}^{\boldsymbol{Lag}}\right)^T}_{(n_{ns}^{Lag}\mathrm{x}1)} \underbrace{\boldsymbol{n}^T}_{(1\mathrm{x}2)} \underbrace{\boldsymbol{B}_{\boldsymbol{ns}}^{\boldsymbol{Lag}}}_{(2\mathrm{x}n_{ns}^{Lag})} + \underbrace{\left(\boldsymbol{B}_{\boldsymbol{ns}}^{\boldsymbol{Lag}}\right)^T}_{(n_{ns}^{Lag}\mathrm{x}2)} \underbrace{\boldsymbol{n}}_{(2\mathrm{x}1)} \underbrace{\boldsymbol{M}_{\boldsymbol{ns}}^{\boldsymbol{Lag}}}_{(1\mathrm{x}n_{ns}^{Lag})}, \tag{3.129}$$

$$\underbrace{\boldsymbol{f}_{\boldsymbol{ns}}^{\boldsymbol{Lag}}}_{(n_{ns}^{Lag}\mathrm{x}1)} = g_0 \underbrace{\left(\boldsymbol{M}_{\boldsymbol{ns}}^{\boldsymbol{Lag}}\right)^T}_{(n_{ns}^{Lag}\mathrm{x}1)}. \tag{3.130}$$

Once $\boldsymbol{K}_{\boldsymbol{ns}}^{\boldsymbol{Lag}}$ is linear, the same matrix can be used within a Newton-Rapshon incremental process. In this case, for a deformed configuration of the element given by $\bar{\boldsymbol{d}}$, one has to consider the contribution $\boldsymbol{K}_{\boldsymbol{ns}}^{\boldsymbol{Lag}}\bar{\boldsymbol{d}}$ in the residual vector.

For the penalty based formulation, according to (2.69), one defines the vector of elemental unknowns

$$\underbrace{\boldsymbol{d}_{\boldsymbol{ns}}^{\boldsymbol{pen}}}_{(n_{ns}^{pen}\mathrm{x}1)} = \begin{Bmatrix} \boldsymbol{d}_{\boldsymbol{ns}} \\ \boldsymbol{d}_{\boldsymbol{tgt}} \end{Bmatrix}, \tag{3.131}$$

in which $n_{ns}^{pen} = n_{ns} + n_{tgt}$.

Performing similar derivations, one obtains

$$\delta W^{\text{contact},pen} = \boldsymbol{\delta d}_{\boldsymbol{ns}}^{\boldsymbol{pen}}\, \boldsymbol{K}_{\boldsymbol{ns}}^{\boldsymbol{pen}}\, \boldsymbol{d}_{\boldsymbol{ns}}^{\boldsymbol{pen}} + \boldsymbol{f}_{\boldsymbol{ns}}^{\boldsymbol{pen}}, \tag{3.132}$$

$$\underbrace{\boldsymbol{K}_{\boldsymbol{ns}}^{\boldsymbol{pen}}}_{(n_{ns}^{pen}\mathrm{x}n_{ns}^{pen})} = \varepsilon_N \underbrace{\left(\boldsymbol{B}_{\boldsymbol{ns}}^{\boldsymbol{pen}}\right)^T}_{(n_{ns}^{pen}\mathrm{x}2)} \underbrace{\boldsymbol{n}}_{(2\mathrm{x}1)} \underbrace{\boldsymbol{n}^T}_{(1\mathrm{x}2)} \underbrace{\boldsymbol{B}_{\boldsymbol{ns}}^{\boldsymbol{pen}}}_{(2\mathrm{x}n_{ns}^{pen})}, \tag{3.133}$$

$$\underbrace{\boldsymbol{f}_{\boldsymbol{ns}}^{\boldsymbol{pen}}}_{(n_{ns}^{pen}\mathrm{x}1)} = \varepsilon_N\, g_0 \underbrace{\left(\boldsymbol{B}_{\boldsymbol{ns}}^{\boldsymbol{pen}}\right)^T}_{(n_{ns}^{pen}\mathrm{x}2)} \underbrace{\boldsymbol{n}}_{(2\mathrm{x}1)}. \tag{3.134}$$

For the Lagrange multipliers based node-to-segment contact element formulation, the contact traction estimative is given by (3.120), considering the values of the Lagrange multipliers GDOF resulting from the system of equations.

Regarding the penalty based formulation, a rough estimative of the contact traction is given by the product of the penalty parameter with the computed gap at the contacting points. It is important to remember that the penalty based strategy does not guarantee that the gap is null, thus this product does not result as zero.

One has to bear in mind that the presented formulation is directly applicable to the Signorini problem, considered as a particular case in which the target is substituted by a rigid obstacle, with no DOF associated to it. The required particularization in the formulation is straightforward and is not shown here. In the developed computational code, the contact with a rigid obstacle is automatically considered depending if the node-to-segment contact element contact a target or a rigid obstacle.

b) The segment-to-segment contact element

The segment-to-segment contact element is attached to the side of plane elements. The derived formulation for the GFEM is based in the mortar contact elements applied for the FEM, see Wriggers (2006) and Fisher and Wriggers (2005).

Since the contact element is not associated to a single point, the normal gap and the traction vary along the line segment of the element. The element geometry and its displacement interpolation are defined using one-dimensional partitions of unity (L2, L3 or L4).

According to Fisher and Wriggers (2005), the mortar method is a discretization technique based on the Lagrange multipliers method, through which the contact constraints are fulfilled in a weak sense.

In the mortar method, the Lagrange multipliers field is interpolated using a one-dimensional PU associated to the contact element. For the FEM, it is written as

$$\lambda\left(\zeta\right)=\sum_{i=1}^{n_{nd}}\lambda_i M_i\left(\zeta\right). \tag{3.135}$$

As for the GFEM, the Lagrange multipliers field can be enriched, resulting that

$$\lambda\left(\zeta\right)=\sum_{i=1}^{n_{nd}}\sum_{j=1}^{n_{ndi}^{enr}}\lambda_j^i M_i\left(\zeta\right)L_j\left(X\left(\zeta\right),Y\left(\zeta\right)\right). \tag{3.136}$$

In the present text we employ the same nomenclature adopted in Wriggers (2006). In accordance with it, the contact element is referred to as *non-mortar* segment, and the target is referred to as *mortar* segment. In Figure 3.12, both elements are depicted for the first degree interpolation (straight element) and higher order interpolation (curved element).

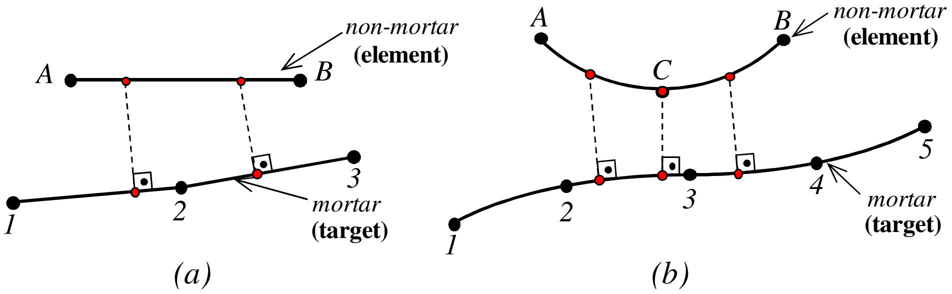


Figure 3.12. Mortar contact element and its target for first degree interpolation (a) and higher order interpolation (b).

Similar interpolations, including enrichment function, can be applied to the displacement fields along the mortar contact element and its targets, i.e.,

$$
\begin{aligned}
x(\zeta) &= X(\zeta) + u(\zeta) = \sum_{i=1}^{n_{nd}} X_i N_i(\zeta) + \sum_{i=1}^{n_{enr}} u_i^{enr} N_i^{enr}(\zeta),\\
y(\zeta) &= Y(\zeta) + v(\zeta) = \sum_{i=1}^{n_{nd}} Y_i N_i(\zeta) + \sum_{i=1}^{n_{enr}} v_i^{enr} N_i^{enr}(\zeta).
\end{aligned}
\tag{3.137}
$$

The integrations arising in the mortar formulation are hereby performed numerically by the Gauss-Legendre quadrature. In this context, the gap function is computed in the integration points distributed along the contact element (red circles in the non-mortar segment), as shown in Figure 3.12. From this figure, one can notice that for the same contact element, different targets can be associated. For each integration point ζ_i at the non-mortar segment, a corresponding contacting point ξ_i is defined at the mortar segments, in accordance with (3.105).

The normal gap value for the segment-to-segment contact element is computed by the numerical integration as well,

$$
g_{ss} = \sum_{i=1}^{n_{GL}} g(\zeta_i, \xi_i) w_i = \sum_{i=1}^{n_{GL}} \left[\underbrace{\big(x(\zeta_i), y(\zeta_i)\big)}_{non-mortar\ segment} - \underbrace{\big(x(\xi_i), y(\xi_i)\big)}_{mortar\,segment} \right] \bullet \underbrace{n_i(\xi_i)}_{mortar\,segment} \; w_i\,, \tag{3.138}
$$

in which n_{GL} is the number of integration points of the Gauss-Legendre quadrature, and w_i are the weights according to this quadrature rule. Obviously, for the GFEM, one considers the enriched displacement field interpolations for computing the current position in both non-mortar and mortar segments considered in (3.138).

If $g_{ss} \leq 0$ in a time instant, the associated contact element is set as active, and the resulting system of equations receives contributions of terms associated to the contact virtual work of such mortar element.

Aiming at a formulation applicable for problems presenting displacements of any magnitude, the linearization of the contact virtual work employing a Newton-Raphson's method is necessary. For the Lagrange multipliers based mortar contact element, the virtual work contribution results as

$$D\delta W^{contact,Lag} = \int_{\Gamma_c^e} \left[D\lambda \;\, \delta g + \lambda \;\, D\delta g + \delta\lambda \;\, Dg \right] d\Gamma . \tag{3.139}$$

In (3.139) the term $D\delta\lambda \;\, g$ is omitted since $D\delta\lambda$ results null.

The linearization of the gap present a similar structure with the virtual gap from (3.112), resulting as

$$Dg = \left[\Delta\boldsymbol{u}\left(\zeta_i\right) - \Delta\boldsymbol{u}\left(\xi_i\right) \right] \bullet \boldsymbol{n}\left(\xi_i\right) . \tag{3.140}$$

The linearization of the Lagrange multipliers field is straightforward, resulting in

$$D\lambda = \sum_{i=1}^{n_{nd}} \sum_{j=1}^{n_{ndi}^{enr}} \Delta\lambda_j^i M_i\left(\zeta\right) L_j\left(X\left(\zeta\right), Y\left(\zeta\right) \right) . \tag{3.141}$$

Certainly the most complex derivation is associated to the linearization of the virtual gap ($D\delta g$), which is discussed in details in Wriggers (2006). After a long sequence of derivations, one obtains that

$$\begin{aligned} D\delta g = & -\left(\frac{\partial \boldsymbol{\delta u}\left(\xi\right)}{\partial \xi} D\xi + \frac{\partial \Delta\boldsymbol{u}\left(\xi\right)}{\partial \xi} \delta\xi + D\xi \frac{\partial^2 \boldsymbol{x}\left(\xi\right)}{\partial \xi^2} \delta\xi \right) \bullet \boldsymbol{n}\left(\xi\right) \\ & + \frac{g}{\left\| \boldsymbol{a}\left(\xi\right) \right\|^2} \left(\frac{\partial \boldsymbol{\delta u}\left(\xi\right)}{\partial \xi} + \frac{\partial^2 \boldsymbol{x}\left(\xi\right)}{\partial \xi^2} \delta\xi \right) \bullet \left[\boldsymbol{n}\left(\xi\right) \otimes \boldsymbol{n}\left(\xi\right) \right] \left(\frac{\partial \Delta\boldsymbol{u}\left(\xi\right)}{\partial \xi} + \frac{\partial^2 \boldsymbol{x}\left(\xi\right)}{\partial \xi^2} D\xi \right) , \end{aligned} \tag{3.142}$$

$$\delta\xi = \frac{1}{\left\| \boldsymbol{a}\left(\xi\right) \right\|^2} \left[\left(\boldsymbol{\delta u}\left(\zeta\right) - \boldsymbol{\delta u}\left(\xi\right) \right) \bullet \frac{\partial \boldsymbol{x}\left(\xi\right)}{\partial \xi} + g\, \boldsymbol{n}\left(\xi\right) \bullet \frac{\partial \boldsymbol{\delta u}\left(\xi\right)}{\partial \xi} \right] , \tag{3.143}$$

$$D\xi = \frac{1}{\left\| \boldsymbol{a}\left(\xi\right) \right\|^2} \left[\left(\Delta\boldsymbol{u}\left(\zeta\right) - \Delta\boldsymbol{u}\left(\xi\right) \right) \bullet \frac{\partial \boldsymbol{x}\left(\xi\right)}{\partial \xi} + g\, \boldsymbol{n}\left(\xi\right) \bullet \frac{\partial \Delta\boldsymbol{u}\left(\xi\right)}{\partial \xi} \right] , \tag{3.144}$$

in which $\boldsymbol{a}\left(\xi\right)$ is the tangent vector at the contacting point located in the mortar segment (target) and $\boldsymbol{n}\left(\xi\right)$ is the unit normal vector at the same point. One must observe that the curvature of the target at the contact pair point also arises in the linearization of the virtual gap (second derivative with respect to ξ). For the FEM, this term vanishes for straight elements. For the GFEM this is not true, since the enrichment over the displacement field can result in curved geometry of the deformed configuration, even for first degree approximation elements.

For the GFEM, all these terms must be computed considering the enriched interpolations, what turns the computations yet more cumbersome.

The computation of the tangent matrix and residual vector associated to mortar contact element is performed by means of a numerical integration, similar to the one presented in (3.138). It results as

$$\boldsymbol{K}_{ss} = \sum_{i=1}^{n_{GL}} \boldsymbol{K}_{css}\left(\xi_i, \zeta_i\right) J\left(\zeta_i\right) w_i , \tag{3.145}$$

$$\boldsymbol{f}_{ss} = \sum_{i=1}^{n_{GL}} \boldsymbol{f}_{css}\left(\xi_i, \zeta_i\right) J\left(\zeta_i\right) w_i , \tag{3.146}$$

$$J\left(\zeta_i\right) = \sqrt{\left.\frac{\partial X\left(\zeta\right)}{\partial \zeta}\right|_{\zeta_i} + \left.\frac{\partial Y\left(\zeta\right)}{\partial \zeta}\right|_{\zeta_i}} \ . \tag{3.147}$$

The computation of the tangent matrix and the residual vector contributions $\boldsymbol{K}_{css}\left(\xi_i, \zeta_i\right)$ and $\boldsymbol{f}_{css}\left(\xi_i, \zeta_i\right)$ is more efficiently performed both for the FEM and GFEM using matrix terms. In order to do so, one defines the vector

$$\underbrace{\boldsymbol{d}_{ss}^{Lag}}_{(n_{ss}^{Lag}\text{x}1)} = \begin{Bmatrix} \boldsymbol{d}_{ss} \\ \boldsymbol{d}_{tgt} \\ \boldsymbol{\Lambda}_{ss} \end{Bmatrix} , \tag{3.148}$$

in which n_{ss}^{Lag} is the number of GDOF associated to the generalized mortar contact element, resulting from the sum of the number of displacement GDOF associated to the non-mortar segment (n_{ss}), the number of GDOF associated to the mortar segment (n_{tgt}), and the number of GDOF associated to the Lagrange multipliers field (n_{lm}).

An equivalent vector $\boldsymbol{\delta d}_{ss}^{Lag}$ is defined for the virtual GDOF.

Analogously to the node-to-segment derivations, one defines a matrix $\boldsymbol{B}_{\boldsymbol{i}}^{Lag} = \boldsymbol{B}^{Lag}\left(\zeta_i, \xi_i\right)$ such that

$$\left(\boldsymbol{u}\left(\zeta_i\right) - \boldsymbol{u}\left(\xi_i\right)\right) = \underbrace{\boldsymbol{B}_{\boldsymbol{i}}^{Lag}}_{(2\text{x}n_{ss}^{Lag})} \underbrace{\boldsymbol{d}_{ss}^{Lag}}_{(n_{ss}^{Lag}\text{x}1)} , \tag{3.149}$$

$$\boldsymbol{B}_{i}^{Lag} = \left[\underbrace{\boldsymbol{B}\left(\zeta_{i}\right)}_{(2xn_{ss})} \quad \underbrace{\boldsymbol{B}\left(\xi_{i}\right)}_{(2xn_{tgt})} \quad \underbrace{\boldsymbol{0}_{Bss}}_{(2xn_{lm})}\right]. \tag{3.150}$$

One also defines a matrix $\boldsymbol{B}_{i,\xi}^{Lag} = \boldsymbol{B}_{,\xi}^{\;Lag}\left(\xi_{i}\right)$ used to compute the derivative of $\Delta\boldsymbol{u}\left(\xi\right)$ with respect to ξ resulting as

$$\boldsymbol{B}_{i,\xi}^{Lag} = \left[\underbrace{\boldsymbol{0}_{nm}}_{(2xn_{ss})} \quad \underbrace{\boldsymbol{B}_{,\xi}\left(\xi_{i}\right)}_{(2xn_{tgt})} \quad \underbrace{\boldsymbol{0}_{Bss}}_{(2xn_{lm})}\right], \tag{3.151}$$

in which $\boldsymbol{B}_{,\xi}\left(\xi_{i}\right)$ is a matrix containing the derivatives of the enriched interpolation functions of $\boldsymbol{B}\left(\xi_{i}\right)$ with respect to ξ.

Similarly to the node-to-segment contact element, the Lagrange multiplier is written in terms of the vector $\boldsymbol{d}_{ss}^{Lag}$ as

$$\lambda = \underbrace{\boldsymbol{M}_{i}^{Lag}}_{(1\text{x}n_{ss}^{Lag})} \underbrace{\boldsymbol{d}_{ss}^{Lag}}_{(n_{ss}^{Lag}\text{x}1)} = \left[\boldsymbol{0}_{Mss} \; \boldsymbol{\Lambda}_{ss}\right]\boldsymbol{d}_{ss}^{Lag}, \tag{3.152}$$

in which $\boldsymbol{0}_{Mss}$ is a $\left(1\text{x}\left(n_{ss} + n_{tgt}\right)\right)$ matrix full of zeros.

One employs the equivalent notation for the following entities: $\boldsymbol{n}_{i} = \boldsymbol{n}\left(\xi_{i}\right)$, $\boldsymbol{a}_{i} = \boldsymbol{a}\left(\xi_{i}\right)$, $\lambda_{i} = \lambda\left(\zeta_{i}\right)$, $g_{i} = g\left(\zeta_{i}, \xi_{i}\right)$ and $\boldsymbol{c}_{i} = \boldsymbol{c}\left(\xi_{i}\right)$ (curvature of the mortar segment, given by the second derivative with respect to ξ).

Having defined all these matrices, the tangent matrix for the Lagrange multipliers formulation is computed by

$$\begin{aligned}
\boldsymbol{K}_{css}^{Lag}\left(\zeta_{i}, \xi_{i}\right) = {} & \left(\boldsymbol{B}_{i}^{Lag}\right)^{T} \boldsymbol{n}_{i}\boldsymbol{M}_{i}^{Lag} + \left(\boldsymbol{M}_{i}^{Lag}\right)^{T} \boldsymbol{n}_{i}^{\;T}\boldsymbol{B}_{i}^{Lag} \\
& -\Bigg[\left(\boldsymbol{B}_{i,\xi}^{Lag}\right)^{T} \boldsymbol{n}_{i}\,\boldsymbol{a}_{i}^{\;T}\boldsymbol{B}_{i}^{Lag} + \left(\boldsymbol{B}_{i}^{Lag}\right)^{T} \boldsymbol{a}_{i}\boldsymbol{n}_{i}^{\;T}\boldsymbol{B}_{i,\xi}^{Lag} \\
& +g_{i}\left(\boldsymbol{B}_{i,\xi}^{Lag}\right)^{T} \boldsymbol{n}_{i}\boldsymbol{n}_{i}^{\;T}\,\boldsymbol{B}_{i,\xi}^{Lag} + \frac{\boldsymbol{c}_{i}^{\;T}\boldsymbol{n}_{i}}{\left\|\boldsymbol{a}_{i}\right\|^{2}}\left(\boldsymbol{B}_{i,\xi}^{Lag}\right)^{T} \boldsymbol{a}_{i}\boldsymbol{a}_{i}^{\;T}\,\boldsymbol{B}_{i,\xi}^{Lag}\Bigg]\frac{\lambda_{i}}{\left\|\boldsymbol{a}_{i}\right\|^{2} - g_{i}\boldsymbol{c}_{i}^{\;T}\boldsymbol{n}_{i}}.
\end{aligned} \tag{3.153}$$

Equivalently, the residual vector contact contribution is computed by

$$\boldsymbol{f}_{css}^{Lag}\left(\zeta_{i}, \xi_{i}\right) = \left[\left(\boldsymbol{B}_{i}^{Lag}\right)^{T} \boldsymbol{n}_{i}\boldsymbol{M}_{i}^{Lag} + \left(\boldsymbol{M}_{i}^{Lag}\right)^{T} \boldsymbol{n}_{i}^{\;T}\boldsymbol{B}_{i}^{Lag}\right]\bar{\boldsymbol{d}}_{ss}^{Lag}, \tag{3.154}$$

in which $\bar{\boldsymbol{d}}_{ss}^{Lag}$ is a vector containing the current value of the GDOF associated to the element.

Even though the method was initially conceived for the Lagrange multipliers, one can extend the method's fundamental ideas to derive a penalty based mortar contact element (in the present work, a generalized penalty based mortar contact element). For this case, the vector of generalized unknowns is

$$\underbrace{\boldsymbol{d}_{ss}^{pen}}_{(n_{ss}^{pen}\text{x}1)} = \begin{Bmatrix} \boldsymbol{d}_{ss} \\ \boldsymbol{d}_{tgt} \end{Bmatrix}, \tag{3.155}$$

Performing equivalent derivations one obtains

$$\begin{aligned}\boldsymbol{K}_{css}^{pen}\left(\zeta_i,\xi_i\right) = \varepsilon_N \Bigg[& \left(\boldsymbol{B}_i^{pen}\right)^T \boldsymbol{n}_i \boldsymbol{n}_i^T \boldsymbol{B}_i^{pen} + \\ & -\Bigg[\left(\boldsymbol{B}_{i,\xi}^{pen}\right)^T \boldsymbol{n}_i\, \boldsymbol{a}_i^T \boldsymbol{B}_i^{pen} + \left(\boldsymbol{B}_i^{pen}\right)^T \boldsymbol{a}_i \boldsymbol{n}_i^T\, \boldsymbol{B}_{i,\xi}^{pen} \\ & +g_i \left(\boldsymbol{B}_{i,\xi}^{pen}\right)^T \boldsymbol{n}_i \boldsymbol{n}_i^T\, \boldsymbol{B}_{i,\xi}^{pen} + \frac{\boldsymbol{c}_i^T \boldsymbol{n}_i}{\left\|\boldsymbol{a}_i\right\|^2}\left(\boldsymbol{B}_{i,\xi}^{pen}\right)^T \boldsymbol{a}_i \boldsymbol{a}_i^T\, \boldsymbol{B}_{i,\xi}^{pen}\Bigg] \frac{g_i}{\left\|\boldsymbol{a}_i\right\|^2 - g_i \boldsymbol{c}_i^T \boldsymbol{n}_i}\Bigg].\end{aligned} \tag{3.156}$$

Equivalently, the residual vector contact contribution is computed by

$$\boldsymbol{f}_{css}^{pen}\left(\zeta_i,\xi_i\right) = \varepsilon_N \left[\left(\boldsymbol{B}_i^{pen}\right)^T \boldsymbol{n}_i \boldsymbol{n}_i^T \boldsymbol{B}_i^{pen}\right] \bar{\boldsymbol{d}}_{ss}^{pen}. \tag{3.157}$$

Regarding the traction condition evaluation, the same strategy applied for the node-to-segment contact element, both for Lagrange multipliers and penalty based formulation, can be adapted for the segment-to-segment contact element.

For the Lagrange multipliers, the traction can be evaluated by the numerical integration

$$t_{ss}^{Lag} = \sum_{i=1}^{n_{GL}} \lambda\left(\zeta_i\right) J\left(\zeta_i\right) w_i. \tag{3.158}$$

For the penalty, a rough estimative for the mean traction over the contact element is given by

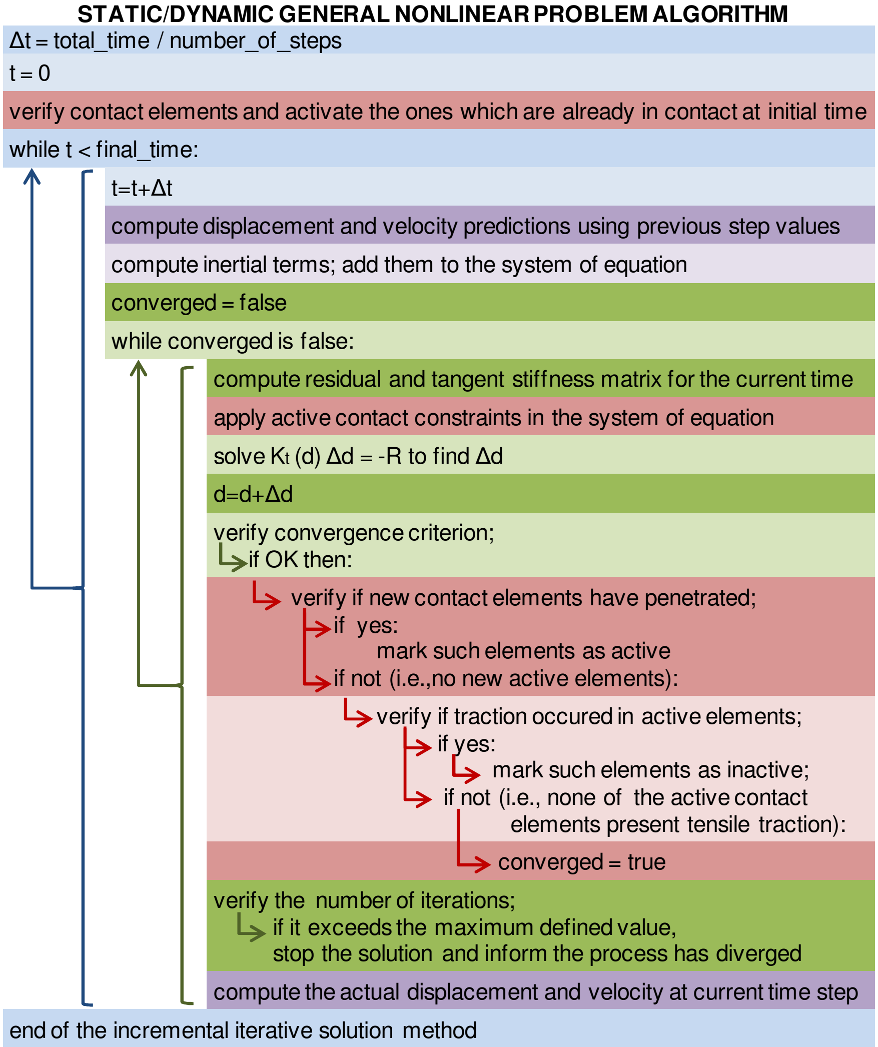
$$t_{ss}^{pen} = \varepsilon_N \sum_{i=1}^{n_{GL}} g\left(\zeta_i,\xi_i\right) J\left(\zeta_i\right) w_i. \tag{3.159}$$

3.5. The general nonlinear analysis framework

After the static and dynamic analyses algorithm and all nonlinear phenomena solution scheme have been discussed separately, it is possible to define a general purpose analyses algorithm, according to which one can solve linear and nonlinear static and dynamics analyses. Such scheme is presented in Figure 3.13.

According to this algorithm, the lines in purple color are referent to the dynamical effects. The tasks included in these lines must be performed only if one intends to consider the inertial effects in the analysis. By ignoring these commands, one falls into a static analysis. It is important to notice that such dynamical analyses apply both for linear or nonlinear analyses. In the first case, the Newton-Raphson method converges after a single iteration for each time step performed.

The magenta color lines are associated with contact problems, and apply only if contact elements, targets and/or obstacles are defined for the problem. Like it happens for the dynamical tasks, these lines' tasks must be executed only if a contact analyses is intended to be performed.



```
STATIC/DYNAMIC GENERAL NONLINEAR PROBLEM ALGORITHM
Δt = total_time / number_of_steps
t = 0
verify contact elements and activate the ones which are already in contact at initial time
while t < final_time:
    t=t+Δt
    compute displacement and velocity predictions using previous step values
    compute inertial terms; add them to the system of equation
    converged = false
    while converged is false:
        compute residual and tangent stiffness matrix for the current time
        apply active contact constraints in the system of equation
        solve Kt (d) Δd = -R to find Δd
        d=d+Δd
        verify convergence criterion;
        ↳if OK then:
            ↳ verify if new contact elements have penetrated;
                ↳if yes:
                    mark such elements as active
                ↳if not (i.e.,no new active elements):
                    ↳verify if traction occured in active elements;
                        ↳if yes:
                            ↳ mark such elements as inactive;
                        ↳ if not (i.e., none of the active contact
                            elements present tensile traction):
                            ↳ converged = true
        verify the number of iterations;
        ↳ if it exceeds the maximum defined value,
          stop the solution and inform the process has diverged
    compute the actual displacement and velocity at current time step
end of the incremental iterative solution method
```

Figure 3.13. Static/dynamic general nonlinear problem algorithm. The purple lines must be performed only for dynamical analyses; the magenta lines must be performed only for contact analysis.

The general algorithm presented in Figure 3.13 was implemented in the developed computational framework, more specifically in the structural problem class, to be discussed in the next chapter. Only with an exception of the linear elastic static problems, presented in Chapter 5, all the problems presented in the present thesis were solved by means of such algorithm.

4 - Computational Implementation

The present chapter is devoted to describe the general structure of the developed computational framework. Its implementation represents one of the contributions of the present research, mainly due to the fact that the developed Object-Oriented (OO) design presents original issues related to the GFEM, which allow one to efficiently deal with the method's characteristics.

The choice of the Object-Oriented Programming (OOP) for developing the code comes from several advantages one finds in such paradigm. These advantages are discussed in Section 4.1, in which one introduces the OOP particularities. In the same section, the general characteristics of the adopted programming language are briefly presented, aiming to allow the reader to understand possible notations and data structures employed in the description of the developed framework. The arguments presented in this sub-section must be understood not as a programming language vindication, but as a justification of the technical choice made by the code developer.

Once the OOP essentials are presented, the general class structure usually adopted for developing Finite Element Analysis frameworks is presented. Then, departing from this generic class structure, the specific data structure demanded to efficiently deal with the GFEM issues is presented. Within this initial description, one focuses in the linear analysis framework, which constitutes the basis for developing the nonlinear analysis functionalities.

Once the computational framework basis is stated, one indicates the demanded modification and inclusions in order to introduce nonlinear analysis capabilities in the computational program.

4.1. The Object-Oriented Programming (OOP)

In general terms, a paradigm is a pattern for guiding a specific activity. When related to the programming activity, the paradigm can be understood as a set of concepts and programming practices for guiding the design and implementation of a computational code. Obviously, each paradigm is conceived aiming specific objectives, presenting advantageous characteristics and also some possible disadvantages. According to Tucker and Noonan

(2008), in the past decades, four different programming paradigms were conceived: Imperative, Object-Oriented, Functional and Logic programming.

The imperative programming paradigm is the oldest of them, being grounded in the classical computational 'von Neumann-Eckert' model, as indicated in Tucker and Noonan (2008). The paradigm approach is based in the definition of a set of variables and procedures aiming a specific objective. The program, in general lines, is a list of tasks to be executed. This approach does not exclude the use of flow control structures, which are used for defining paths to be followed by the commands defined in the code. Fortran and C ANSI are examples of programming languages which support the imperative programming paradigm.

Due to historical aspects, the imperative programming was broadly applied for developing scientific and engineering computational applications, and presents several advantages in this situation. On the other hand, along the time, the demand of features for such applications changed, and the OOP emerged as an alternative for developing this kind of application.

The Object-Oriented Programming (OOP) is based in a model in which the tasks are performed by defining a set of objects interacting one to each other, interchanging messages in order to perform the tasks requested by the software user. Such computational objects must be understood as abstractions representing real world objects or theoretical entities.

The OOP provides a logical framework in which the data manipulation is restricted to the module in which the data is defined. One consequence of such approach is that a detailed knowledge about the module internal implementation is no longer necessary to external actors. In this case, only the definition of the module public methods interface is sufficient for using it.

According to Tucker and Noonan (2008), an important difference between the imperative programming and the OOP is that the Object-Oriented (OO) data structures are less passive than the ones defined in the imperative paradigm, since the message passing among objects turn such entities active in the computational framework. Smalltalk, C++ and Python are some languages that support the OOP paradigm.

It is important to mention that a given language is not necessarily attached to a unique paradigm. In fact, some programming languages are designed to support more than one programming paradigm, as for instance C++, which supports both the imperative and the OO programming. These programming languages are also called multi-paradigm programming languages.

On what follows, a brief discussion on the OOP characteristics is presented.

4.1.1. The OOP paradigm essentials

The OOP is a paradigm in which one creates abstractions of real world objects or theoretical entities, aiming to represent them within a computational model. These abstractions are defined by creating a description of the data structures to represent such entities. Such description is named class, consisting in a set of variables definitions (attributes) associated procedures (methods). This concept of associating the attributes and methods within a single structure is referred in the OOP as encapsulation.

Once a class is defined, it is possible to create instances according to the class definition, generating objects. Metaphorically explaining, the class may be viewed as the blueprint, while an object is a product created according to it.

Within the OO philosophy, all tasks related to a given object must be performed following the related class description. This means that changing the value of an object's attribute or performing computations using its values is an exclusive competence of the class. Moreover, the internal attributes are protected from actions of other classes' entities, according to the concept of information hiding, which arises in the OOP paradigm.

In fact, the internal details of the class methods are not known by other classes, which can ask tasks for its objects by calling their methods. In this context, an important issue is the definition of the class interface, which describes how one must call the methods of the class to ask specific tasks.

This strategy provides modularity to the code, once the class depends only on itself to perform tasks strictly associated to its instances. The interest of achieving modularity is that if the classes are adequately designed, it is possible to work separately in one class without depending in the implementation details of other classes. Of course, the other classes' interface is important, such that one can access their methods.

Several advantages arise from the modularity. Probably one of the most remarkable is that it allows one to easily implement new functionalities in the code, demanding at most little changes in the rest of the code. This character is especially important in a context in which a group of different developers work over the same computational code, which is also known as collaborative programming.

Within this framework, the computational program can be viewed as a collection of objects, acting in accordance with its respective class definition, which, by it turns, is designed to perform specific actions. In several situations, a given object needs to require actions or information to other objects, which is done by sending messages to them. The characteristics of the relation among different objects take a relevant position in the OOP paradigm.

The simplest relation among objects is the association, in which one object simply uses other object to perform a task. In other cases, the relation among objects goes deeper, such that an object has other object as an attribute, resulting in a relation named aggregation. Moreover, if the ‘owned’ object is indispensable for the existence of the object that ‘owns’ it, an especial case of aggregation is configured, which is named composition.

Going further in the analysis of the objects characteristics, it is possible to notice that, likewise in the real world, some objects are special types of others, a fact that can be efficiently explored in the OOP design. In this case, a class can be represented in the computational model as a special case of another class, inheriting a part of its data structure.

This situation is known in the OOP paradigm as inheritance. In this case, the resulting class, referred as subclass, is defined by changing and extending attributes and methods of the other class, which is named super-class. The inheritance is one of the fundamental concepts of the OOP, and provides code reuse, improving the efficiency of the computational implementation process.

Yet regarding the inheritance feature, in some cases several classes share similar characteristics, but their similar counterpart does not constitute a conventional class in the strict sense, since it is not possible to instantiate an object of such generic object. This super-class, from which one cannot generate objects, is just a definition of a data structure to be extended by the subclasses, constituting an abstract class.

It is also common in OO codes that distinct classes share methods to perform similar tasks with different internal implementation, i.e., they share methods with equal name and similar signature, but that are essentially different in their internal details. The related classes constitute a polymorphic set of classes, which allows one to call generic signature methods regardless of which of the classes it belongs.

In this situation, the computational code can be implemented such that it allows one to call a given object’s method without taking into account for which of these classes the object belongs. The polymorphism is another essential feature of the OOP, providing flexibility for the resulting code.

As one can observe, the OOP paradigm constitute a considerable complex methodology to write a computational code, presenting a higher level of abstraction in comparison to the imperative programming paradigm. In spite of this, the resulting code encompasses several advantages if it is properly designed. Specifically for the FEM codes, Cross et al (1999) addresses several advantages to be achieved from the OOP paradigm.

In fact, the OOP demands remarkable time investments in the code design in order to achieve such goals. Developing an efficient OO general class design for a specific kind of application is an object of study known as Design Patterns.

The main objective of the present chapter is to present the conceived design for implementing a GFEM analysis framework according to the OOP paradigm. Even though the OO design is not attached to a specific programming language, a brief discussion of the used language allows one to explain the code details in a simpler manner.

In the present work, the adopted programming language to develop the computational code is Python. The next section is devoted to introduce some particularities of the Python programming language.

4.1.2. A brief presentation of the adopted programming language

Python is a high-level programming language conceived in the late 1980's by Guido von Rossum. It is a multi-paradigm programming language, supporting both the Imperative and Object-Oriented Programming paradigms discussed previously. It is a script language, which means that the resulting code is interpreted during run-time and not compiled, as required by FORTRAN or C languages, for instance. It also presents an interactive mode, which allows one to type commands and instantly receive their results.

According to Langtangen (2008), the higher abstraction level inserted in scripting can make programming more convenient since "… scripts are often considerably faster to develop than the corresponding programs in traditional languages such as FORTRAN, C or C++, or Java, and the code is normally much shorter". The same author also advocates that "Python stands out as the language of choice for scripting in computational science because of its very clean syntax, rich modularization features, good support for numerical computing, and rapidly growing popularity".

Actually, scripting languages allow for connecting different applications, as scripts are efficient in receiving inputs and formatting them for generating outputs to another application. This explains the observed better interaction achieved when using an interpreter instead of a compiler. In fact, these are extremely attractive and desirable characteristics for scientific computational purposes. Such use of Python is applied, for instance, by Layman, Missoum and Geest (2010). In fact Python is nowadays being used by many Finite Element Packages such as Abaqus, as reported by Kuutti and Kolari (2012).

Among several advantages, probably the one most relevant for defining Python as the programming language to develop the GFEM computational code is related to its dynamic nature. First of all, Python is a dynamic typing language. This means that a variable can change its type in run-time, depending on the attributed value.

Python also presents automatic memory management, providing an efficient way to develop efficient codes. Memory management is critical task in computational applications, demanding great efforts in order to avoid problems like computational performance degradation, memory fault and stack overflow. An additional remarkable advantage of Python is its multi-platform feature, meaning that a given code can run on different Operating System, such as GNU/Linux, Windows and Mac/OS.

Furthermore, one of the greatest interests in Python is related to its native support for several different types of collections, fitting exactly with the GFEM nature, in which the number of degrees of freedom and enrichment functions changes considerably in different situations.

The simplest, but yet powerful type of collection in Python is the tuple. A tuple is syntactically defined by means of parenthesis, and allows grouping any quantity of variables and objects, regardless of their type. A tuple can also contain other tuples. The tuples' items can be referenced by means of a numerical index, starting from zero, according to the sequence in which such items were defined in the tuple. In spite of these advantageous characteristics, the tuple data structure its static, i.e., once the tuple is defined, their values cannot be changed.

The list is another collection natively support in Python, presenting a dynamic nature, in the sense that new items can be inserted in the list. Likewise, items also can be removed from it. In addition to this, the list data type is provided with several methods that allows one to easily perform operations over it, like concatenations, sorting and many others. The list is defined in Python syntax by using square brackets. This syntax is used in the present chapter in order to explain the data structure of the conceived GFEM OO framework.

A third powerful collection data structure is available in Python, being named dictionary. In Python, a dictionary is defined syntactically by means of curly brackets. Differently from the tuple and list, in the dictionary, the reference to one of its elements is not performed by means of the item's index. In the dictionary, each item is attached to another object, named its 'key', that must be unique in the dictionary, such that one can access a given item by means of this key. This key can even be an object defined by the user, with the only restriction that the key cannot be a mutable variable. As it happens for the list, several native methods to efficiently access the dictionary items are available in the language.

Taking into account the GFEM characteristics presented in the previous chapter, it turns clear that the just mentioned characteristics are strongly favorable for programming the method. This does not mean at all that any other programming language cannot be employed for developing a GFEM computational code. However, we believe that these arguments suffice for justifying the author's technical choice of using Python for developing the code. Moreover, another strong argument for using a script language for developing the GFEM framework is presented in Section 4.3.

Despite its flexibility and other additional advantages not mentioned herein, in general, script languages present less efficiency in performing computation, when compared to compiled languages. It is worth mentioning that the computational performance is considered a major requirement for numerical applications.

Regarding this issue, Python presents a powerful set of available numerical support libraries. These libraries are in general created from well consolidated codes written in high performance programming languages, like Fortran and C, and allows one to call their routines from Python in a direct fashion.

For instance, the library NumPy (The SciPy Community (2011)), is in practice a standard library for performing matrix computations in Python. It is not unusual achieving speed ups of tens or hundreds for a computation performed using a NumPy subroutine, when compared with the same computation performed in pure Python.

Moreover, additional numerical applications can be found in other libraries, as SciPy (The SciPy community (2010)). For instance, the sparse system of equation resulting from the FEM and GFEM models, in the present work, were efficiently assembled and solved by means of the subroutines available in the module 'sparse' from SciPy. Essentially, it uses the UMFPACK solver, written in C language (Davis (2004a), Davis (2004b), Davis and Duff (1997) and Davis and Duff (1999)).

Regarding the results analysis, it is important also to cite matplotlib (Hunter (2007)), a Python 2D plotting library providing tools for generating high quality graphics. It is worth to mention that all the result graphics and the displacement, stress and strain fields color-maps used in the present work were generated by means of matplotlib functionalities.

In what follows, we advance in the description of the developed code structure.

4.2. The traditional OO approach for the FEM programming

The OOP application for finite element analysis codes started to be presented in the technical literature in the beginning of the 1990s, as for instance, in Forde, Foschi and Stiemer (1990) and Alves Filho and Devloo (1991). In the first paper, the basic concepts of the OOP were introduced, followed by a description of an OO design for the FEM. The described OO code is also compared to an equivalent procedural program, identifying the advantages of the OO approach.

Such works were followed by Zimmermann, Dubois-Pérelin and Bomme (1992) and Dubois-Pélerin, Zimmermann and Bomme (1992), in which the paradigm is described in more details using Smalltalk programming language. Next, Dubois-Pélerin and Zimmermann (1993) have published a paper describing an efficient OO implementation using C++. The use of advanced features of C++ to implement finite element classes is presented later in Bittencourt (2000).

Many other papers on the use of the OOP for numerical analysis were published, like, for instance, Mackie (2000), and the subject has been intensively discussed along the last decades. Skipping from a detailed review of the OOP for FEM, which is not the focus of the present text, we just refer to Mackerle (2000), which lists hundreds of references on the subject, also including a list of OOP applications to the Boundary Element Method (BEM).

The application and detailed description of the OOP for other variants of non-conventional numerical methods is clearly limited in the available literature. For instance, among the works reviewed regarding exclusively to the GFEM, its implementation by OOP was found by the authors only in one dissertation, Pereira (2004), written in Portuguese, and in which one finds a description of one possible structure of classes composing this kind of computational framework. Regarding technical periodicals, the only reference addressing to a closer method to the GFEM, the eXtended Finite Element Method (XFEM), is presented by Bordas et al (2007).

The GFEM OO design conceived during the present research is based in many features of the traditional FEM OO design. Figure 4.1 depicts a simple generic class design for a finite element linear analysis code.

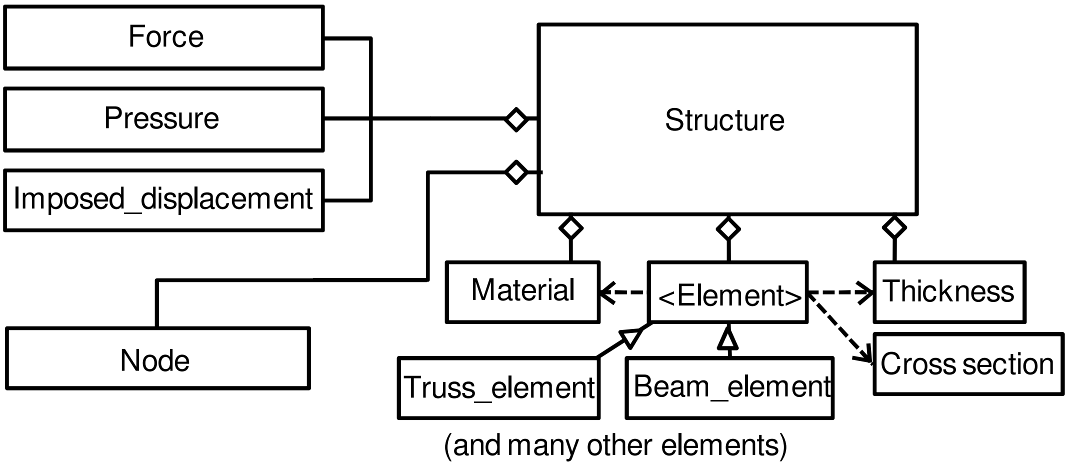


Figure 4.1. Generic class design for a finite element linear analysis code

In such scheme, the diamond symbol in the connecting lines indicates a composition, relation, while the triangle symbol indicates a relation of inheritance. It is worth noting that in the presented scheme, the generic element does not exist in practice, i.e., it is an abstract class, which is presented in Figure 4.1 by putting its name between angle brackets (<Element>).

Regarding the dashed line arrows that connect the element class to the ‘material’, ‘thickness’ and ‘cross section’ classes, they represent a composition by reference. This relation means that the thickness and cross sections instances are not instantiated inside element instances, but associated to them by means of a pointer.

In fact, one can notice that those classes are related to the structure class in a composition relation (diamond symbol). This kind of relation is useful when several objects share the same object (a given material, for instance), since only one instance of such class need to be instantiate. In this case, all the elements made out of such material just point to the same instance, instantiated in the structure class object.

Regarding to the generic FEM OO design of Figure 4.1, within such scheme, the node class is one of the main actors, representing the discrete points of the continuum at which the desired result is computed. Therefore, each node is represented by its Cartesian coordinates and must store the results achieved by means of the numerical method.

The continuum regions between the nodes constitute finite dimension elements, which define another fundamental class in the FEM/OOP approach. Beyond the set of nodes defining it, the elements instances must also hold information about the material and additional geometric characteristics, such as its thickness (plane and shell elements) or cross section data (beam and truss elements). As already mentioned, such characteristics are defined by means of additional classes as ‘thickness’/‘cross section’, which also constitute the data structure presented in Figure 4.1.

Also boundary conditions data are necessary to define the problem, such that classes to describe them are demanded. For solid mechanics problems using displacement based formulation, classes like ‘force’, ‘pressure’ and ‘imposed displacement’ are related to the boundary conditions.

This basic class framework design is completed by the ‘structure’ class. The structure is the main core of the data structure, and constitutes the interface for the rest of the code. In fact, it is used to define the finite element model characteristics and to manage the other classes’ instances. Nevertheless, depending on the desired usage for the code, a more general name for such class is ‘structural set’ or ‘structural problem’. This nomenclature makes sense if one thinks about problems treating a set of solids, as, for instance, the contact problems. Departing from the general OO design (Figure 4.1), the modifications demanded in order to comport the GFEM are presented in the following section.

4.3. An Object-Oriented class design for the GFEM

The major changes in the present FEM structure in order to support the GFEM are related to the node class. As already discussed, in the GFEM, the number of degrees of freedom (DOF) associated to a given node varies according to the number of enrichment functions applied to it. Furthermore, for all computations, each specific enrichment data must be directly associated to the new DOF.

In fact, it turns out that the enrichment function, which does not even exists for the FEM, is an important actor within this framework. Moreover, given its generic nature, such entity can be better described by means of defining an abstract <enrichment> class representing the basis for a polymorphic set of different enrichment classes.

By taking all these aspects into account, in the proposed OO design, a set of classes replaces the FEM node class, as indicated in Figure 4.2.

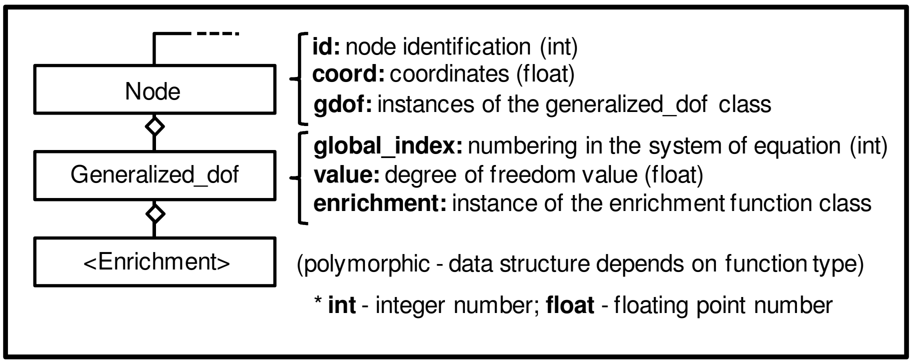

Node
Generalized_dof
<Enrichment>
id: node identification (int)
coord: coordinates (float)
gdof: instances of the generalized_dof class
global_index: numbering in the system of equation (int)
value: degree of freedom value (float)
enrichment: instance of the enrichment function class
(polymorphic - data structure depends on function type)
* **int** - integer number; **float** - floating point number

Figure 4.2. Classes for modeling the node and its enriched degrees of freedom in the GFEM.

In order to associate a new nodal degree of freedom with enrichment function instances, we propose the definition of a ‘generalized degree of freedom’ class. Regarding its functionality, this class basically holds: a) the scalar value of the degree of freedom, computed by means of the numerical method; b) its numbering in the system of equations (global_index); c) an instance of the ‘enrichment function’ class. Obviously, if no enrichment is applied, such instance must be null, resulting in a regular FEM degree of freedom.

Since the new data structure for the node class must be able to hold a variable number of generalized degrees of freedom, it demands the usage of a dynamic data structure. Here, we have used Python’s list to hold the nodal degrees of freedom. For other programming languages, a linked list must be implemented.

In addition to this variable number of DOF, the ‘node’ class also demands a data structure that is capable to hold any type of degree of freedom, such as displacement, temperature or magnetic field, for instance.

In order to do so, the proposed design uses a Python dictionary named ‘gdof’. The ‘gdof’ dictionary keys represent the type of DOF, while the related value is a Python list in which one can insert any number of ‘generalized degree of freedom’ (GDOF) instances, depending on the number and type of enrichment defined by the user. Such scheme is indicated in Figure 4.3.

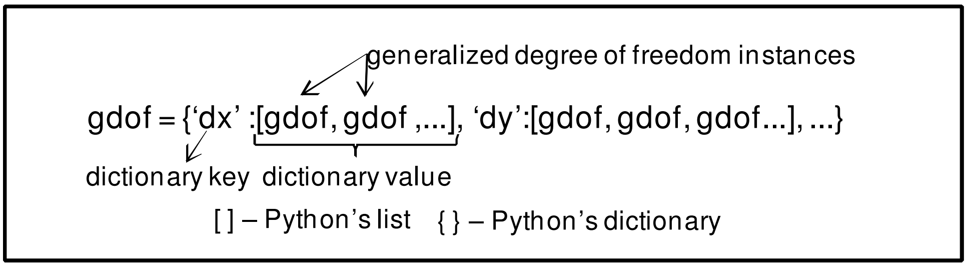

generalized degree of freedom instances
gdof = {‘dx’ :[gdof, gdof ,...], ‘dy’:[gdof, gdof, gdof...], ...}
dictionary key
dictionary value
[ ] – Python’s list
{ } – Python’s dictionary

Figure 4.3. The gdof node attribute, defined by means of a Python dictionary.

In Figure 4.3, ‘dx’ and ‘dy’, for instance, stand for ‘displacement in x direction’ and ‘displacement in y direction’, respectively. As already mentioned, this allows one to include any other type of DOF in such data structure. Specifically, we find out in our implementation that such tool is efficient for holding Lagrange multipliers variables used for imposing displacements efficiently in the GFEM. Such variables are also used in contact problems.

The proposed data structure allows that the generalized element’s stiffness matrix and load vector computations still being performed by means of the same strategy of the traditional FEM, as already discussed in the previous chapter. The enriched terms contributions is taken into account in the routine that computes the B matrix, in which it is verified whether the generalized degree of freedom is enriched or not. If none of the DOF of a given element is enriched, the FEM B matrix is obtained.

Finally, an important aspect towards a generic framework is the fact that the enrichment function is represented by a set of polymorphic classes. Such an abstract representation allows that virtually any type of function can be included in the code. This advantage is especially convenient for scripting languages, like Python, since one can include new enrichment function without needing to recompile the code.

Another important aspect focused in the proposed class design is related to the shape functions used as partitions of unity. As discussed previously, if one follows strictly the definition of partition of unity, only the linear (triangular) or bi-linear (quadrilaterals) Lagrangian shape functions can be used as PU for the GFEM. Even though the GFEM polynomial enrichment functions improve considerably the accuracy of the results, these linear PU are not efficient for describing curved geometries.

In this context, an aimed feature for a GFEM framework is the capability of using different order PU to describe the element's geometry (initial shape) and physical behavior (interpolated fields). In order to do so, the proposed OO design solution is based on the definition of a polymorphic class named 'Partition of Unity', defining generic partitions of unity for any desired domain (one-dimensional, two-dimensional, and so on).

Such a mixed approximation approach is performed by defining different PU instances inside the element data structure, as indicated in Figure 4.4. As it can be noticed, by adopting this design, both isoparametric FEM elements and curved geometry linear GFEM PU can be supported using the same code framework.

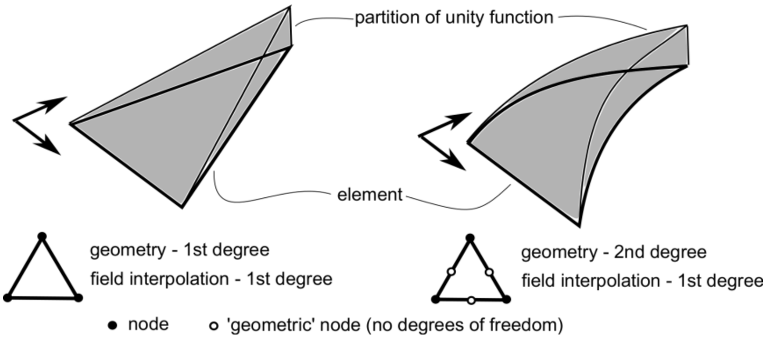

partition of unity function
element
geometry - 1st degree
field interpolation - 1st degree
geometry - 2nd degree
field interpolation - 1st degree
● node
○ 'geometric' node (no degrees of freedom)

Figure 4.4. Triangular elements using the traditional linear interpolation and an element mixing different approximation PU for describing the element's geometry and its structural behavior.

When considered altogether, the hereby proposed set of changes in the classical FEM OO class design allows that basically the same framework can be efficiently extended to the GFEM purposes with no loss for the generality of the original methods. In fact, the other classes presented in Figure 4.1 remain practically unchanged, requiring at most little changes in order to support both FEM and GFEM models. Figure 4.5 presents the complete proposed OO class design for the GFEM.

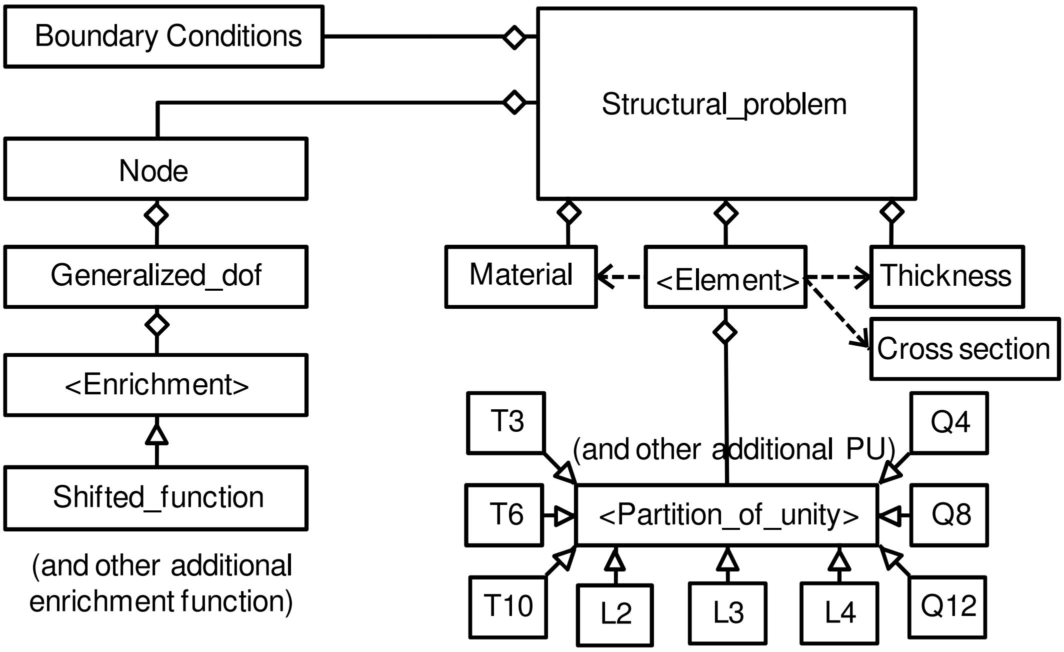


Figure 4.5. General representation for the proposed OO design for the GFEM.

As it can be noticed we also propose the union of the force, pressure and imposed displacement classes in a single class named ‘boundary condition’, searching to achieve a more generic representation of such data. This unification is a straightforward modification and present no major details to be described, but improves significantly the code generality, especially if one aims to employ the same framework not only for solid mechanics analysis code, but also to solve other types of partial differential equation problems.

4.3.1. Including the nonlinear analysis capabilities

The OO design presented in the last section can be used as the basis for developing a FEM/GFEM nonlinear analysis framework. As already discussed in the previous chapter, the adopted nonlinear analysis approach is mainly based on applying load increments along a period of time. Obviously the time demands a data structure to represent it within a nonlinear context. In fact, if the velocities observed in the model give rise to inertial forces that cannot be neglected, even linear analyses demand a control on the model’s characteristics attached to the time evolution.

The ‘time line’ class is conceived to provide capabilities related to time. Beyond a simple object to control the evolution of time, other additional features are included, like, for instance, capabilities to define time functions and to store the history of convergence parameters.

The control over the time evolution is simply performed by defining the total time of the analysis and the number of load steps in which one intends to discretize the time. The time evolution is considered by simply adding time step increments until the total time is reached, which configures the end of the analysis.

According to the current time value, functions resulting in scalar values can be defined, aiming to describe the variation of a given load during the solution process. These functions are here referred to as time functions. Moreover, aiming to yield a higher flexibility in the definition of time functions, the conceived data structure allows the user to define functions composed by other piecewise continuous functions, as for instance, the ones depicted in Figure 4.6.

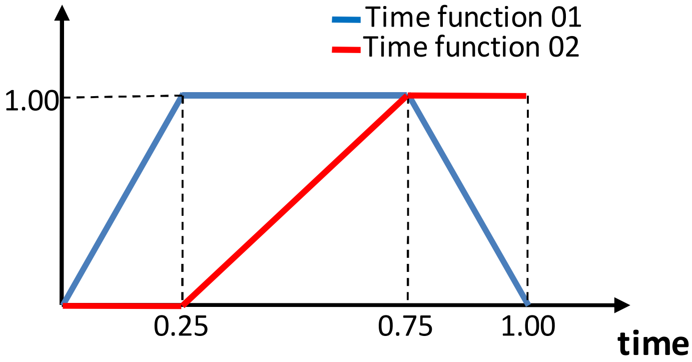


Figure 4.6. Example of time functions defined in a time_line object.

Specifically in the proposed implementation, the piecewise continuous time functions are defined by means of Python’s dictionary, in which the time intervals are determined by a string describing the time interval, and its associated value is the function in such interval. For instance, the time functions indicated in Figure 4.6 would be defined by creating the dictionaries shown in Figure 4.7.

Definition of Time function 01

```
tf1={} # tf1 - time function 01
tf1[“t < 0.25”] = “t/0.25”
tf1[“0.25 <= t <0.75” ] = “1.0”
tf1[“0.75 <= t <1.00”]  = “1.0-(t-0.75)/0.25”
```

Definition of Time function 02

```
tf2={} # tf2 - time function 02
tf2[“t < 0.25”] = “0.0”
tf2[“0.25 <= t < 0.75” ] = “(t-0.25)/0.5”
tf2[“0.75 <= t < 1.00”]  = “1.0”
```

Figure 4.7. Definition of time functions 01 and 02 using Python’s dictionary.

In Figure 4.7 'tf1' and 'tf2' are dictionaries representing the time functions. If a given key is true for the current time, the associated function is evaluated. These evaluations can be easily performed over the strings representing the time interval and time function by means of Python's 'eval()' command. Further details on this function are found in Rossum (2011).

The presented time functions can be applied to the boundary conditions, which are designed to be associated with these functions. For instance, the time functions presented in Figure 4.6 can be used for defining the illustrative example presented in Figure 4.8.

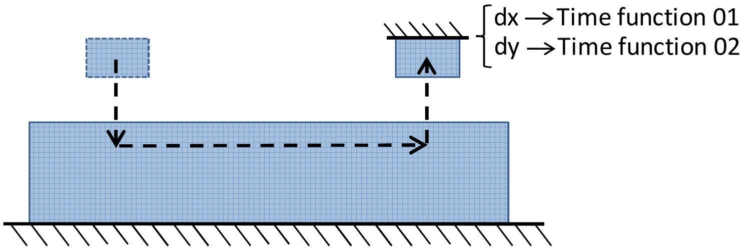


Figure 4.8. Example of the use of time function associated to displacement boundary conditions.

In this example, the proposed time functions are associated to displacement boundary conditions applied in one of the solids to make it move in the 2D space, sliding over the other solid. The resulting deformed configuration in some time instants are presented in Figure 4.9.

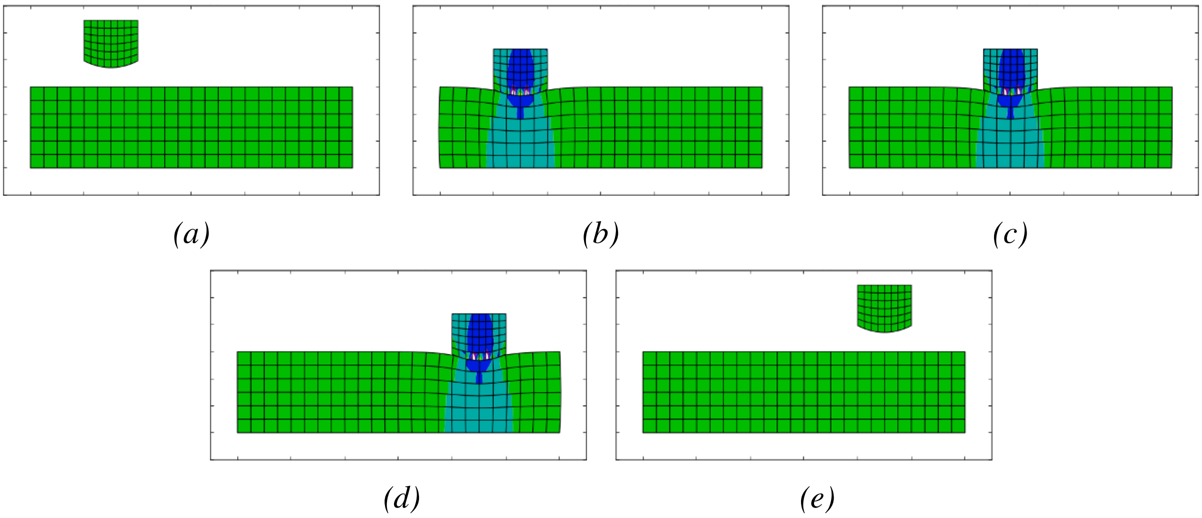


Figure 4.9. Deformed configuration for different time instants of the illustrative example indicated in Figure 4.8, for time instants 0.0 (a), 0.25 (b), 0.50 (c), 0.75 (d) and 1.00 (e).

Once the time line is defined in the structural problem object, some changes in the element and material classes are demanded in order to support both the kinematic and material nonlinear behavior.

Probably the nonlinear description of the solid's kinematic is one of the most straightforward adaptations in the linear analysis framework. These changes are mainly performed directly in the element class, in which the strain is computed, according to the adopted constitutive model. In this case, the nonlinear kinematic relation presented in Chapters 2 and 3 are directly implemented in this class.

On the other hand, the nonlinear material behavior demands major changes in the existing code. In fact, like different behavior materials are supported, the linear elastic material class must be extended to generate other material models. In the proposed implementation, the set of classes to represent the material models results as depicted in Figure 4.10.

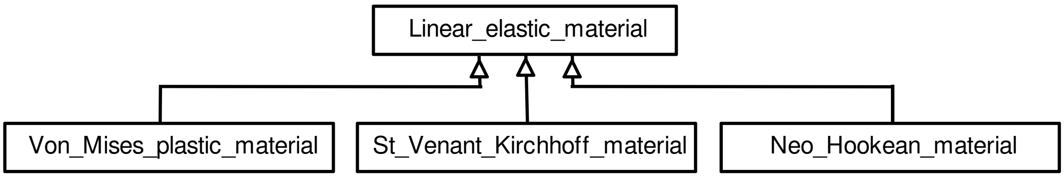


Figure 4.10. The material classes in the nonlinear analysis scheme.

Beyond the methods usually defined in a class to set and get object attributes, the main method in these material classes are related to the computing of the tangent constitutive tensor. In the case of elastoplastic models, the hardening law and its derivative computation are also provided by the respective material class.

In these cases, the computations also depend on the current strain state at a given point of the domain, including information on the evolution of other parameters. As already discussed in Chapter 3, these values are computed at integration points of each of the elements. This demands a data structure to hold such values, according to the theoretical model defined in Chapter 2.

These functionalities are provided by the ‘stress strain state’ class. For each integration point defined in an element, a ‘stress/strain state’ object is instantiated inside element objects, when the model is built. Along the solution process, these data is updated for each of the iterations. It is worth mentioning that this class is not exclusively employed by elasto-plastic models. In fact, it is also useful for conservative models, like for instance, the Neo-Hookean material, for which the tangent constitutive tensor depends on the current strain state.

Regarding the contact problem functionalities, new classes for defining contact elements and targets are necessary. Taking into account the 2D idealization adopted for the model, these targets are geometrically described by defining line segments.

The simplest kind of target is related to the Signorini problem, in which it is not deformable. Hereby those rigid targets are associated to a class named ‘obstacle’, defined by means of one-dimensional partition of unities, resulting in linear, quadratic and cubic interpolations geometries.

For the case of contact among deformable bodies, a similar class named ‘target’ is defined. Targets are also geometrically defined by line segments, being attached to plane elements side. Differently from the obstacle, the target instances move and change their shape during the solution process. Targets and obstacles perform similar tasks (similar signature methods) resulting in a polymorphic set of classes.

The contact phenomenon is basically considered by means of contact elements, which are attached to the deformable solid’s boundaries. Such contact element must perform the contact identification and compute the contact constraints, if it is activated in a given instant of the solution process.

The contact identification is based in the condition stated in (2.61). Following the numerical solution process discussed in Chapter 3, all elements are initially set as not active. These elements remain with this status until it detects penetration. This penetration is evaluated by computing the gap value in relation to the target and obstacle objects defined in a given model. In this situation, the contact element status is set as ‘activated’. While it is active, it returns constraints terms to be accounted in the system of equation, in order to avoid the penetration.

According to the numerical solution strategy presented in Chapter 3, the contact element is also responsible for evaluating the tractions at the position in which it is applied, which is necessary to verify condition (2.64). If tensile tractions are detected, the contact element status is set as ‘deactivated’.

Since in the conceived code both node-to-segment and segment-to-segment contact elements are defined, a set of two different classes are defined: ‘node-to-segment’ and ‘segment-to-segment’. The first is associated to nodes positioned in the boundary of the deformable body, while the second is associated to plane element’s side at such boundary.

Having briefly discussed the general aspects of the computational implementation, it is possible to affirm that the proposed framework supports linear and nonlinear, static and dynamic numerical analysis using both the FEM and the GFEM. Some of these analyses results are presented in the following two chapters.

5 - Numerical validation

The past chapters were devoted to describe the theoretical foundation, the numerical approach and the computational implementation for performing static and dynamic, linear and nonlinear solid analysis using the GFEM. Next, we present and analyze numerical examples, proposed for validating the implemented formulation, evaluating its efficiency, robustness and accuracy. Theoretical and numerical results, computed using the tradition FEM, are used in order to achieve such objective.

5.1. GFEM - enrichment accuracy (linear elasticity)

In order to evaluate the results achieved both for the polynomial and shifted enrichment functions defined in the previous chapters, a 24.0x6.0x1.0 (dimensionless) cantilever beam, subjected to a distributed load q=1.0 is stated, as indicated in Figure 5.1. Plane stress hypothesis is assumed. The load due to self weight is neglected.

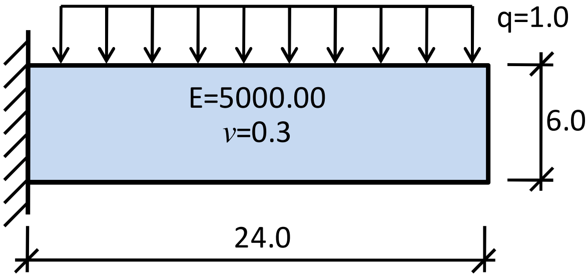
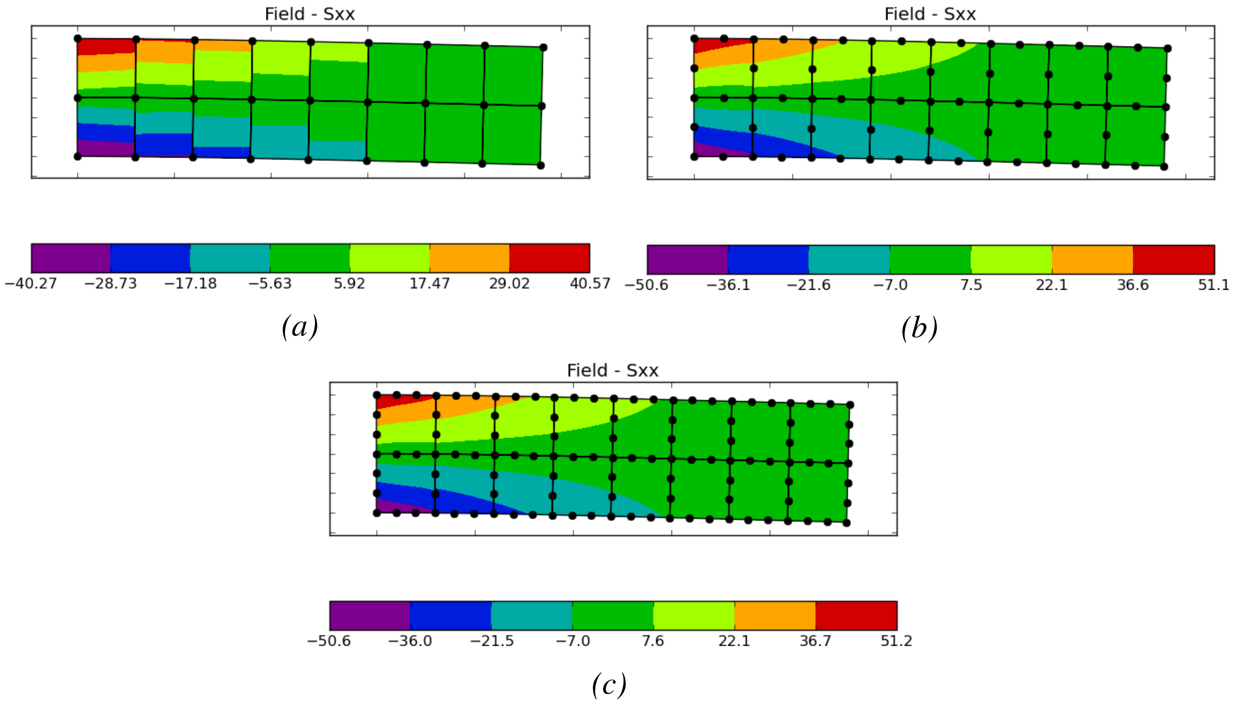


Figure 5.1. Cantilever beam scheme.

In order to verify the accuracy of the implemented conventional isoparametric finite element formulation, the structure is modeled by means of quadrilateral regular meshes containing 16 elements (bi-linear, bi-quadratic and bi-cubic). The horizontal stress component (S_{xx}) results are indicated in Figure 5.2.



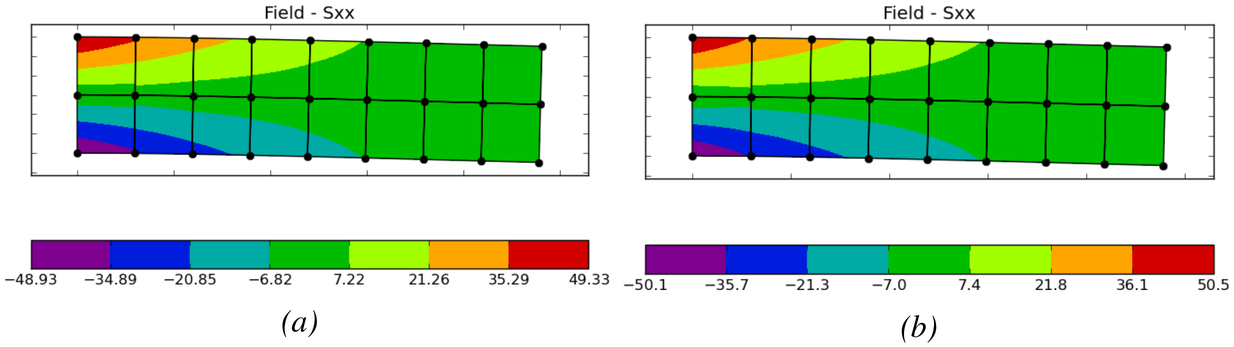
(a)

(b)

(c)

Figure 5.2. Horizontal stress (Sxx) results for bi-linear (a), bi-quadratic (b) and bi-cubic (c) isoparametric quadrilateral elements.

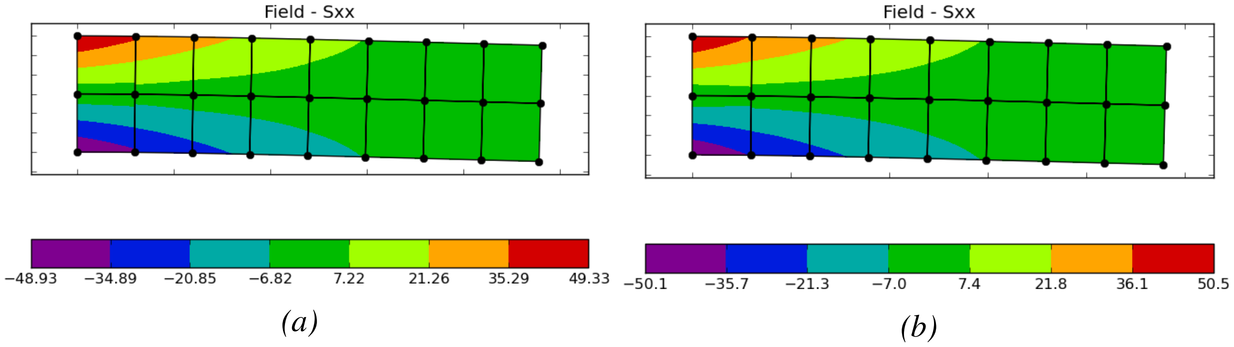
The GFEM results both for the polynomial and shifted enrichment are presented, respectively, in Figure 5.3 and Figure 5.4.



(a)

(b)

Figure 5.3. Horizontal stress component (Sxx) results for first (a) and second (degree) 'polynomial' enrichment.



(a)

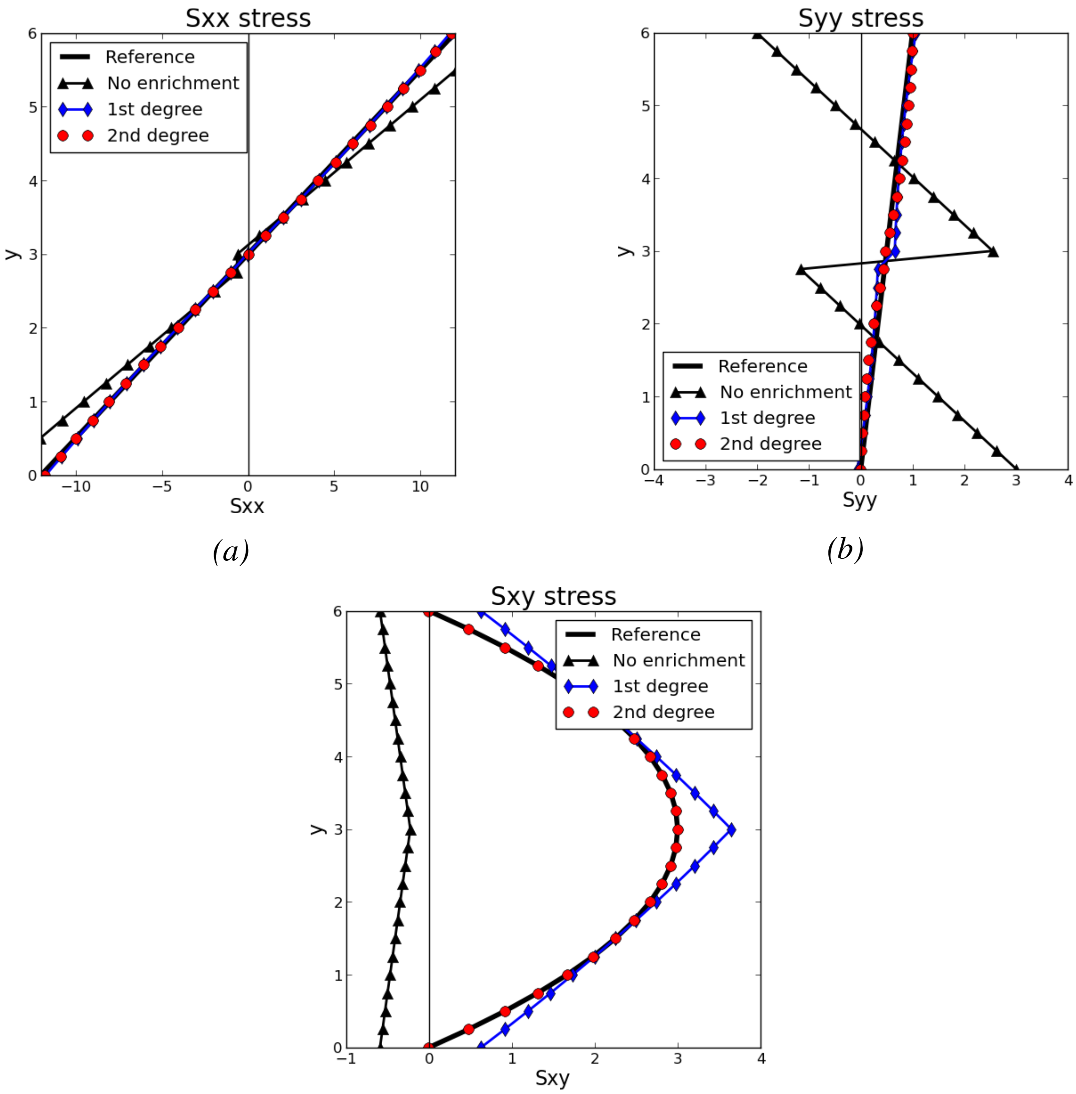
(b)

Figure 5.4. Horizontal stress component (Sxx) results for first (a) and second (degree) 'shifted' enrichment.

The results presented in Figure 5.3 and in Figure 5.4 were achieved by enriching all the cantilever nodes, except the ones in which the Dirichlet boundary conditions were applied (nodal enforcement of the displacement). The reason for not enriching them is commented at the end of the present example.

As it can be noticed, for the presented field, both polynomial and shifted enrichment result in very similar stress distribution. Also, little changes in the results are noticed in the field maps of the first and second degree enrichment.

In order to better evaluate the GFEM stress results, the stress distribution along the cross section position, at the middle of the cantilever span, is presented in Figure 5.5.



(a)

(b)

(c)

Figure 5.5. Horizontal 'Sxx' (a), vertical 'Syy'(b) and shear 'Sxy' stress components at the cross section positioned at the middle of the cantilever beam span.

The results presented in Figure 5.5, were achieved for a set of shifted enrichment function, both for the first and second degree enrichment. The reference was computed using the traditional Euler-Bernoulli beam theory. As it can be noticed, the second degree enrichment provides stress results very close to the theoretical reference. Such stress field distribution improvement is more noticeable for the vertical (Syy) and shear (Sxy) stress components.

Taking into account that the developed computational framework supports both FEM and GFEM analysis, the present example can be used to compare the traditional hierarchic (h) and polynomial (p) refinement of the FEM with the enrichment feature of the GFEM, here referred as selective (s) refinement. The convergence of the maximum vertical displacement of the cantilever beam for the 'h', 'p' and 's' refinements is presented in Figure 5.6.

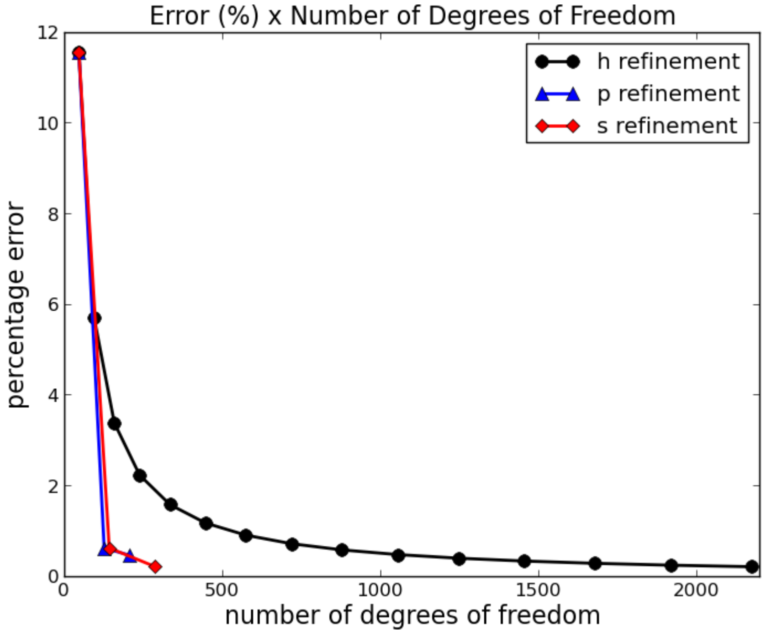


Figure 5.6. Maximum vertical displacement convergence for the different refinement strategies.

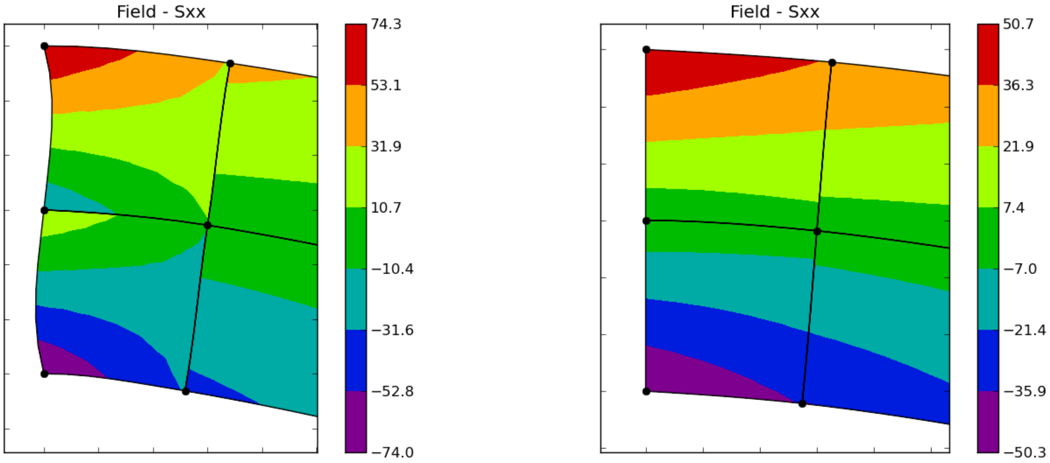
The selective 'refinement' presents a convergence rate very close to the tradition polynomial refinement. The hierarchic refinement demands considerable more degrees of freedom (dofs) in order to result in a similar percentage error value.

Finally, is important to comment the reason for not enriching the nodes in which the Dirichlet boundary conditions are applied.

As already discussed in the previous chapters, one of the difficulties faced in the GFEM is the enforcement of displacement (Dirichlet) boundary conditions. This occurs due to the fact that, differently of the displacement based FEM, the degrees of freedom in general do not represent directly the nodal displacements. On the contrary, the displacement of any point in the problem domain, for a generic enrichment, is obtained by combining several degrees of freedom resulting from the enrichment.

In fact, one of the advantages of the shifted function enrichment is preserving the original nodal displacement meaning of the original nodal degree of freedom, since such enrichment results as zero at the nodes. In spite of this fact, even this enrichment changes the element's original displacement field in the rest of its domain. Due to this reason, when a node at the boundary is enriched, the nodal enforcement of the displacement does not guarantee that this value is enforced in between the nodes.

This effect can be observed if one enriches the nodes at the clamped end of the cantilever proposed in the present example. The deformed shape at the clamped region allows one to verify that such displacement enforcement is not observed at the element's face, as depicted in Figure 5.7(a).



(a) *(b)*

Figure 5.7. Deformed configuration of the cantilever clamped end when enrichment is applied for the nodes in such boundary; nodal enforcement of displacement (a); displacement enforcement along the elements' side (b) (obs.: the deformed configuration in (a) and (b) is scaled by a factor of 10).

On the other hand, if one intends to enrich such nodes, the displacement enforcement must be performed along the whole element's side. Figure 5.7 (b) depicts the deformed configuration in such situation, demonstrating that this approach is able to enforce the displacement field along the element side.

5.2. Distortion and curvature sensibility (linear elasticity)

It is known that the FEM is sensible to distorted meshes. In the other hand, it is also known that the GFEM can circumvent such effects when polynomial enrichments are employed. In order to evaluate the GFEM model sensibility to mesh distortions, the same cantilever beam employed in example 5.1 is used to solve the problem proposed in Figure 5.8.

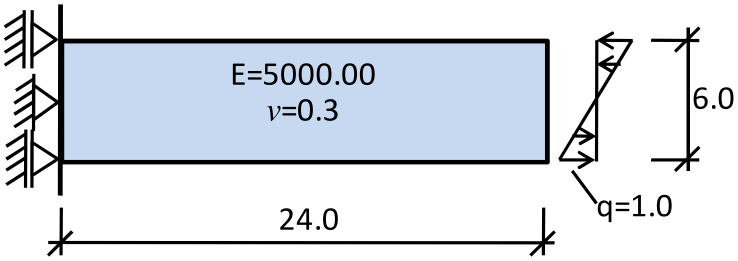


Figure 5.8. Cantilever beam subjected to a ‘pure’ moment load.

The FEM horizontal stress component results (Sxx) for different mesh distortions are depicted in Figure 5.9.

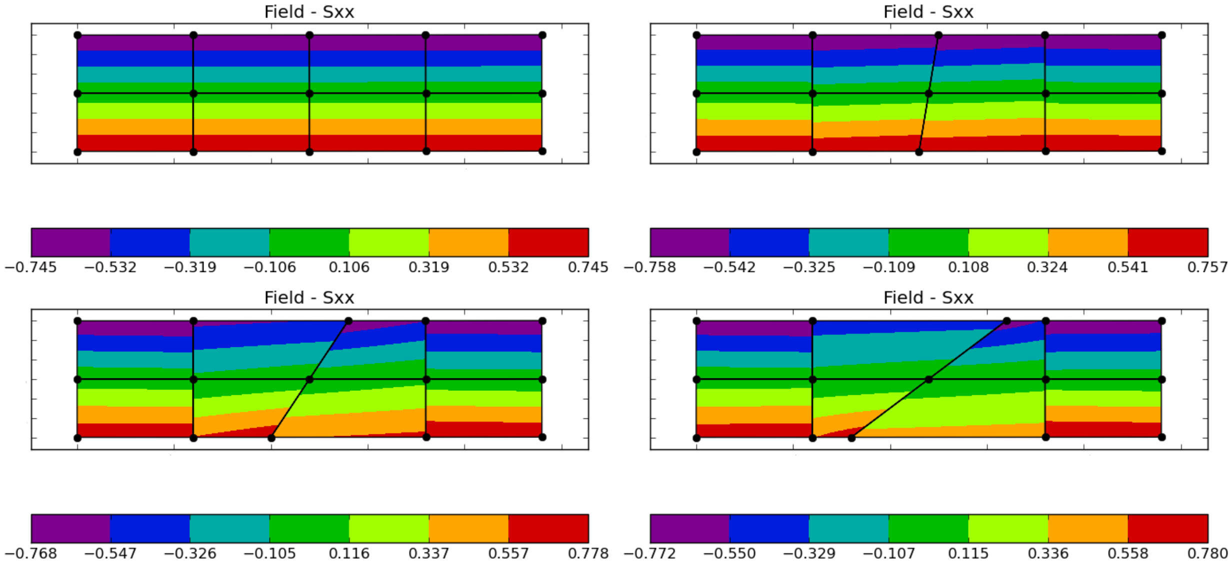
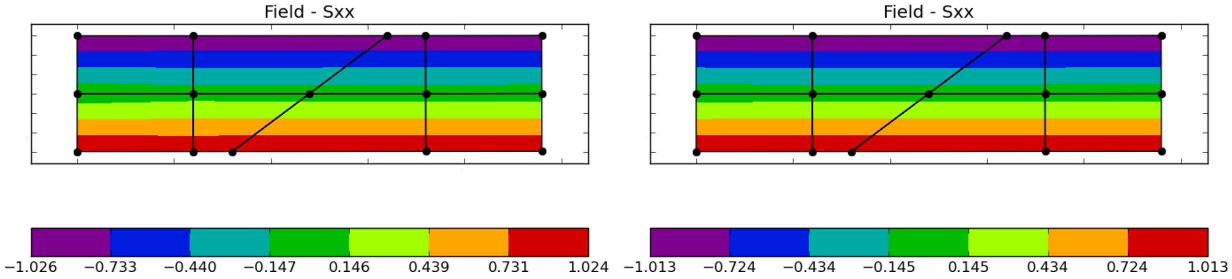


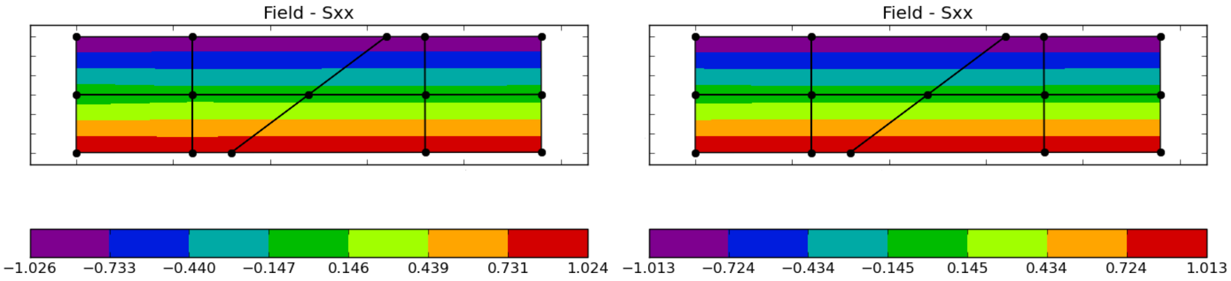
Figure 5.9. FEM horizontal stress results for different ‘distorted’ meshes.

Even for the more distorted mesh, both polynomial and shifted enrichment functions improve the horizontal stress results in the GFEM model, as depicted in Figure 5.10 and Figure 5.11.



(a) *(b)*

Figure 5.10. GFEM horizontal stress results for polynomial enrichment functions; first (a) and second (b) degree enrichment.



(a) *(b)*

Figure 5.11. GFEM horizontal stress results for shifted enrichment functions; first (a) and second (b) degree enrichment.

Using the developed computational framework capability of mixing different degrees partition of unities to describe the element geometry and structural behavior, we advance in the present study proposing curved meshes, described by means of second degree serendipity partitions of unity (PU), associated to first degree (linear) lagrangian PU to describe the solid structural behavior. The achieved horizontal stress results are presented in Figure 5.12.

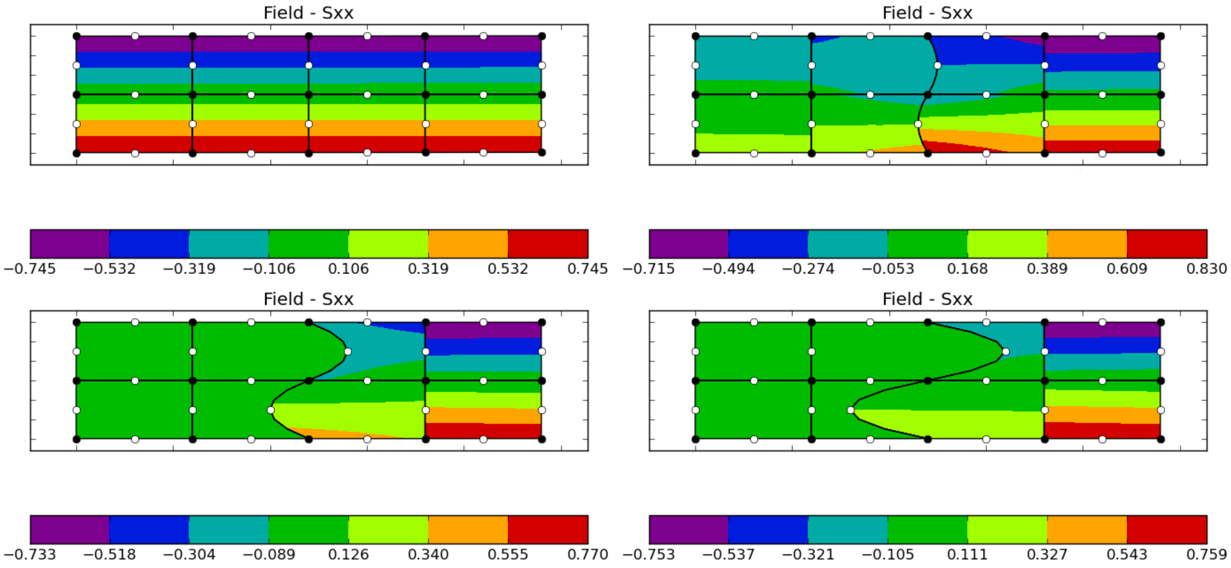
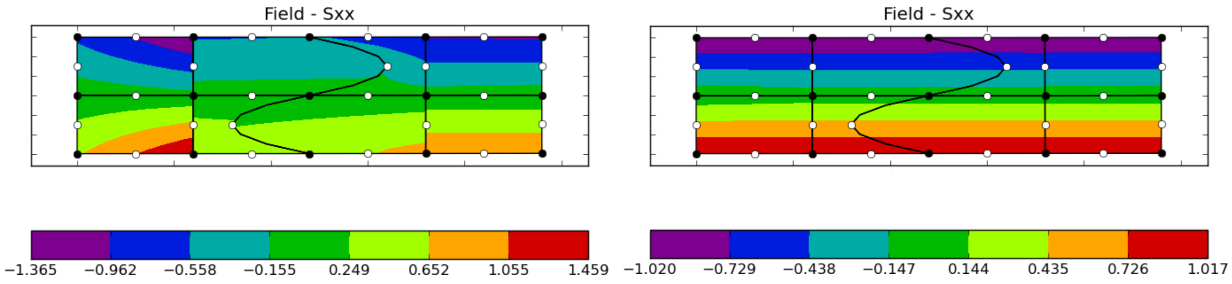


Figure 5.12. FEM horizontal stress results for different ‘curved’ meshes.

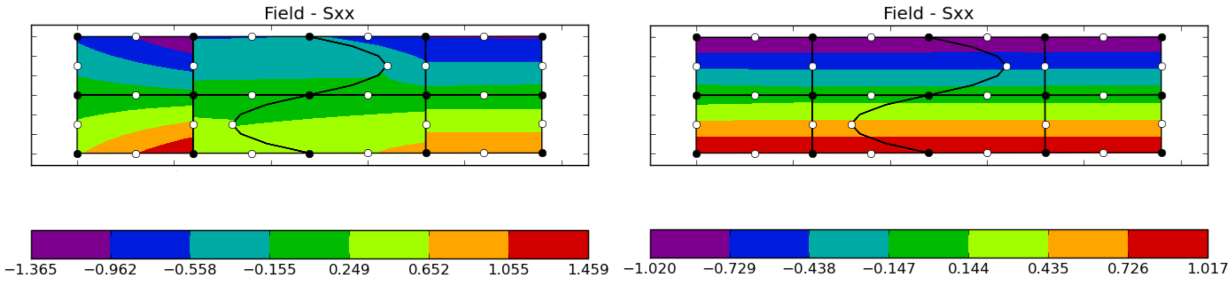
The white nodes in Figure 5.12 do not present degrees of freedom associated to them since they are used only by the geometric PU. These nodes are here referred as ‘geometric nodes’.

For the more curved mesh, both polynomial and shifted enrichment function improve the horizontal stress results of the GFEM models, as depicted in Figure 5.13 and Figure 5.14.



(a) *(b)*

Figure 5.13. Horizontal stress component (Sxx) results achieved using polynomial enrichment functions; first (a) and second (b) degree enrichment.



(a) *(b)*

Figure 5.14. Horizontal stress component (Sxx) results achieved using shifted functions; first (a) and second (b) degree enrichment.

Once again, both polynomial and shifted function enrichments result in similar results. Due to this fact, in what follows, only the results of one of them will be presented, except for the cases in which substantial result differences occur.

It is important to mention that similar meshes tested for the isoparametric FEM elements for equivalent curved meshes do not resulted in similarly good results for the more curved meshes, even for bi-cubic isoparametric elements.

5.3. Linear elastic beam dynamics

Aiming to evaluate the dynamical behavior of the structures modeled using the GFEM, a 2.0 meters cantilever beam is stated, as indicated in Figure 5.15.

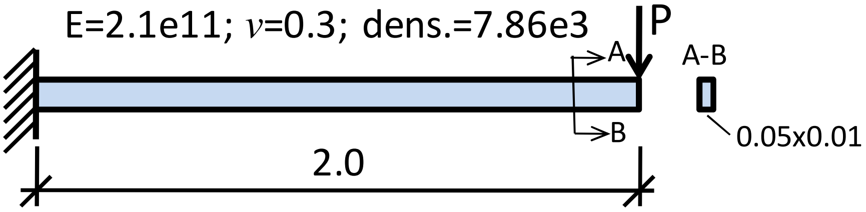


Figure 5.15. Cantilever beam (dynamics evaluation).

As indicated, the beam cross section dimension is 0.05x0.01m (5 centimeters height and thickness of 1 centimeter), such that a plane stress model is adopted. Linear elastic behavior is assumed, for steel like material properties: 2.1 10^{11} N/m2 for the Young modulus, 0.3 for the Poisson's ratio and 7.86 10^3 kg/m^3 for the density. Even though the density is used for computing the inertial terms, the load due to self-weight is not considered for the numerical model.

For the proposed data, natural frequency of vibration of the structure is approximately 10.4373 Hertz (angular frequency ω equals to 65.5793 radians), and its period (T) is equal to 0.09581 seconds.

One supposes that the load **P** is applied according to the following time history:

$$P(t) = P\ \sin(\pi\ {}^{t}\!/_{T}),\ \text{for}\ t \leq T;$$
$$P(t) = 0,\ \text{for}\ t > T.$$

For these conditions, Warburton (1976) presents the following solution for the vertical displacements *v(t)* at the free edge of the cantilever:

$$v(t) = \frac{Pl^3}{EI}\left(0.441\sin(\pi\ {}^{t}\!/_{T}) - 0.216\sin(2\pi\ {}^{t}\!/_{T})\right),\ \text{for}\ t < T;$$
$$v(t) = \frac{Pl^3}{EI}\left(-0.432\sin(\omega_1(t-T)) - 0.0059\sin(\omega_2(t-T)) - 0.0066\cos(\omega_2(t-T))\right), t \geq T.$$

In these equations **P** is the applied load (here adopted as 100 N), *l* is the cantilever span (2.0 meters), *E* is the Young modulus, and *I* is the cross section moment of inertia. Also, ω_1 and ω_2 are, respectively the first and second angular frequencies of the structure.

Taking this solution as a reference for evaluating the dynamical behavior of the proposed structure, several different models are computed using the developed computational framework, both for FEM and GFEM.

5.3.1. FEM - computational code validation

The first model to be evaluated is defined by means of a regular mesh of 640 (4x160) serendipity bi-quadratic quadrilateral elements (Q8), resulting in a FE model containing 4496 degrees of freedom. The dynamic model is integrated for the total time of 0.5 seconds, within 300 time steps, resulting in a time increment equals to 0.001667 seconds. The results were computed both for consistent and lumped mass matrix. The lumped matrix is computed according to the formulation presented in Chapter 3, i.e., proportional to the diagonal terms of the consistent mass matrix.

The achieved results are depicted in Figure 5.16, in which the theoretical solution result is also plotted. As it can be noticed, the results are very close to the theoretical solution, what is expected for such a fine mesh model. For this reason, in further analysis of the present section, the results obtained for this mesh are considered as a numerical reference.

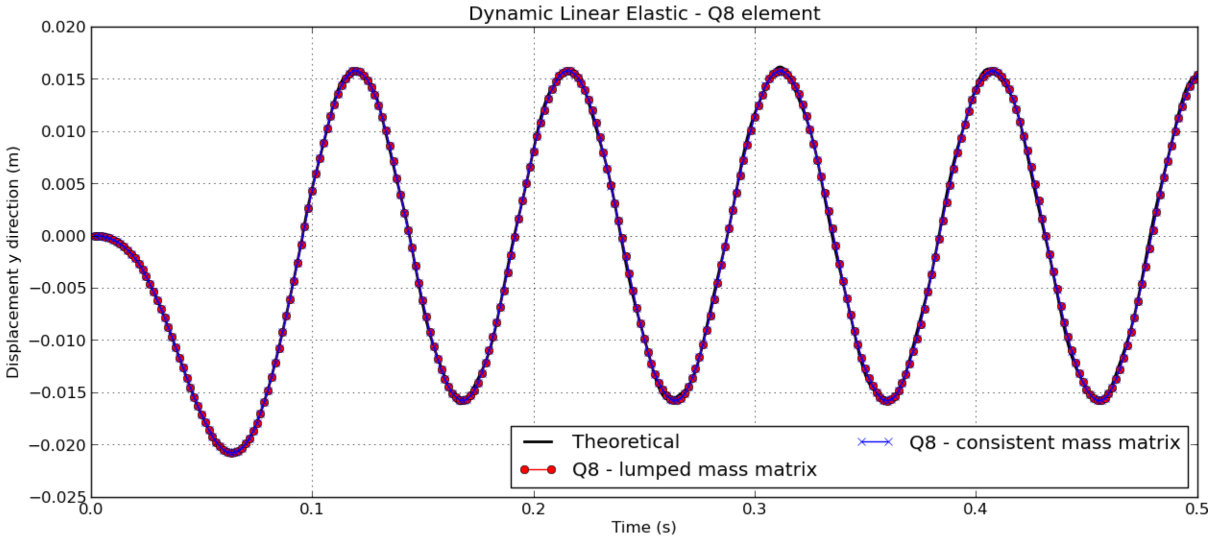


Figure 5.16. Vertical displacement of the free edge of the cantilever beam, for a 640 bi-quadratic (Q8) element model.

Still for the traditional finite element model, a hierarchical refinement is adopted employing meshes of bi-linear quadratric (Q4) elements. Starting from a very coarse mesh of 40 (2x20) elements, for which the relation between the elements' sides is 4, one defines other meshes of 80, 160 and 640 bi-linear elements. The vertical displacements result for each of these meshes is depicted in Figure 5.17. The same time increment of the previous example is employed.

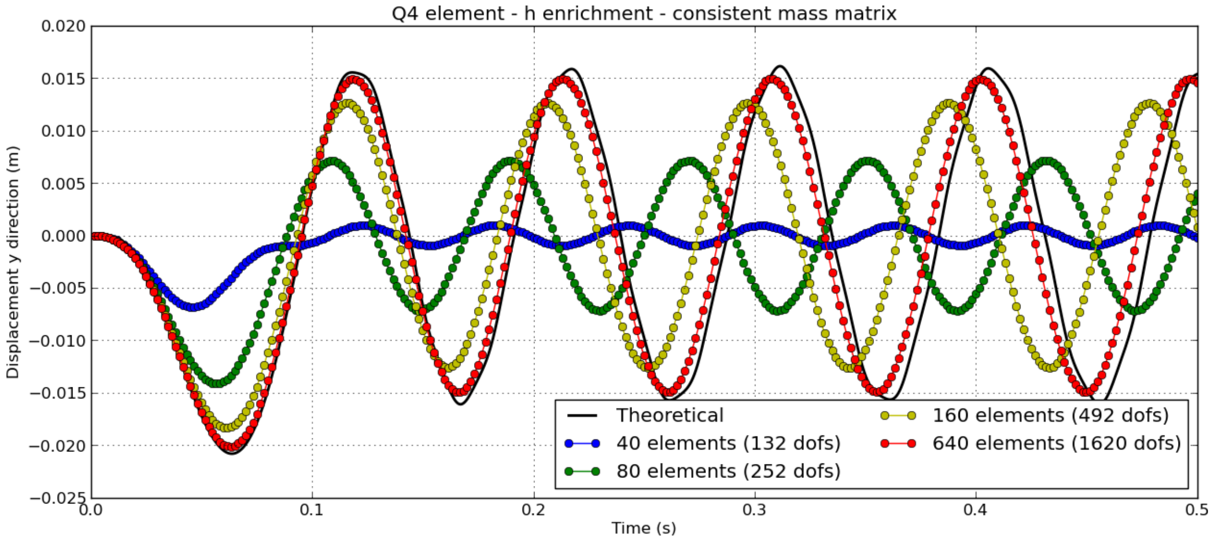


Figure 5.17. Vertical displacement of the free edge of the cantilever beam, for several bi-linear (Q4) element models (consistent mass matrix).

The results achieved for the same meshes, computed using lumped mass matrix are practically the same, as shown in Figure 5.18.

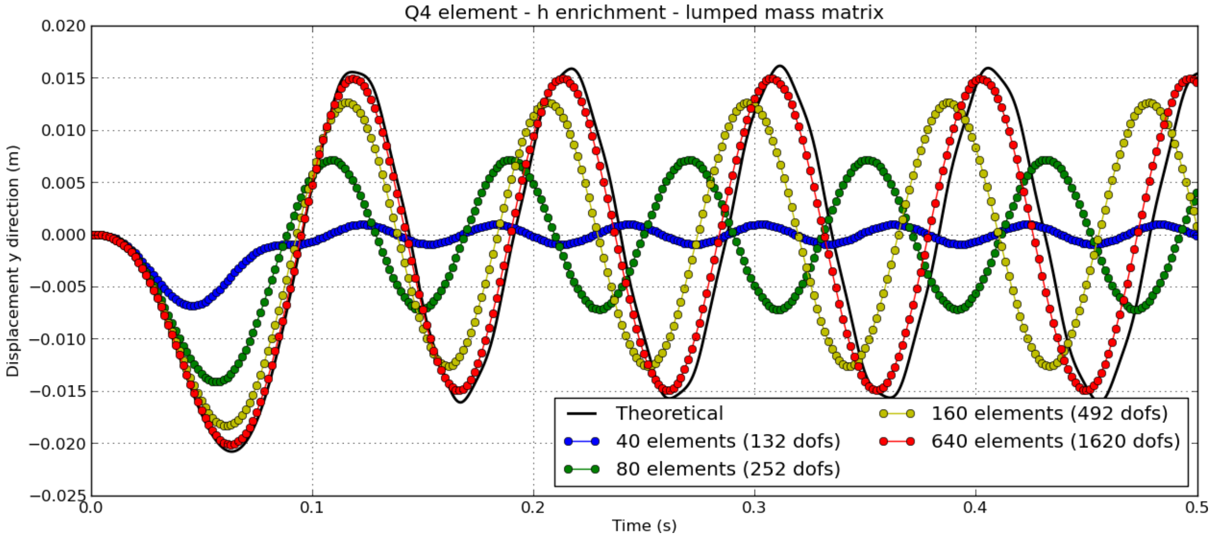


Figure 5.18. Vertical displacement of the free edge of the cantilever beam, for several bi-linear (Q4) element models (lumped mass matrix).

5.3.2. GFEM - polynomial and shifted enrichment

For the evaluation of the GFEM results, the coarser Q4 mesh (40 elements) is used. The enrichment is applied over all nodes, except the ones in which Dirichlet boundary conditions are applied. For the first degree enrichment, it results in 372 degrees of freedom. For the second degree enrichment, the resulting number of degrees of freedom is 732.

The free edge vertical displacement for the polynomial enrichment are presented in Figure 5.19. A consistent mass matrix considering the enriched degrees of freedom is used.

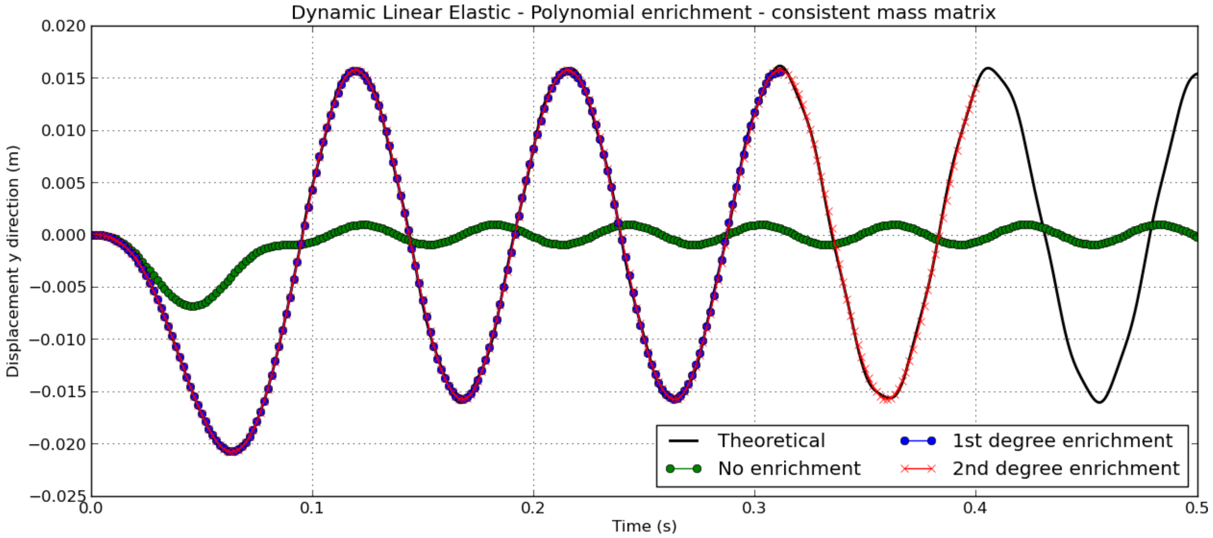


Figure 5.19. displacement of the free edge of the cantilever beam, for polynomial enrichment function (consistent mass matrix).

Even though remarkable improvements in the solution are observed, one notices that the results for the first degree enrichment are available only up to 0.3 seconds, and up to 0.4 seconds for the second degree. For later time instants, the models have diverged.

If the same enriched models are computed using a lumped mass matrix, the transient analysis do not diverges, but the results are inaccurate, as depicted in Figure 5.20.

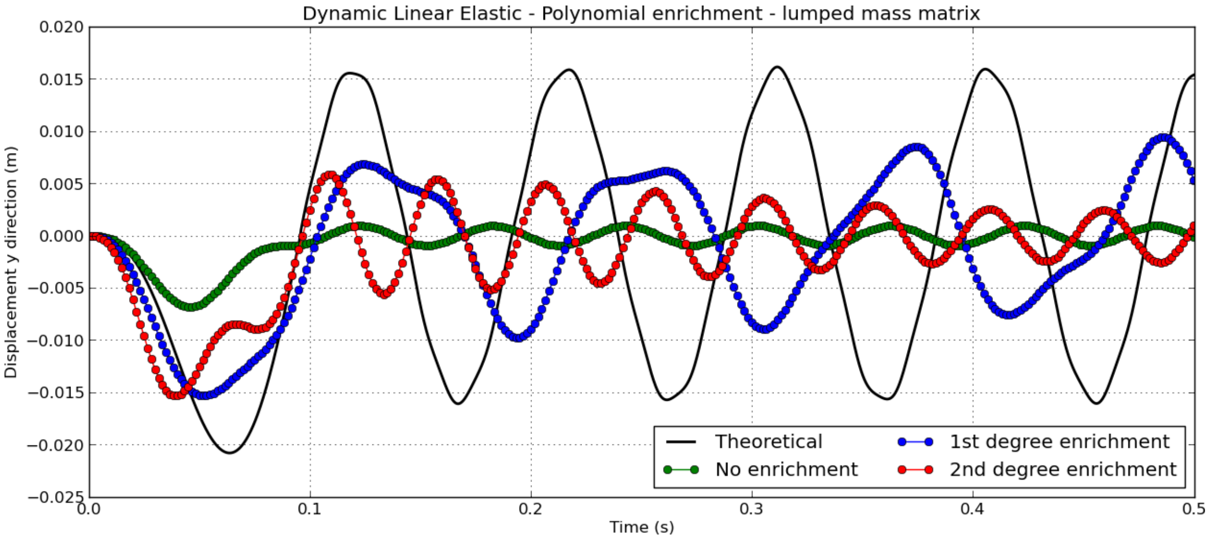


Figure 5.20. Vertical displacement of the free edge of the cantilever beam, for polynomial enrichment function GFEM model (lumped mass matrix).

The shifted enrichment function were also evaluated, both for the first and second degree enrichment. The shifted enrichment functions results for the consistent mass matrix are even less stable than the ones obtained with the polynomial enrichment functions (see Figure 5.21).

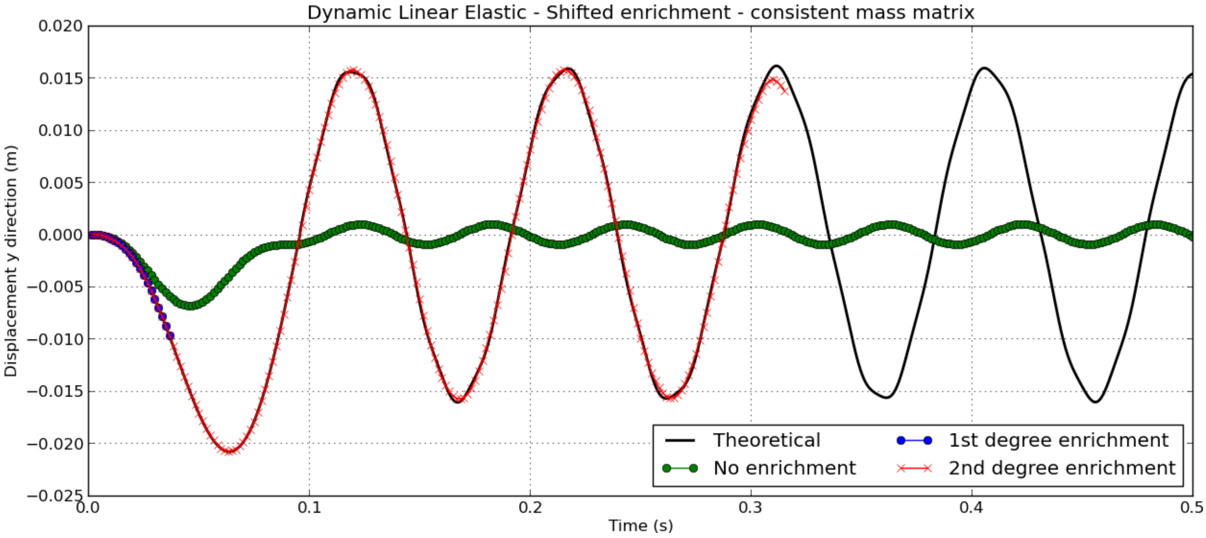


Figure 5.21. Vertical displacement of the free edge of the cantilever beam, for shifted enrichment function GFEM model (consistent mass matrix).

Also for the shifted enrichment functions, the results obtained for the lumped mass matrix are more stable, but are also not accurate, especially for the second degree enrichment (Figure 5.22).

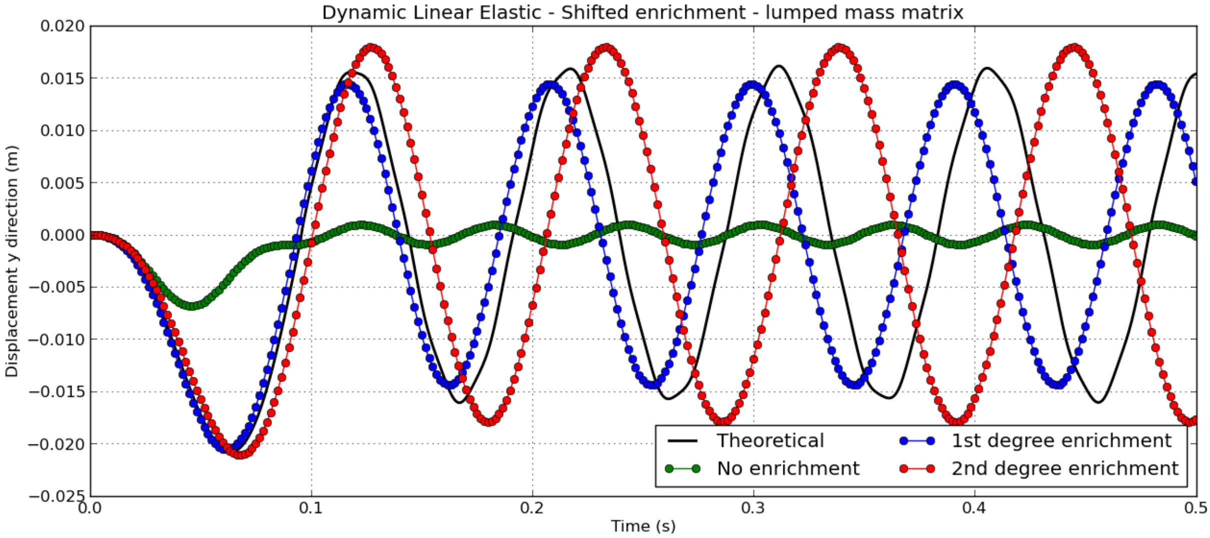


Figure 5.22. Vertical displacement of the free edge of the cantilever beam, for shifted enrichment function GFEM model (lumped mass matrix).

5.3.3. GFEM - Trigonometric enrichment functions

As observed in the previous section, the preliminary results of the dynamical behavior of the GFEM model are not sufficiently accurate and stable for employing the method to solve linear dynamics problems. The present section aims to evaluate the dynamical behavior of the GFEM models using trigonometric enrichment functions.

The referred trigonometric enrichment functions are defined in Chapter 3, in which they are referred as 'double sine' enrichment functions. Obviously such nomenclature is adopted due to the fact that they are computed by performing the product of the sine function of the coordinates, both in horizontal and vertical directions.

In the present section such functions are used to improve the vertical displacements. For this reason, they are applied only over the degrees of freedom related to such field. Also, taking into account the analytical solution of such kind of problem, the enrichment is only computed according to the sine of the horizontal coordinate position, i.e., the sine related to the vertical coordinate is computed to the power of zero, such that it results always as one.

The results of the vertical displacement of the free edge of cantilever beam are presented in Figure 5.23, for several different trigonometric enrichment functions. The enrichment is applied over all nodes except the ones in which Dirichlet boundary conditions are applied. It was used the same time increment of the previous examples and the results are computed using a consistent mass matrix.

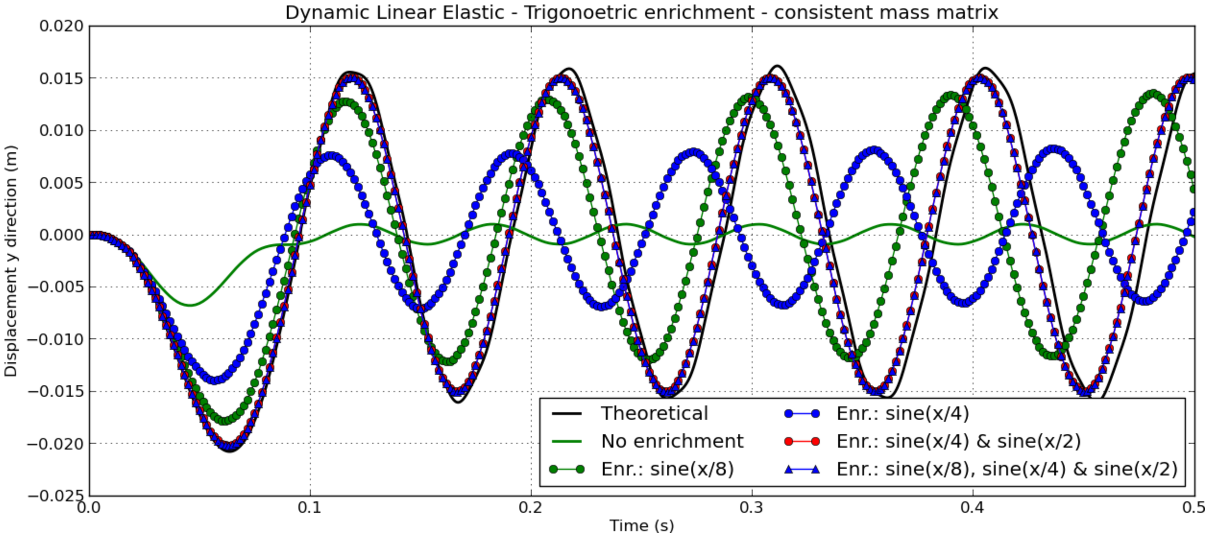


Figure 5.23. Vertical displacement of the free edge of the cantilever beam, for several trigonometric (double sine) enrichment functions GFEM model (consistent mass matrix).

As it can be noticed, the vertical displacement results are improved using the proposed enrichment, especially when more than one enrichment function is employed (for instance, the last enrichments presented in Figure 5.23). On the other hand, one has observed that further enrichment schemes (three or more enrichment functions over the same node) do not improve the results for the proposed example.

For the case of lumped mass matrix, one does not observe suitable results.

5.3.4. Other lumping technique evaluation

Advancing in the tests of the dynamical behavior of the GFEM model, other numerical tests are proposed aiming to improve the unstable and inaccurate results of the previous sections.

The first proposed change is to evaluate the dynamical behavior of the GFEM model using a different lumping technique for the mass matrix computation. Here we propose a new technique according to which the lumped mass matrix is computed by considering the diagonal term of the lumped mass matrix as the sum of all the terms of the respective row of the consistent matrix.

When used for the shifted enrichment functions, such ‘new’ lumping technique does not result in stable models, as indicated in Figure 5.24.

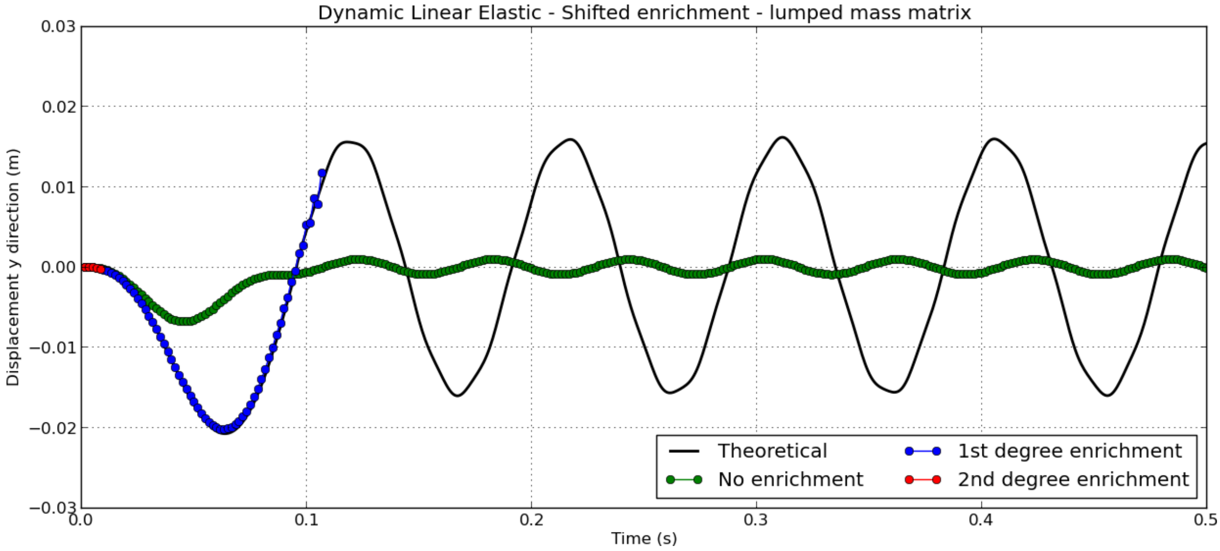


Figure 5.24 Vertical displacement of the free edge of the cantilever beam, for shifted enrichment function GFEM model (lumped mass matrix – new lumping technique).

In spite of this fact, the results achieved using the 'new' lumping technique for the polynomial enrichment functions are stable and yet accurate, especially for the second degree enrichment, as it can be observed in Figure 5.25.

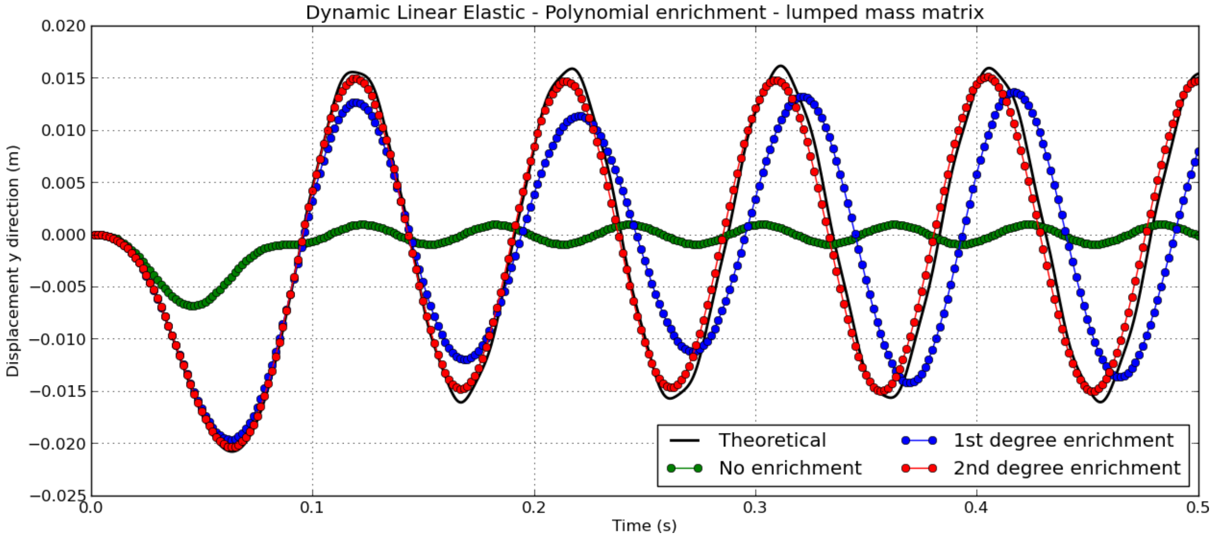


Figure 5.25. Vertical displacement of the free edge of the cantilever beam, for polynomial enrichment function GFEM model (lumped mass matrix – new lumping technique).

For the traditional FEM, the results achieved using the 'new' proposed technique are very close to the ones obtained previously, as shown in Figure 5.26.

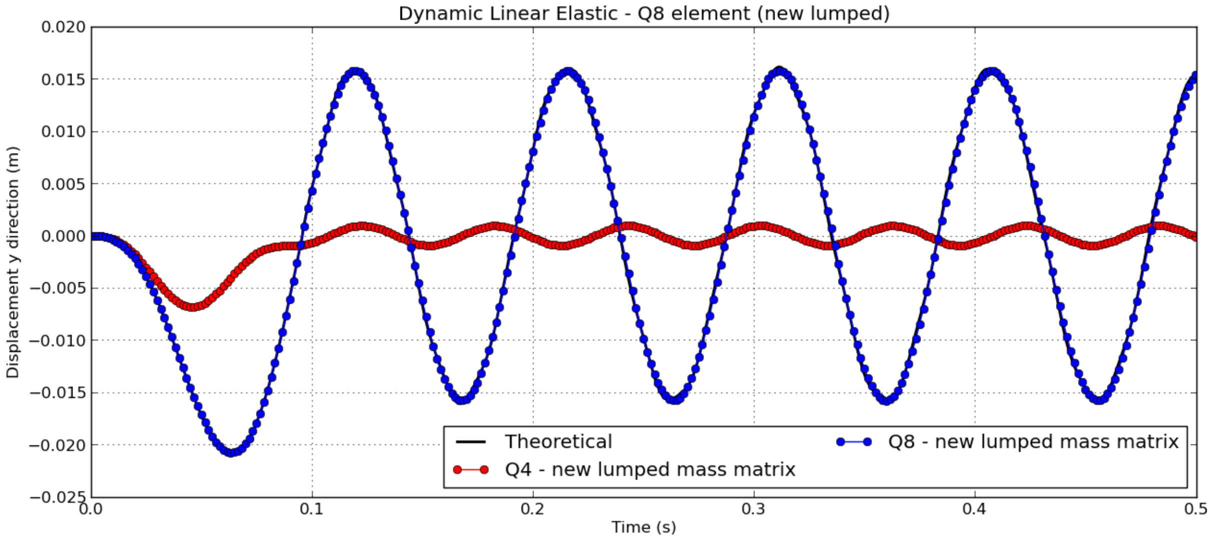


Figure 5.26. Vertical displacement of the free edge of the cantilever beam the traditional FEM model (lumped mass matrix – new lumping technique).

5.3.5. GFEM dynamics for not enriched mass matrix

Advancing in the efforts for efficiently modeling the dynamical behavior of solids using the GFEM, numerical tests are processed using mass matrices computed without considering the enrichment related degree of freedom.

For the polynomial enrichment functions the results achieved for the consistent and lumped mass matrices are depicted, respectively in Figure 5.27and Figure 5.28.

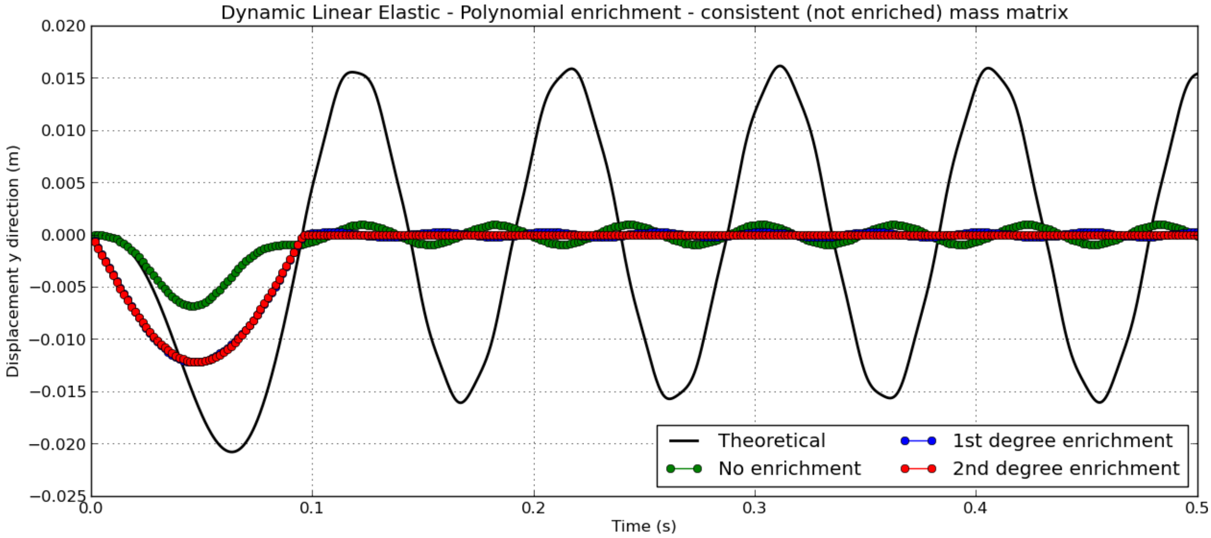


Figure 5.27. Vertical displacement of the free edge of the cantilever beam, for polynomial enrichment function GFEM model (consistent -not enriched - mass matrix).

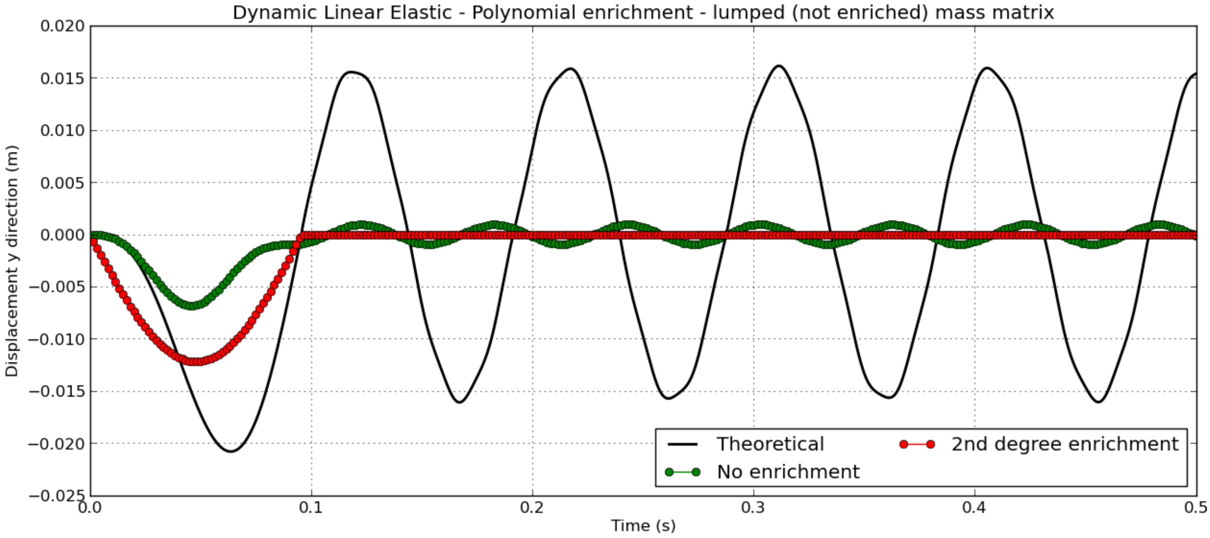


Figure 5.28. Vertical displacement of the free edge of the cantilever beam, for polynomial enrichment function GFEM model (lumped -not enriched - mass matrix).

It is clear that such strategy is not suitable for such kind of enrichment, since the structure present no dynamic response. In fact, it behaves like if it was totally dumped.

Despite of this fact, for the shifted enrichment functions using a consistent mass matrix, the results are stable and accurate as shown in Figure 5.29.

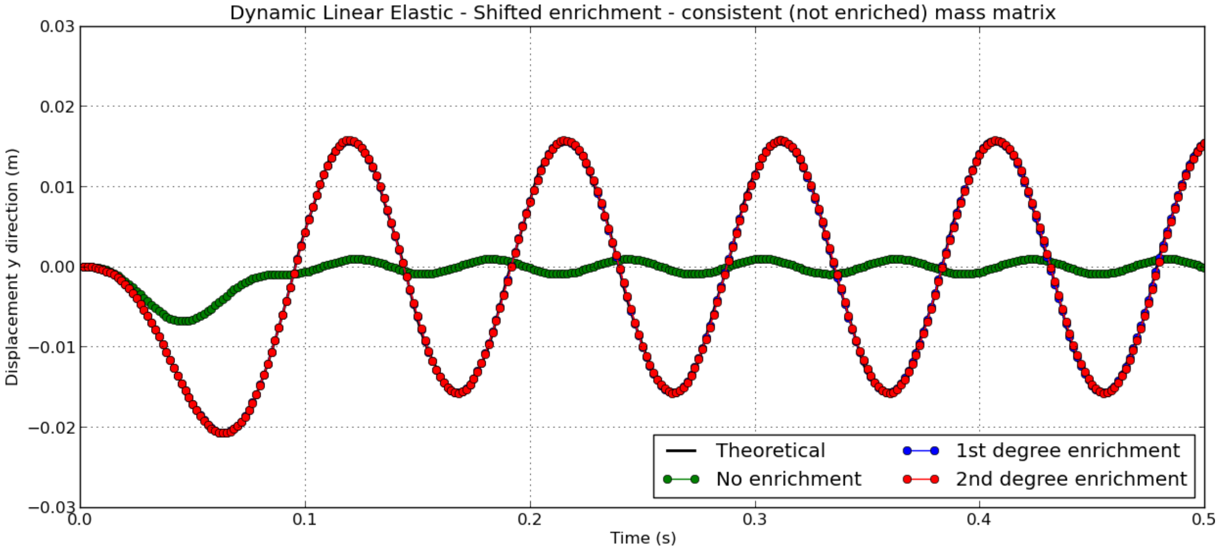


Figure 5.29. Vertical displacement of the free edge of the cantilever beam, for shifted enrichment function GFEM model (consistent -not enriched - mass matrix).

For the lumped mass matrix, the transient solution diverges for the first degree enrichment, but it converges and is stable and accurate for the second degree enrichment (see Figure 5.30).

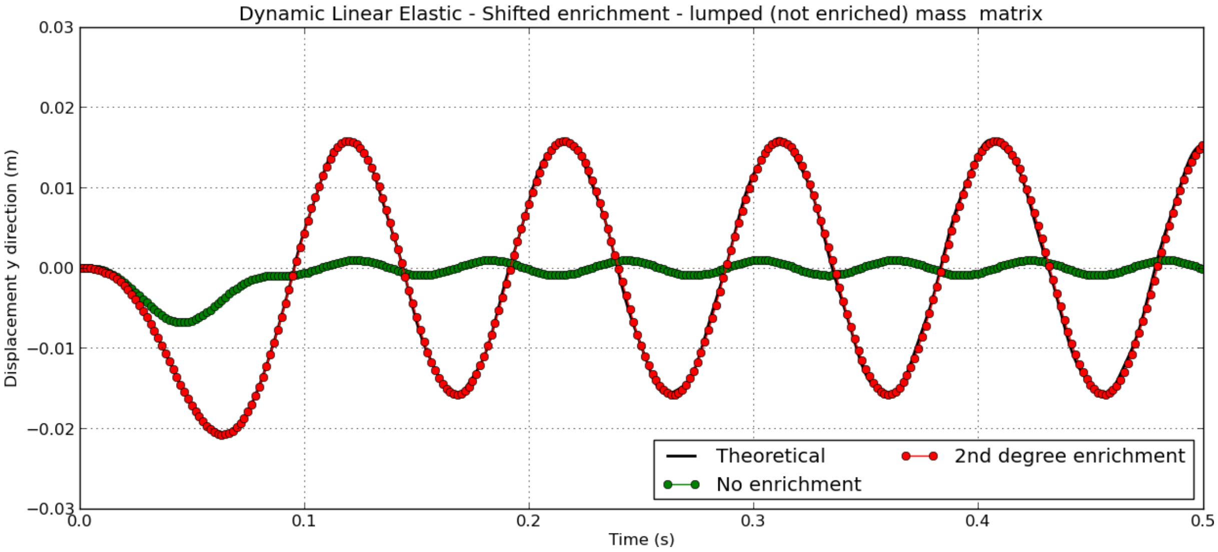


Figure 5.30. Vertical displacement of the free edge of the cantilever beam, for shifted enrichment function GFEM model (lumped -not enriched - mass matrix).

5.3.6. Stability after several steps increments

Once suitable linear dynamic results are achieved for the GFEM, a step forward in the validation of the proposed GFEM formulation is to evaluate if it indeed remain stable after a considerable number of time increments. In order to perform such test, we extend the previous example to longer time integration scheme (3.0 seconds). The time step interval remains the same, so that the model is computed in 1800 time steps.

The GFEM model using consistent enriched mass matrix are accurate at initial instants, but unstable after some time steps, as illustrated in Figure 5.31.

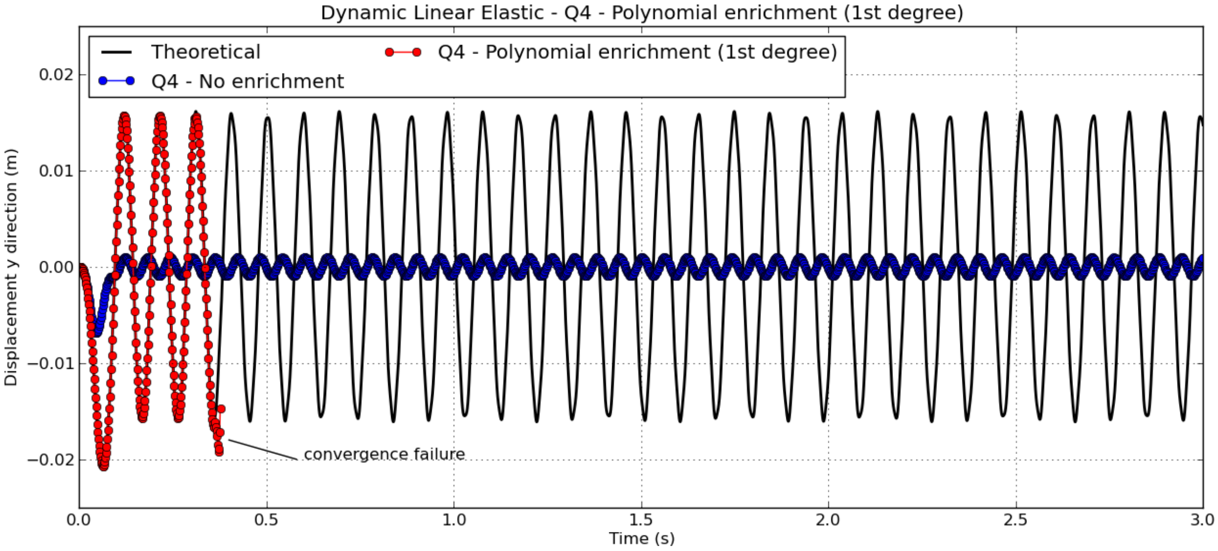


Figure 5.31 Vertical displacement of the free edge of the cantilever beam, for shifted enrichment function GFEM model (consistent- enriched - mass matrix); 1800 time steps.

The trigonometric (double sine) enrichment was also tested, using the enrichment scheme represented by red circles in Figure 5.23, i.e., sine(x/4) and sine (x/2). The achieved results are presented in Figure 5.32.

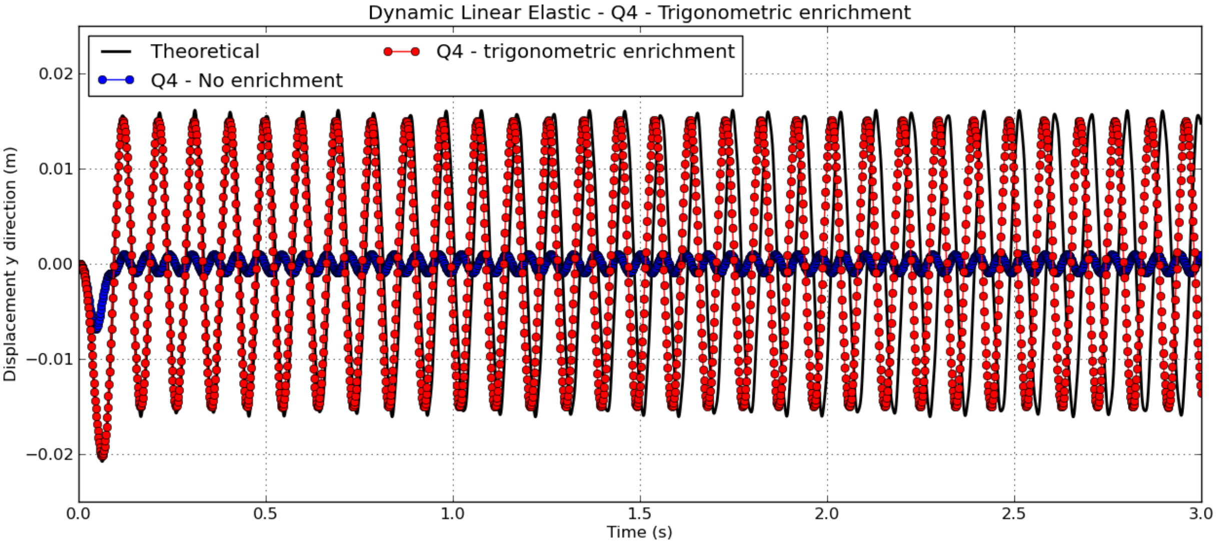


Figure 5.32. Vertical displacement of the free edge of the cantilever beam, for trigonometric (double sine) enrichment function GFEM model (consistent- enriched - mass matrix); 1800 time steps.

For the proposed trigonometric enrichment, the transient solution in 1800 time steps is stable and remarkable displacement result improvements are observed. In spite of these facts, the period of vibration is slightly smaller than the one of the theoretical solution, causing a discrepancy after several time steps.

For the shifted enrichment model employing not enriched mass matrix, the coarser mesh (40 elements) is used for the proposed 1800 time steps integration. The results remain accurate even at the end of the processing, as it can be observed in Figure 5.33.

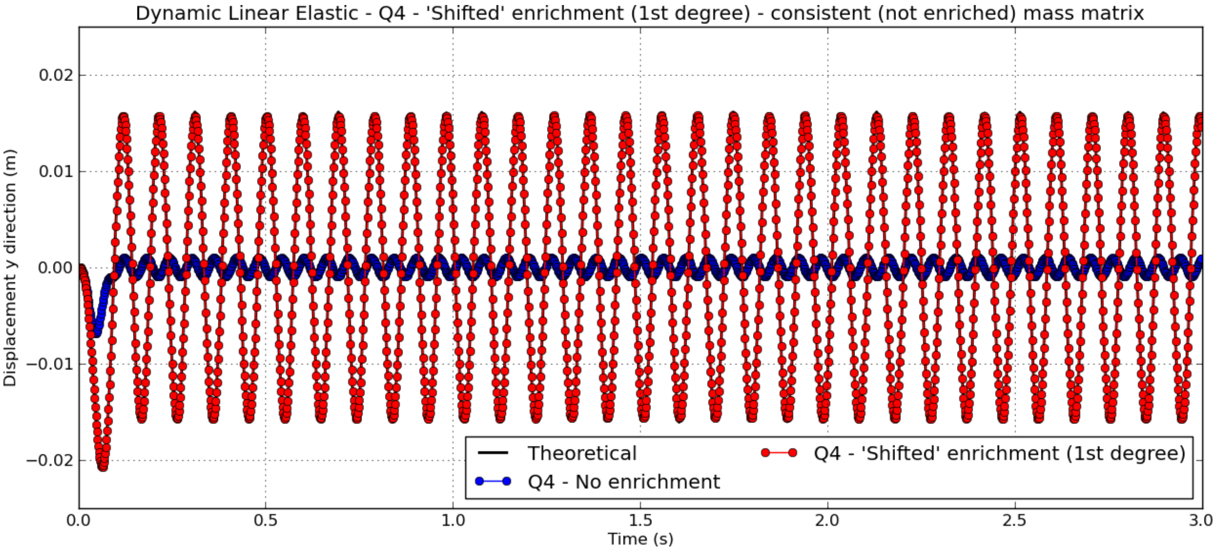


Figure 5.33. Vertical displacement of the free edge of the cantilever beam, for shifted enrichment function GFEM model (consistent- not enriched - mass matrix); 1800 time steps.

Finally, the bi-quadratic (Q8) FEM model with 640 elements is also evaluated. The results are accurate and stable, as depicted in Figure 5.34.

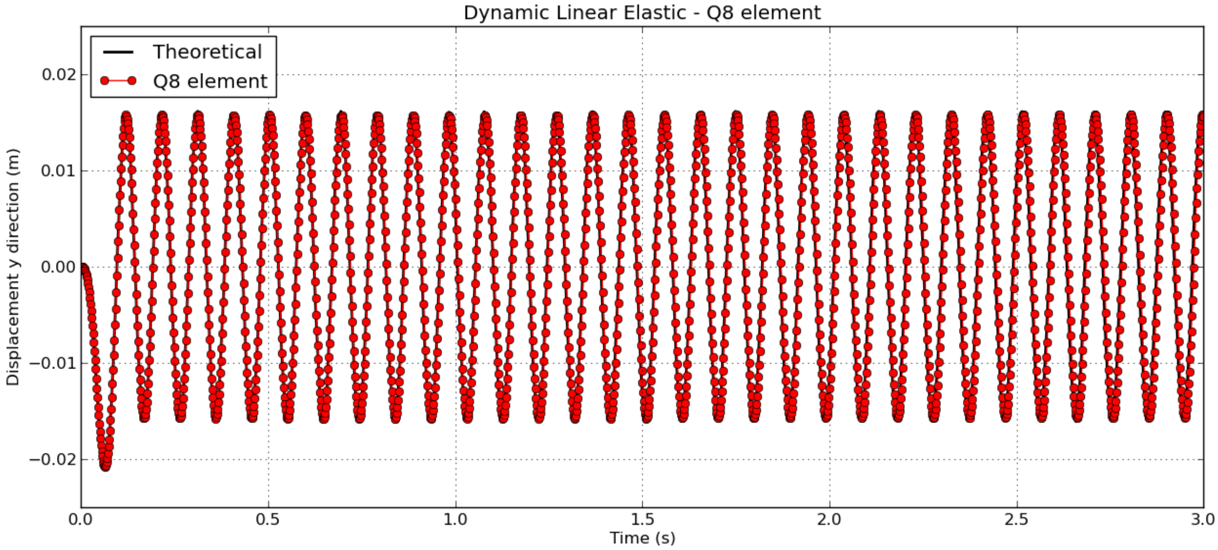


Figure 5.34. Vertical displacement of the free edge of the cantilever beam, for FEM - Q8 mesh (consistent mass matrix); 1800 time steps.

5.3.7. GFEM – dynamics for harmonic applied forces

Advancing in the linear dynamics analysis evaluation, one evaluates the dynamical behavior of the GFEM models for structures subjected to harmonic applied forces.

One proposes to evaluate the vertical displacement of the free edge of the cantilever beam for a 10 N force applied at the same position as the one considered in the previous linear dynamics examples (see Figure 5.15), oscillating in four different frequencies: 6.0 Hertz, 8.0 Hertz, 10.2 Hertz and 12.0 Hertz. It is worth to note that the natural frequency of vibration of the proposed structure is 10.4373 Hertz, i.e., the third proposed harmonic load is purposely close to such value.

The numerical solutions obtained for the 640 bi-quadratic (Q8) elements are considered the reference for comparing the results achieved by means of the GFEM.

The results achieved for the trigonometric enrichment scheme evaluated in the previous section (sine(x/4) and sine(x/2)), for the proposed frequencies, are indicated in the following four figures (Figure 5.35 to Figure 5.38).

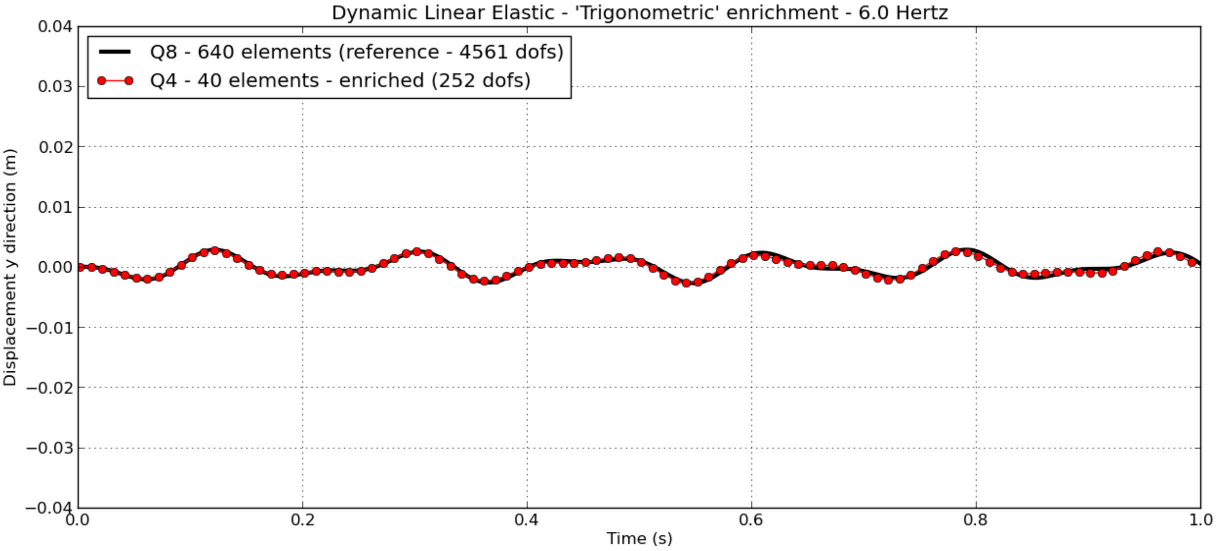


Figure 5.35. Vertical displacement of the free edge of the cantilever beam, for trigonometric (double sine) enrichment GFEM model; harmonic load – 6.0 Hertz.

For the frequencies of 6.0 and 8.0 Hertz the results are, in general, close to the reference solution. It is worth mentioning that the results achieved using the same coarse mesh, for no enrichment, are very far from the ones of the reference solution. In some time instants of the solution, the GFEM solutions differ slightly from the reference solution, but the model seems to represent the expected dynamical behavior, in a general sense.

For the frequency of 10.2 Hertz (close to the proposed structure natural frequency), the results are very close to the reference solution (Figure 5.37). At further time instants, the GFEM model fails in representing accurately the amplitude of vibration of the reference solution.

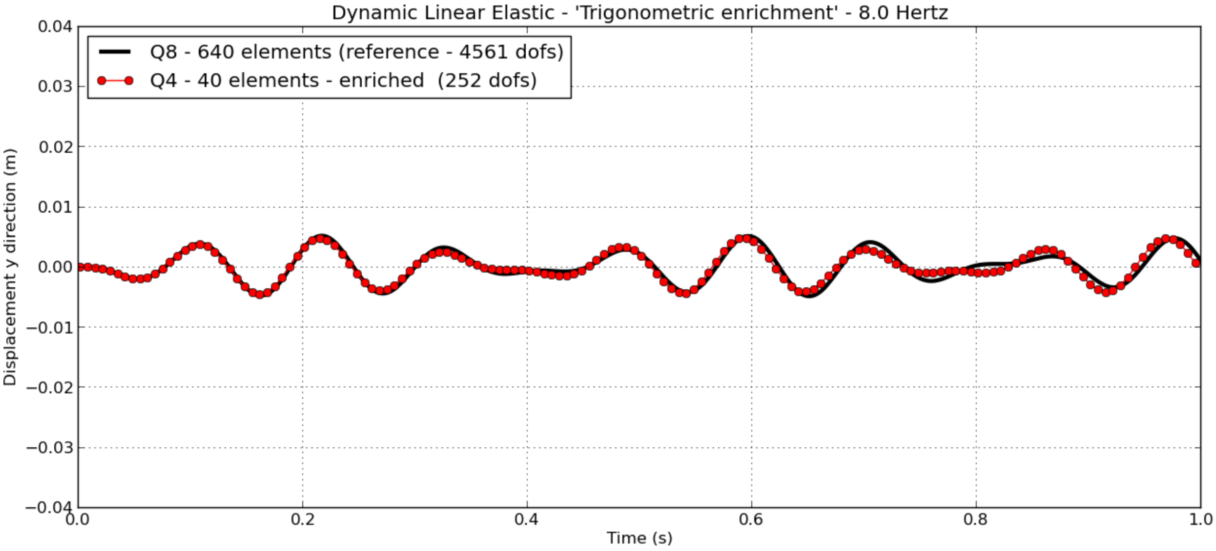


Figure 5.36. Vertical displacement of the free edge of the cantilever beam, for trigonometric (double sine) enrichment GFEM model; harmonic load – 8.0 Hertz.

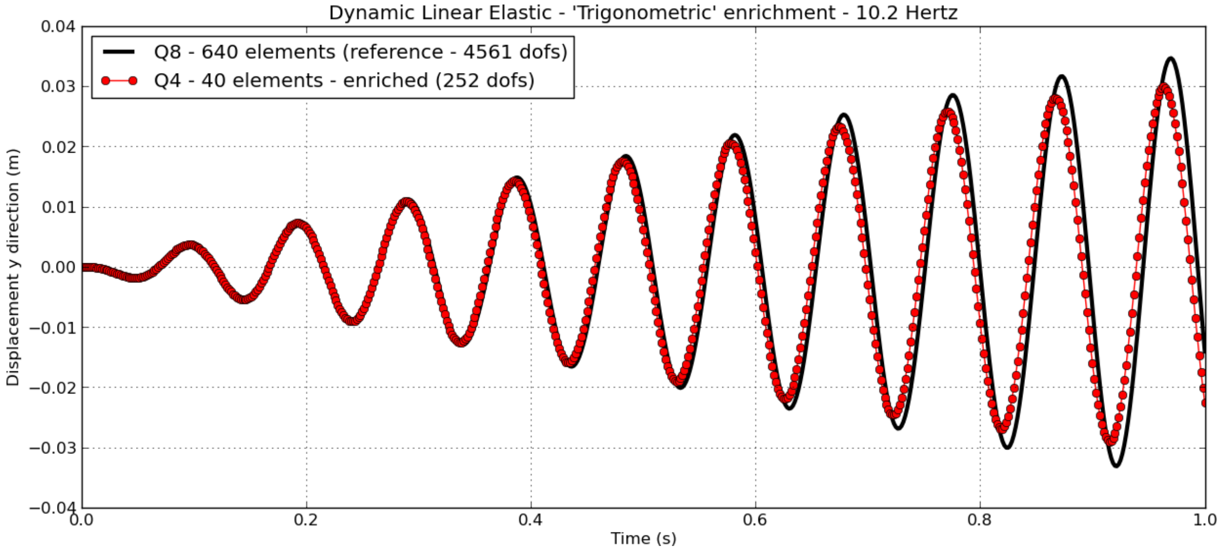


Figure 5.37. Vertical displacement of the free edge of the cantilever beam, for trigonometric (double sine) enrichment GFEM model; harmonic load – 10.2 Hertz.

For the frequency of 12.0 Hertz, the proposed trigonometric enrichment GFEM model seems to present the same dynamical behavior observed for the lower frequencies tested previously (Figure 5.38).

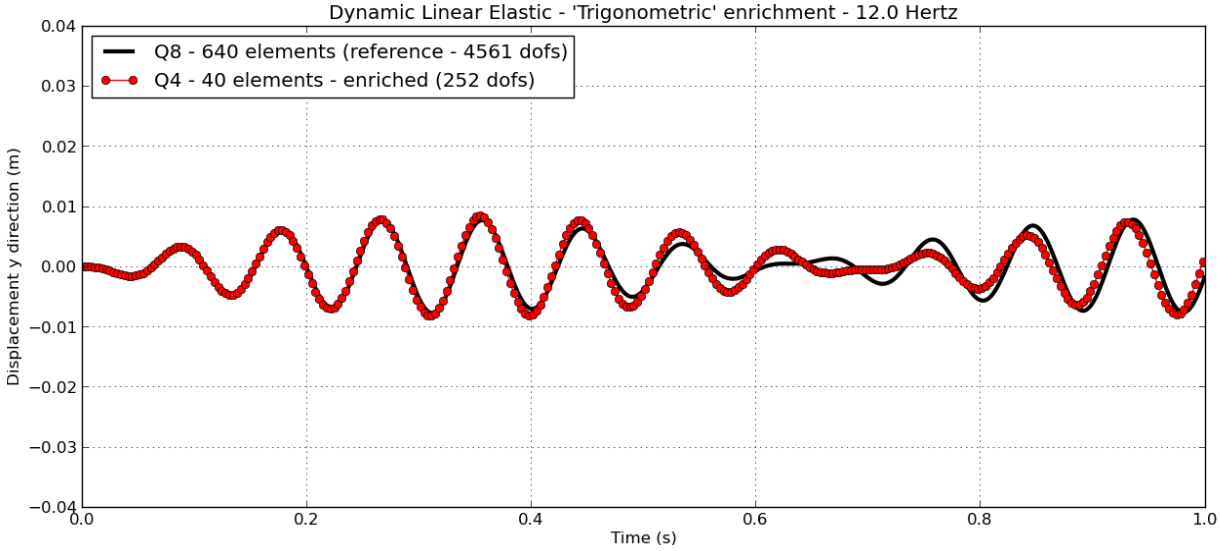


Figure 5.38. Vertical displacement of the free edge of the cantilever beam, for trigonometric (double sine) enrichment GFEM model; harmonic load – 12.0 Hertz.

Advancing in the GFEM linear dynamics behavior evaluation, one finally tests the model subjected to the same frequencies for a first degree shifted enrichment scheme, using consistent not enriched mass matrix to represent the structure inertial behavior. The achieved results are present in the four following figures (Figure 5.39 to Figure 5.41.)

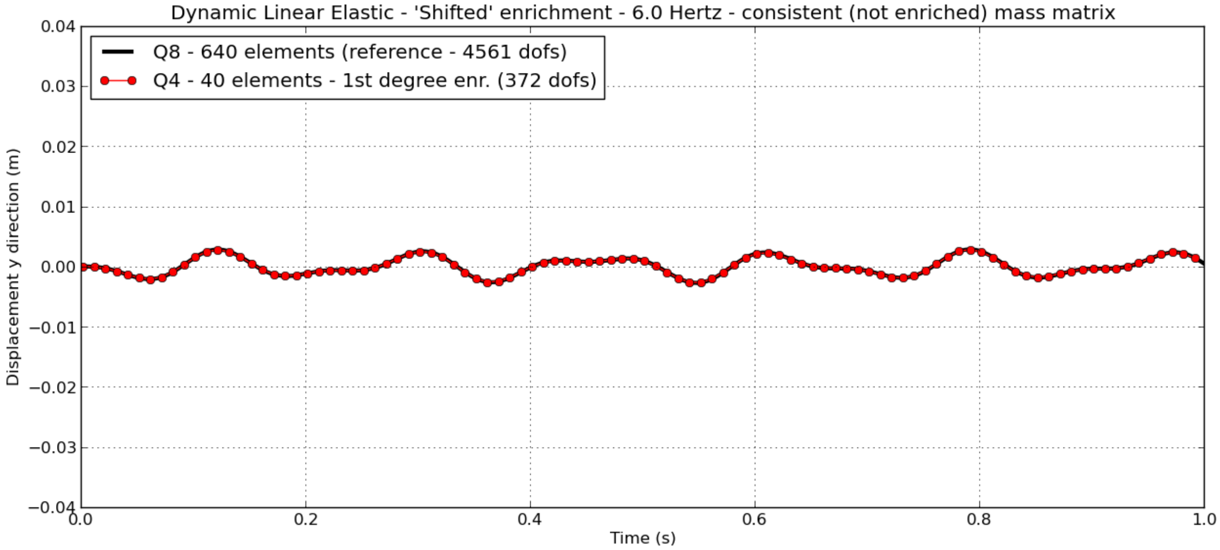


Figure 5.39. Vertical displacement of the free edge of the cantilever beam, for a first degree shifted GFEM model (not enriched mass matrix); harmonic load – 6.0 Hertz.

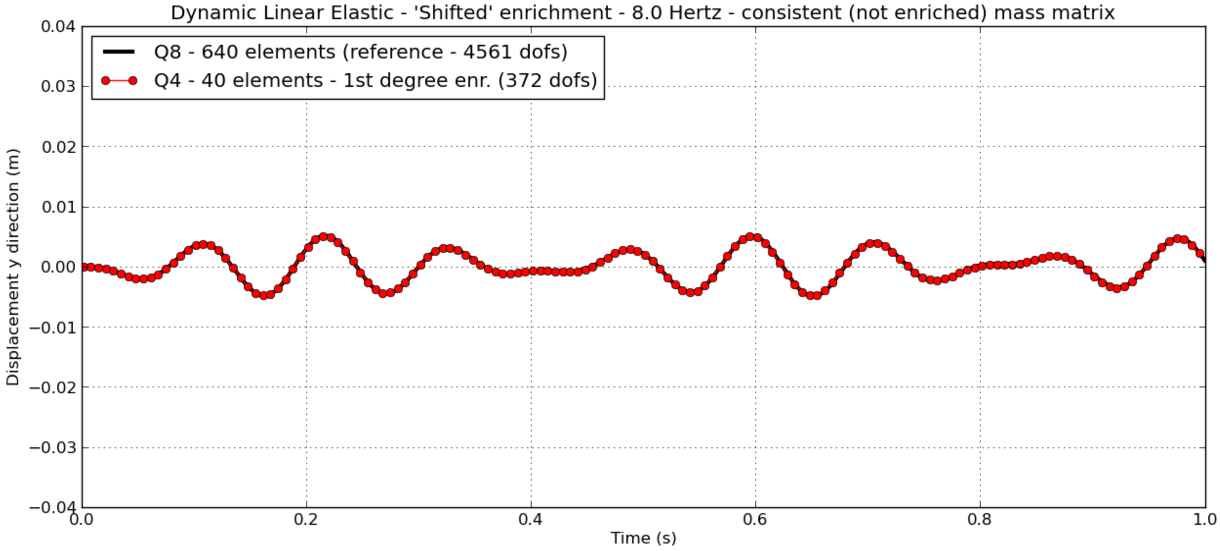


Figure 5.40. Vertical displacement of the free edge of the cantilever beam, for a first degree shifted enrichment GFEM model (not enriched mass matrix); harmonic load – 8.0 Hertz.

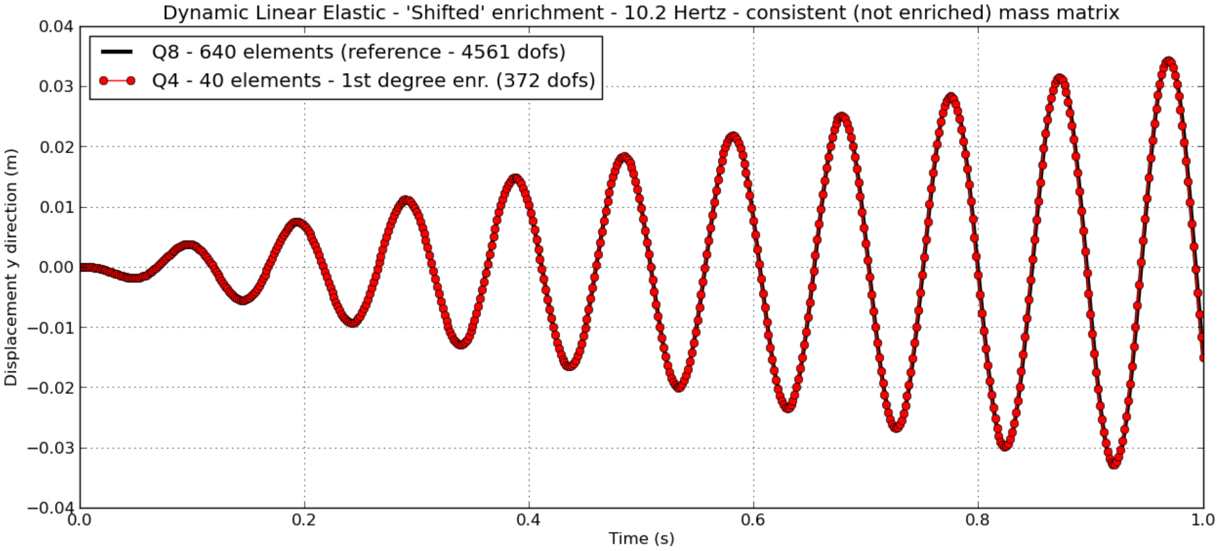


Figure 5.41. Vertical displacement of the free edge of the cantilever beam, for a first degree shifted enrichment GFEM model (not enriched mass matrix); harmonic load – 10.2 Hertz.

The observed results are very close to the reference solution, for all tested frequencies, as it can be observed in the presented results.

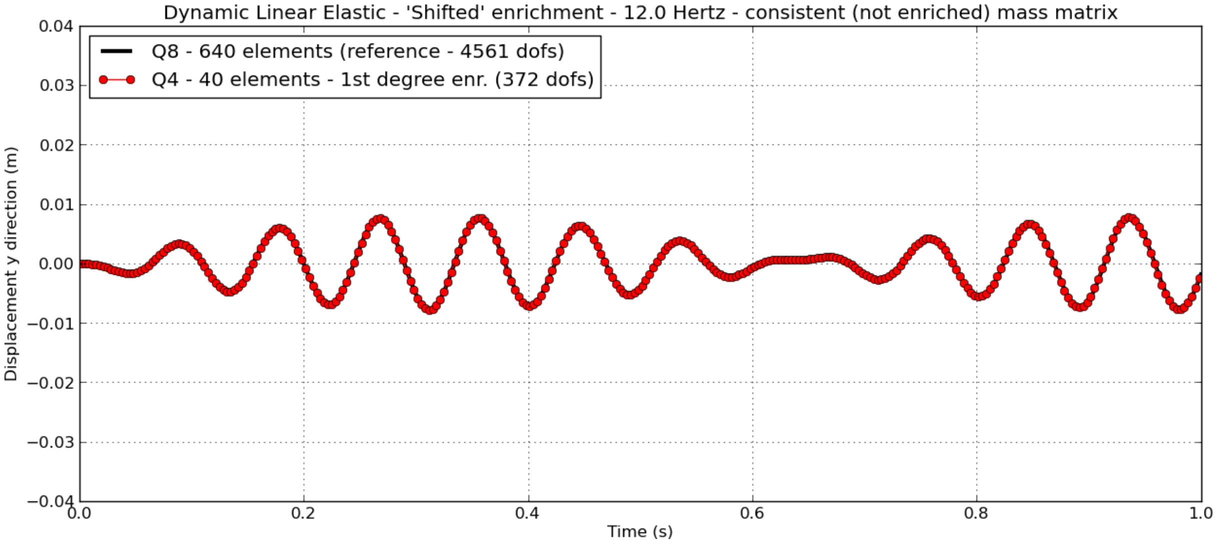


Figure 5.42. Vertical displacement of the free edge of the cantilever beam, for a first degree shifted enrichment GFEM model (not enriched mass matrix); harmonic load – 12.0 Hertz.

5.3.8. A brief discussion on the system of equation condition number

The linear dynamic analysis performed by means of implicit time integration methods depends on the equilibrium configuration convergence, which, by it turns, is related to the associated system of equation condition number. It is known that the shifted and polynomial enrichment functions can introduce linear dependency on the resulting system of equation, demanding especial techniques to solve it.

The loss of stability of the solution process, observed in the present example, is related to such system of equation conditioning. So, an analysis on the resulting condition number of the tested models is important to explain the instabilities observed in the previously presented results.

The condition number for the FEM models is presented in Table 5.1.

Table 5.1. Condition number for the FEM models.

type	n. elements	n. dof	mass matrix	condition number
Q4	40	126	consistent	1.296877e+05
			lumped – technic 1	1.294200e+05
			lumped – technic 2	1.294200e+05
Q4	80	246	consistent	1.301963e+05
			lumped – technic 1	1.301182e+05
			lumped – technic 2	1.301182e+05
Q4	160	486	consistent	1.318546e+05
			lumped – technic 1	1.318233e+05
			lumped – technic 2	1.318233e+05
Q4	640	1610	consistent	5.242989e+05
			lumped – technic 1	5.242679e+05
			lumped – technic 2	5.242679e+05
Q8	640	4498	consistent	4.319351e+06
			lumped – technic 1	4.319238e+06
			lumped – technic 2	4.319351e+06

In Table 5.1, we refer as lumping technic 1 the one presented in Chapter 3 (the lumped terms are computed proportional to the consistent matrix diagonal terms). The lumping technic 2 is the one tested in Section 5.3.4, for which the lumped term is computed by the sum of the terms of the specific consistent mass matrix row. It is important to mention that the presented condition numbers are computed using a NumPy routine (The SciPy Community (2011)), in which such number is computed according to Strang (1980).

It is also important to explain that the presented condition numbers are computed over the scaled form of the system of equation matrix K^*, computed by

$$K^* = DKD,\ D_{ij} = \frac{1}{\sqrt{K_{ij}}} \text{ if } i = j;\ D_{ij}=0 \text{ if } i \neq j.$$

For some of the GFEM models, the observed condition number is clearly much higher, as indicated in Table 5.2.

Table 5.2. Condition number for the GFEM models.

enrichment	n. dof	mass matrix	condition number
shifted – 1st degree	366	enriched – consistent	3.777804e+17
		enriched – lumped – technic 1	9.263366e+05
		enriched – lumped – technic 2	1.684145e+09
		not enriched – consistent	7.057268e+17
		not enriched – lumped – technic 1	6.142784e+17
shifted – 2nd degree	726	enriched – consistent	4.789890e+17
		enriched – lumped – technic 1	1.225849e+06
		enriched – lumped – technic 2	1.066182e+18
		not enriched – consistent	2.103090e+18
		not enriched – lumped – technic 1	2.463852e+18
polynomial – 1st degree	366	enriched – consistent	2.160766e+18
		enriched – lumped – technic 1	3.878980e+05
		enriched – lumped – technic 2	3.232698e+05
		not enriched – consistent	1.262585e+08
		not enriched – lumped – technic 1	1.259327e+08
polynomial – 2nd degree	726	enriched – consistent	4.570755e+18
		enriched – lumped – technic 1	1.839213e+06
		enriched – lumped – technic 2	6.003907e+05
		not enriched – consistent	4.841566e+18
		not enriched – lumped – technic 1	2.673353e+18
sine(x/8)	186	enriched – consistent	9.426896e+08
sine(x/4)	186	enriched – consistent	1.042058e+08
sine(x/4) and sine(x/2)	246	enriched – consistent	5.108613e+09
sine(x/8), sine(x/4) and sine(x/2)	306	enriched – consistent	4.943142e+15

It can be found, by comparing the condition number present in Table 5.2 to the results presented in the previous sections, that such high values of the condition number are, in general, associated to the models for which the dynamic analysis failed to find a solution. In spite of that, some GFEM models that present high condition number shown to be stable, as for instance, the ones enriched by shifted function, using not enriched consistent mass matrices.

Such a stability issue is also observed in the nonlinear analysis to be presented in the following examples, and causes loss of convergence in some of the nonlinear problems.

5.4. Cylinder - internal pressure (small-strain plasticity)

Aiming to evaluate the accuracy of the GFEM model to represent small strain elastoplastic material behavior, a simple example of an internally pressurized cylinder is proposed, for which an analytical solution derived by Hill (1950) is present in Souza Neto, Perić and Owen (2008). The cylinder cross section geometry is presented in Figure 5.43.

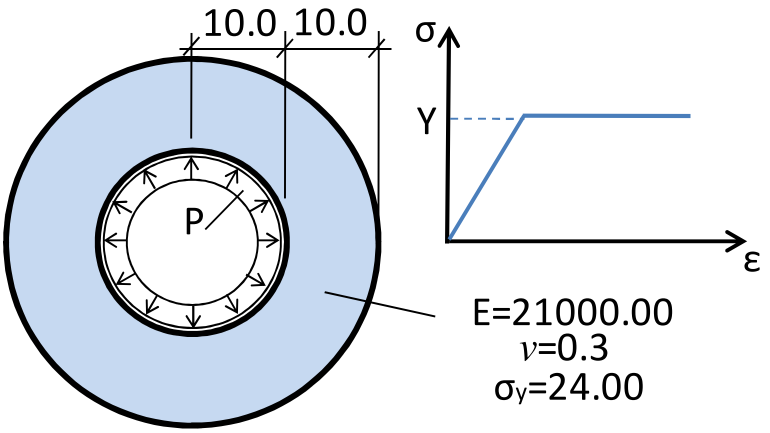


Figure 5.43. Internally pressurized cylinder example scheme.

It is assumed a von Mises stress based model for the perfectly plastic behavior material. The material properties are: Young's modulus equal to 21000 kN/cm^2, Poisson's ratio equal to 0.3 and uniaxial yield stress equal to 24 kN/cm^2.

The Hill solution relates the internal pressure *P* to the plastic front radius r_p as

$$\frac{P\sqrt{3}}{\sigma_y} = \ln\left(\frac{r_p}{r_i}\right) + \tfrac{1}{2}\left(1 - \frac{r_p^2}{r_e^2}\right),$$

in which r_i and r_e are, respectively, the internal and external cylinder radius.

One finds the position of the plastic front by means of the presented nonlinear equation, from which is possible to determine r_p. Plastic yield starts when the plastic front radius is equal to the cylinder internal radius. For the proposed data, it starts with a pressure equals to 10.39 kN/cm^2. The limit pressure, for which the cylinder collapses, is determined by means of the same equation, by imposing that the plastic yield radius is equal to external cylinder radius. For the proposed data, this value is equal to 19.21 kN/cm^2.

Once the elastic front radius is positioned, different expressions describe the stress components, in cylindrical coordinate system, i.e., σ_r (radial direction) and σ_θ (normal to the radial direction):

- For the elastic behavior region:

$$\sigma_r(r) = -\frac{\sigma_y\, r_p^2}{2\sqrt{3}\, r_e^2}\left(\frac{r_e^2}{r^2} - 1\right), \sigma_\theta(r) = \frac{\sigma_y\, r_p^2}{2\sqrt{3}\, r_e^2}\left(\frac{r_e^2}{r^2} + 1\right), r_i \le r \le r_p.$$

- For the plastic region:

$$\sigma_r(r) = \frac{\sigma_y}{\sqrt{3}}\left[-\tfrac{1}{2} - \ln(\tfrac{r_p}{r}) + \tfrac{r_p^2}{2r_e^2}\right], \sigma_\theta(r) = \frac{\sigma_y}{\sqrt{3}}\left[\tfrac{1}{2} - \ln(\tfrac{r_p}{r}) + \tfrac{r_p^2}{2r_e^2}\right], r_p \le r \le r_e.$$

The Finite Element (FE) model proposed to solve the problem is defined by a 192 (12x16) elements mesh of curved (isoparametric) bi-quadratic (Q8) elements, resulting in a model with 264 degrees of freedom. Only a quarter of the cylinder is represented in the FE model, since the it is symmetric. Plane strain hypothesis is assumed. The stress results along the cylinder radius for values of the pressure **P** equal to 10, 12 and 14 kN/m^2 are presented, respectively in Figure 5.44, Figure 5.45and Figure 5.46.

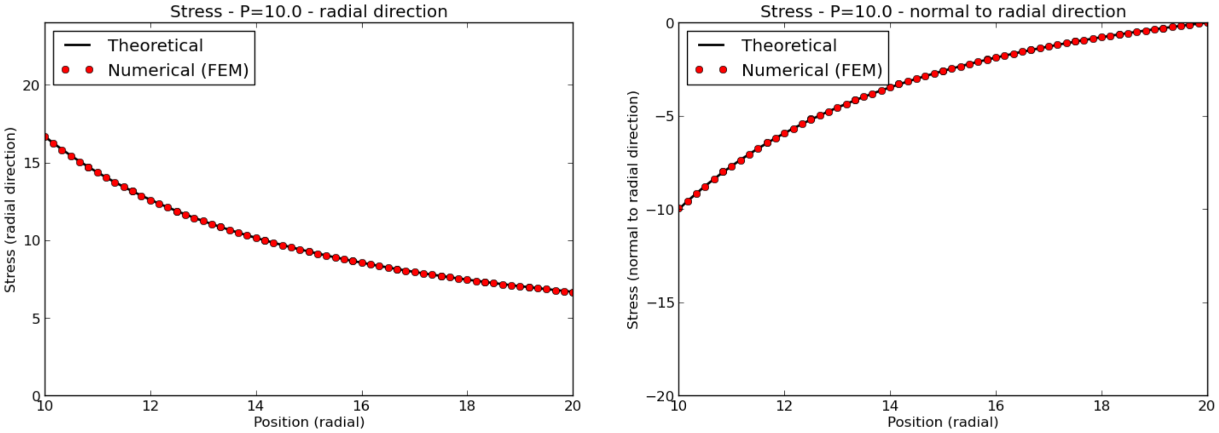


Figure 5.44. Stresses in the radial and normal to the radial directions along the cylinder radius for a pressure P=10 kN/cm^2.

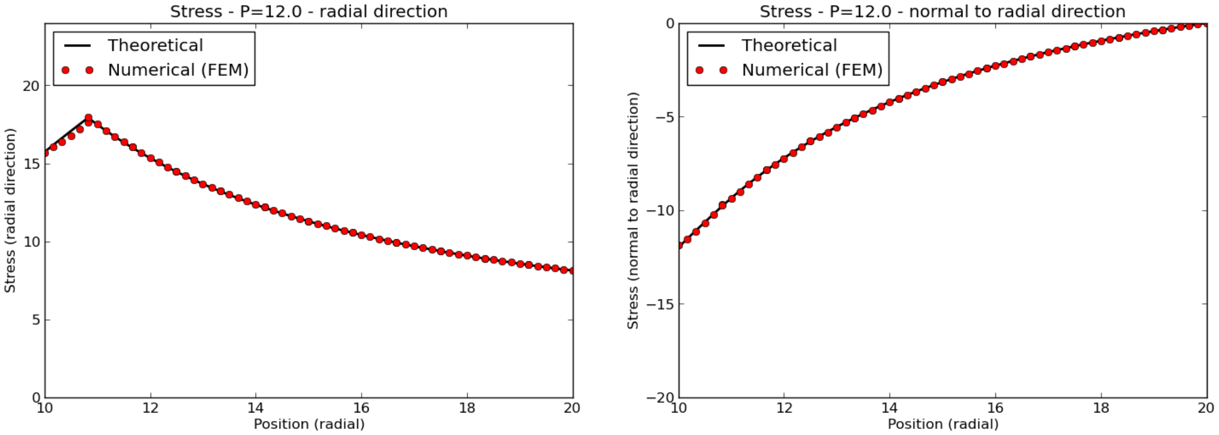


Figure 5.45. Stresses in the radial and normal to the radial directions along the cylinder radius for a pressure P=12 kN/cm^2.

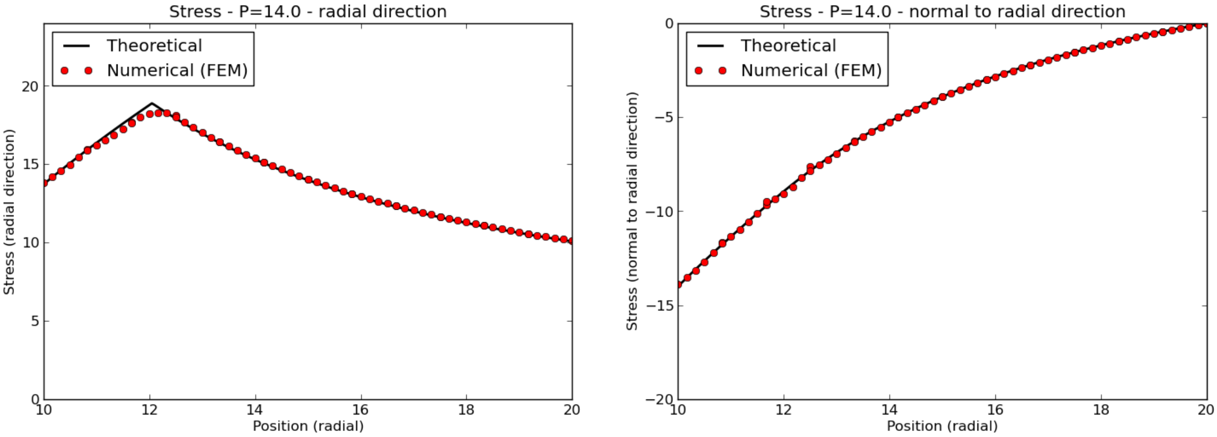


Figure 5.46. Stresses in the radial and normal to the radial directions along the cylinder radius for a pressure P=14 kN/cm^2.

The GFEM model used to simulate the elastoplastic example is defined by means of a mesh of 12 quadrilateral elements. Their geometry is described by means of bi-quadratic (Q8) partitions of unity (PU), while the PU used to describe the displacement fields is bi-linear (Q4).

All the nodes are enriched using shifted functions, up to the second degree, resulting in 120 degrees of freedom. The numerical integration is performed using 6x6 points (Gauss-Legendre quadrature). The Dirichlet boundary conditions, used to simulate the symmetries, are applied by imposing the displacement fields along the whole elements' sides.

Some of the displacement, stress and plastic strain fields obtained both for the FEM and the GFEM model are presented in the following figures (Figure 5.47 to Figure 5.52).

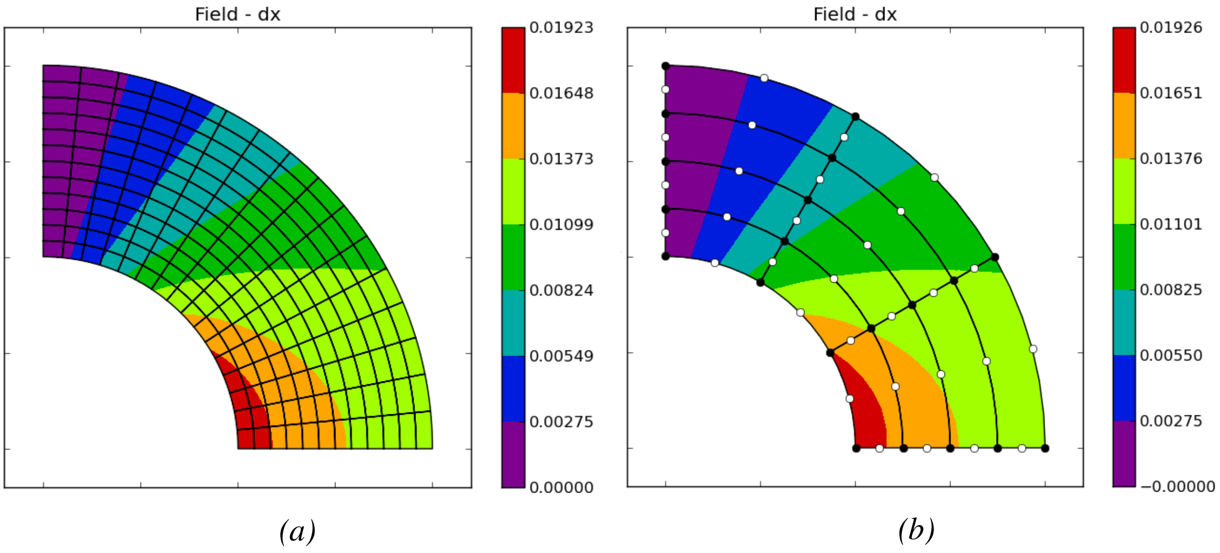
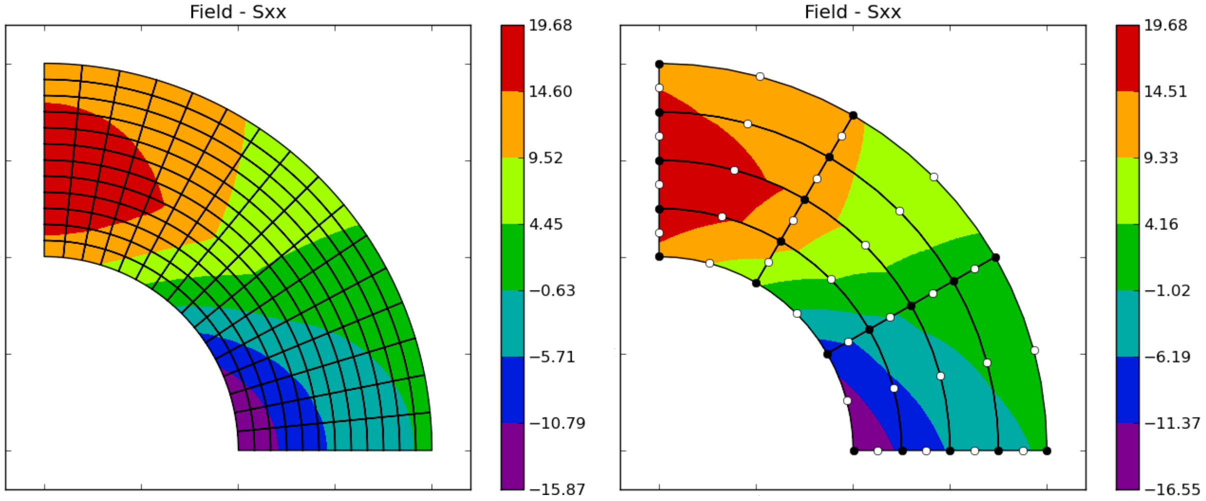
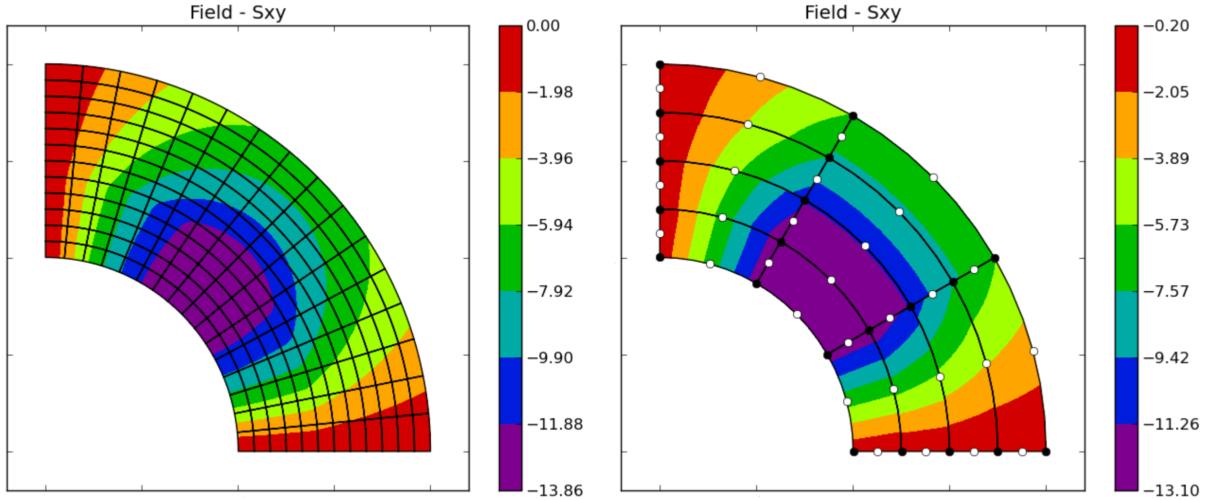


Figure 5.47. Horizontal displacement (dx) results for the FEM model (a) and the GFEM model (b).



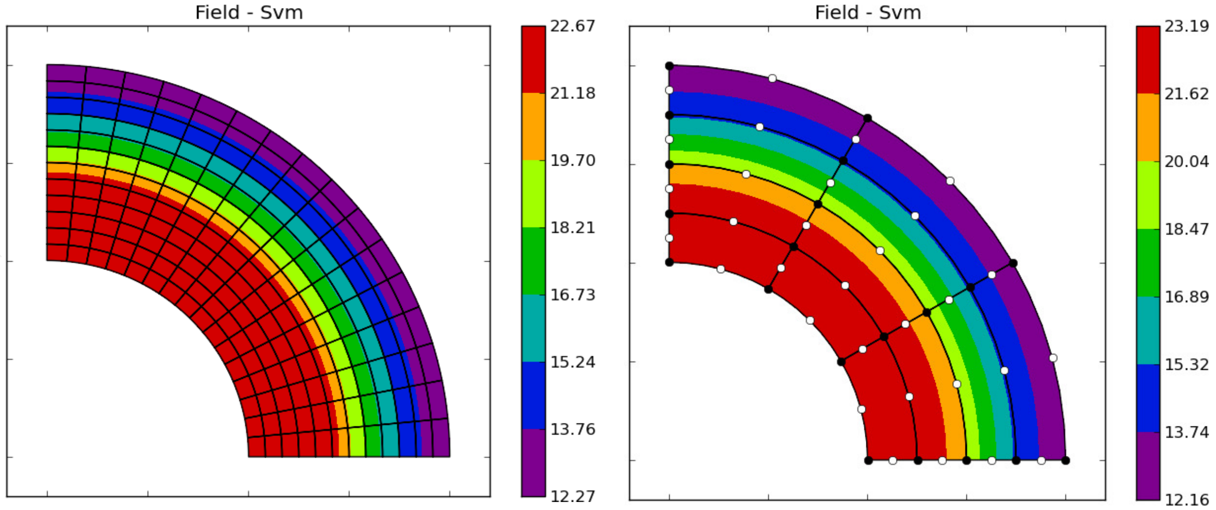
(a) *(b)*

Figure 5.48. Horizontal stress (Sxx) results for the FEM model (a) and the GFEM model (b).



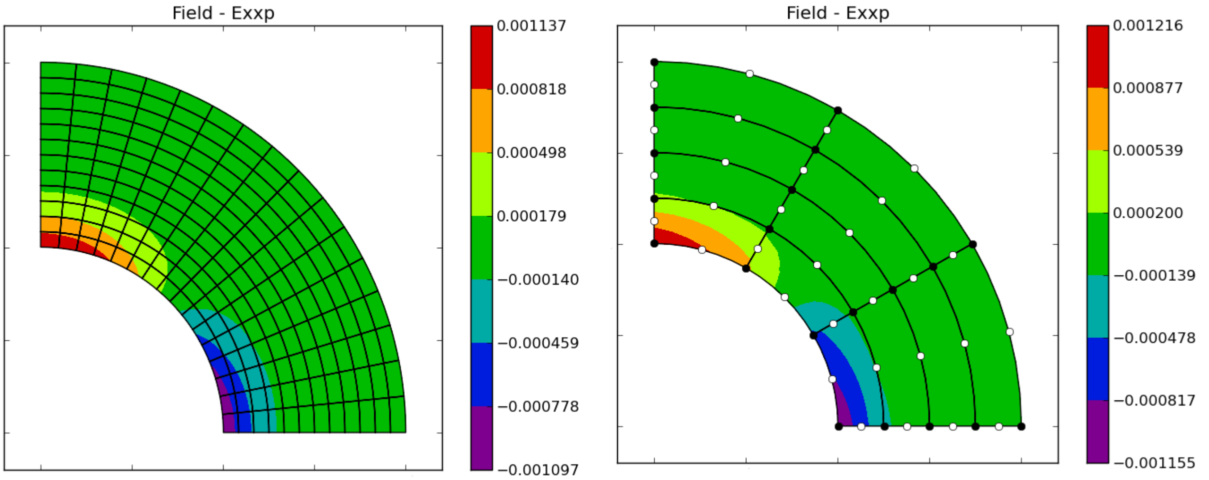
(a) *(b)*

Figure 5.49. Shear stress (Sxy) results for the FEM model (a) and the GFEM model (b).



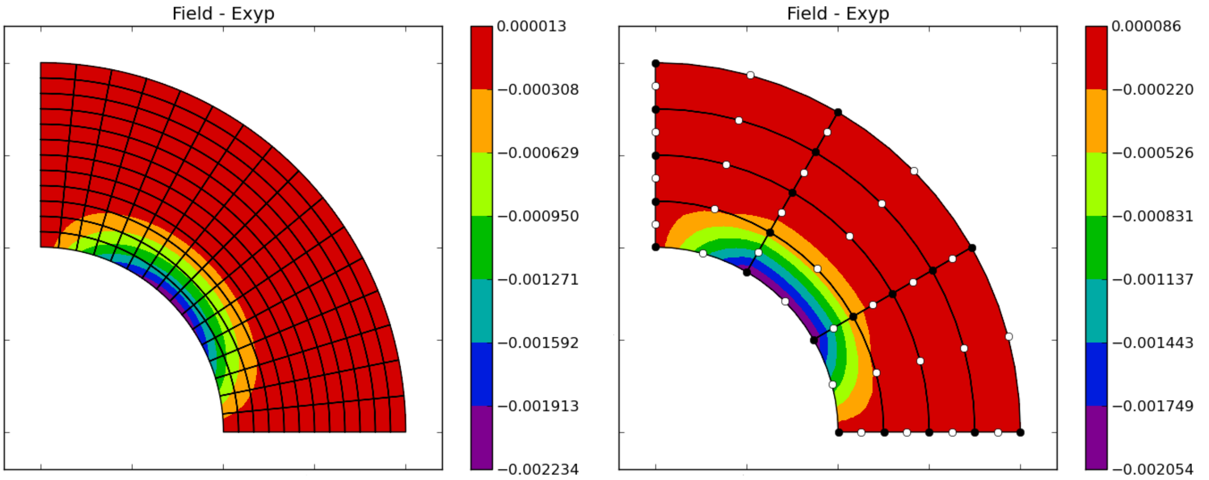
(a) *(b)*

Figure 5.50. Von Mises stress (Svm) results for the FEM model (a) and the GFEM model (b).



(a) *(b)*

Figure 5.51. Plastic strain in horizontal direction (Exxp) results for the FEM model (a) and the GFEM model (b).



(a) *(b)*

Figure 5.52. Plastic shear strain component (Exyp) results for the FEM model (a) and the GFEM model (b).

Taking into account the difference in the number of elements in the meshes, one observes that the GFEM model represents all the presented fields with sufficient accuracy.

It is worth to comment that for the GFEM model, the nodes represented in white color are only used by the geometric partition of unity. Therefore, this nodes present no degrees of freedom related to them.

5.5. Simple bar (small-strain plasticity)

Advancing in the elastoplastic material model evaluation, a simple bar example is stated, aiming to evaluate the computational code capability of representing both bi-linear and nonlinear hardening laws described in Chapter 3. The ability to correctly model the elastic behavior in the unloading stage of cyclic loads is also evaluated.

The proposed bar length is 5 (dimensionless). A unitary are (1x1) cross section is adopted. The proposed mesh is defined by 10 bi-linear (Q4) elements, as indicated in Figure 5.53.

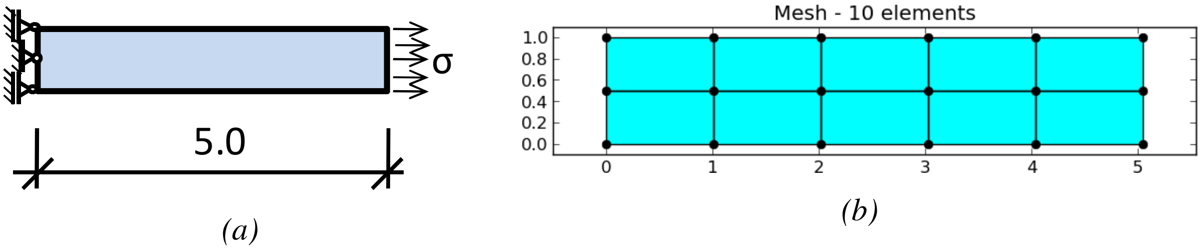


Figure 5.53. Simple bar structure (a) and proposed mesh for modelling it (b).

For the bi-linear hardening material the adopted material properties are indicated in Table 5.3. The stress/strain relation both for the FEM and the GFEM model, is indicated in Figure 5.54.

Table 5.3. Material properties for the proposed bi-linear hardening material (dimensionless).

Young's modulus	7000.00
Poisson's ratio	0.3
Yield stress	10.00
Infinity stress	10.00
Isotropic hardening modulus	1.0
Exponential factor	-

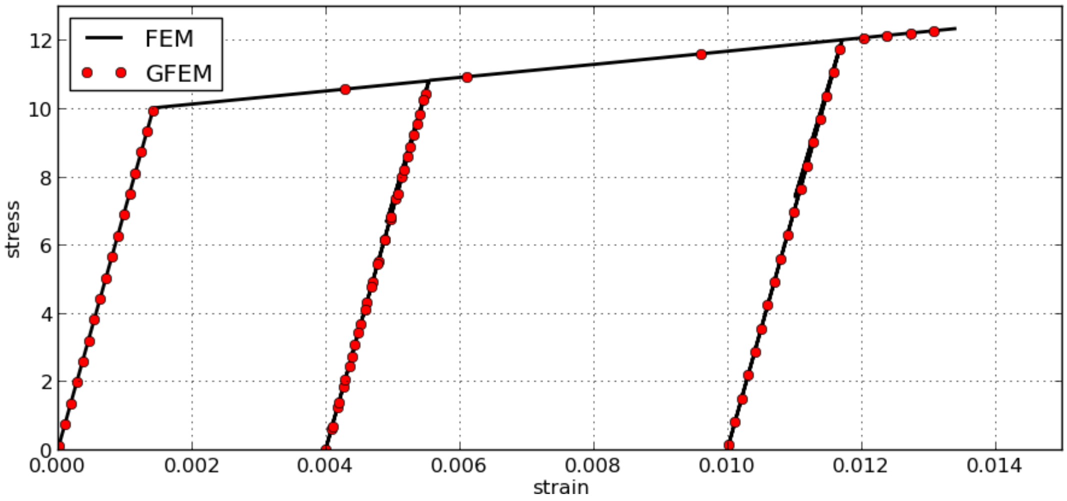


Figure 5.54. Bi-linear hardening elastoplastic material – stress/strain relation.

For the nonlinear hardening material the adopted properties are indicated in Table 5.4. The stress/strain relation, both for the FEM and the GFEM model, is indicated in Figure 5.55.

Table 5.4 Material properties for the proposed nonlinear hardening material (dimensionless).

Young's modulus	21000.00
Poisson's ratio	0.3
Yield stress	25.00
Infinity stress	40.00
Isotropic hardening modulus	2.5
Exponential factor	20.0

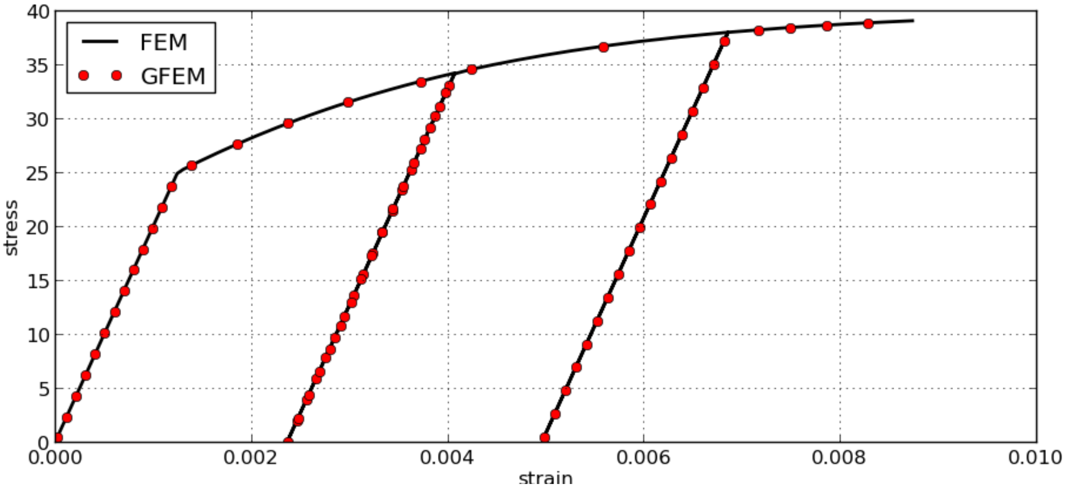


Figure 5.55. Nonlinear hardening elastoplastic material – stress/strain relation.

5.6. Von Mises truss (nonlinear kinematics)

Aiming to evaluate the solution of nonlinear kinematic stability problems, a simple truss example is stated, as indicated in Figure 5.56.

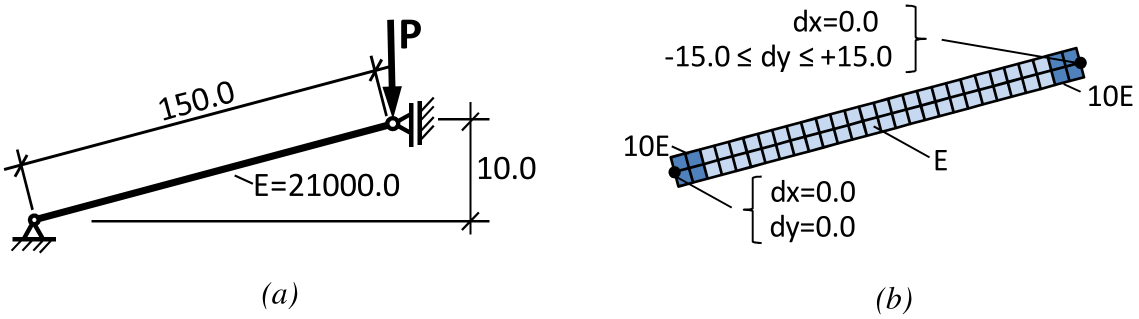


Figure 5.56. Simple truss problem scheme (a) and the respective proposed 2D FE model (b).

The proposed example is a limit point instability problem, also referred as von Mises truss (Bazant and Cedolin (2010)). The instability occurs due to the fact that an abrupt change in the bar configuration is observed, depending on the value of the load P.

The analytical solution for the problem can be easily derived (see Proença (2010)). The nonlinear equation relating the applied force P and the vertical displacement of the node in which such force is applied (v) is

$$P = \frac{EA}{2l_0^3}\left[\left(h+v\right)^3 + \left(h+v\right)a^2 - \left(h+v\right)l_0^2\right], a = \sqrt{l_0^2 - h^2} \, .$$

In such equation, l_0 is the initial length of the bar (150.0), h is the initial vertical distance between the bar edges (10.0), and A is the bar cross sectional area.

The 2D finite element (FE) model scheme used to model such structure is indicated in Figure 5.56 (b). A regular mesh of 120 bi-linear elements (Q4) is used to defined the bar geometry. The bar hinges are simulated by imposing displacements at single nodes at the edges of the mesh.

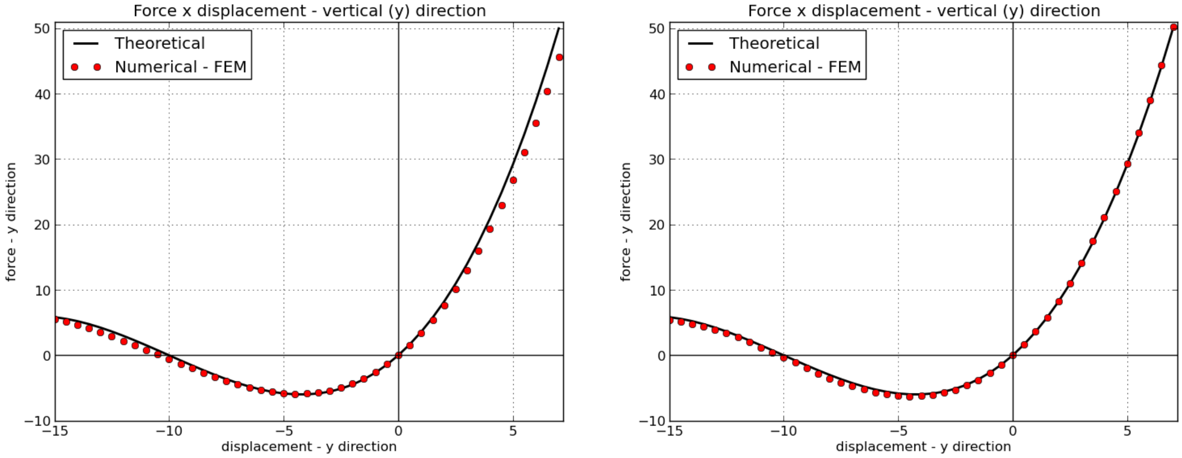
As the problem can present instabilities depending on the applied force, the force is applied by imposing displacements at the 'top hinge' node, and its value is evaluated by computing the vertical reaction observed in such node.

It is also important to take into account that in some steps of the nonlinear process, the forces applied over those nodes is very high, resulting in a crushing zone in their 'neighborhood'. This is the reason why the elements positioned near those nodes are defined using a Young's modulus ten times higher than the one defined for the rest of the bar (se Figure 5.56 (b)).

The results obtained for the FE model are presented in Figure 5.57.

The crushing at the 'hinge' nodes is evidenced by analyzing the deviation of the numerical solution from the theoretical one, in Figure 5.57 (a). The stress and strain concentration at such regions is also noticed by observing the von Mises stress field presented in the edge elements, as depicted in Figure 5.58.

On the other hand, the results present in Figure 5.57 (b) demonstrate that using a higher Young modulus at the edge elements allows one to achieve for the 2D FE model exactly the same results as the ones predicted by the theoretical solution.



(a) *(b)*

Figure 5.57. Vertical force/displacement relation obtained for 2D FE model; (a) theoretical solution presenting deviation due to crushing at the 'simulated' hinge; (b) solution achieved when 'reinforced' elements are positioned at the edges of the mesh in which the displacements are applied.

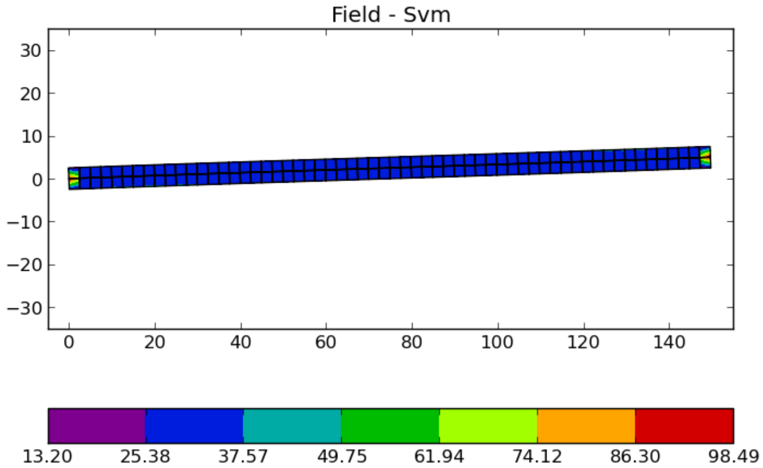


Figure 5.58. Von Mises stress resusts for the FE model (stress concentration at the edge elements).

Since the proposed FEM model already represents exactly the theoretical results, we propose a mesh 'coarsen' in order to evaluate the GFEM model.

The proposed mesh for the GFEM model is depicted in Figure 5.59.

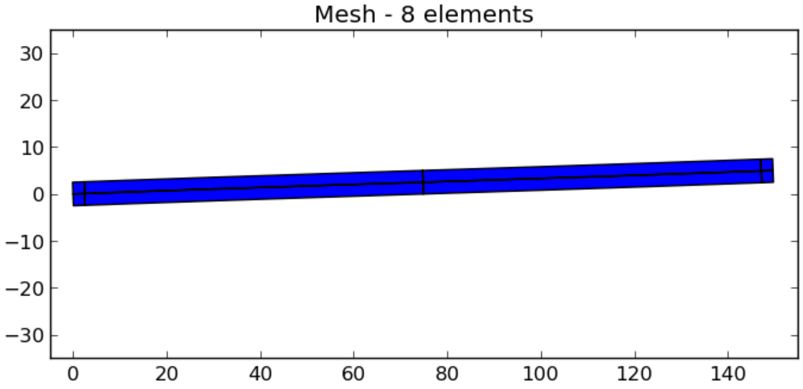


Figure 5.59. Eight bi-linear (Q4) element mesh proposed for the GFEM model.

The results achieved for both first and second degree shifted enrichment functions applied over all nodes, excluding the ones in which the boundary condition is applied, are indicated in Figure 5.60.

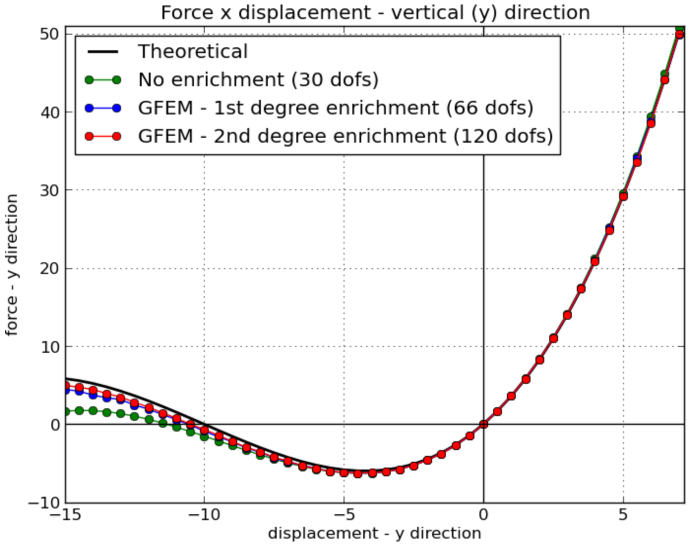
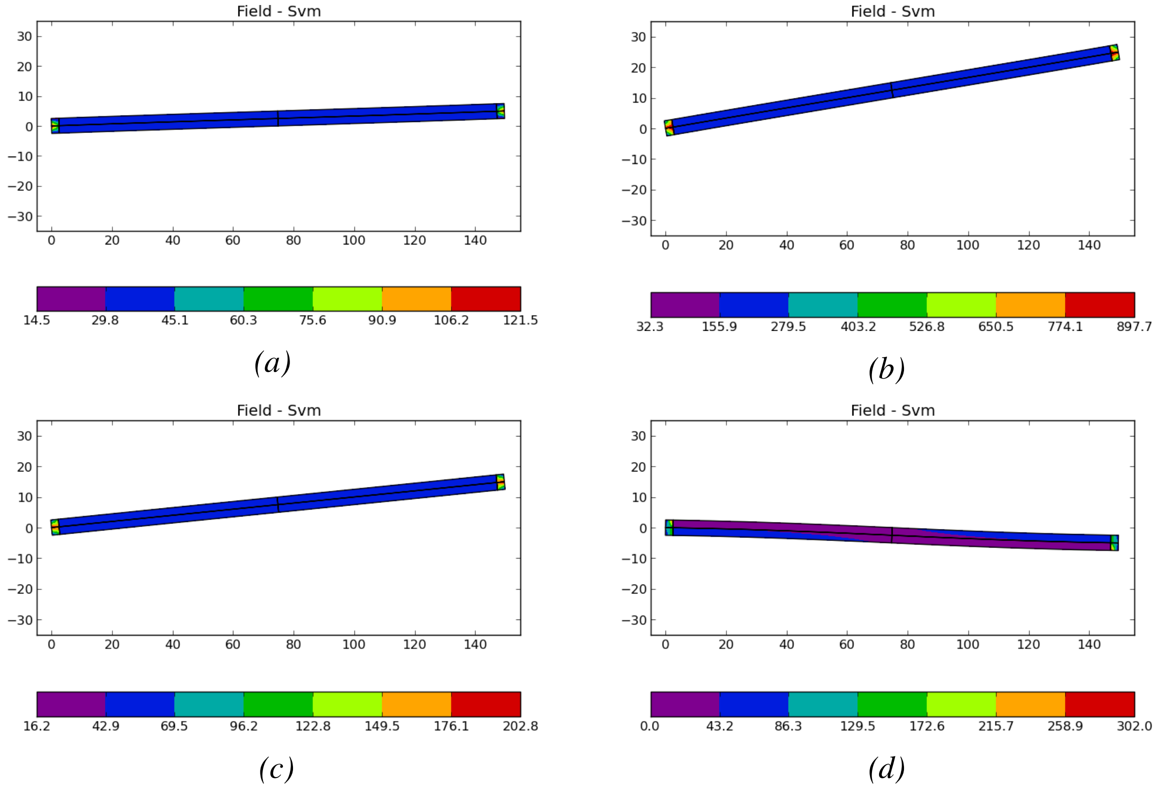


Figure 5.60. Vertical force/displacement relation obtained for 2D GFEM shifted enrichment functions model.

Figure 5.61 illustrates the GFEM model deformed equilibrium configuration for some steps of the proposed nonlinear problem.



(a)

(b)

(c)

(d)

Figure 5.61. GFEM model deformed equilibrium configuration at some steps of the solution.

5.7. Euler column (nonlinear elastic instability)

Continuing the evaluation of the GFEM model for the nonlinear kinematic model, one proposes the classical Euler column problem (bifurcation stability). The proposed problem adopted data and 2D plane model mesh scheme is depicted in Figure 5.62.

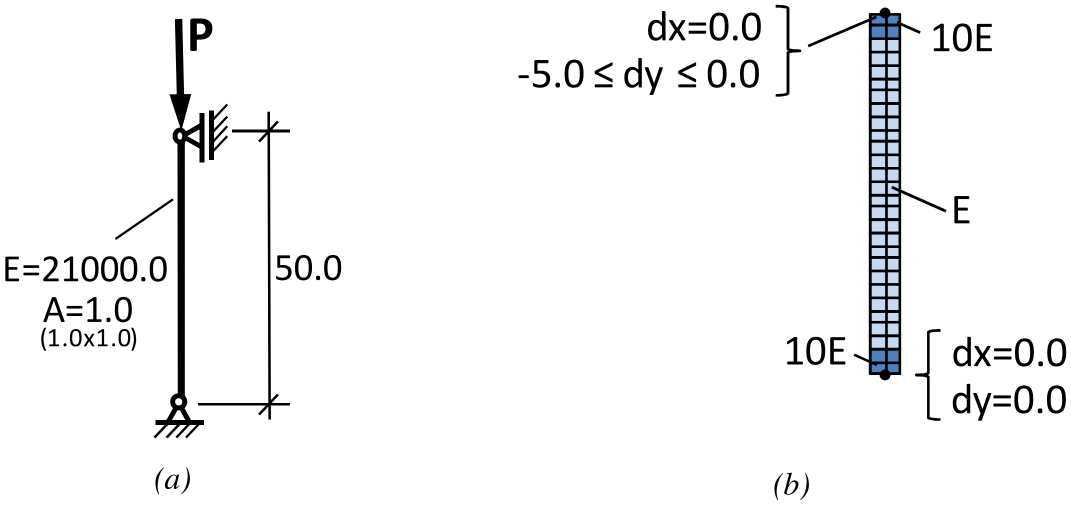


Figure 5.62. Euler column problem proposed data (a) and proposed 2D plane model (b).

The prismatic bar elastic buckling theory allows one to compute the critical load by

$$P_{cr} = \frac{\pi^2 EI}{L^2},$$

in which P_{cr} is the critical load, E is the Young's modulus, I is the prismatic bar moment of inertia and L is its length (see Timoshenko and Gere (1985)). For the proposed data, the computed value for P_{cr} is 6.9087 (dimensionless).

Such critical load can be used as a reference for the proposed nonlinear elastic instability problem, which occurs when the model presents, for instance, an initial geometric imperfection.

After the critical load is reached, one observes an abrupt lateral displacement (equilibrium bifurcation), which can be estimated by means of

$$d(P) = \frac{8\sqrt{\frac{P}{P_{cr}} - 1}}{\pi^2 L}, P > P_{cr},$$

in which, d is the lateral displacement and P is the applied force.

Similarly to the previous example, the 1D model (prismatic bar) hinge is simulated by imposing displacements in a single node of the 2D plane model mesh, which can cause crushing and demands the same strategy of using a different Young's modulus for the elements surrounding such nodes (edge of the mesh). Due to stability issues, the force is applied by enforcing vertical displacements at the node, and it is evaluated by computing the reaction force, in a similar fashion that it was performed for the previous example.

For the FE model, a 300 (2x150) regular bi-linear (Q4) elements mesh is defined. Such mesh is defined such that the initial position of the nodes defines a structural element with a deviation ‘e’ of the perfectly straight geometric configuration. The graphics relating the applied vertical force to the lateral displacement at the middle height of the column, for several different values of geometric ‘imperfections’ e, is depicted in Figure 5.63.

The results of Figure 5.63 demonstrate that the FE model represents the global behavior expected for the elastic stability problem, and that the nonlinear kinematic model approaches to the results expected by the elastic buckling theory.

Indeed, the accuracy of the results can be improved by means of a polynomial refinement, i.e., defining a FE model of 300 elements bi-quadratic (Q8) elements. Keeping all the other parameters already used for the bi-linear elements mesh, the results achieved for the Q8 elements mesh are depicted in Figure 5.64.

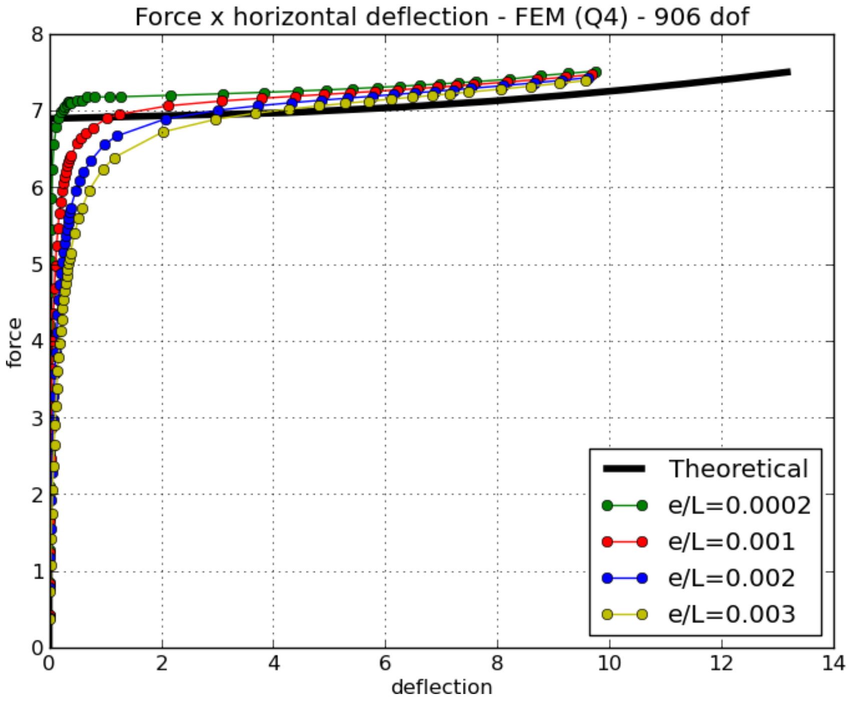


Figure 5.63. Vertical force x horizontal (lateral) displacement at the middle height of the column, for several initial geometric ‘imperfections’ e - Q4 (bi-linear) elements.

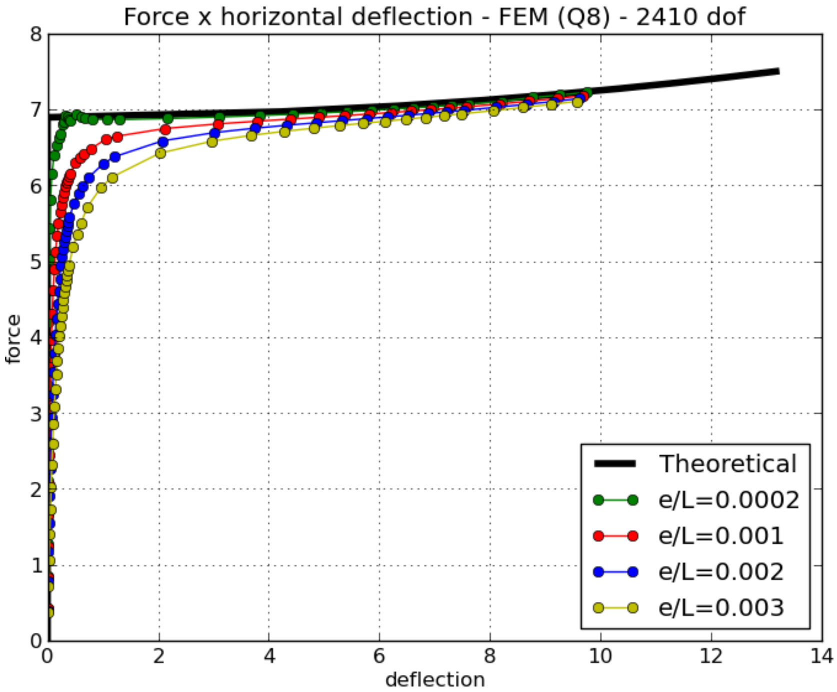


Figure 5.64. Vertical force x horizontal (lateral) displacement at the middle height of the column, for several initial geometric 'imperfections' e - Q8 (bi-quadratic) elements.

The data presented in Figure 5.64 reinforces the expectations that for the smaller value of geometric imperfection, the results are close to the ones predicted by means of the elastic buckling theory. This is the value of imperfection to be used in the GFEM model.

Advancing for the GFEM model evaluation, the same strategy of mesh 'coarsen' is applied for the present example, since the bi-linear mesh elements results are already close to the theoretical reference ones.

A mesh of 44 (2x22) bi-linear elements are defined for GFEM model. Shifted functions are used for the model enrichment, excluding the nodes in which the displacements are applied. Both first and second degree enrichments are tested.

It is worth to mention that such mesh results in elements measuring 0.5x0.25 length units, i.e., the relation of the elements size is 5, which can compromise its performance for the FEM. The objective here is to evaluate the GFEM gains in such situation, based on the results observed in the linear elastic results presented in the example 5.2.

The GFEM results are indicated in Figure 5.65. As expected, the not enriched model, equivalent to the FEM, presented poor quality results, while both first and second degree enrichments presented accurate results.

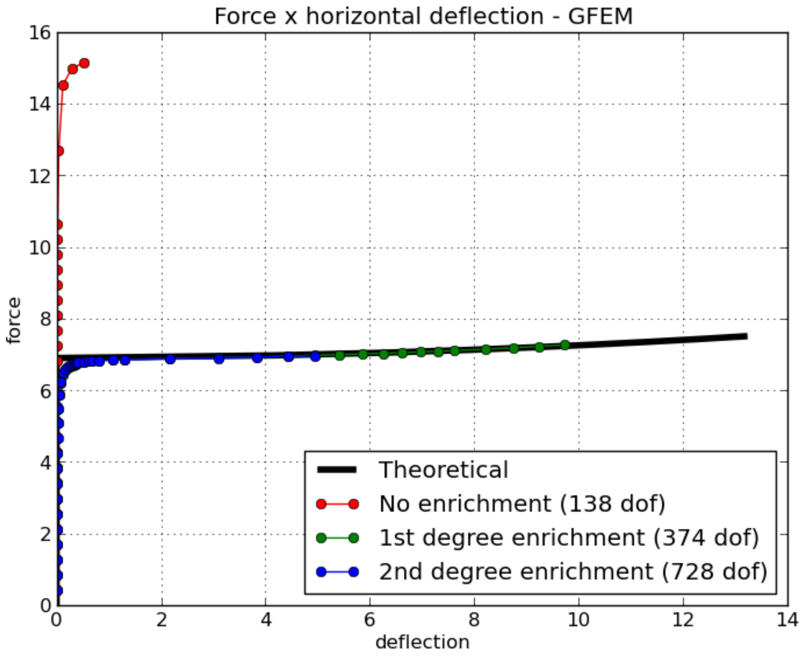
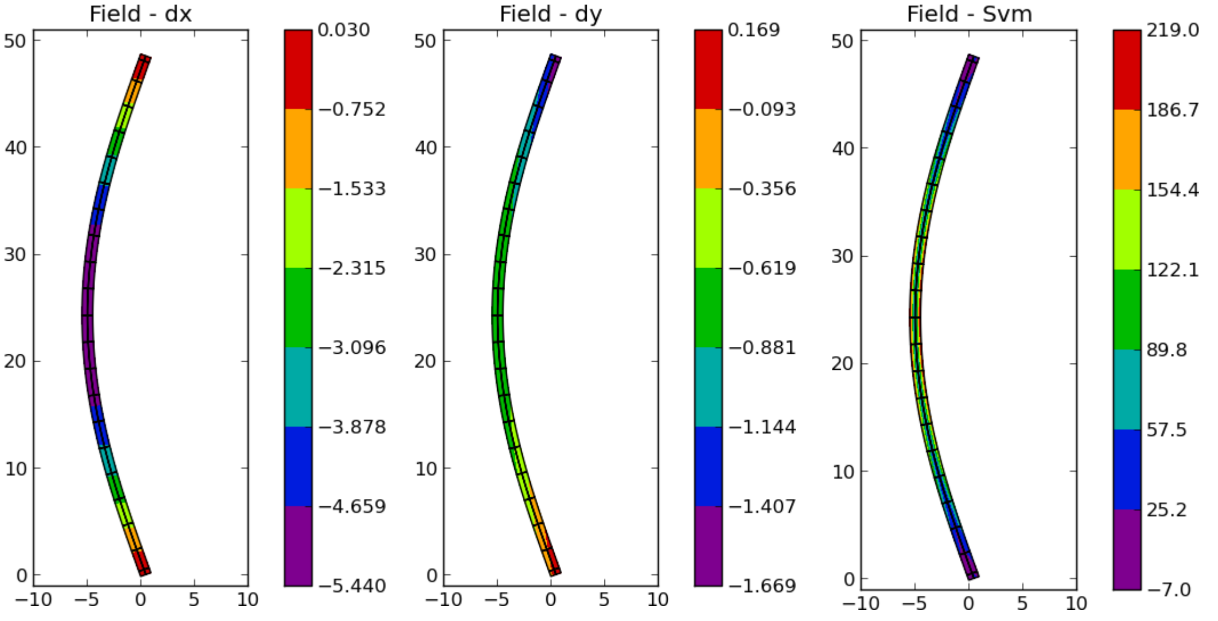


Figure 5.65. Vertical force x horizontal (lateral) displacement at the middle height of the column – GFEM model

It is also worth to observe that the GFEM model present a considerably reduced number of degrees of freedom, when compared to the tested FEM models.

Aiming to illustrate the deformed configuration of the model for a force higher than the critical load value, the resulting horizontal and vertical displacement, and the von Mises stress field are presented in Figure 5.66.



(a) *(b)* *(c)*

Figure 5.66. Deformed equilibrium configuration of the proposed GFEM Euler column model: (a) horizontal displacements; (b) vertical displacements; (c) von Mises stress field.

5.8. Neo-Hookean solid simple bar (hyperelastic material)

The problems proposed in the two last examples present relatively large displacements. Despite of this fact, the material stress/strain relation considered for the model is constant.

On the other hand, when larger strains are observed, the Saint-Venant Kirchhoff model can fail in converging, demanding constitutive model supporting large strains (hyperelasticity). The simplest of them is the Neo-Hookean material, already discussed in Chapter 3.

In order to illustrate such hyperlastic behavior, the same mesh employed in example 5.5 is used to simulate the force and displacement relation for the Neo-Hookean material model. The adopted material parameters are: Young modulus equal to 1000.00 and Poisson's ratio equal to 0.499. The computed results are presented in Figure 5.67.

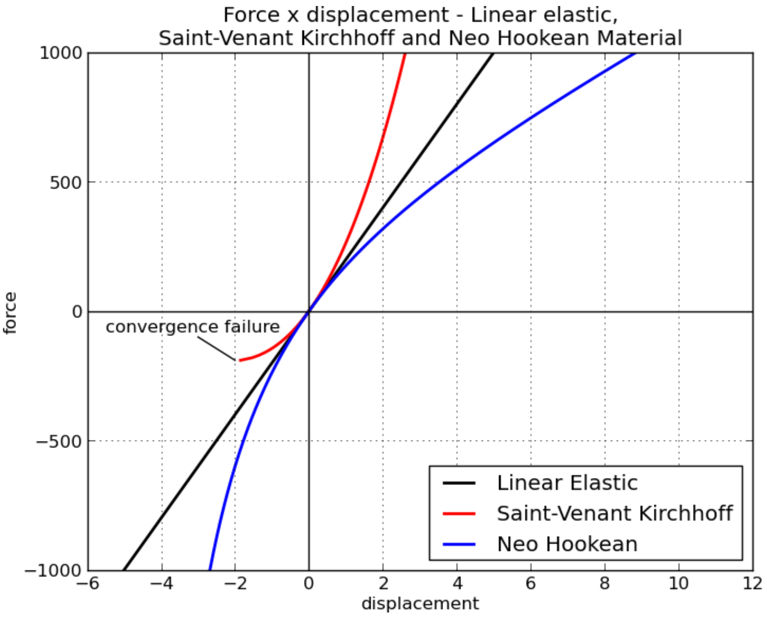


Figure 5.67. Force x displacement relation for a simple bar model, for linear elastic, Saint-Venant Kirchhoff and Neo-Hookean material constitutive models.

The same model is also used to evaluate Neo-Hooekan material structural response when the Poisson ratio's changes. The results are indicated in Figure 5.68.

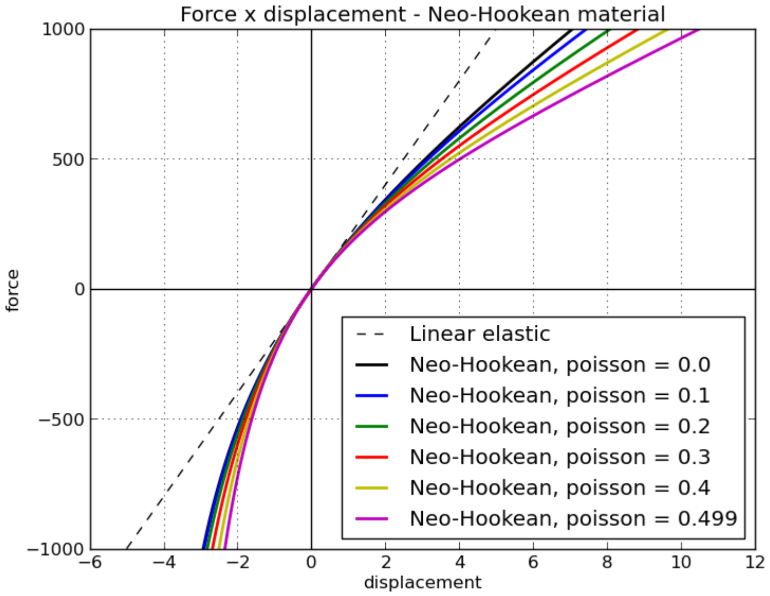
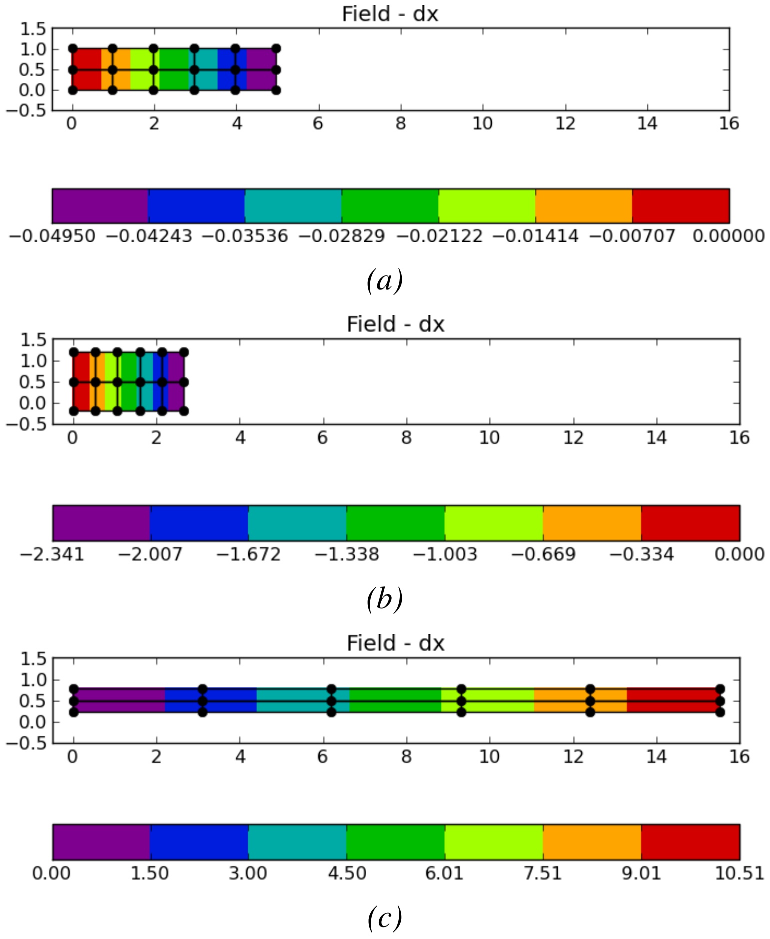


Figure 5.68. Hyperlastic bar for different values of Poisson's ratio.

Finally, Figure 5.69 illustrates different deformed configuration of the hyperelastic bar model.



(a)

(b)

(c)

Figure 5.69. Deformed configuration of the hyperlastic bar: (a) small shortening; (b) large shortening and (c) large stretching.

5.9. Simple Signorini contact test (rigid obstacle contact)

A simple example of contact between a square shaped 2D solid and a rigid support is proposed. The frictional forces observed between the solid and the rigid support are neglected, resulting in the so-called Signorini contact problem (Laursen (2002)). The proposed geometry, material parameters (dimensionless) and additional data are presented in Figure 5.70.

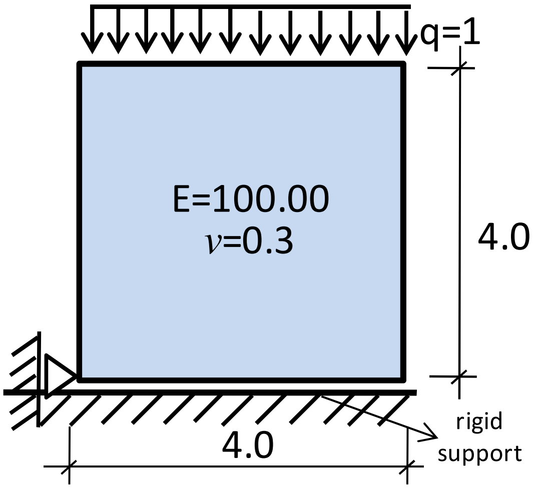


Figure 5.70. Proposed simple Signorini contact problem scheme.

One starts evaluating the node-to-segment contact element derived in Chapter 3, for the FE models. The proposed model is a 16 (4x4) quadrilateral elements mesh, defined for bi-linear (Q4), bi-quadratic (Q8) and bi-cubic (Q12) serendipity elements. The vertical displacement results for such models are shown in Figure 5.71.

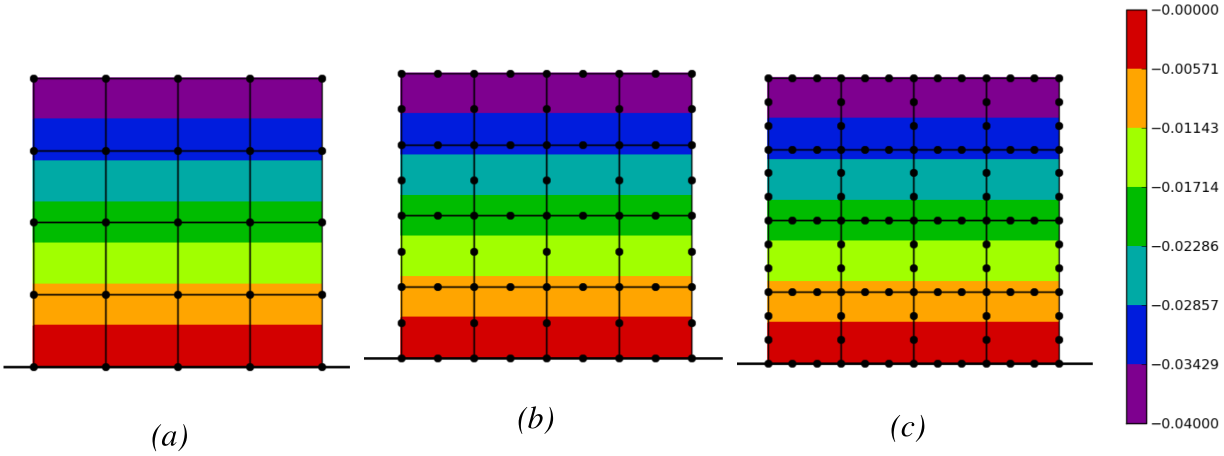


Figure 5.71. Vertical displacement field (dy) for (a) bi-linear, (b) bi-quadratic and (c) bi-cubic FE elements models - node-to-segment contact elements.

As it can be noticed, for the Signorini problem, the FEM vertical displacement results achieved using node-to-segment contact elements are accurate, for all the tested elements.

Also the other fields (displacements and stresses) are regular, as expected for the proposed problem.

On the other hand, for the GFEM, such results are not accurate. Figure 5.72 indicates the results for a GFEM model in which first and second degrees shifted enrichment functions are applied to all the solids nodes.

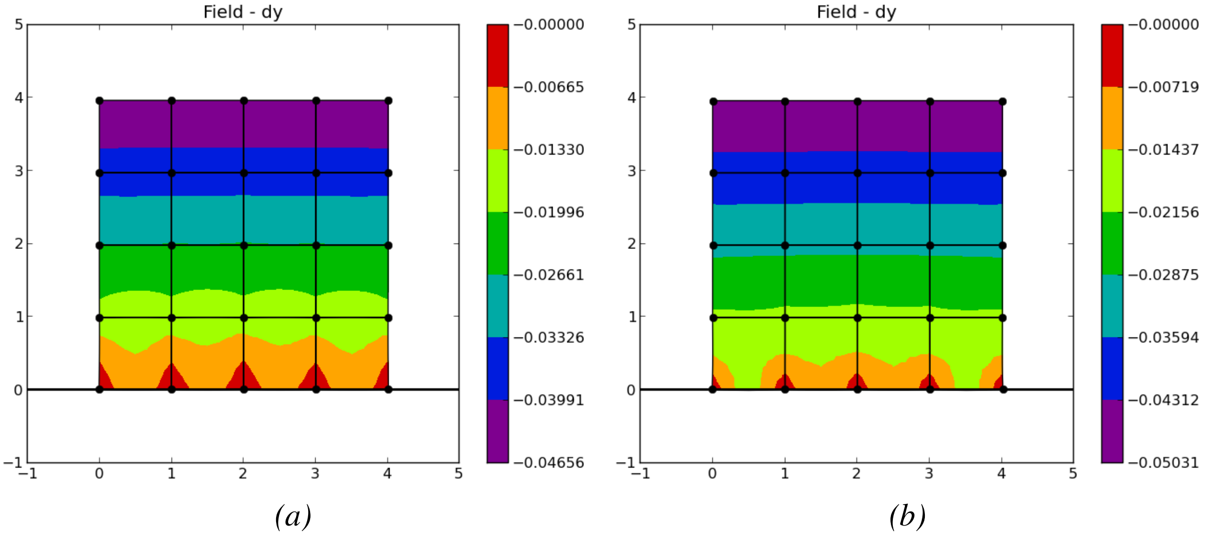


Figure 5.72. Vertical displacement field for a first (a) and second (b) degree enrichment function GFEM model, using node-to-segment contact elements.

The field deviations observed in Figure 5.72 is due to penetration in the elements side, as it can be observed in a closer view of the contact region, as shown in Figure 5.73.

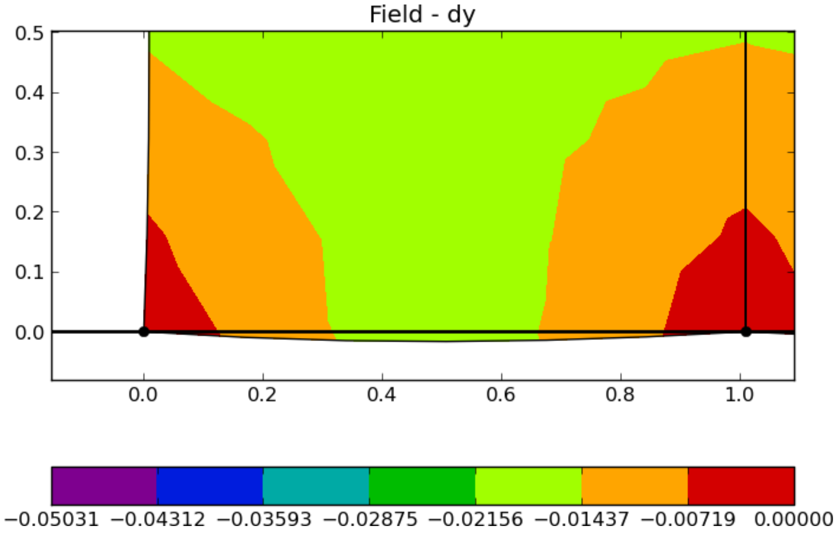


Figure 5.73. Closer view of the contact region of the GFEM model computed using node-to-segment contact elements.

This effect is similar to the one discussed in example 5.1, in which the displacement boundary conditions must be imposed over the whole element side if enrichments are applied over the boundary nodes.

The distribution of gap values and contact tractions along the contact interface are depicted in Figure 5.74 and Figure 5.75, respectively.

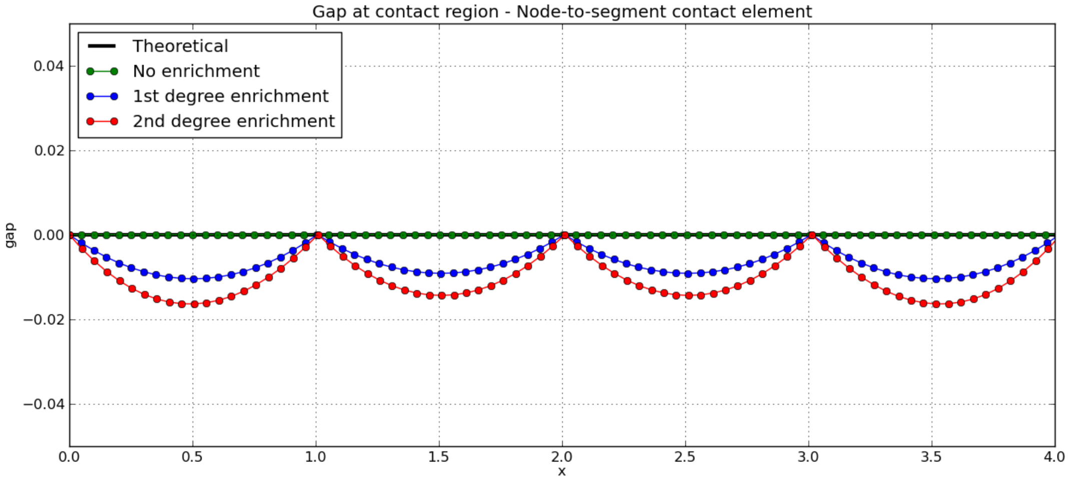


Figure 5.74. Contact gaps at the contact region for the GFEM model.

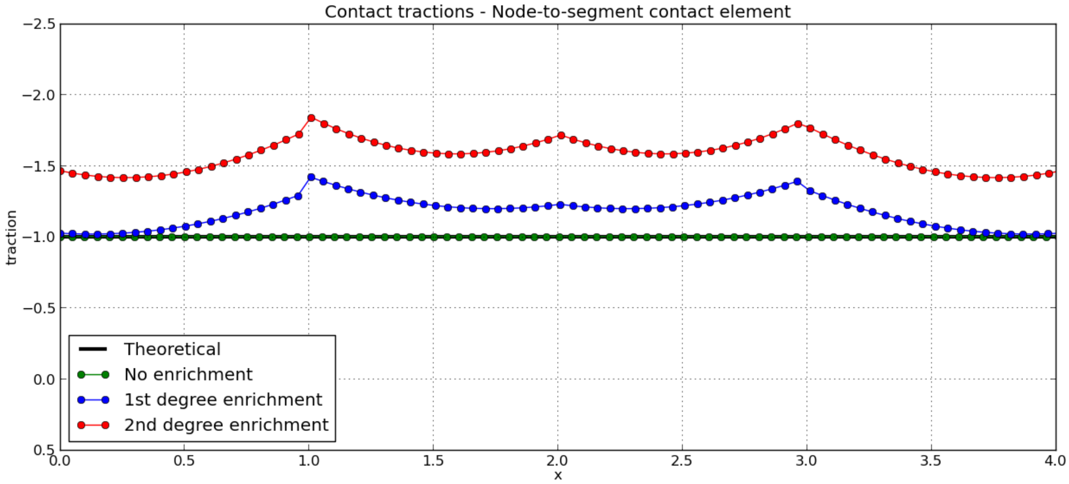


Figure 5.75. Contact tractions at the contact region for the GFEM model.

Like it happens for the displacement boundary conditions, such results enforce the fact that for the GFEM, the contact conditions enforcement needs to be performed in an element wise fashion, i.e., the conditions must be enforced along the whole element side, when such nodes are enriched. In this context, the segment-to-segment contact elements derived in Chapter 3 are the ones that must be used for accurately computing GFEM contact models.

In fact, when applied to the proposed problem, the segment-to-segment contact elements resulted in the expected displacement and stresses fields, preventing penetrations at the contact region in between nodes, and recovering the contact vertical stress field (Syy). In Figure 5.76, a closer view of the same region indicated in Figure 5.73 is illustrated, for the case of a shifted second degree enrichment GFEM model, using the derived segment-to-segment contact element.

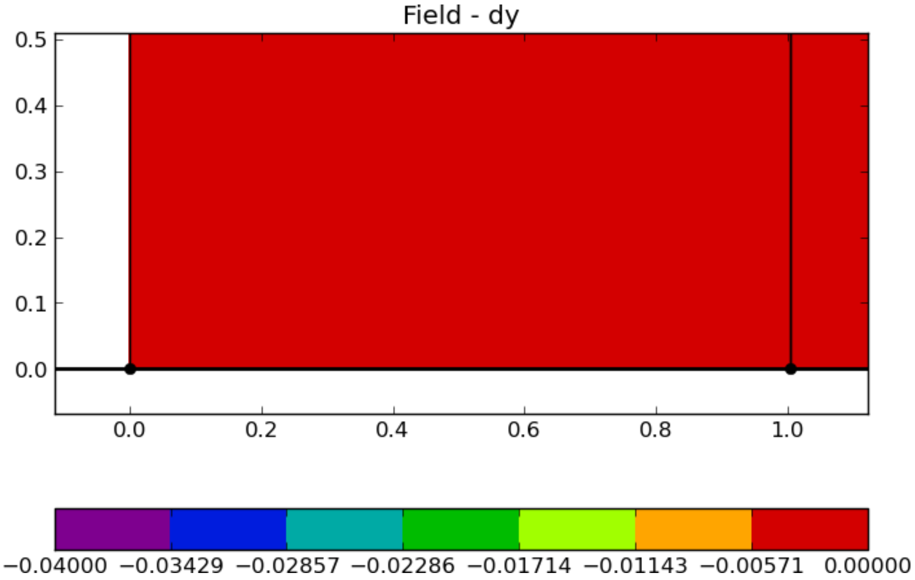


Figure 5.76. . Closer view of the contact region of the GFEM model computed using segment-to-segment contact elements.

It is important to mention that both penalty and Lagrange multipliers based contact elements resulted in similarly accurate results.

For the case o Lagrange multipliers contact elements, for all the tested enrichments, the same degree of enrichment was also applied to enrich the Lagrange multipliers field.

5.10. Contact patch-test (deformable bodies contact)

Continuing the contact analysis evaluation, a simple patch test of two rectangular deformable bodies is proposed, aiming to evaluate the node-to-segment and the segment-to-segment contact elements, both for the FEM and the GFEM. The contact patch test data is presented in Figure 5.77.

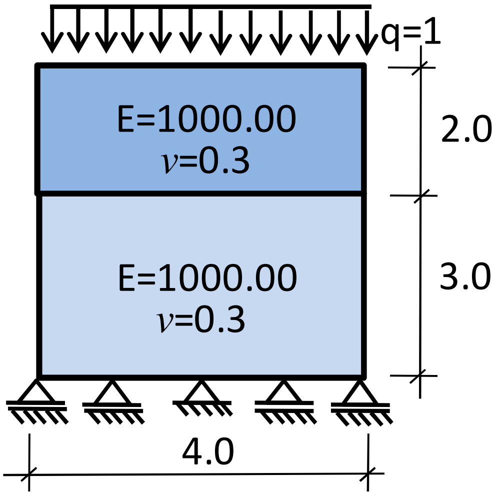
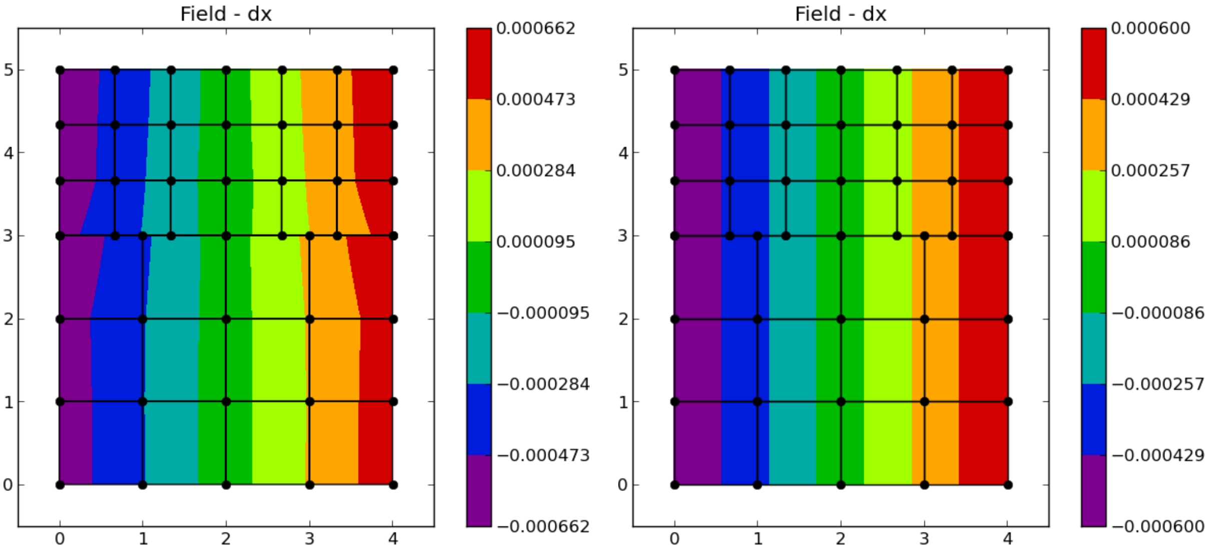


Figure 5.77. Contact patch test scheme.

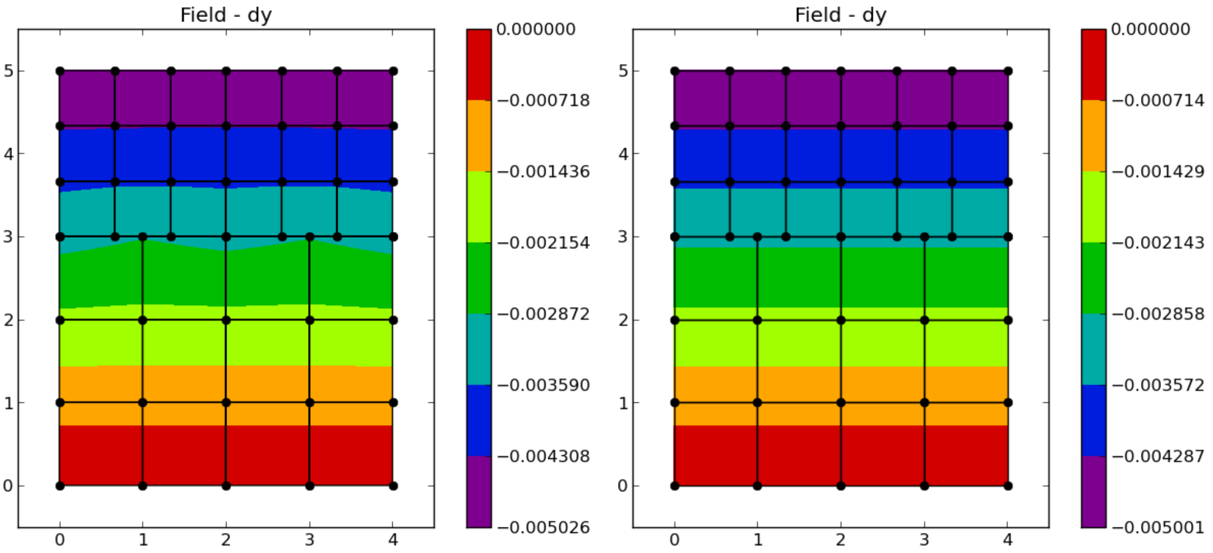
The proposed example consists of two rectangular solids composed of the same material, one positioned over the other, such that contact is observed between them. The adopted meshes are defined such that the elements are not coincident at the contact region.

For the FEM model, the horizontal and vertical displacement fields obtained using node-to-segment and segment-to-segment contact elements are presented in Figure 5.78 and 5.79.



(a) *(b)*

Figure 5.78. Horizontal displacement (dx) field obtained using (a) node-to-segment and (b) segment-to-segment (mortar) contact elements.

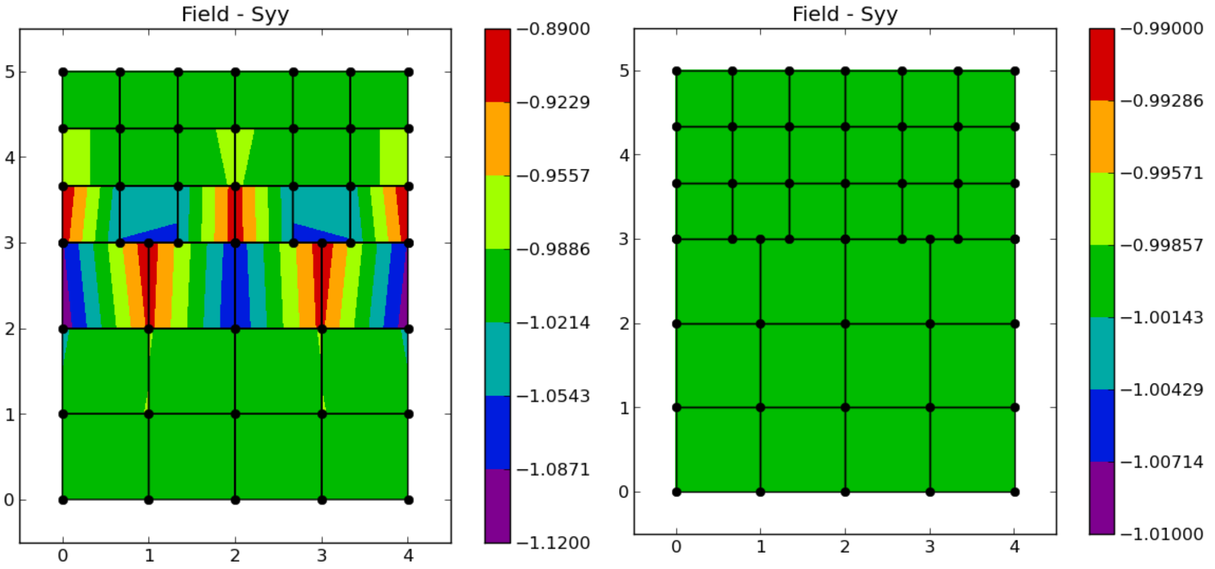


(a) *(b)*

Figure 5.79. Vertical displacement (dy) field obtained using (a) node-to-segment and (b) segment-to-segment (mortar) contact elements.

One observes that the use of the derived node-to-segment contact elements does not result in regular displacement fields, as expected for the proposed patch test. In spite of that, the mortar contact element results in regular displacement fields and accurate results.

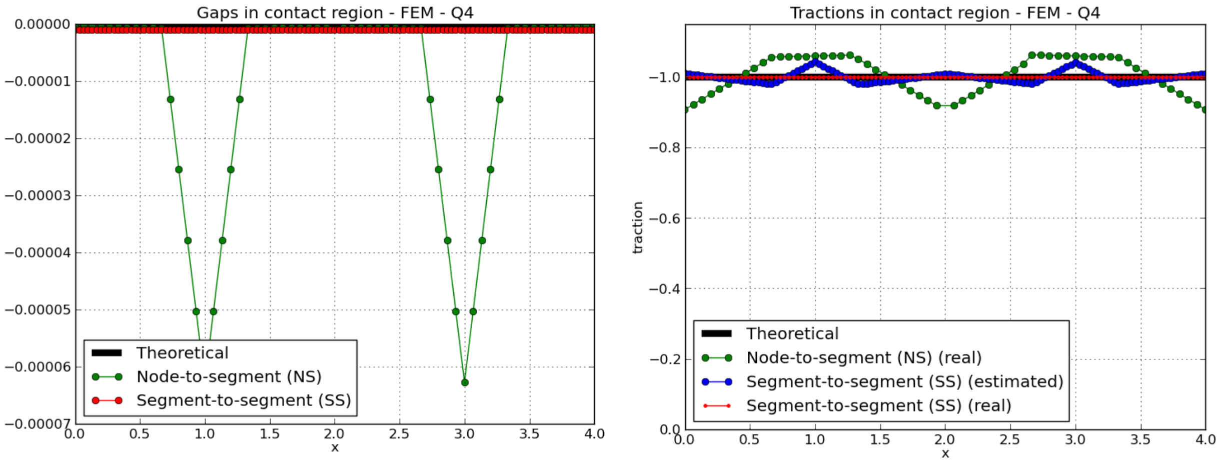
Obviously the same field perturbations are reflected in the stresses fields. The vertical stress (Syy) component fields for both contact elements are depicted in Figure 5.80.



(a) *(b)*

Figure 5.80. Vertical stress field (Syy) obtained using (a) node-to-segment and (b) segment-to-segment (mortar) contact elements.

The distribution of gap values and contact tractions along the contact region are depicted in Figure 5.81.



(a) *(b)*

Figure 5.81. Contact gaps (a) and traction (b) for model computed using node-to-segment contact elements and segment-to-segment (mortar) contact elements – Lagrange multipliers formulation.

Regarding the gaps, one observes that, contrary to the mortar contact segment, the node-to-segment contact elements model presents variations in the gap along the contact interface, especially near the nodes at the position x=1.0 and x=3.0.

Before advancing, it is important to explain the terms ‘real’ and ‘estimated’ associated to the traction results presented in Figure 5.81(b). Here one refers to ‘estimated tractions’ for the values computed in the contact elements, in accordance with relations (3.158) and (3.159). The term ‘real traction’ refers to the tractions computed along the element side using the Cauchy relation.

As observed, even though the estimated traction varies along the contact interface, the computed real tractions are quite constant and close to the value expected for the patch test.

Since the node-to-segment results are not accurate for the FEM, it is clear that such contact elements are not suitable for GFEM models. However, for illustrative purposes, the gaps and tractions obtained in such situation, when the nodes at the contact interface are enriched, are presented in Figure 5.82.

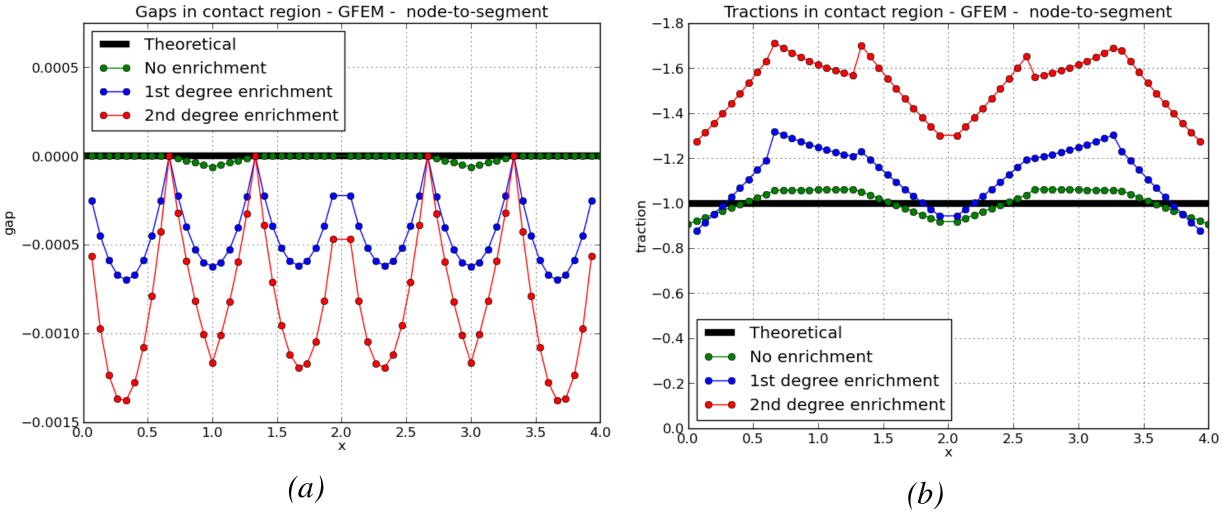
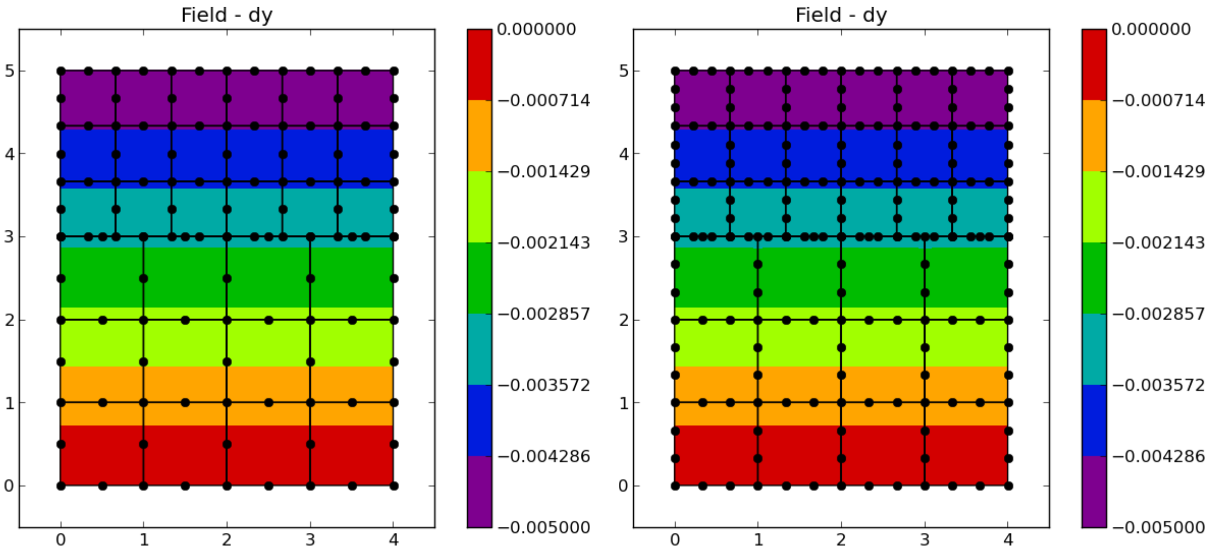


Figure 5.82. Contact gaps (a) and traction (b) for model computed using node-to-segment contact elements – GFEM shifted enrichment functions- penalty formulation.

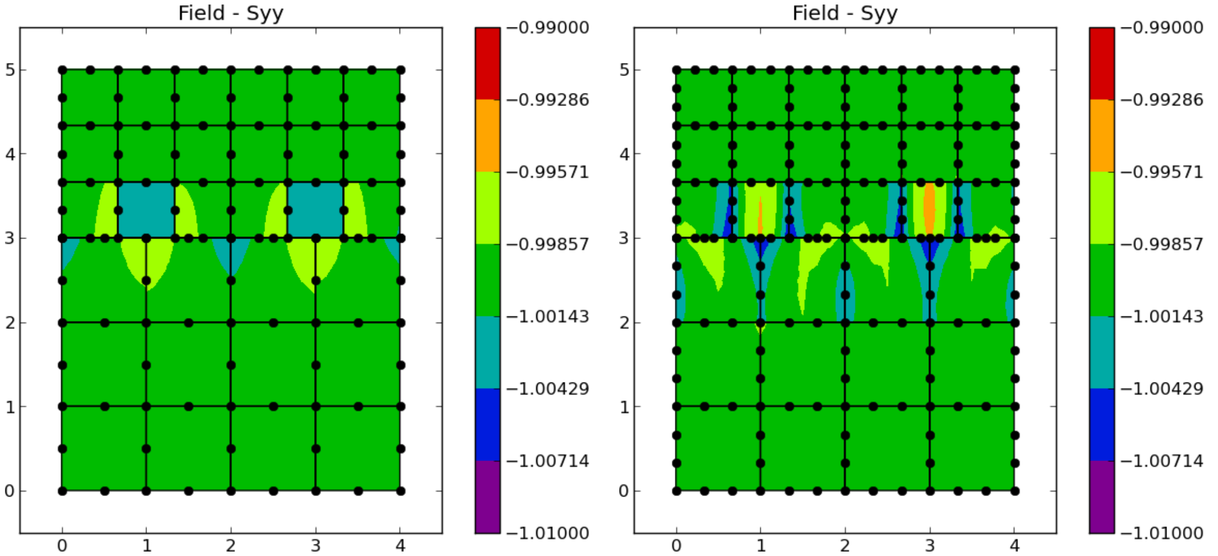
Once the node-to-segment is evaluated, the mortar segment-to-segment contact elements are also evaluated for the FEM higher order interpolations. For doing so, the patch test problem is remodeled for meshes of bi-quadratic (Q8) and bi-cubic (Q12) finite elements. The vertical displacement fields are presented in Figure 5.83.



(a) *(b)*

Figure 5.83. Vertical displacement (dy) field for bi-quadratic (a) and bi-cubic (b) elements meshes – mortar segment-to-segment contact element – Lagrange multipliers formulation.

In spite of the good displacement results, it is possible to verify that the vertical stress component field (Syy) is not constant for the FEM higher order interpolations (Figure 5.84).

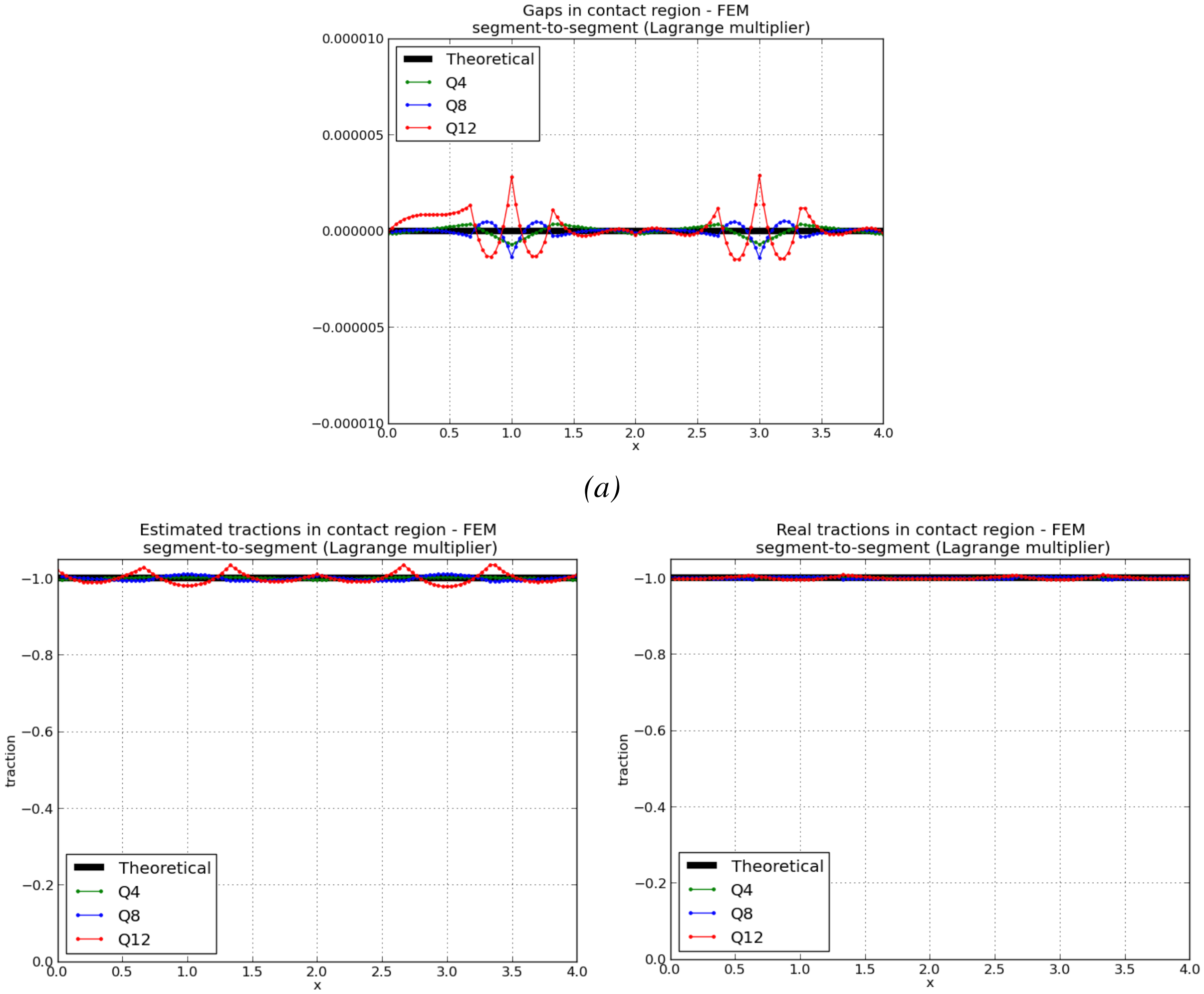


(a) *(b)*

Figure 5.84. Vertical component stress field (Syy) for bi-quadratic (a) and bi-cubic element meshes (b) – mortar contact element – Lagrange multipliers formulation.

It is worth mentioning that such oscillations were not caused by poor numerical integrations, since several integration schemes containing sufficient points were evaluated for these models.

The contact gaps and tractions distribution along the contact interface, for the mortar contact element based on the Lagrange multipliers method, is depicted in Figure 5.85.



(a)

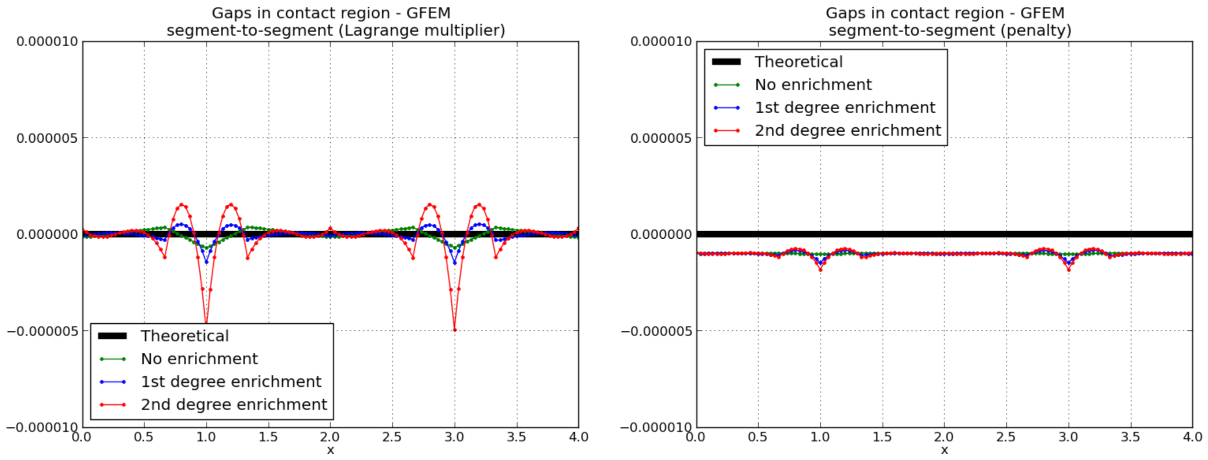
(b) *(c)*

Figure 5.85. Contact gaps (a), estimated tractions (b) and real tractions (c) distributions along the contact area – mortar segment-to-segment contact element – Lagrange multipliers formulation.

One observes that the gaps variations over the contact interfaces are larger for the higher degrees interpolation, especially the cubic one. Regarding the estimated traction value, the same behavior is observed, i.e., its distribution varies more for the cubic interpolation.

Finally, the evaluation of the results achieved for the proposed enriched mortar contact elements is performed, using the bi-linear elements mesh presented at the beginning of the current example (see Figure 5.78, Figure 5.79 and Figure 5.80).

Both the penalty and the Lagrange multipliers based formulation were evaluated. For the first one, a 10^6 value is used for the penalty parameter. For the Lagrange multiplier, the same degree employed for enriching the displacement fields is used for enriching the Lagrange multipliers field. Likewise in the FEM models, six integration points are used in the mortar elements computation. Shifted enrichment functions were applied. The contact gaps distribution results are presented in Figure 5.86.

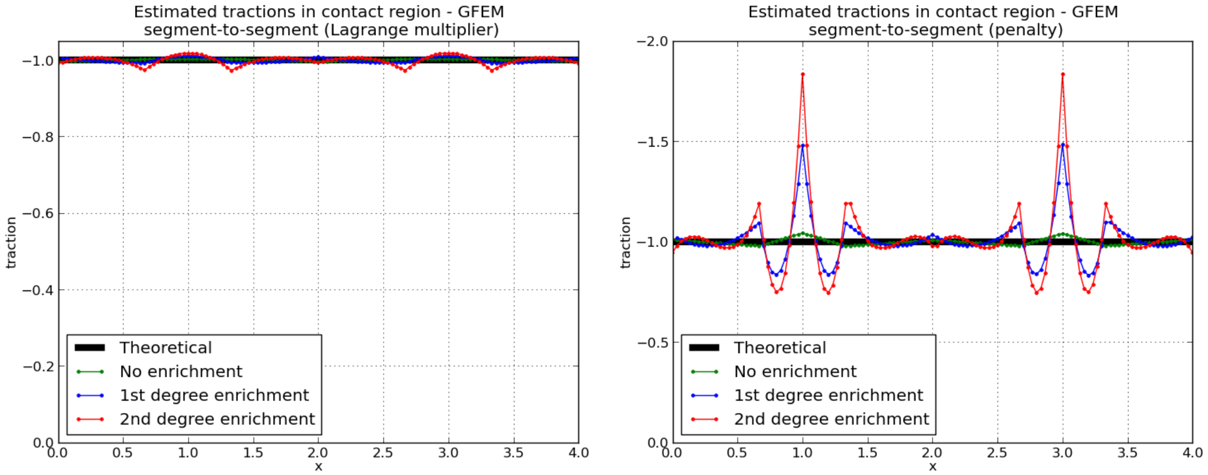


(a) *(b)*

Figure 5.86. Contact gaps distribution for shifted enrichment functions GFEM model, both for Lagrange multipliers (a) and penalty (b) formulation.

For the presented example, the enriched mortar contact elements based in the Lagrange multipliers formulation seems to results in gaps that oscillate more than the ones observed for the penalty formulation. On the other hand, the mean gap value achieved by means of the Lagrange multipliers formulation is closer to zero, if compared to the mean gap value computed using the penalty method.

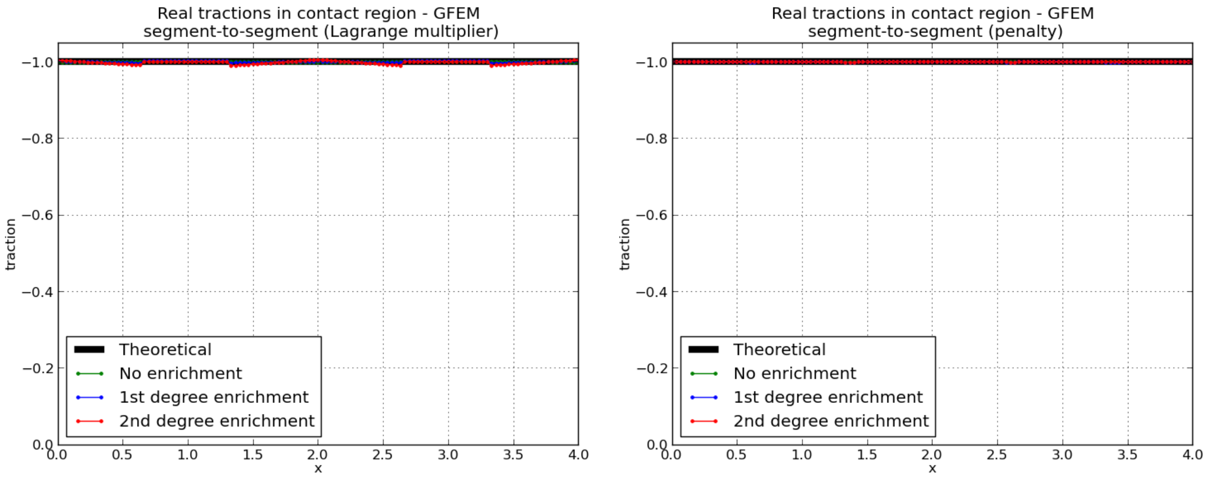
Regarding the estimated traction distribution, the Lagrange multipliers formulation seems to be sufficiently representative, especially for the first and second degree enrichments. For the penalty formulation, the estimated traction value varies considerably (see Figure 5.87 (b)).



(a) *(b)*

Figure 5.87. Contact estimated tractions distribution for shifted enrichment functions GFEM model, both for Lagrange multipliers (a) and penalty (b) formulation.

Finally, taking into account the real traction distributions, both formulations presented sufficiently accurate results for the proposed patch test (Figure 5.88).



(a) *(b)*

Figure 5.88. Contact real tractions distribution for shifted enrichment functions GFEM model, both for Lagrange multipliers (a) and penalty (b) formulation.

It is important to mention that the vertical stress component field, obtained by the enriched mortar contact elements, is sufficiently accurate and close to the expected value (Figure 5.89), especially if one compares it with ones obtained for higher order mortar contact elements of the traditional FEM (Figure 5.84).

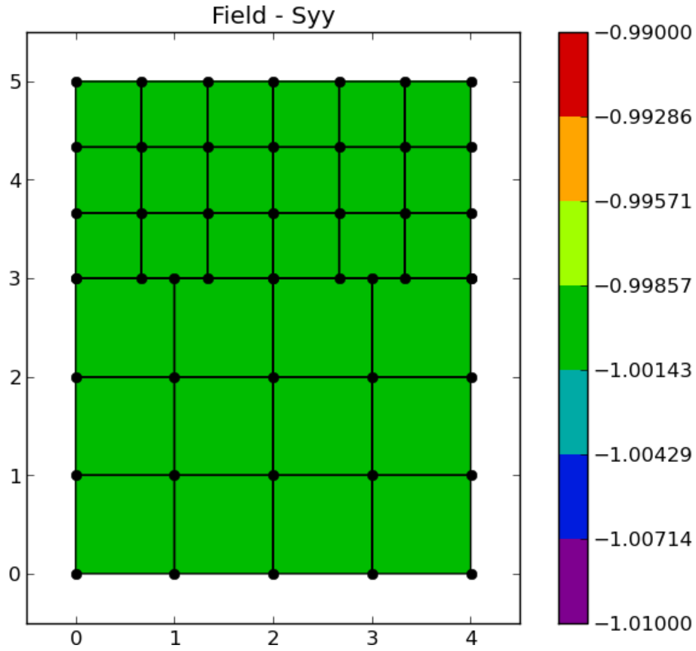


Figure 5.89. Vertical stress component field (Syy) obtained for a shifted functions enrichment GFEM model (second degree enrichment) - Lagrange multipliers formulation.

5.11. Hertz problem (small displacement contact)

Taking advantage of the existence of an analytical solution of the contact tensile distribution for the well-known Hertz problem, we propose the evaluation of the generalized segment-to-segment contact element for the problem indicated in Figure 5.90.

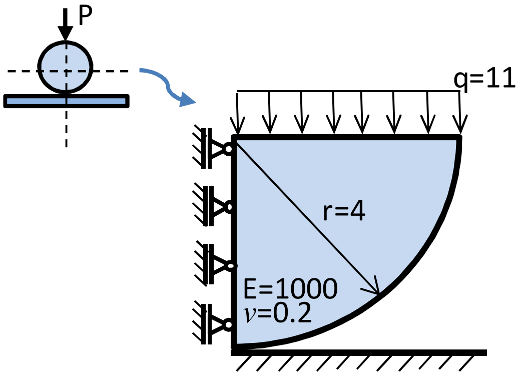


Figure 5.90. Proposed data for a Hertz contact problem scheme.

The proposed problem simulates the frictionless contact of a deformable cylinder and a rigid support, resulting in a Signorini problem. Due to the problem characteristics, a plane strain behavior is assumed. A linear elastic model is considered for the cylinder material. No dimensions are associated to the data presented in Figure 5.90.

The theoretical traction values along the contact region are defined by

$$t_n(x) = \frac{4rq}{\pi b^2}\sqrt{b^2 - x^2}\ ,$$

in which r is the cylinder radius, q is the distributed load, and b is half the length of the contact area in the idealized 2D model, which, by its turn, is computed by

$$b = 2\sqrt{\frac{2r^2 q\left(1 - \upsilon^2\right)}{E\pi}}\ .$$

Among other references, such formula is found in Dias (2013).

For the proposed problem, the value for b is approximately 0.656, and the maximum traction value is circa 85.41.

Despite it seems simple, this problem present some difficulties in order to result in accurate results regarding the traction distributions. After several bad succeeded tests, the topology used by Laursen (1992) and Dias (2013) seems to be the best choice for modeling such problem. The adopted mesh, based on those ones, is presented in Figure 5.91.

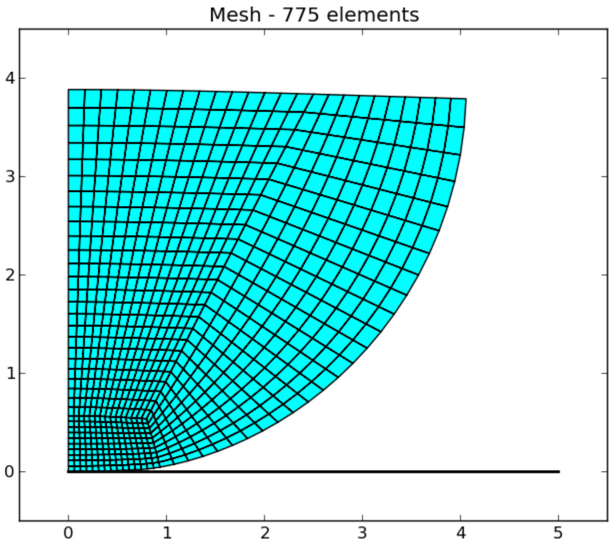


Figure 5.91. Bi-linear quadrilateral (Q4) elements mesh proposed for modeling the Hertz contact problem.

First, the problem is computed not enriching the mesh nodes, i.e., resulting in a model equivalent to the tradition FEM. The estimated contact tractions results are presented in Figure 5.92.

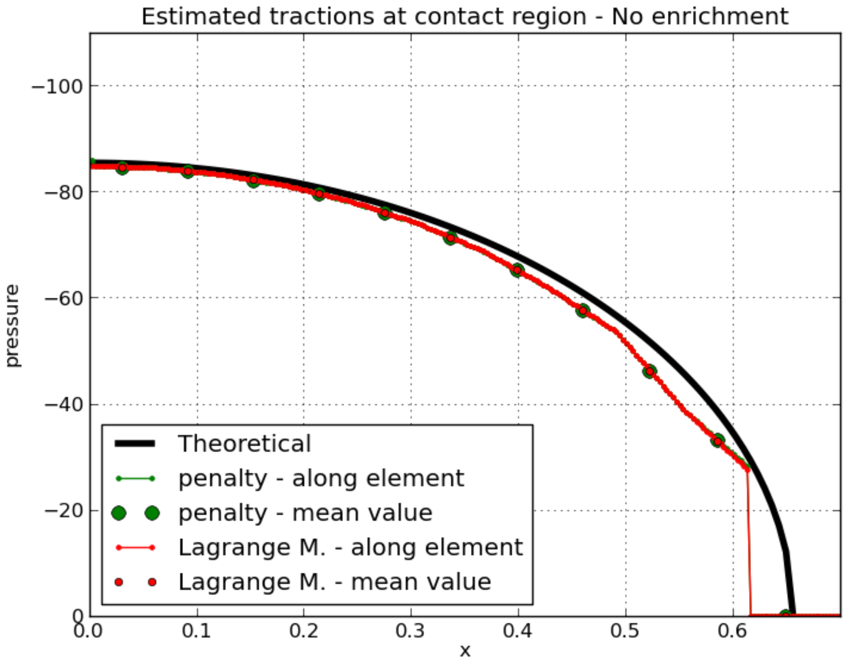


Figure 5.92. Estimated contact tractions results for the Hertz contact problem, both for the Lagrange multiplies and penalty formulation – no enrichment model.

Both Lagrange multipliers and penalty based segment-to-segment contact formulations present quite similar results regarding the estimated traction, for the not enriched mesh. It is worth to note that the mean value for each of the elements is also plotted in Figure 5.92, and they are also close one to each other in both formulations.

For the first degree enrichment (shift functions), the estimated traction values oscillate around the theoretical expect value, as it is shown in Figure 5.93.

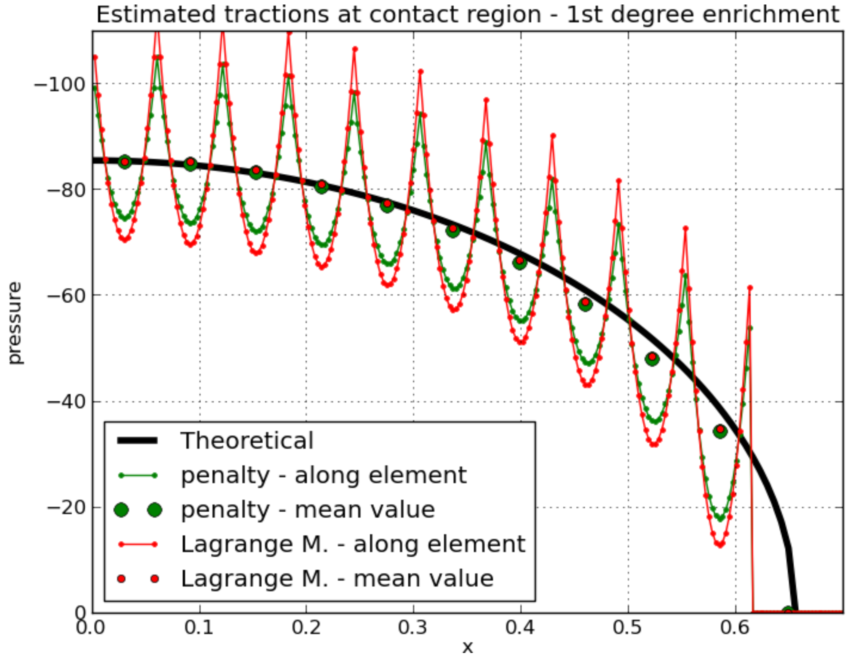


Figure 5.93. Estimated contact tractions results for the Hertz contact problem, both for the Lagrange multiplies and penalty formulation – shifted enrichment (first degree enrichment).

The mean estimated traction value, for each segment-to-segment contact element is also plotted in Figure 5.93. Those mean tractions present similar values for both Lagrange multipliers and penalty based formulation, and they are close to the theoretical value.

Such results enforce the expectations that the use of the mean traction value, estimated at each segment-to-segment contact element, are sufficiently accurate for the method proposes.

Regarding the real tractions at the contact interface, both first degree enrichment and not enriched models presented similar results, as shown in Figure 5.94. On the other hand, the stress fields of the enriched model are smoother and more continuous across the elements edges, as it can be observed in Figure 5.95.

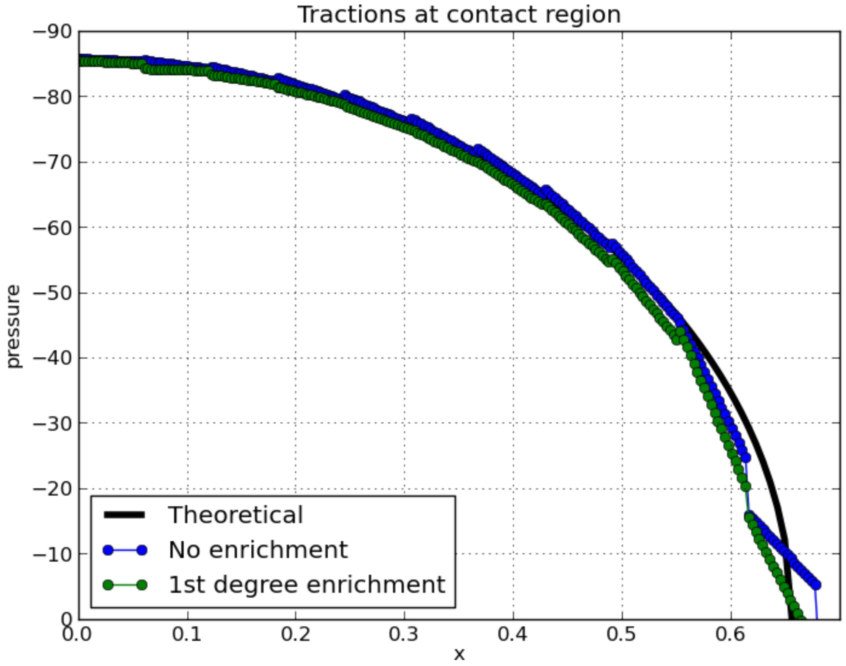
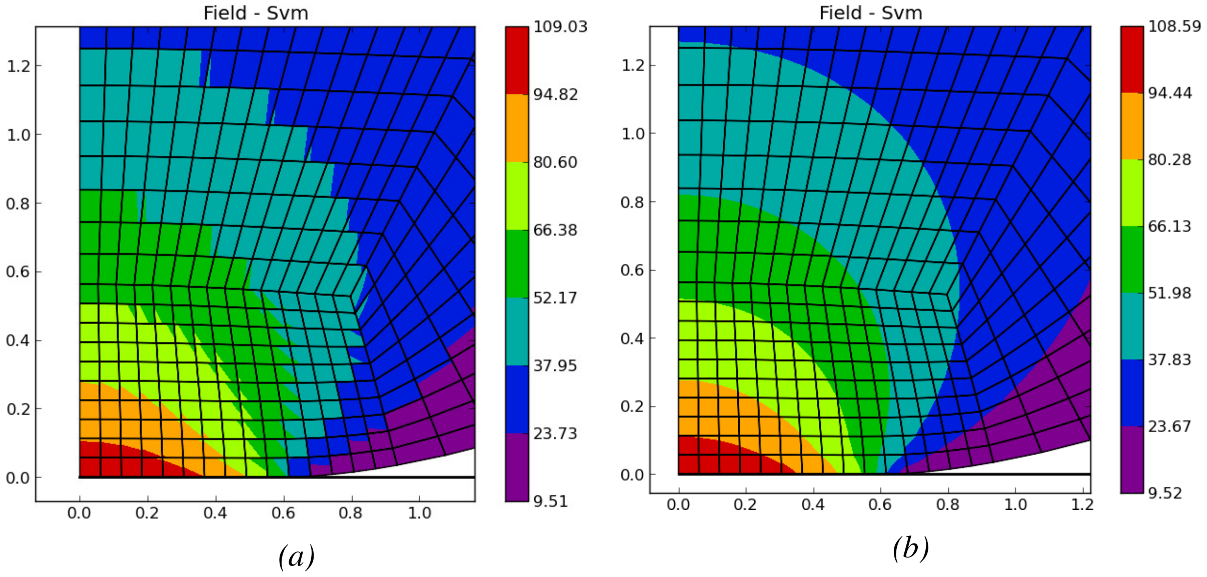


Figure 5.94. Real contact tractions results for the Hertz contact problem, for shifted functions first degree enrichment and for not enriched models - Lagrange multiplies formulation.

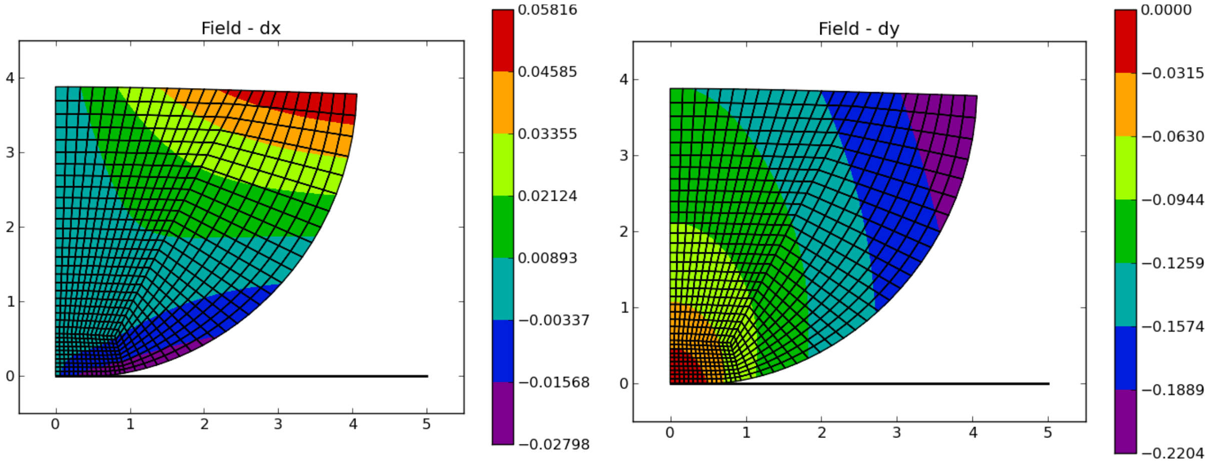


(a)

(b)

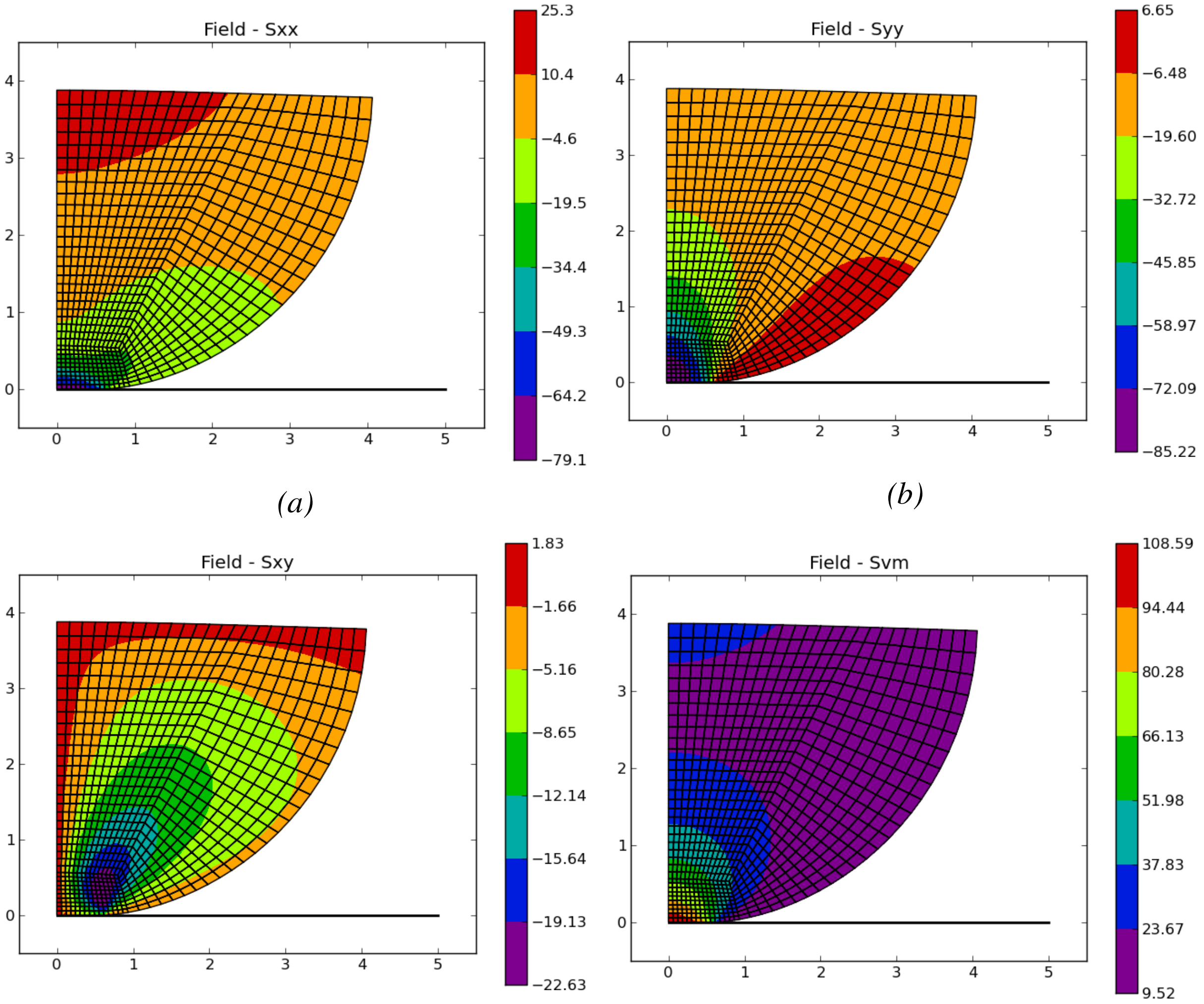
Figure 5.95. Von Mises stress (Svm) field in a closer view next to the contact region: (a) not enriched model, (b) first degree shifted functions enrichment.

In order to illustrate the achieved results, the displacement field results, for the enriched model, are presented in Figure 5.96. The stress field results are presented in Figure 5.97.



(a) *(b)*

Figure 5.96. Horizontal (a) and vertical (b) field results for the Hertz contact problem.



(a) *(b)*

(c) *(d)*

Figure 5.97. Stress field results for the Hertz contact problem: (a) horizontal component stress (Sxx), (b) vertical component stress (Syy), (c) shear stress (Sxy) and (d) von Mises stress fields.

6 - Numerical examples

In Chapter 5, several simple examples were proposed both for demonstrating the correctness of the developed framework, using the Finite Element Method, and to evaluate the accuracy of the nonlinear proposed formulation using the Generalized Finite Element Method. Beyond that, several observed characteristics of the methods were commented, constituting useful information for solving general nonlinear problem using the GFEM formulation.

The present chapter is devoted to advance in such direction, proposing more complex examples and solving them using the developed GFEM framework. Additional important conclusions can be observed by analyzing the achieved results.

Since no theoretical solutions are available for the following proposed problems, as it happens for the most of the problems proposed in Chapter 5, the results achieved for Finite Element Method models, computed using very refined mesh models, are used as the reference solution.

6.1. Solids with hole

The advantage of mixing different order approximation PU for describing the solid geometry and structural behavior becomes clear in the following proposed example, in which one aims to evaluate the GFEM polynomial and shifted enrichment functions results for problems in which the modeled solid present a less trivial geometry than the ones presented in the previous examples. In order to do so, two different problems of solids with holes are proposed in Figure 6.1.

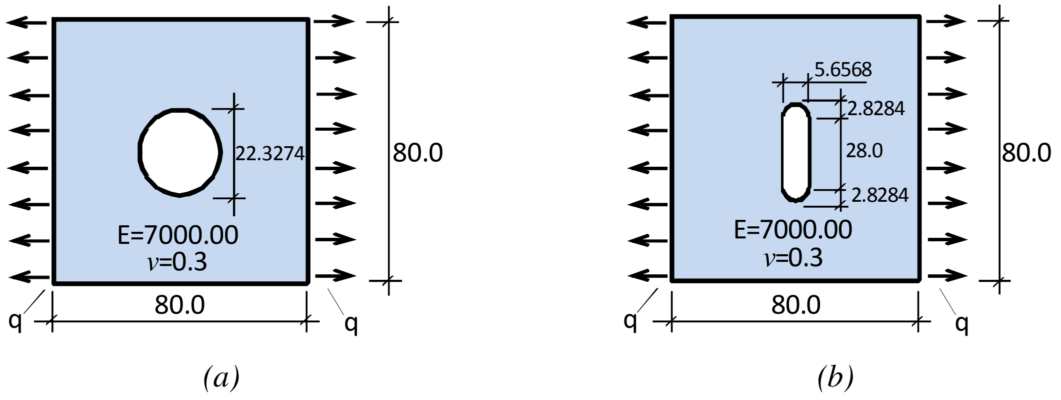


Figure 6.1. Solid with holes scheme: circular hole (a) and ‘long’ hole (b).

For both solids it is considered a linear elastic behavior material with Young modulus 7000.00 (dimensionless) and Poisson's ratio equal to 0.3. The solids geometric characteristics (dimensionless) are indicated in Figure 6.1. It is considered plane stress hypothesis and unitary thickness. It is considered a distributed force q equals to 10.0 (dimensionless).

The circular hole solid was modeled taking into account the problem symmetry. The considered reference solution is achieved by means of a finite element model containing 4650 triangular quadratic elements, resulting in 42320 degrees of freedom model.

The GFEM model contains 24 quadrilateral elements. The elements surrounding the hole are defined using serendipity bi-cubic partitions of unity, in order to describe the hole geometry. Second degree enrichment using shifted functions is applied to all nodes of the solid, including the ones in which Dirichlet boundary conditions are applied to consider the symmetry, resulting in 420 degrees of freedom. The Dirichlet boundary conditions are applied over the elements' sides, using penalty method (penalty parameter equals to 10^{10}).

Both meshes are depicted in Figure 6.2.

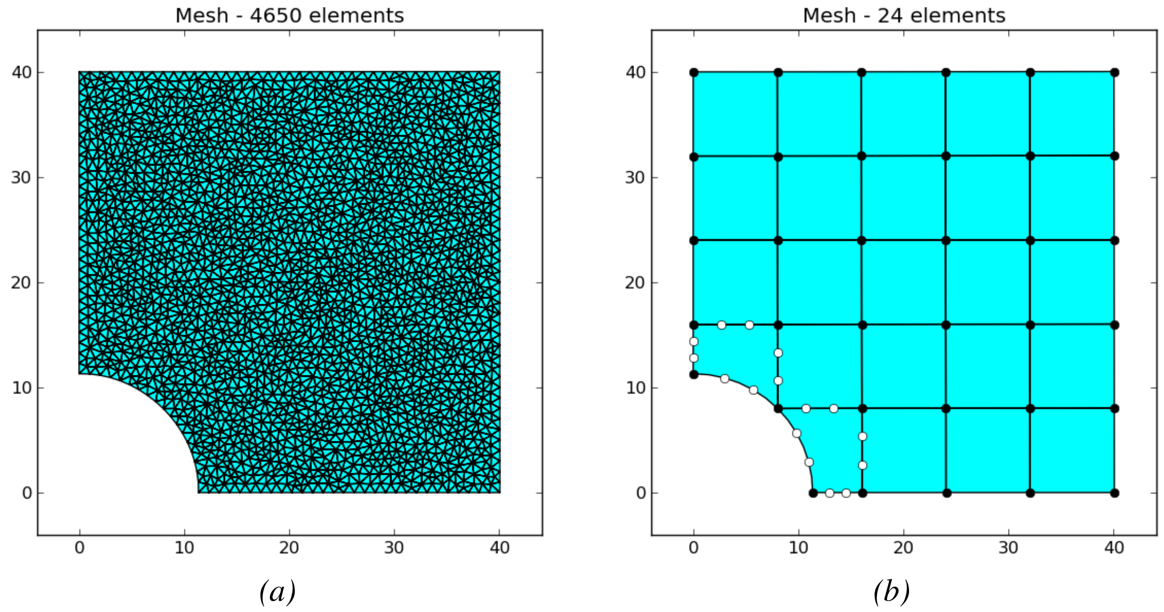
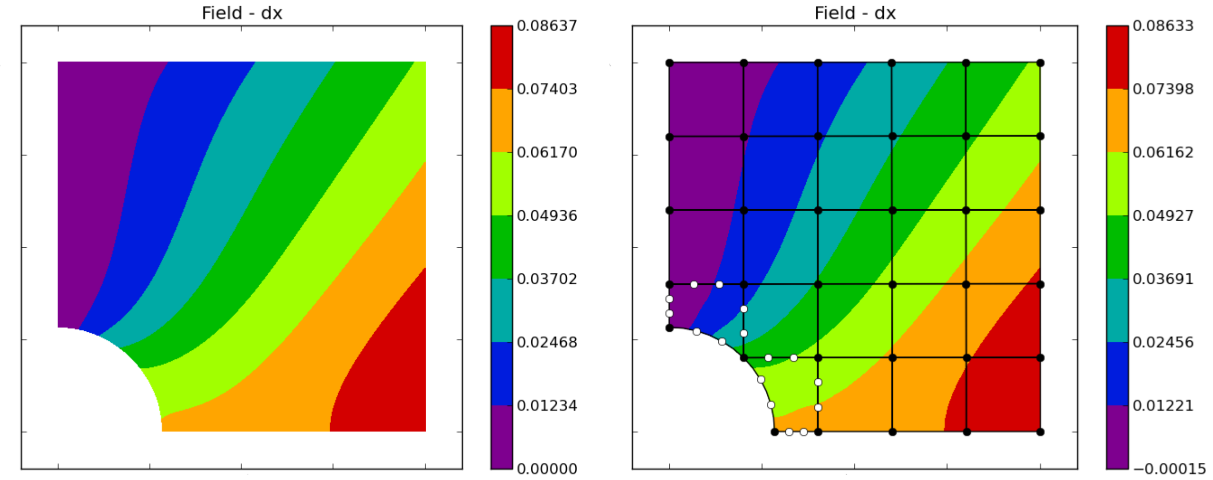


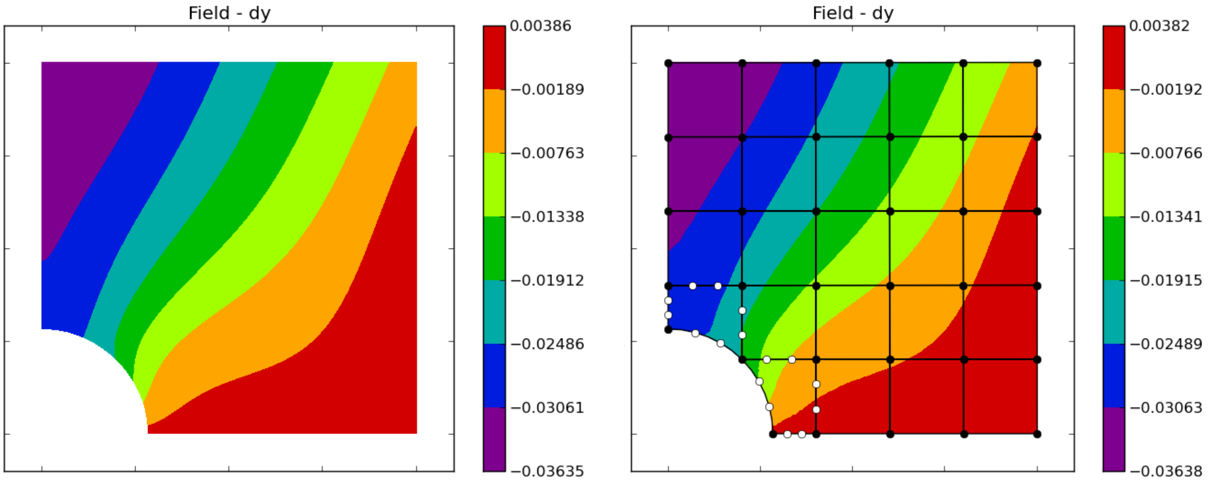
Figure 6.2. (a) FEM overkill mesh (reference numerical solution) containing 4650 elements (42320 dofs); and (b) GFEM solution mesh, containing 24 elements (420 dofs for the applied enrichment).

The displacement and stress fields resulting for both the FEM reference and the GFEM model are depicted in the color maps presented from Figure 6.3 to Figure 6.8.



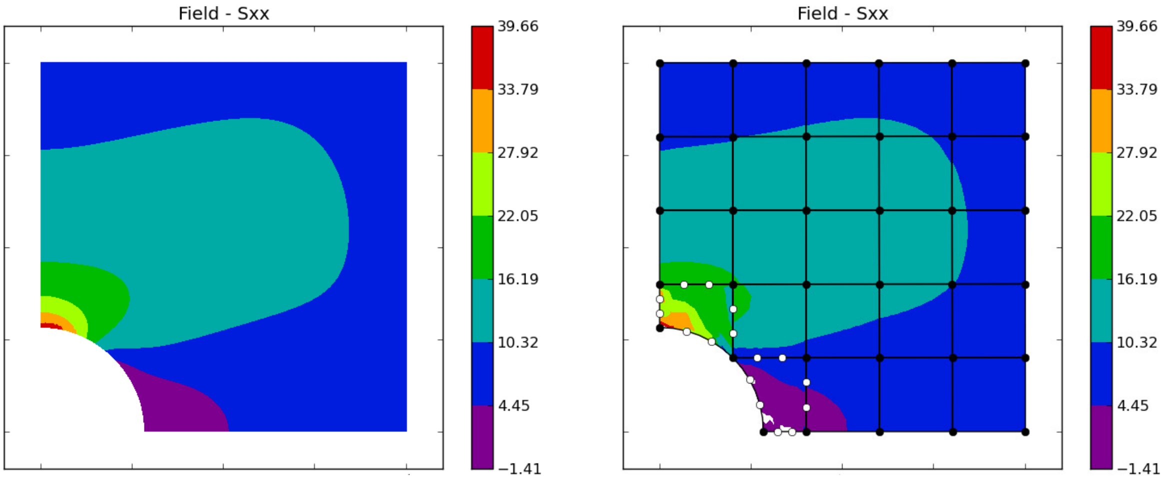
(a) *(b)*

Figure 6.3. Horizontal displacement (dx) for: (a) finite element (reference) solution; (b) GFEM solution.



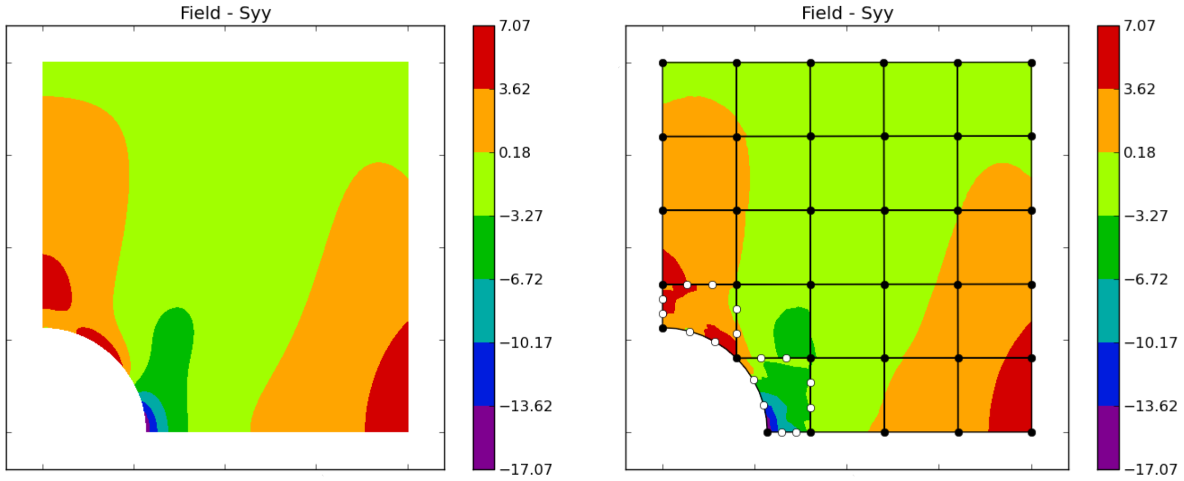
(a) *(b)*

Figure 6.4. Vertical displacement (dy) for: (a) finite element (reference) solution; (b) GFEM solution.



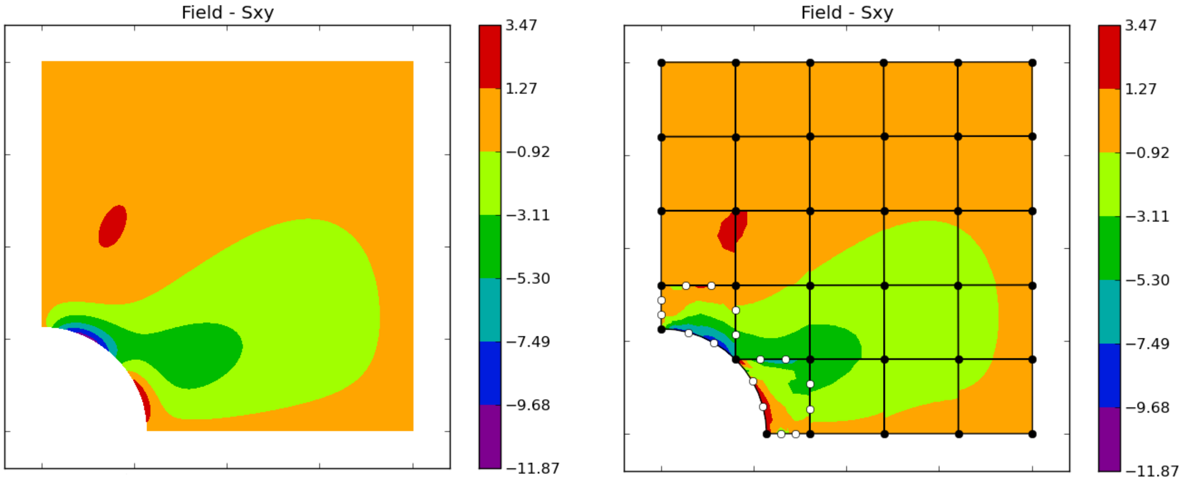
(a) *(b)*

Figure 6.5. Horizontal stress (Sxx) for: (a) finite element (reference) solution; (b) GFEM solution.



(a) *(b)*

Figure 6.6. Vertical stress (Syy) for: (a) finite element (reference) solution; (b) GFEM solution.



(a) *(b)*

Figure 6.7. Shear stress (Sxy) for: (a) finite element (reference) solution; (b) GFEM solution.

The ‘long’ hole solid is also modeled taking into account the problem symmetry. The considered reference solution is achieved by means of a finite element model containing 4028 triangular quadratic elements, resulting in 36722 degrees of freedom model. Such mesh is presented in Figure 6.2 (a).

The GFEM model contains 33 quadrilateral elements. Two of the four elements surrounding the hole are defined using a serendipity bi-cubic partition of unity, in order to describe the hole. Second degree enrichment using shifted functions is applied to all nodes of the solid, including the ones in which Dirichlet boundary conditions are applied to consider the symmetry, resulting in 420 degrees of freedom. The Dirichlet boundary conditions area applied over the elements’ sides, using penalty method (penalty parameter equals to 10^{10}).

The reason of modeling this second problem is that the mesh proposed for describing the ‘long’ hole geometry results in some elements with even less conventional shapes, as indicated in Figure 6.8 (b).

The displacement and stress fields resulting for both the FEM reference and the GFEM model are depicted in the color maps presented from Figure 6.9 to Figure 6.13.

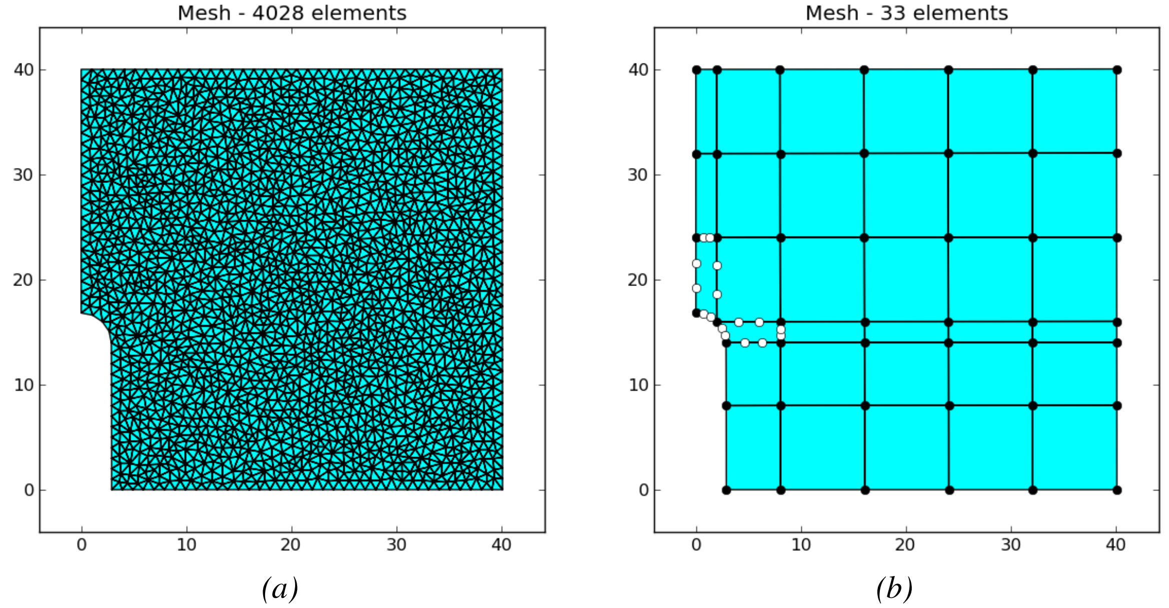


Figure 6.8. (a) FEM overkill mesh (reference numerical solution) containing 4028 elements (36722 dofs); and (b) GFEM solution mesh, containing 24 (552 dofs for the applied enrichment).

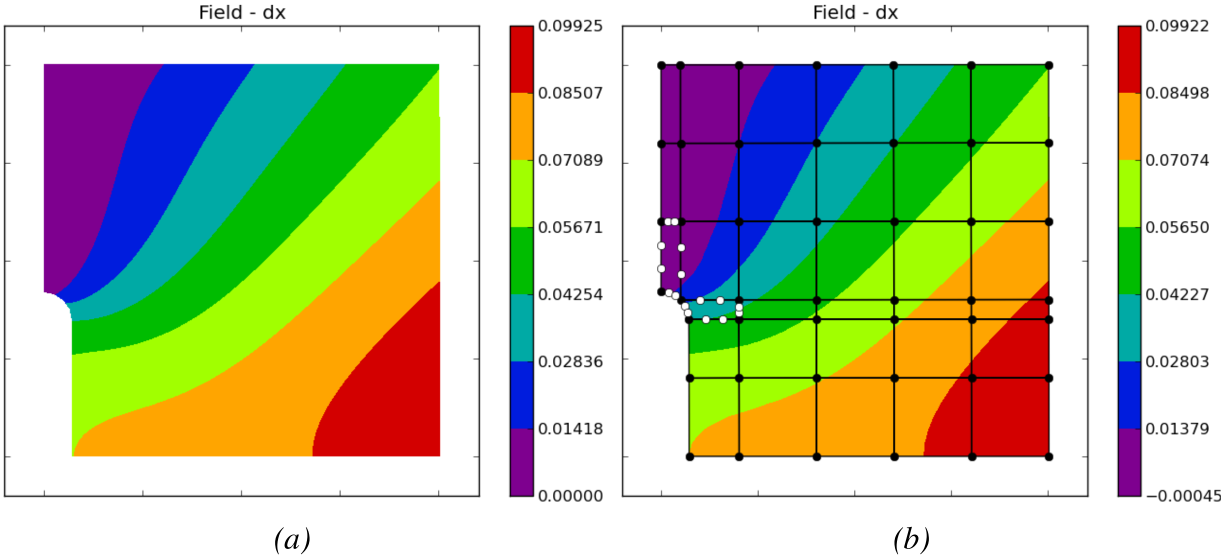
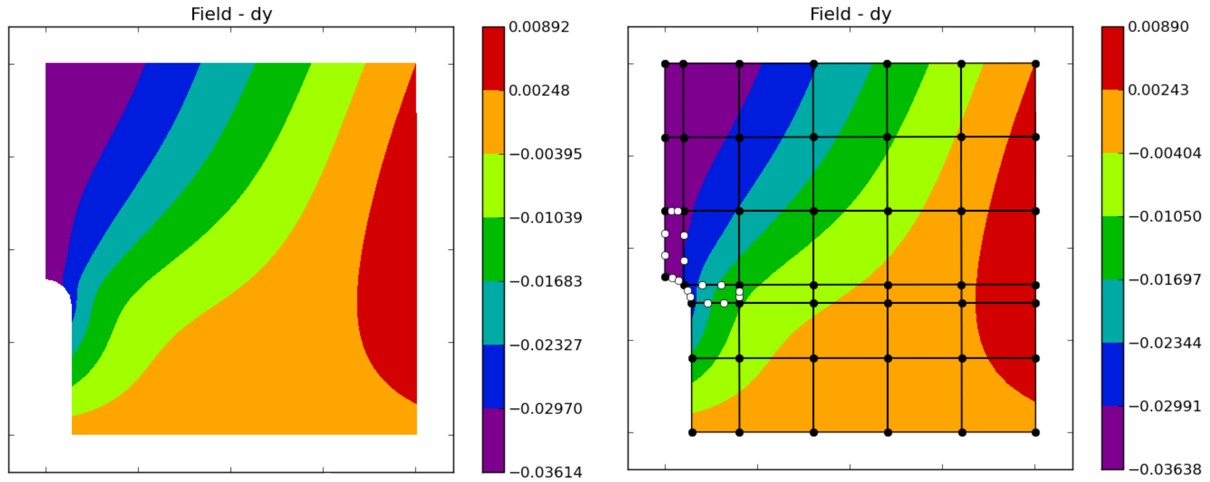
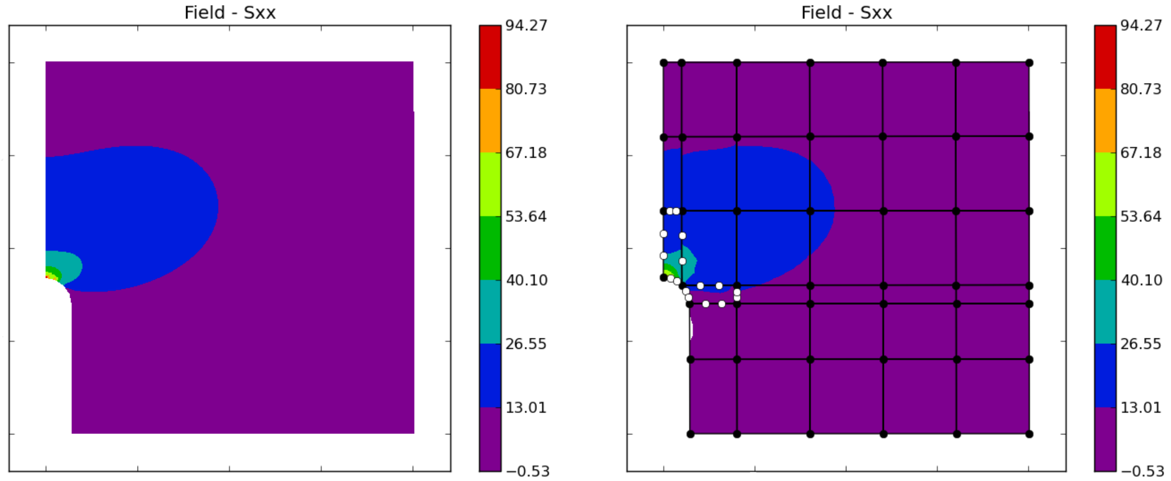


Figure 6.9. Horizontal displacement (dx) for: (a) finite element (reference) solution; (b) GFEM solution.



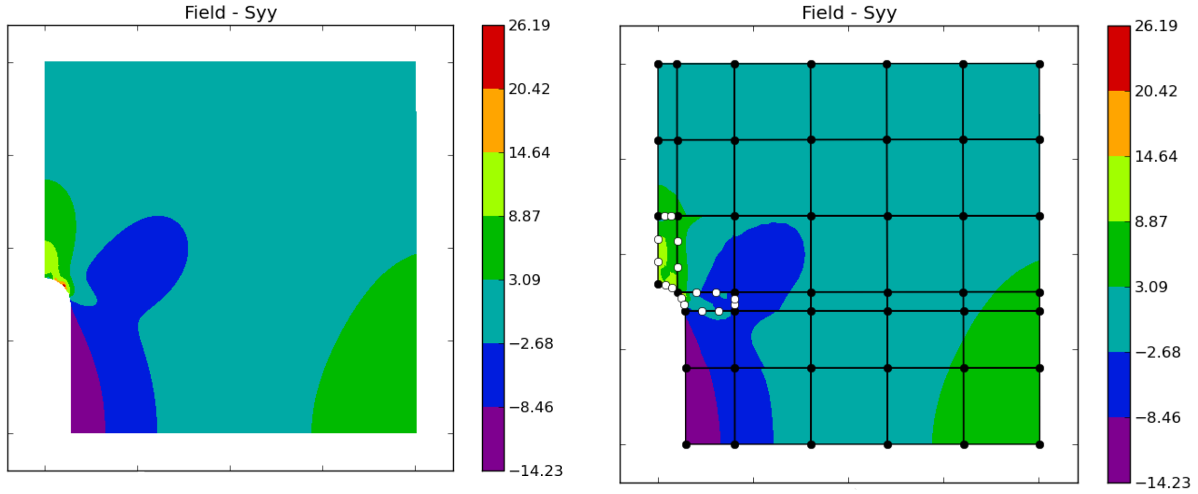
(a) *(b)*

Figure 6.10. Vertical displacement (dy) for: (a) finite element (reference) solution; (b) GFEM solution.



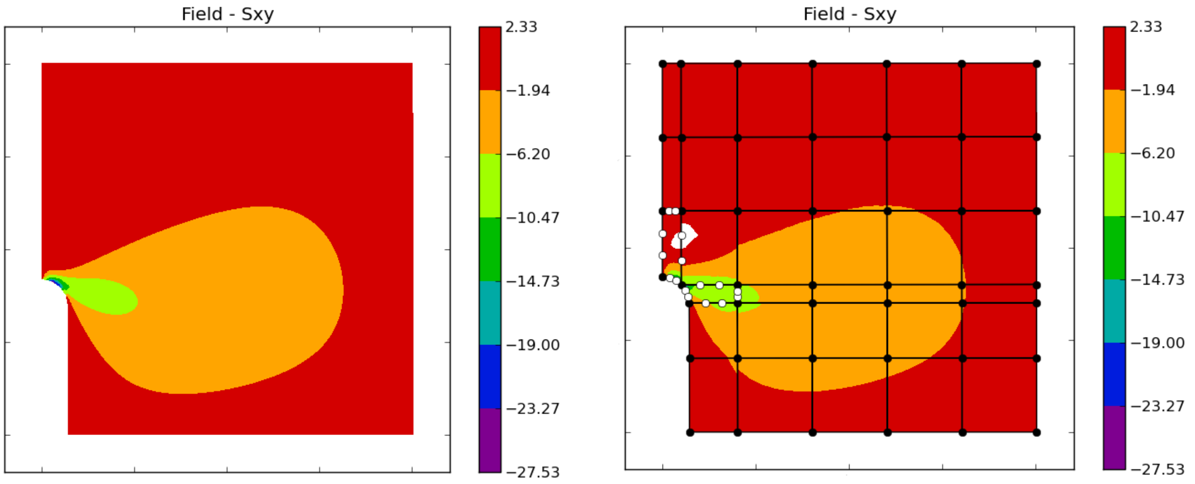
(a) *(b)*

Figure 6.11. Horizontal stress (Sxx) for: (a) finite element (reference) solution; (b) GFEM solution.



(a) *(b)*

Figure 6.12. Vertical stress (Syy) for: (a) finite element (reference) solution; (b) GFEM solution.

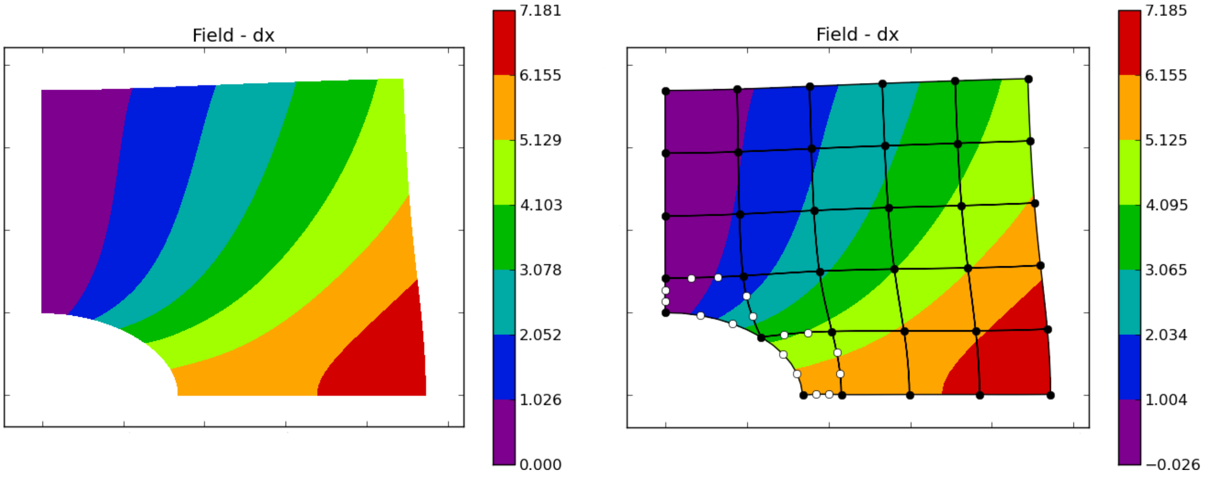


(a) *(b)*

Figure 6.13. Shear stress (Sxy) for: (a) finite element (reference) solution; (b) GFEM solution.

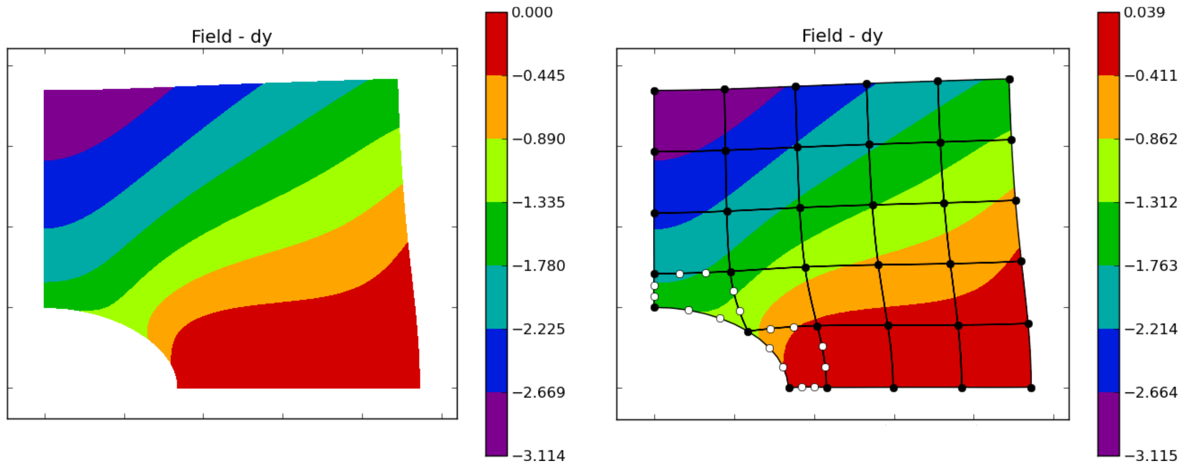
For both problems, the GFEM results are very close to the ones of reference solution. The presented stresses fields are generated computing the specific stress component value for several points in each of the elements domain. Due to this reason, such fields can present discontinuities in the elements boundaries. Despite of this fact, in general, the stresses fields of the GFEM do not present remarkable discontinuities, as one would expect for such a coarse meshes.

After evaluated for the linear elastic material model, the circular hole model is also processed using the hyperelastic Neo-Hookean material model, aiming to evaluate if the same accuracy is also observed for such nonlinear model. The pressure *q* is increased to the value of 100. The results are indicated in the following figures (Figure 5.14 to Figure 5.18).



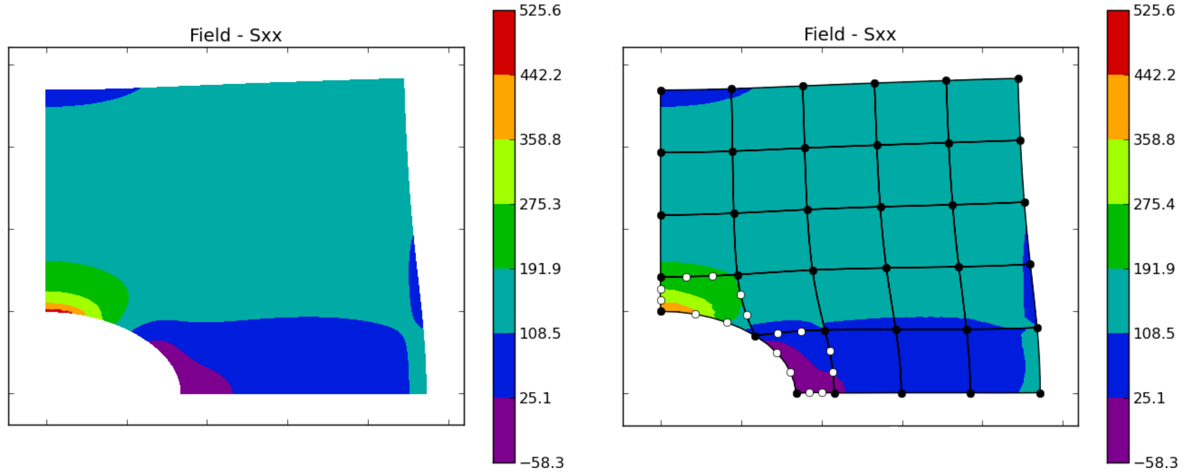
(a) *(b)*

Figure 6.14. Horizontal displacement (dx) for: (a) finite element (reference) solution; (b) GFEM solution (Neo-Hookean material).



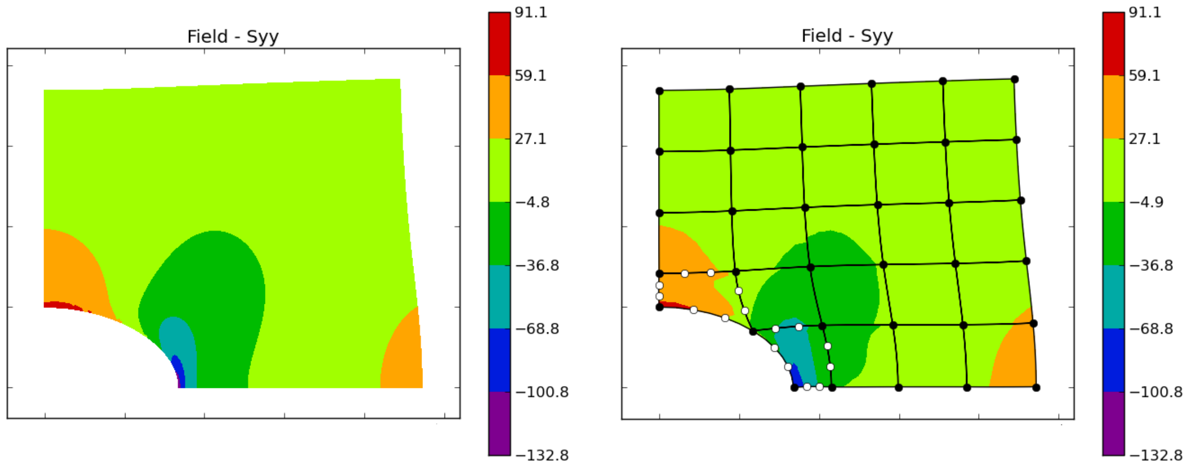
(a) *(b)*

Figure 6.15. Vertical displacement (dy) for: (a) finite element (reference) solution; (b) GFEM solution (Neo-Hookean material).



(a) *(b)*

Figure 6.16. Horizontal stress (Sxx) for: (a) finite element (reference) solution; (b) GFEM solution (Neo-Hookean material).



(a) *(b)*

Figure 6.17. Vertical stress (Syy) for: (a) finite element (reference) solution; (b) GFEM solution (Neo-Hookean material).

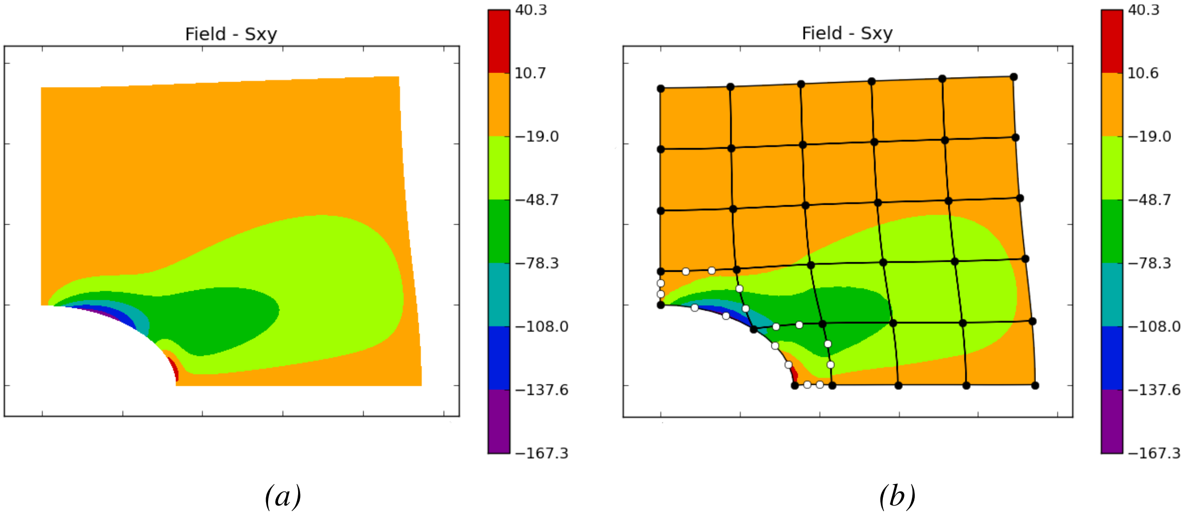


Figure 6.18. Shear stress (Sxy) for: (a) finite element (reference) solution; (b) GFEM solution (Neo-Hookean material).

6.2. Beam - plastic hinge

Testing the performance of the shifted function GFEM model for elastoplastic analyses is the objective to be achieved by means of the present proposed example. In order to do so, a beam clamped in both side is defined, according to the geometry presented in Figure 6.19. A vertical displacement is applied at the left clamped edge, such that when the stress fields evolve, one can observe a phenomenon similar to the one predicted by the plastic hinge theory.

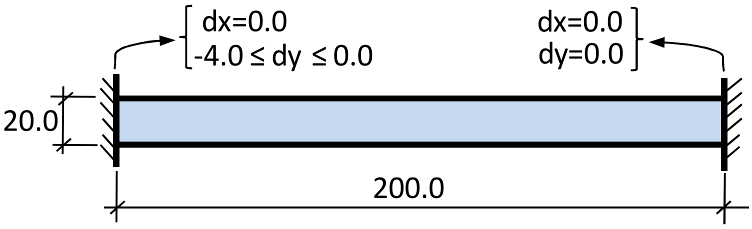


Figure 6.19. Proposed scheme for the beam clamped at both edges.

Beyond the beam height (20) and the beam span (200), it is important to inform that its thickness is 1 (all dimensionless). The maximum value for the vertical applied displacement is 4, in downward direction. The material properties used in the model are the one at Table 5.4. The loads due to self-weight are neglected.

The reference solution is computed by means of a FEM model; a 1000 bi-quadratic (Q8) elements regular mesh (10x100) is used, resulting in 6442 degrees of freedom. Plane stress hypothesis is assumed. The nonlinear incremental solution is performed in 4 time steps increments. Figure 6.20 illustrates the horizontal displacement field (dx) obtained using such model.

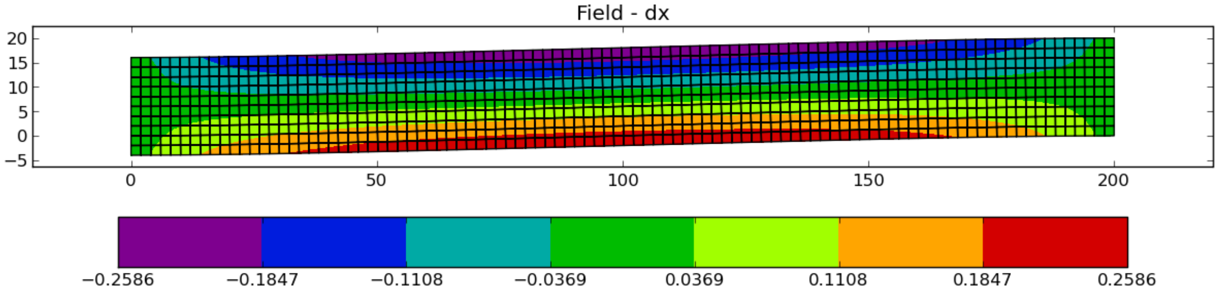


Figure 6.20. Horizontal displacement (dx) field – numerical reference solution.

At the end of the process, remarkable plastic behavior is observed at some parts of the structure, especially the ones near the clamped edges. Interesting information regarding the plastic strains and the stress along the beam cross section at those regions can be obtained from the model results.

For instance, the plastic strain horizontal component (Exxp) along the beam height evolves as the imposed displacement increases during the nonlinear analysis. The horizontal component of the plastic strain value along the beam cross section positioned 10 length units (half of its height) from the left clamped edge is depicted in Figure 6.21.

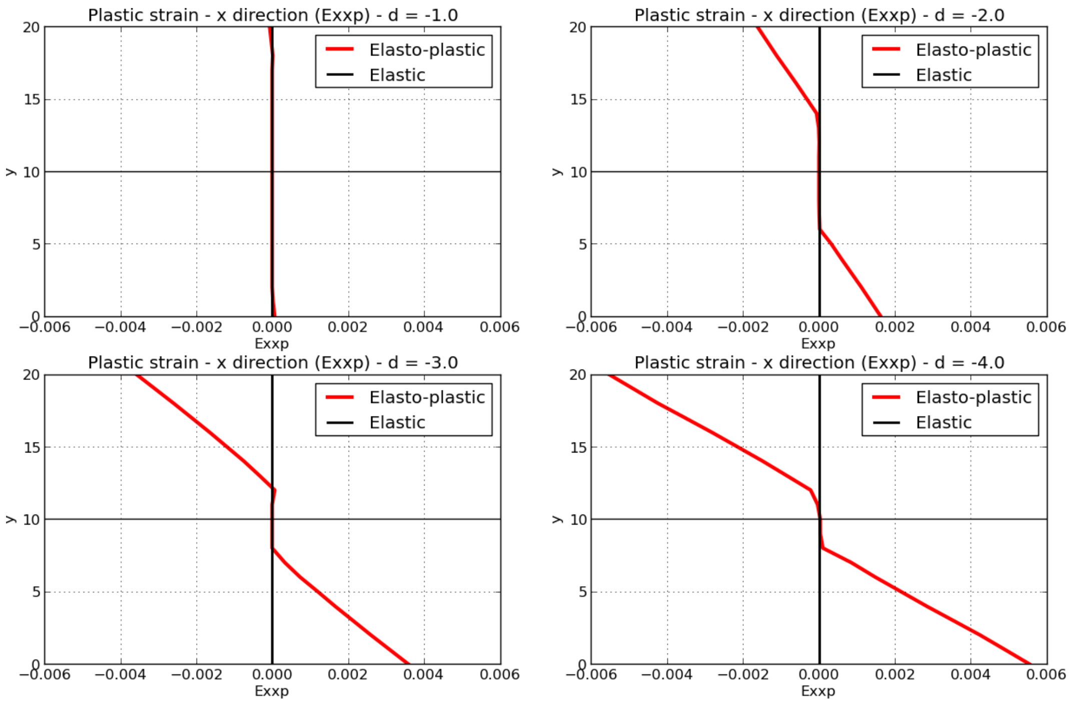


Figure 6.21. Horizontal plastic strain component (Exxp) values along the cross section positioned 10 length units from the left clamped edge, for both linear elastic and elastoplastic material model.

Also, as the plastic strains increase during the structure loading, the stress distributions differ considerably from the ones expected for the linear elastic material model. Such behavior is depicted in Figure 6.22 and Figure 6.23, in which the horizontal stress component

(Sxx) and von Mises stress (Svm) distribution along such cross section is presented. The equivalent values for the linear elastic material model, achieved using the same FE model are also presented at the same figures.

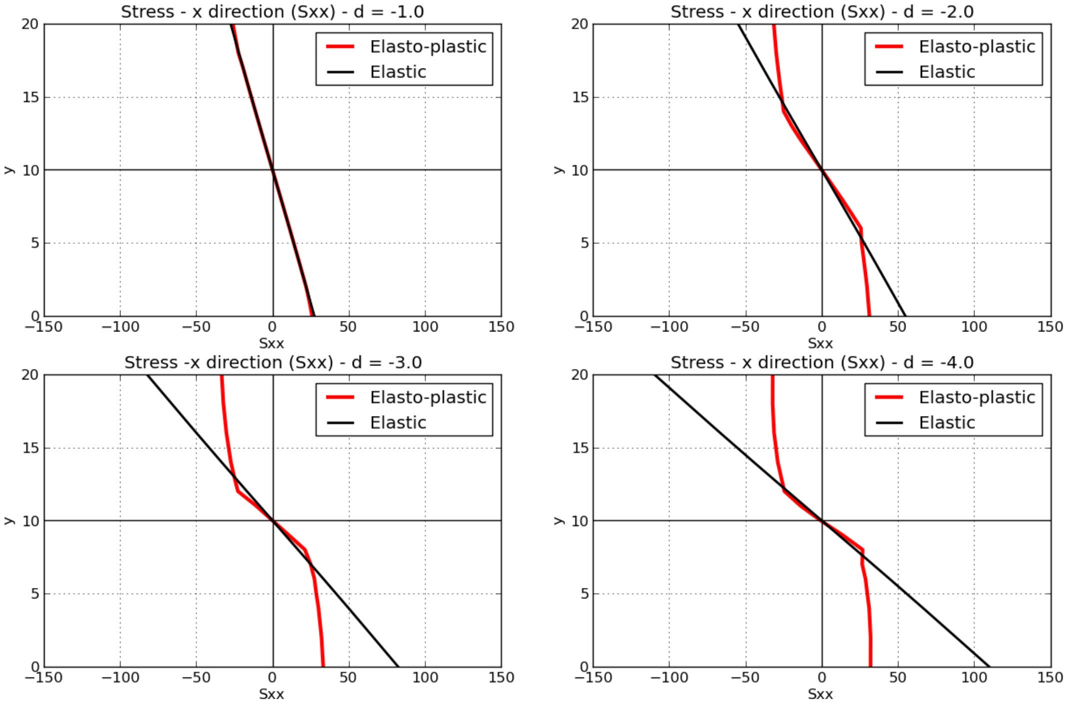


Figure 6.22. Horizontal stress component (Sxx) values along the cross section positioned 10 length units from the left clamped edge, for both linear elastic and elastoplastic material model.

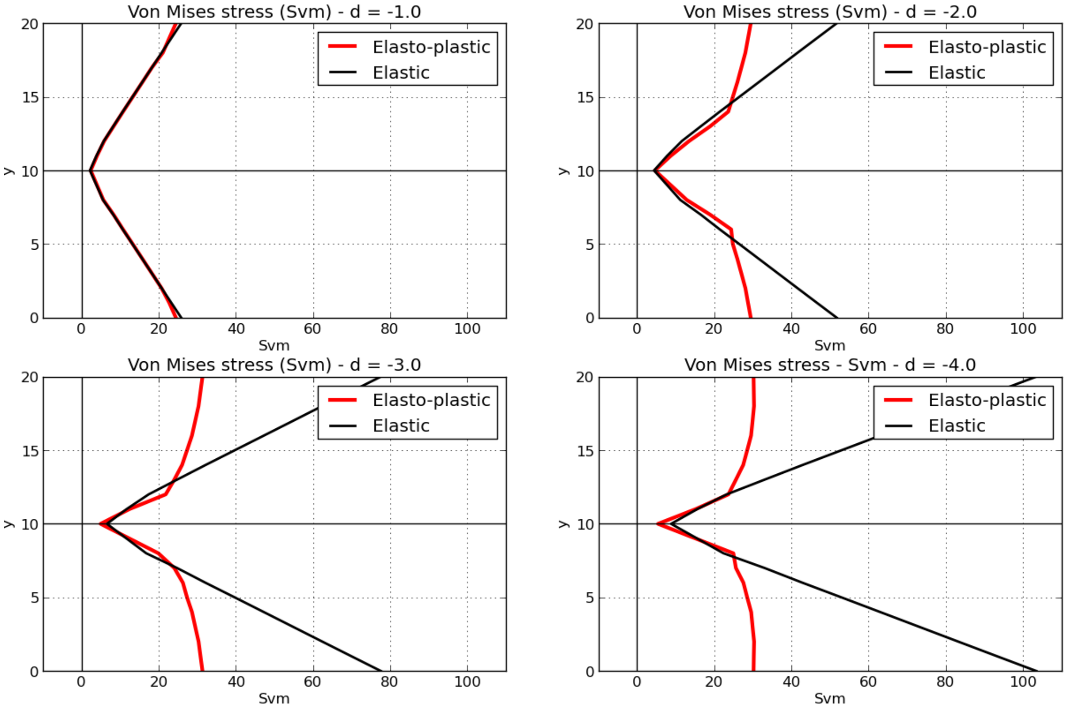


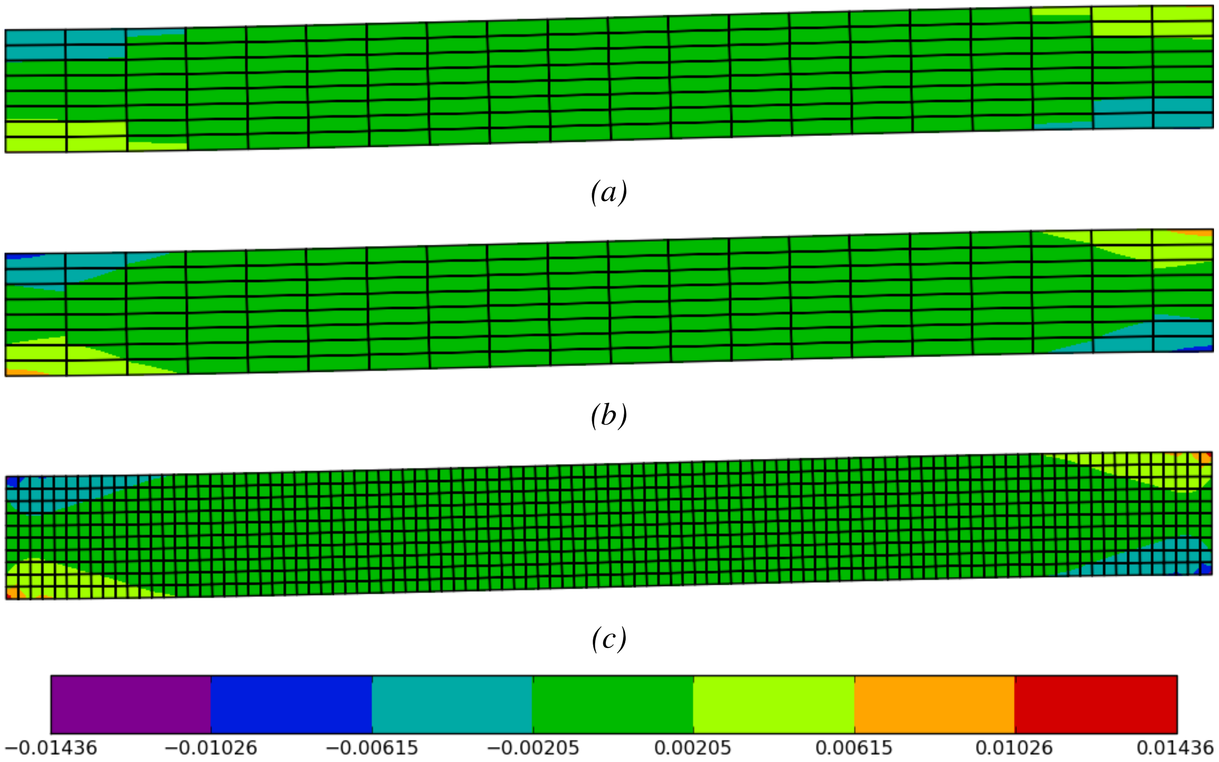
Figure 6.23. Von Mises (Svm) along the cross section positioned 10 length units from the left clamped edge, both for linear elastic and elastoplastic material model.

For the imposed displacement value equal to 1, the structure, still show, in practice, linear elastic behavior in the overall sense, since it presents negligible plastic strains values. For the other presented imposed displacement values, the results are clearly associated to an elastoplastic model, fact that can be evidenced by verifying the difference between the results for the linear elastic and the elastoplastic material model.

The GFEM model is built by defining a mesh of 160 (8x20) bi-linear (Q4) elements mesh, resulting in elements for which the relation between its sides is 4. As already evaluated in the previous chapter, such shape ratio is suitable for the shifted and polynomial GFEM models.

First degree shifted enrichment is used for all the nodes of the GFEM model. The displacement boundary conditions at the clamped edges are applied along the whole elements side, since the nodes at such edge are also enriched. A penalty strategy is used for imposing such displacements conditions. For the GFEM model, 8x8 integration points are used for each element. The remaining conditions are the same used for the reference solution model.

The plastic strain horizontal component (Exxp) for the not enriched and the first degree enrichment GFEM model is presented in Figure 6.24. The results for the horizontal stress (Sxx) and von Mises stress (Svm) is presented in Figure 6.25 and Figure 6.26.

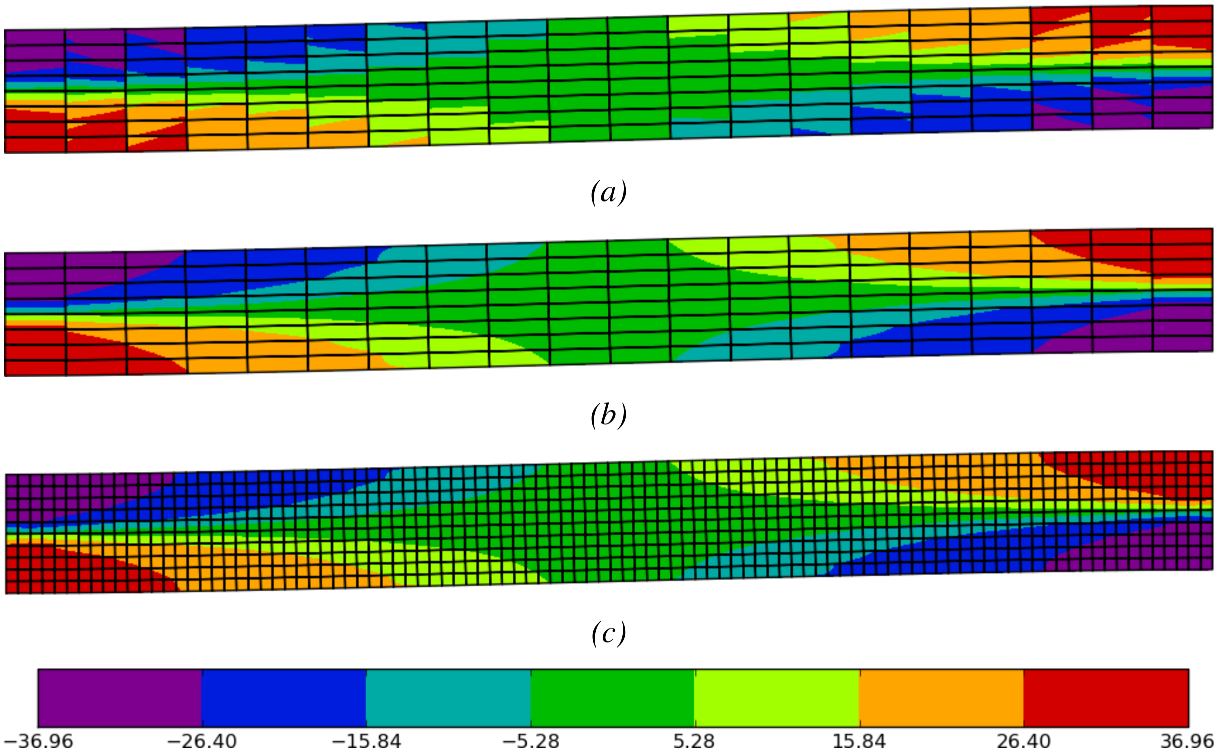


(a)

(b)

(c)

Figure 6.24. Plastic strain horizontal component (Exxp) field; GFEM model, not enriched (a), first degree enrichment (b) and FEM reference solution (c).

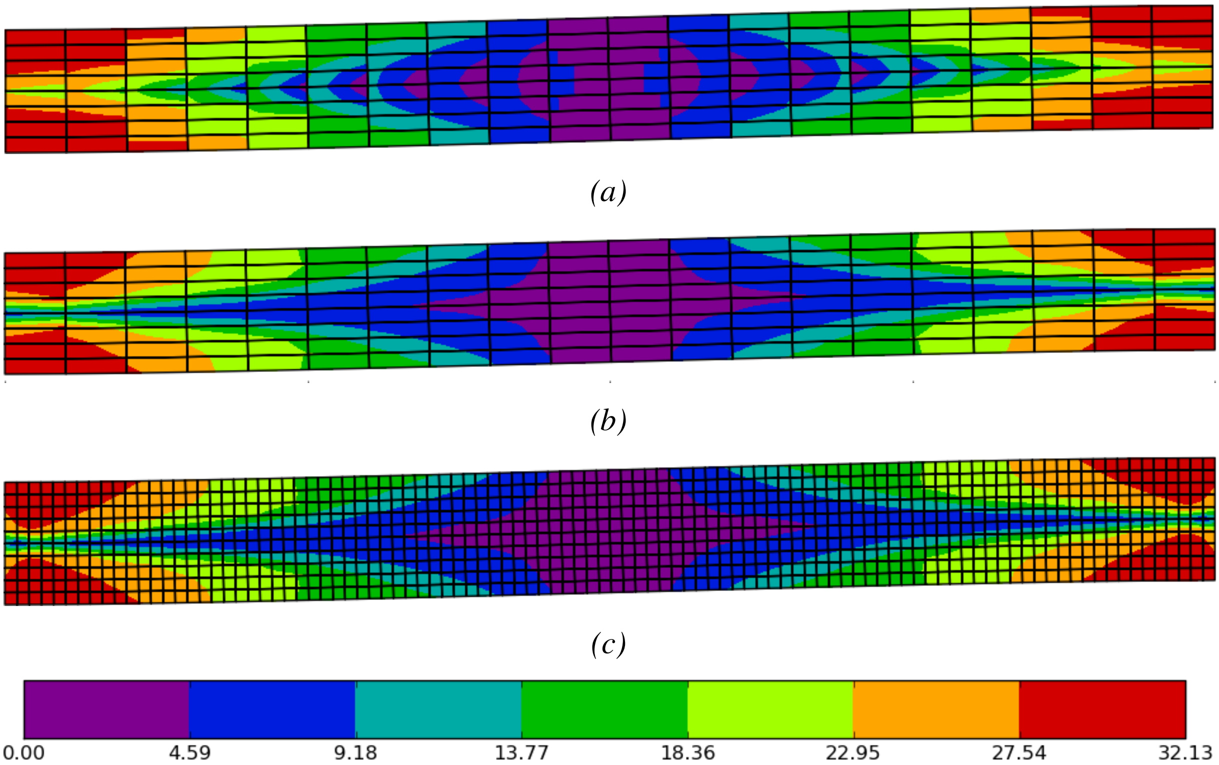


(a)

(b)

(c)

Figure 6.25. Stress horizontal component (Sxx) field; GFEM model, not enriched (a), first degree enrichment (b) and FEM reference solution (c).



(a)

(b)

(c)

Figure 6.26. Von Mises stress (Svm) field; GFEM model, not enriched (a), first degree enrichment (b) and FEM reference solution (c).

The GFEM model, when no enrichment is applied over it, results in 414 degrees of freedom; the first degree enrichment increases the number of degrees of freedom to 1170.

The achieved values for the same fields at the cross section positioned 10 length units from the left clamped edge (Figure 6.27) show that the enrichment promotes improvements in those values.

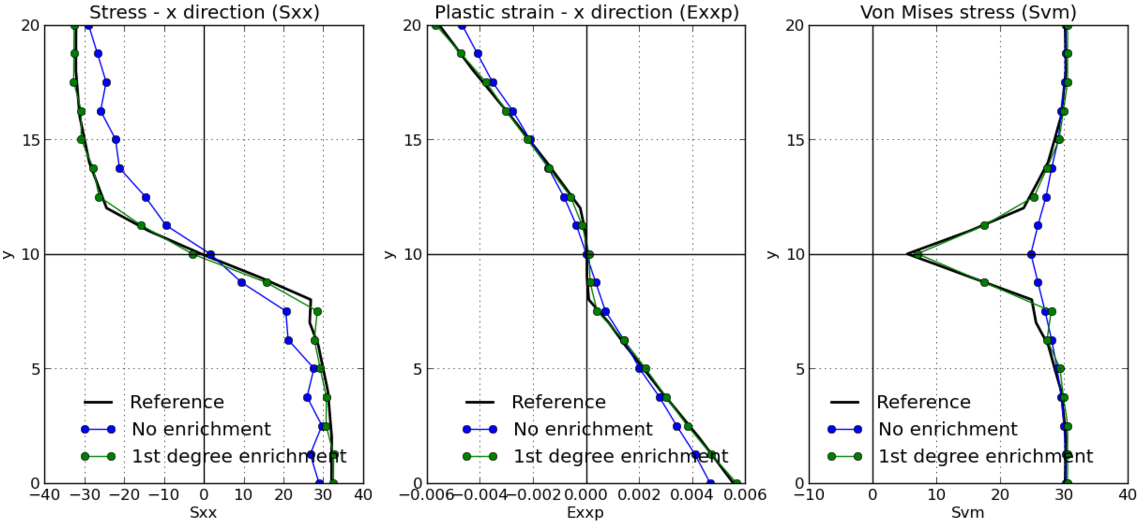


Figure 6.27. Horizontal stress (Sxx), horizontal plastic strain (Exxp) and von Mises stress (Svm) field values across the cross section positioned 10 length units from the left clamped edge of the beam.

The history of several convergence parameters associated to the nonlinear solution process is presented in Figure 6.28. One observes that the residual force and energy L2 norms converges both for the FEM and the GFEM model, while no convergence is observed for the norm of the displacement value for the GFEM model.

As already discussed in Chapter 3, this occurs due to the fact that the displacement vector in the GFEM presents both enriched and not enriched degrees of freedom. The enriched ones (generalized degrees of freedom) does not represent the displacement value at the model points. As observed, the L2 norm computed for this vector does not tend to zero like for the other convergence parameters, and so, could not be used as a convergence parameter for the GFEM.

On the other hand, the convergence parameter proposed for the GFEM (Chapter 3), indeed represents the convergence, as one can observed in Figure 6.28. For most of the nonlinear examples computed in the present work, we observed that such norm indicates the GFEM nonlinear model convergence as reliably as the other traditionally used parameters.

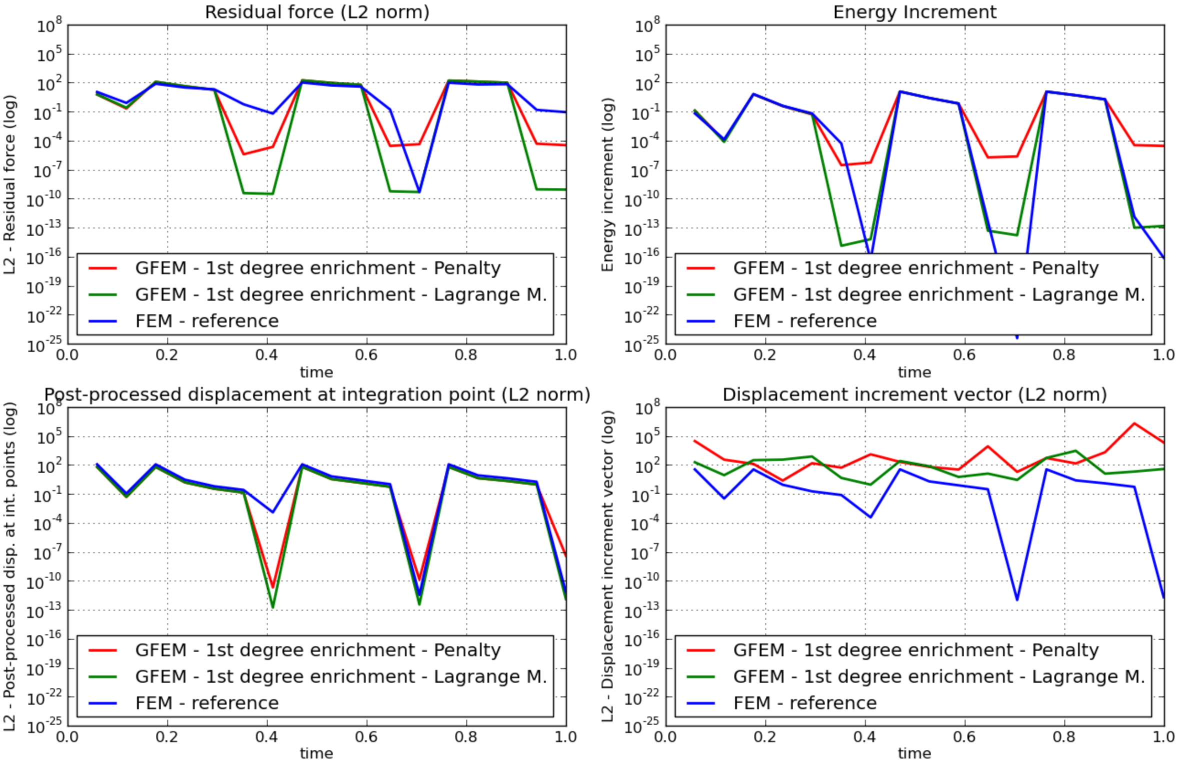


Figure 6.28. Convergence parameters history obtained in the nonlinear incremental iterative solution process – first degree enrichment.

An important aspect to be discussed for the present example regards the nonlinear GFEM models robustness. For the present model, for instance, the first degree enrichment GFEM model converged both using the penalty and the Lagrange multipliers techniques for imposing the displacement boundary conditions. However, when a second degree enrichment scheme is applied, the model for which the penalty method is used did not converge.

On the other hand, the GFEM model converges also for the second degree enrichment if the Lagrange multipliers method is used to enforce the Dirichlet boundary conditions. The convergence parameters history observed for such model is presented in Figure 6.29.

In fact, depending on the nonlinear model complexity and its level of nonlinear response, we observed that higher order enrichments can result in a nonlinear model which does not converge. The system of equations condition number, already discussed in Section 5.3.8, is probably one of the causes of such loss of convergence. Numerical stability techniques to solve such numerical issues are discussed and proposed for further studies in the conclusion chapter.

It is important to mention that for most of the cases, the first degree enrichment GFEM model is accurate, and generally, no remarkable improvements are noted when compared to the solution achieved for the second degree enrichment schemes. For instance, the results for the second degree enrichments, depicted in Figure 6.30, are as good as the ones presented in Figure 6.27.

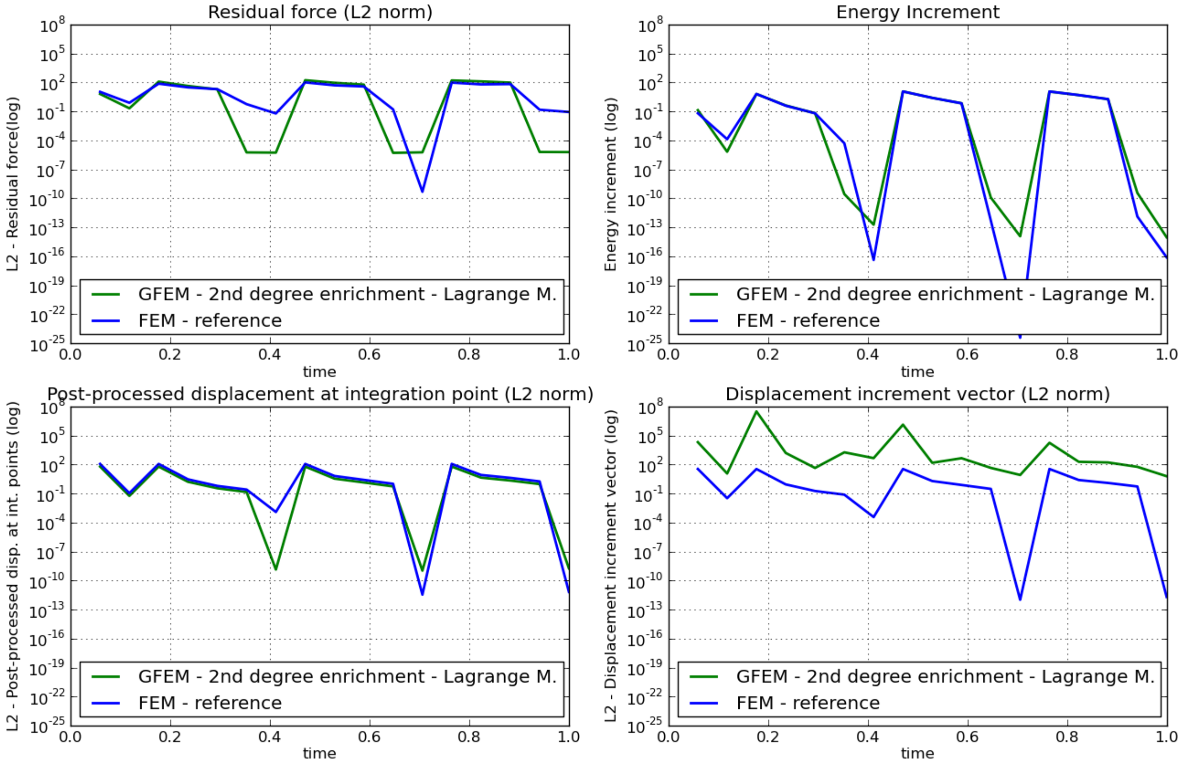


Figure 6.29. Convergence parameters history obtained in the nonlinear incremental iterative solution process

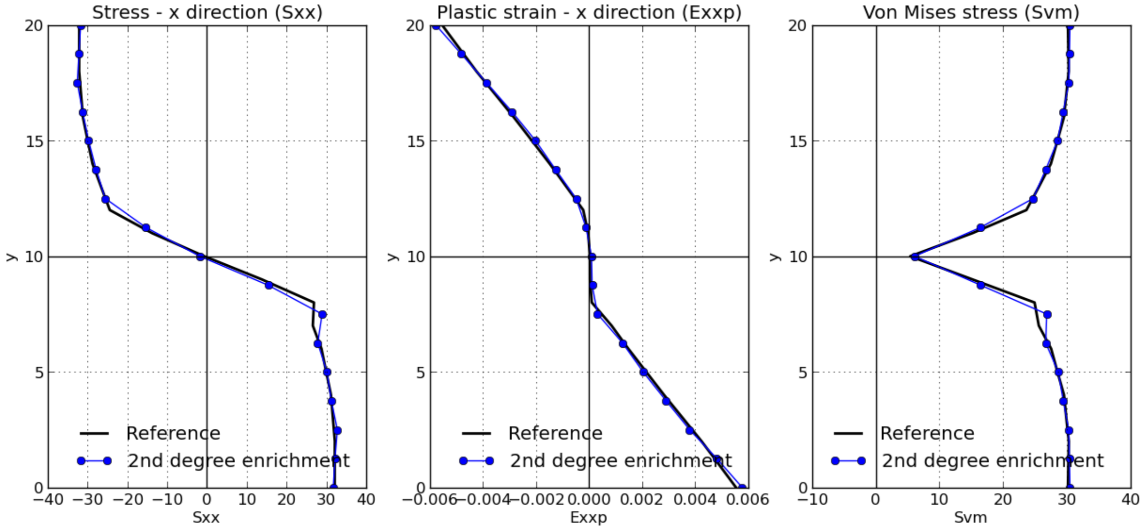


Figure 6.30. Horizontal stress (Sxx), horizontal plastic strain (Exxp) and von Mises stress (Svm) field values across the cross section, position 10 length units from the left clamped, achieved for second degree enrichment GFEM model.

In fact, it seems that the better relation between accuracy and computational cost is achieved with the association of an h-refinement (mesh refinement) to the first degree enrichment, since in general, the second degree enrichment results computationally costly and less stable.

6.3. Two-layered tube

The present example is proposed to evaluate the generalized mortar contact element formulation. In order to do so, a two-layered tube, loaded by an internal constant pressure is modeled. Both tube layers are made of the same material, for which it is considered a linear elastic model. The adopted geometry and material characteristics are presented in Figure 6.31.

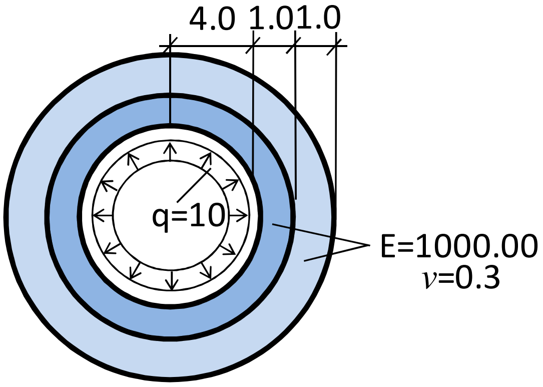


Figure 6.31. Two-layered with internal pressure example - proposed scheme data.

Due to the problem symmetry, just a quarter of the tube is modeled, assuming plane strain theoretical hypothesis for the 2D model. The internal layer is modeled by defining a regular mesh of curved quadrilateral elements, while the external layer is defined by a mesh of curved triangular elements, resulting in 88 elements, depicted in Figure 6.32 (a).

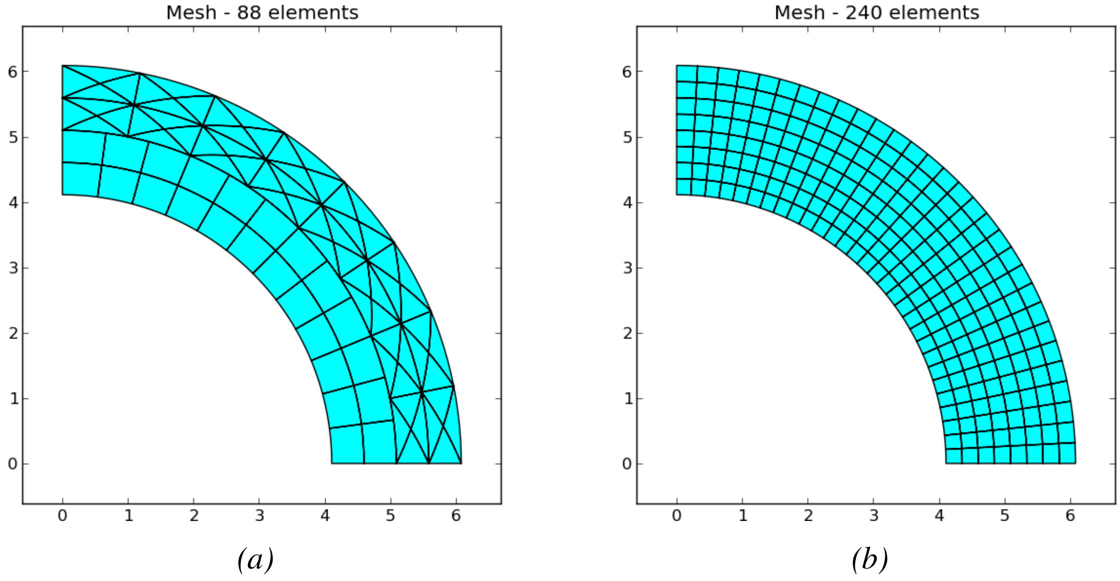
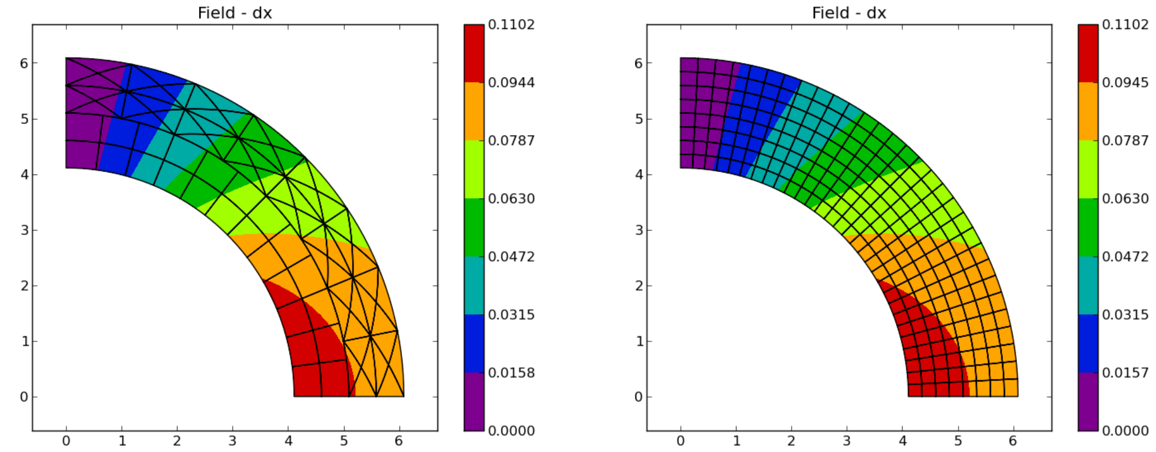


Figure 6.32. Proposed 2D mesh for the two-layered tube (a). A refined mesh of curved quadrilateral elements is defined for an equivalent one layered tube (b), for comparison reasons.

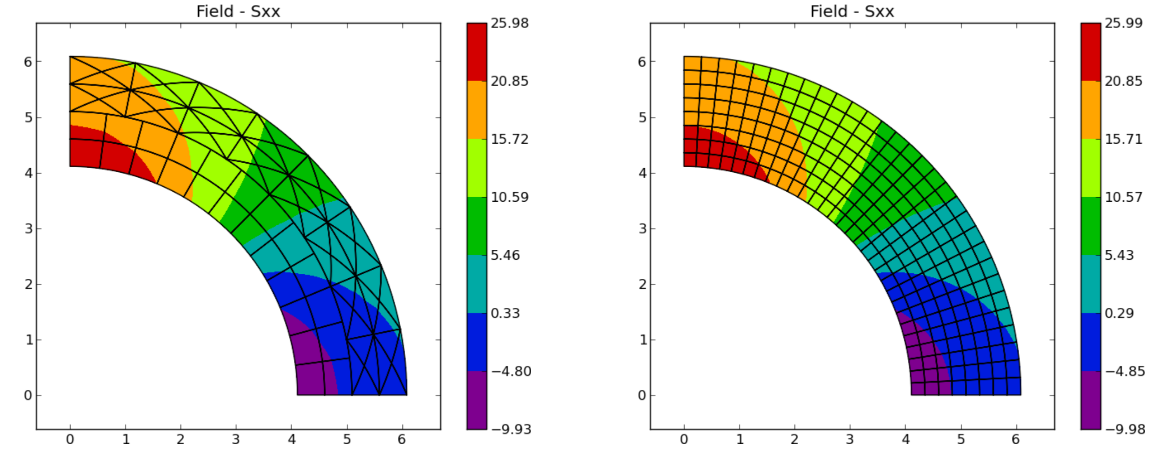
For the GFEM model, second degree partitions of unity (PU) are used for defining the elements geometry (Q8 for the quadrilaterals, T6 for the triangles), while the equivalent first degree PU are used for describing the structural behavior of the elements to be enriched (Q4 for the quadrilaterals, T3 for the triangles). Also, an isoparametric FEM model is build using the same mesh, for comparing the results with the ones achieved using the GFEM model. In this case, of course, the structural behavior is defined by the same PU used in the geometry description.

Finally, an equivalent one-layered tube is modeled using 240 isoparametric bi-quadratic (Q8) elements, aiming to compare the field results of the two-layered GFEM model (Figure 6.32 (b)). The results for both first degree shifted functions GFEM model and the FEM one-layered model are presented in the following figures (Figure 6.33 to Figure 6.36). For the GFEM model, all the nodes are enriched, and the symmetry conditions are applied over the whole elements' sides, using the penalty method.



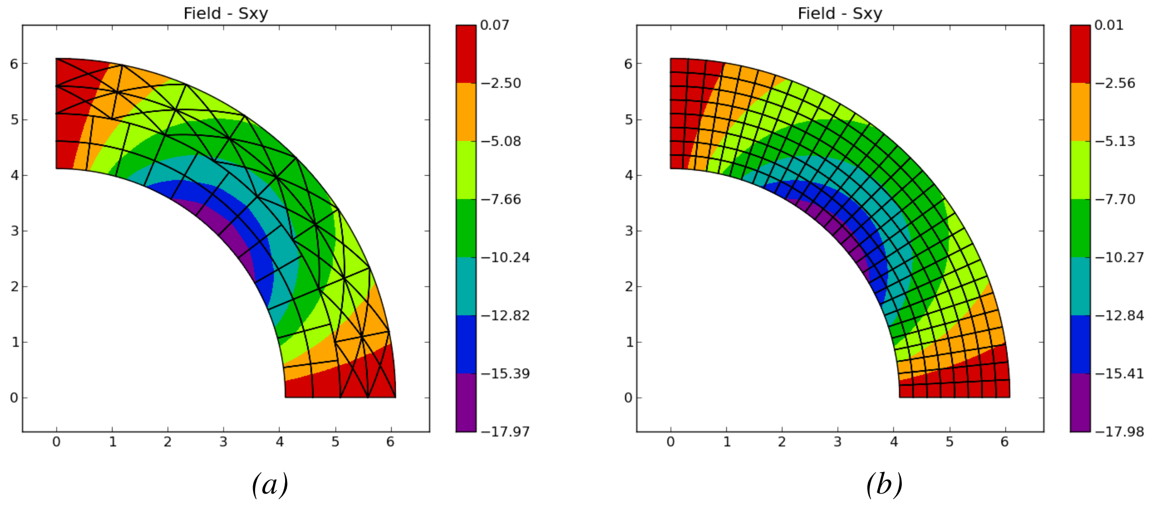
(a) *(b)*

Figure 6.33. Horizontal displacement (dx) field for the first degree enrichment shifted functions GFEM model (a) and the one layered tube isoparametric FEM model (b).



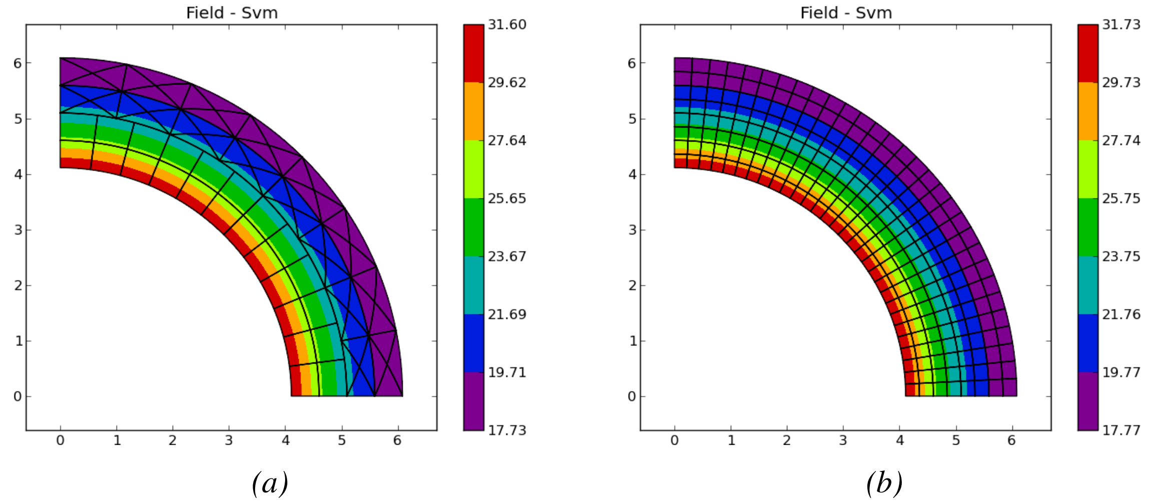
(a) *(b)*

Figure 6.34. Horizontal stress component (Sxx) field for the first degree enrichment shifted functions GFEM model (a) and the one layered tube isoparametric FEM model (b).



(a) *(b)*

Figure 6.35. Shear stress component (Sxy) field for the first degree enrichment shifted functions GFEM model (a) and the one layered tube isoparametric FEM model (b).

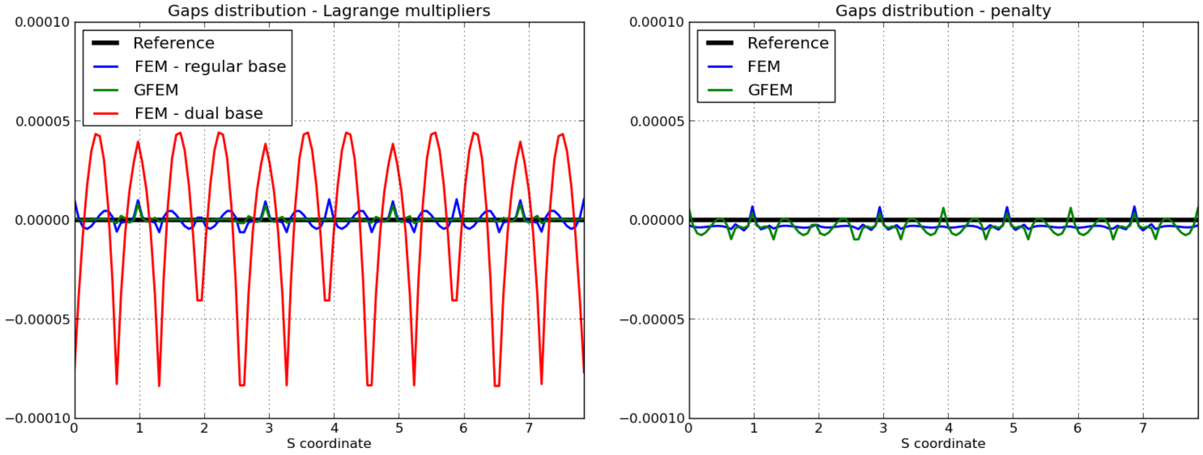


(a) *(b)*

Figure 6.36. Von Mises stress (Svm) field for the first degree enrichment shifted functions GFEM model (a) and the one layered tube isoparametric FEM model (b).

As it can be observed, the generalized mortar contact elements provided the expected field results continuity across the contact interface, resulting in field distributions quite similar to the ones of the one-layered model. Similar field results are achieved for the FEM two-layered tube model.

Both the Lagrange multiplier and penalty mortar formulation were used to compute the FEM and the GFEM model, and the resulting contact gaps, estimated and real traction along the contact interface are depicted in Figure 6.37, Figure 6.38 and Figure 6.39, respectively. In such graphics, the variable in the horizontal axis is the coordinate *S* of the curved line, starting from the left of the quarter solid model.

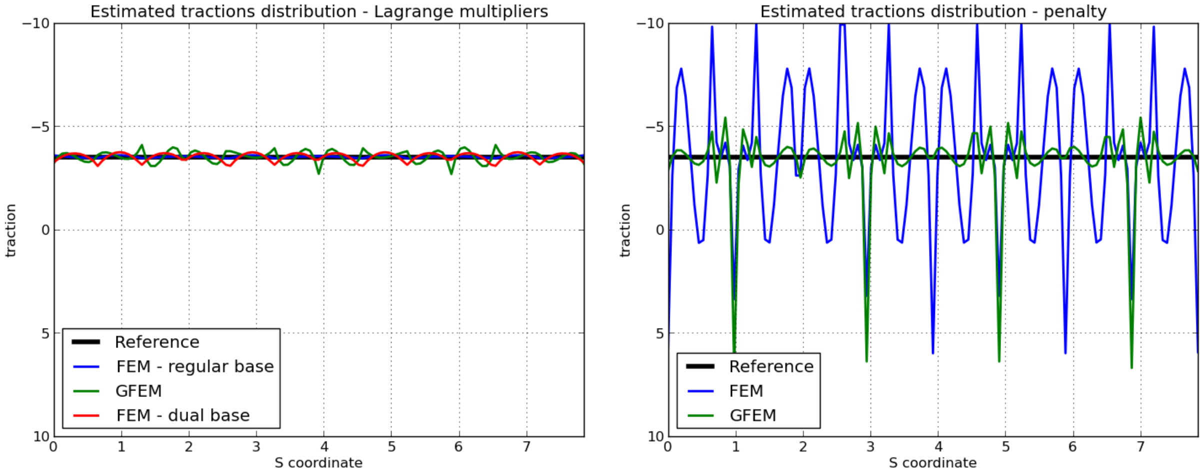


(a) *(b)*

Figure 6.37. Gaps at the contact interface for the Lagrange multipliers (a) and penalty (b) based mortar contact elements formulation.

For the Finite Element model, the Lagrange multipliers formulation is also evaluated using a dual base for describing the Lagrange multipliers field (see Wriggers (2006)). The achieved results are also indicated in the graphics.

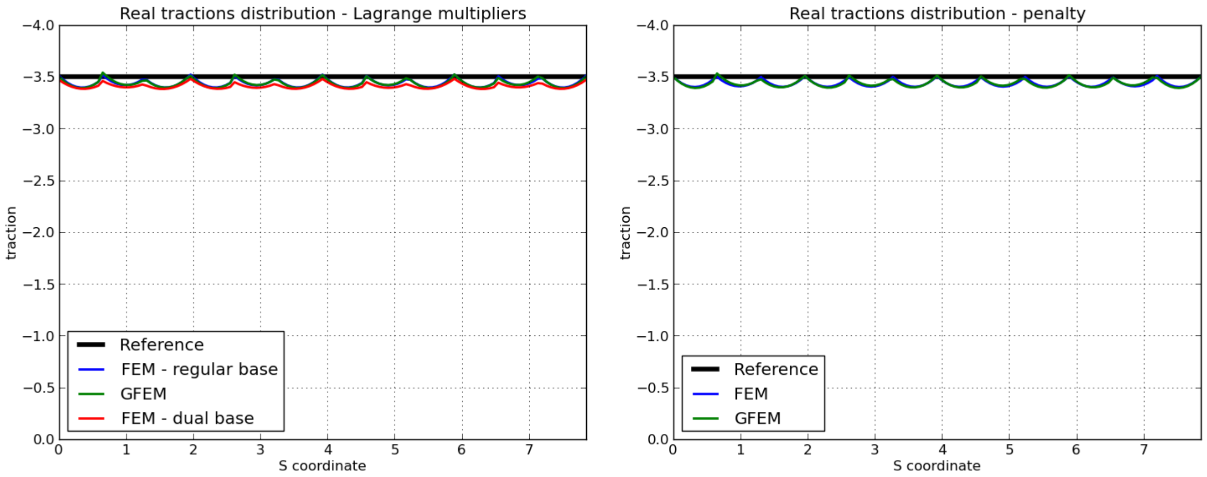
It is worth to inform that such dual base did not result in stable models for the GFEM, at least for the tested models, since they did not converge. Also, for the present model, the FEM Lagrange multipliers based mortar contact element using the dual base resulted in gap results worst than the ones achieved using the regular base for the Lagrange multipliers field.



(a) *(b)*

Figure 6.38. Estimated tractions at the contact interface for the Lagrange multipliers (a) and penalty (b) based mortar contact elements formulation.

Regarding the estimated tractions, all models presented similar results for the Lagrange multipliers methods. For the penalty method, the estimated tractions presented poor quality results, at least regarding the tractions distributions. The real tractions presented similarly good traction distribution results for all cases.

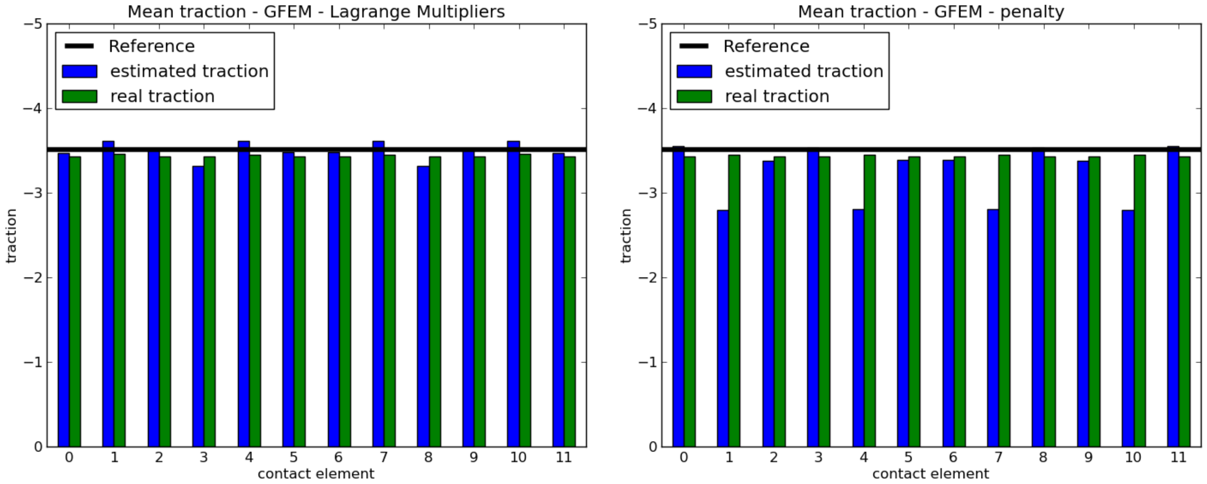


(a) *(b)*

Figure 6.39. Real tractions at the contact interface for the Lagrange multipliers (a) and penalty (b) based mortar contact elements formulation.

Since the mean values of estimated tractions are used by the method to verify the traction condition, it is worth to verify such results for the present example.

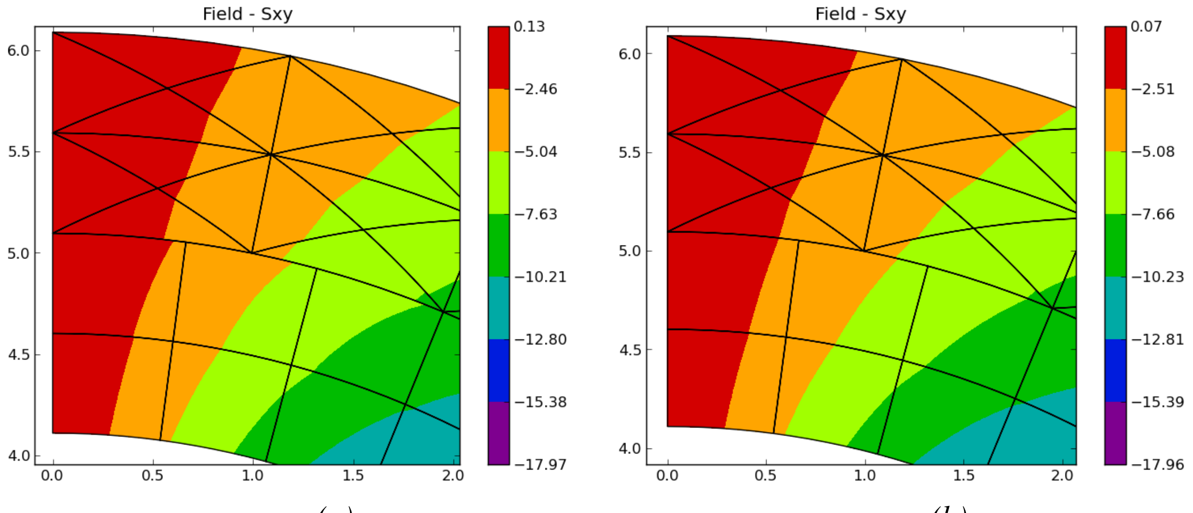
The estimated tractions mean values are shown in Figure 6.40. The mean values of the real tractions are also presented in the same figure. The mortar contact elements are associated to the internal layer elements’ sides, so they are numbered from the left to the right, starting in from zero to eleven.



(a) *(b)*

Figure 6.40. Mean value of the estimated tractions for each of the generalized mortar contact elements, both for the Lagrange multiplier (a) and penalty (b) based formulations.

Finally, regarding the FEM model, it is important to mention that, at least for the present model, some of the fields achieved for the dual base Lagrange multiplier mortar contact element are less continuous across the contact interface than the ones that use the regular base. Such fact can be noted, for instance, in the shear stress field depicted in Figure 6.41.



(a) *(b)*

Figure 6.41. Shear stress field for the dual base (a) and regular base (b) interpolation for the Lagrange multipliers field.

6.4. Beams contact

A large displacement contact problem is proposed in the present example. The structural set is composed by two beams, one positioned over the other, as depicted in Figure 6.42. A large vertical displacement is imposed at the clamped edge of one of the beams, such that it pushes the other beam, resulting in a large sliding contact between the beams surfaces. One considers the Saint Venant-Kirchhoff constitutive model for the material. Self weight is neglected.

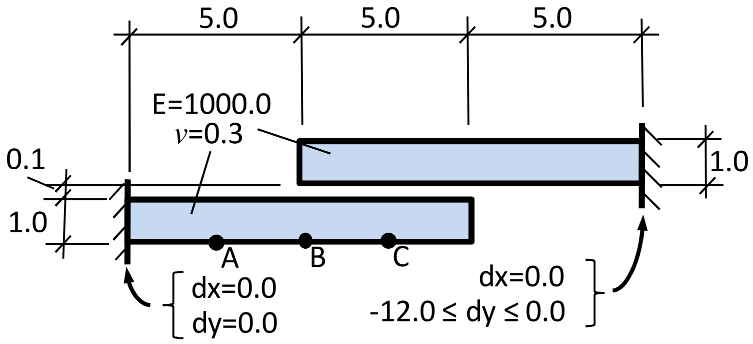


Figure 6.42. Two beams contact problem proposed data.

We consider the reference solution for the proposed problem a FEM model defined by a regular mesh of 160 (4x40) bi-quadratic (Q8) elements for each of the beams, resulting in 2274 degrees of freedom for the whole structural set. Taking into account that for the present example Lagrange multiplier based mortar contact elements are used, such number is increased up to 2329.

It is important to mention that the Dirichlet boundary conditions are applied using the penalty method, such that no additional degrees of freedom are related to it.

The proposed mesh for the GFEM model is composed by 40 (2x20) bi-linear (Q4) elements for each of the beams, resulting in 252 degrees of freedom for the not enriched model. Taking into account the Lagrange multipliers degrees of freedom associated to the mortar contact elements, such number increases up to 263.

The GFEM model is enriched using shifted functions. First degree enrichment is applied over the structural set nodes, except the ones in which Dirichlet boundary conditions are applied, resulting in 743 degrees of freedom.

The displacement is imposed up to the value of 12 length units, in downward direction. The nonlinear solution is computed in 50 time step increments. The adopted convergence criterion is the L2 norm of the post-processed displacement increment at integration points. The final equilibrium configuration for the reference solution model is presented in Figure 6.43.

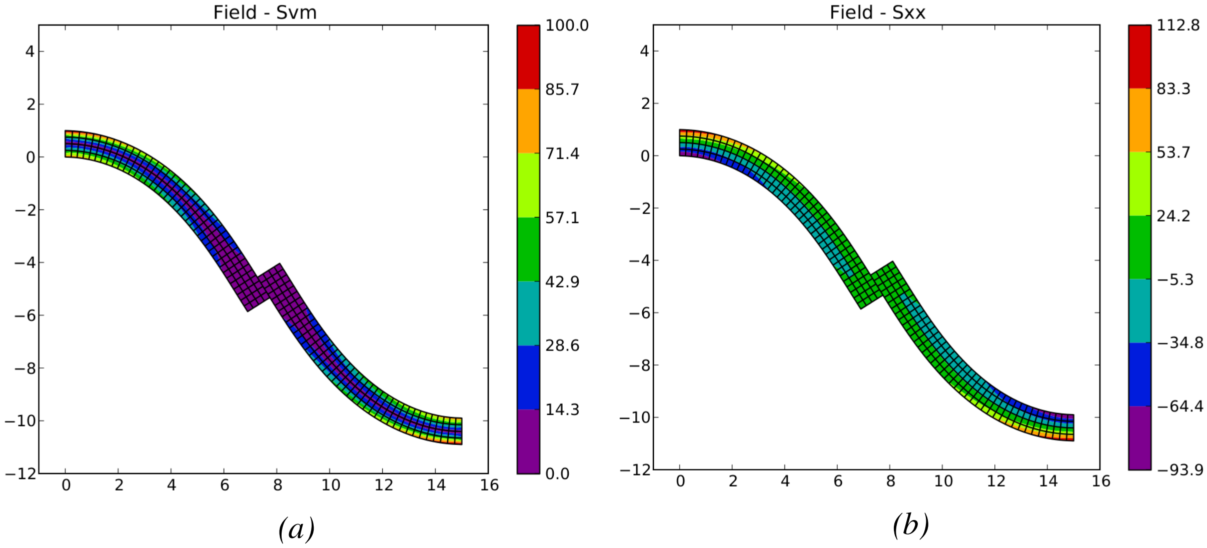


Figure 6.43. Final equilibrium configuration of the proposed problem, achieved for the FEM reference solution model; von Mises stress (a) and horizontal stress component (b) fields.

The same stress fields for the GFEM model are presented in Figure 6.44 and Figure 6.45, both for the not enriched and the first degree enrichment model. As it can be observed, the fields observed for the enriched model are smoother and more continuous across the elements.

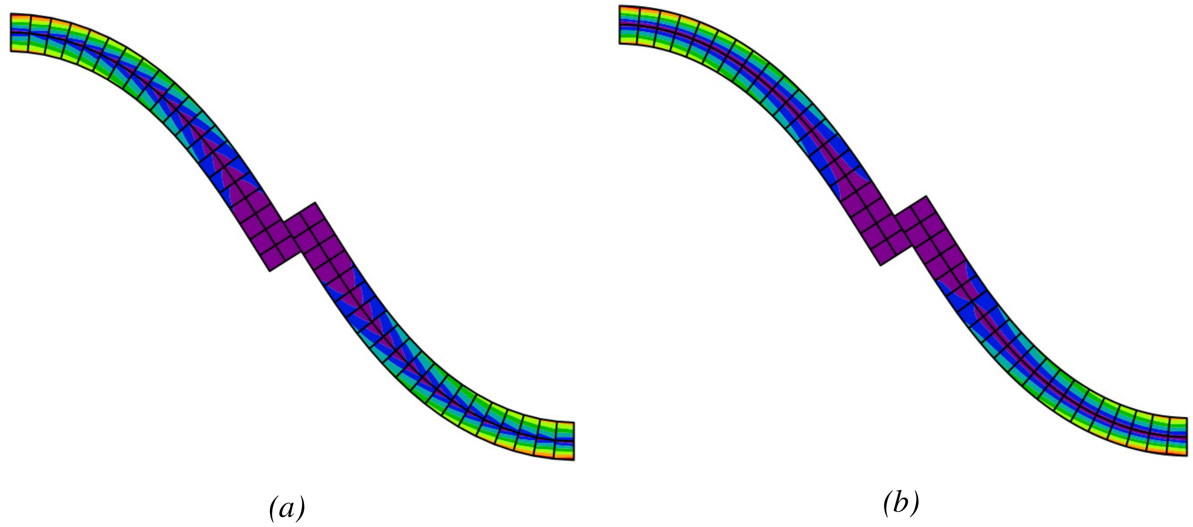


Figure 6.44. Von Mises stress (Svm) field for the not enriched (a) and the first degree enrichment GFEM model (b).

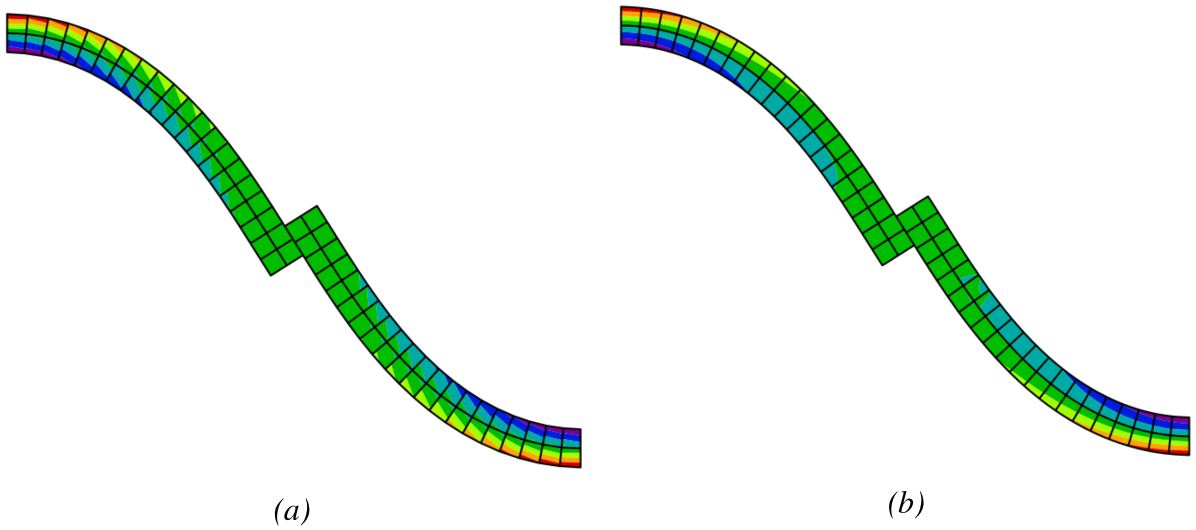


Figure 6.45. Horizontal stress component (Sxx) field for the not enriched (a) and the first degree enrichment GFEM model (b).

Taking into account the considerable change of deformed configuration during the proposed nonlinear problem, it is interesting to depict such deformed configurations along the whole process. Such set of images is depicted in Figure 6.46.

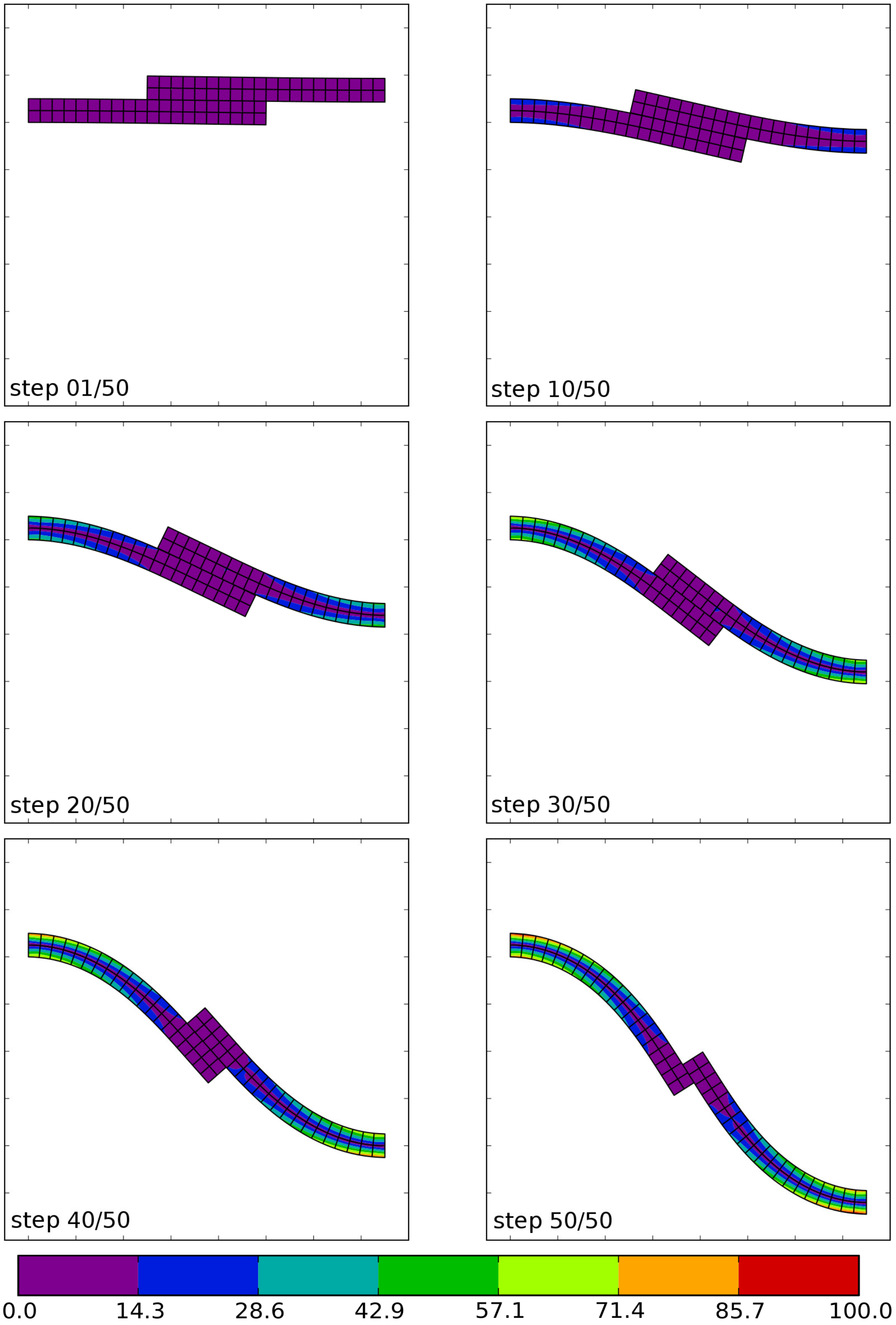


Figure 6.46. Deformed configuration of the structural set along the nonlinear process.

The evolution of the von Mises stress (Svm) during the solution process, until the final time (1.0), for the points A, B and C indicated in Figure 6.42 are depicted in Figure 6.47, Figure 6.48 and Figure 6.49.

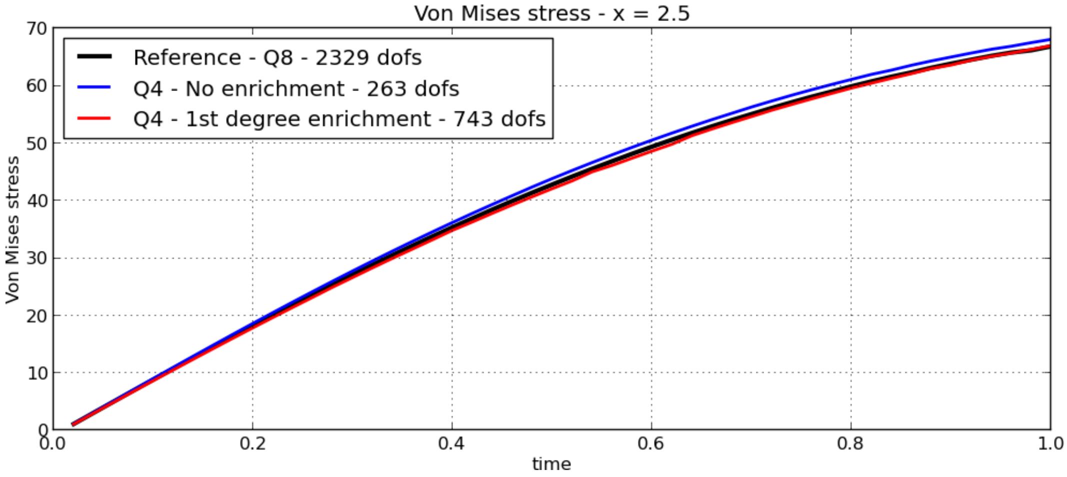


Figure 6.47. Von Mises stress value evolution during the nonlinear process, at point A.

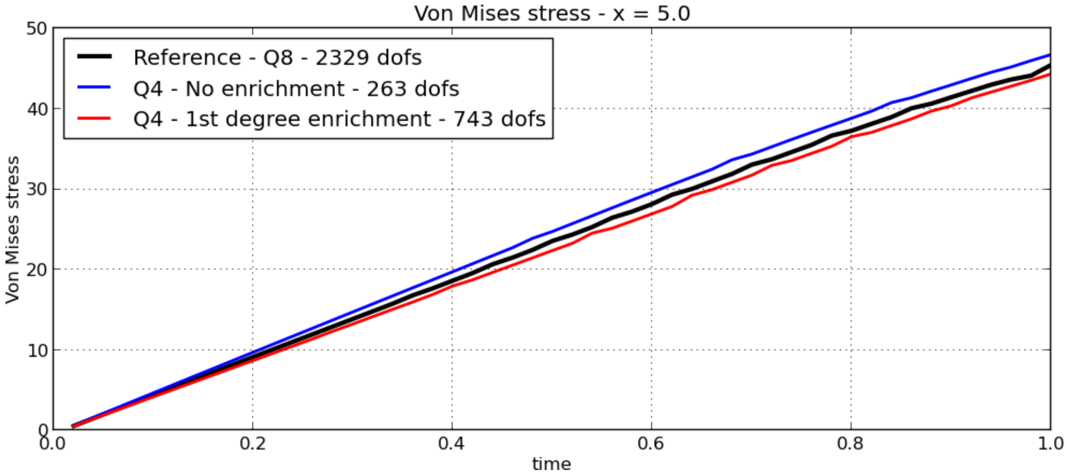


Figure 6.48. Von Mises stress value evolution during the nonlinear process, at point B.

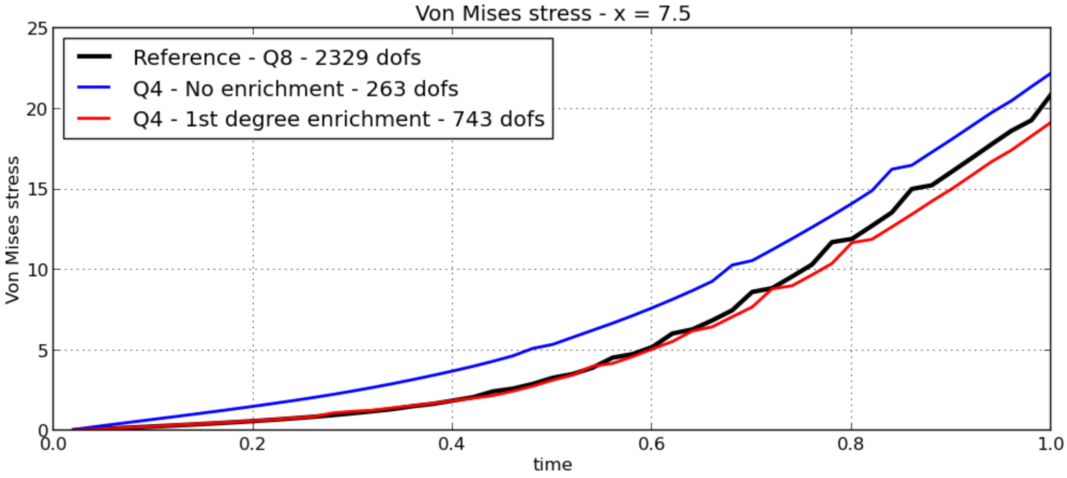


Figure 6.49. Von Mises stress value evolution during the nonlinear process, at point C.

In order to advance in the results analysis, the gaps and real tractions at the contact interface are verified in a time step for which a great amount of the beam surfaces remain in contact. Taking the contact gaps results at time step 10 (vertical imposed displacement equal to 2.4 length units), one notices that all the models present similar results (Figure 6.50).

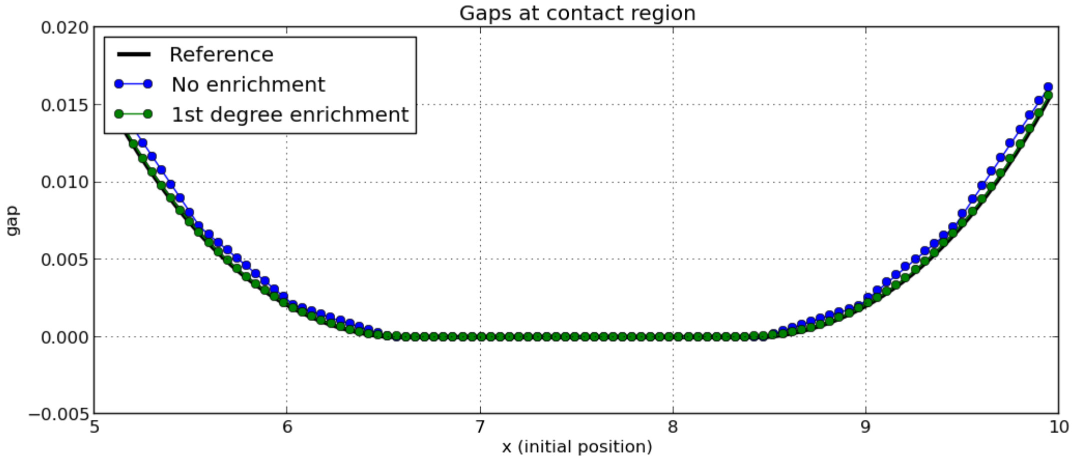


Figure 6.50. Gaps at contact region for time step 10/50.

Even though the gaps results are similar, the real tractions evaluated at the contact interface differs considerably from the not enriched and the first degree enrichment GFEM model, as depicted in Figure 6.51.

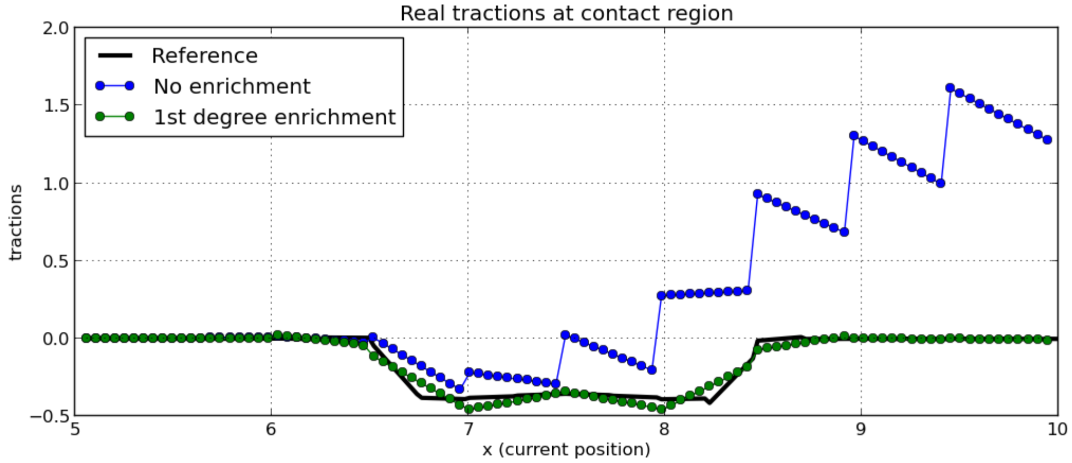


Figure 6.51. Real tractions at contact region for time step 10/50.

By analyzing those results, one notices that the enrichment benefits goes beyond the fields smoothing present in Figure 6.44 and Figure 6.45, since considerable traction error is observed for the not enriched GFEM model, while the enriched model is much closer to the reference solution.

In fact, even if one generates a refined mesh of bi-linear elements, the tractions results remain not correct. For instance, if one takes a model defined by a bi-linear element mesh, similar to the one used for the reference solution (160 elements for each of the beams), the tractions results remain not accurate. The traction results for such refined GFEM model are presented in Figure 6.52.

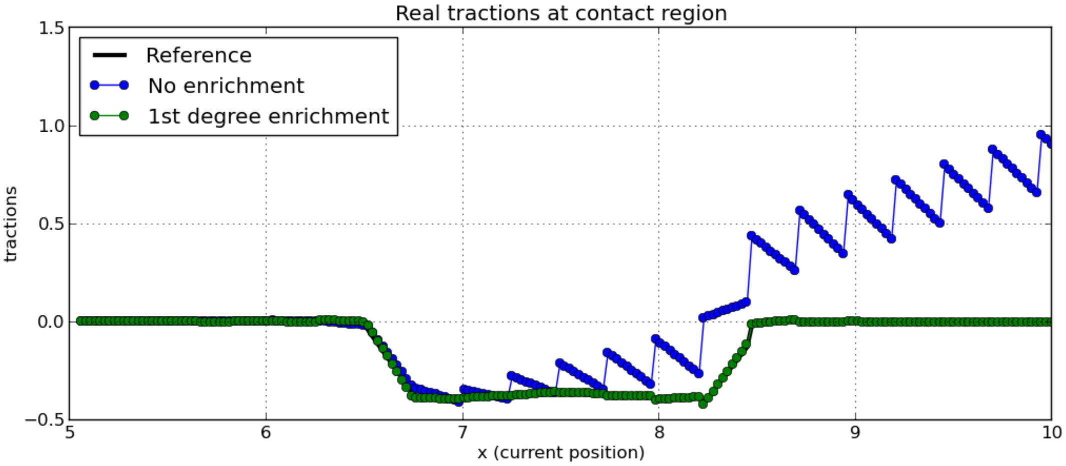


Figure 6.52. Real tractions at contact region for time step 10/50 – refined mesh GFEM model.

6.5. Sliding arcs

A large strain contact problem is proposed in the present example. The structural set is composed by two arcs, one positioned over the other, as depicted in Figure 6.53. A large horizontal displacement is imposed at the clamped edges of the top arc, such that it slides pushing the other's arc external surface. Large sliding contact between the arcs' surfaces is observed. A Neo-Hookean hyperelastic model is adopted for the arcs' material. Self weight is neglected. The present example is inspired in the one presented by Yang et al (2005).

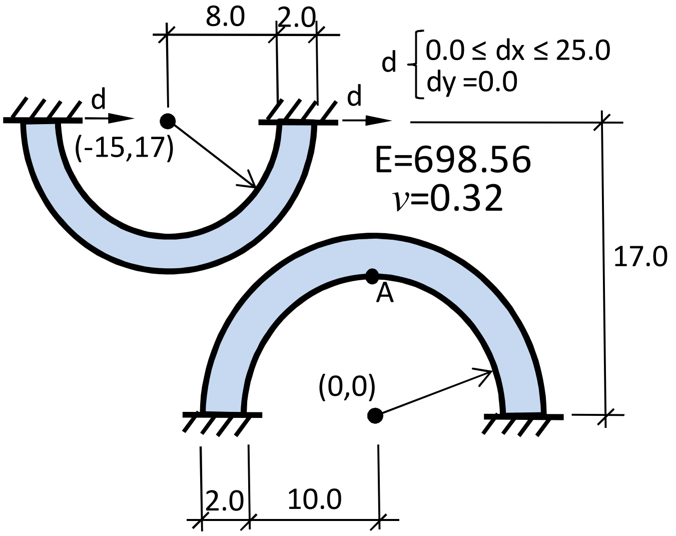


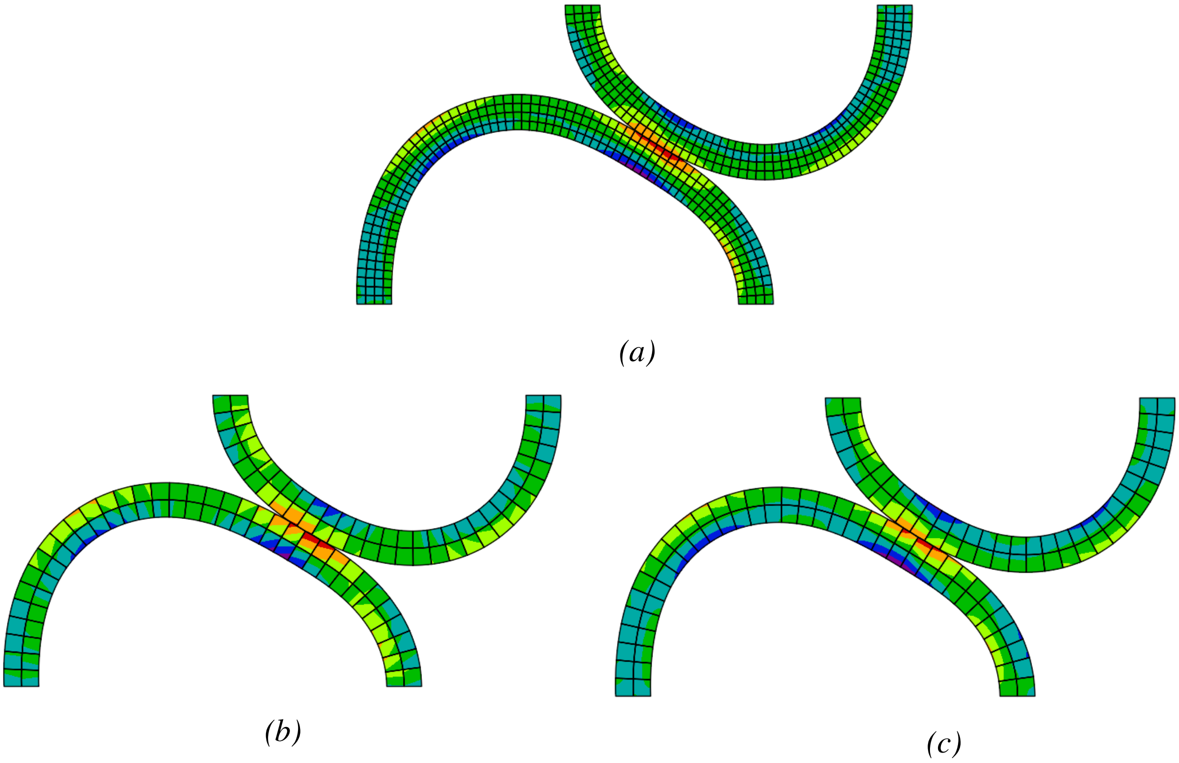
Figure 6.53. Sliding arcs contact problem proposed data.

We consider the reference solution for the proposed problem a FEM model defined by a regular mesh of 520 bi-quadratic (Q8) isoparametric (curved) elements, resulting in 3795 degrees of freedom for the whole structural set, already taking into account the ones related to the Lagrange multiplier based mortar contact elements. The Dirichlet boundary conditions are applied using the penalty method, such that no additional degrees of freedom are related to them.

The proposed mesh for the GFEM model is composed by 130 elements. For such elements, a bi-linear (Q4) PU is defined for describing the structural behavior, while a bi-quadratic PU (Q8) is used to describe the elements curved geometry. It results in a model containing 435 degrees of freedom if no enrichment is applied (including the degrees of freedom associated to the Lagrange multipliers contact elements).

The GFEM model is enriched using shifted functions. First degree enrichment is applied over all nodes, except the ones in which Dirichlet boundary conditions are applied and the ones positioned on the arcs external surface, resulting in 939 degrees of freedom.

The displacement is imposed up to the value of 25 length units, in left to right direction. The nonlinear solution is computed in 200 time step increments. The adopted convergence criterion is the L2 norm of the post-processed displacement increment at integration points. The final equilibrium configuration for the reference solution and the GFEM models is presented in Figure 6.54 and Figure 6.55.


(a)

(b)

(c)

Figure 6.54. Final deformed configuration of the sliding arc problem, for (a) the reference solution model, (b) the GFEM model, not enriched, and (c) first degree enriched GFEM model. The field presented is related to the horizontal stress component (Sxx).

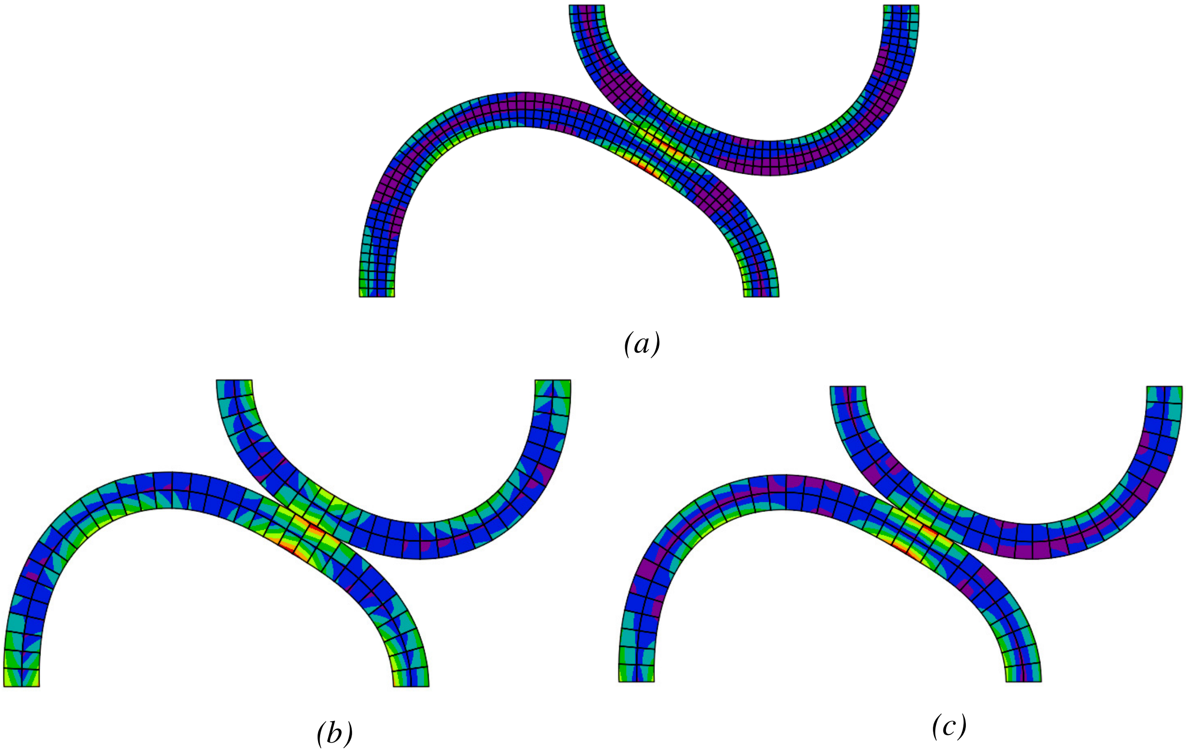


Figure 6.55. Final deformed configuration of the sliding arc problem, for (a) the reference solution model, (b) the GFEM model, not enriched, and (c) first degree enriched the GFEM model. The field presented is related to the von Mises stress value (Sxx).

It is important to explain the reason for not enriching the nodes positioned at the external boundaries of the arcs. As already mentioned earlier, in the examples computed in the present research, we found that, even though better results can be achieved for the GFEM, when highly nonlinear behaviors are observed, the enriched model can fail to find convergence. This is the case of the present example, in which a remarkable nonlinear response is observed in the model.

On the other hand, the proposed problem complexity demanded several parameters adjustments even in the FEM model. In addition to this fact, it is important to inform that the solution diverged for the totally enriched GFEM model only at advanced instants of the nonlinear analysis.

Finally, it is important to comment again that one of the advantages of the Generalized Finite Element Method is related to the fact that it allows one to enrich only specific nodes, so that is possible to avoid enriching regions for which the enrichment may cause undesired effects. The present model is a good example in which such characteristic is useful.

Taking into account the considerable change of deformed configuration during the proposed nonlinear problem, it is interesting to depict such deformed configurations along the whole process. Such set of images is depicted in Figure 6.56.

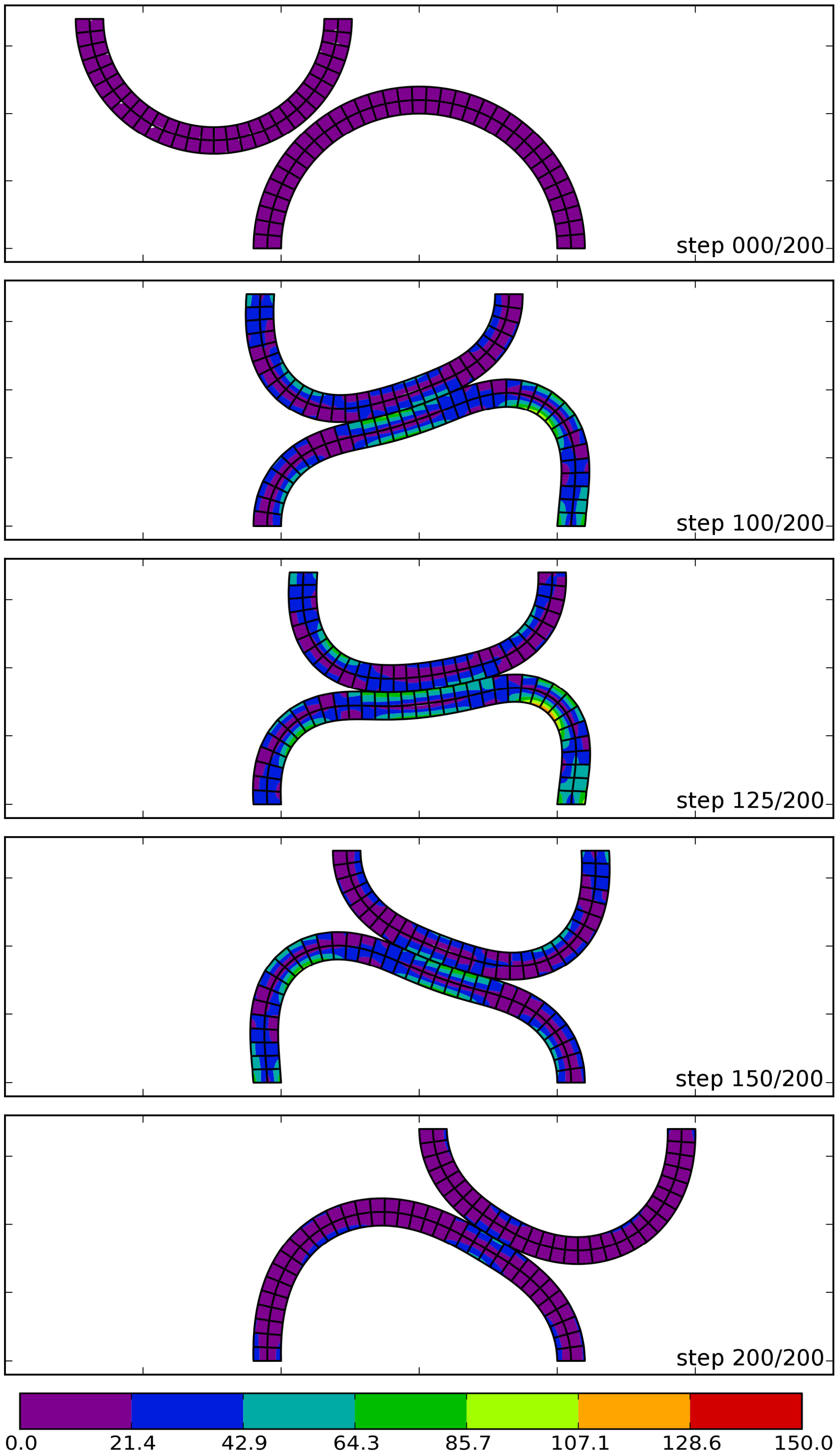


Figure 6.56. Deformed configuration of the structural set along the nonlinear process.

Finally, in order to illustrate the nonlinear behavior observed in the present problem, the vertical displacement and the horizontal component of the stress along the whole nonlinear process (until the final time equal to 1), in point A (see Figure 6.56), is shown in Figure 6.57 and Figure 6.58 , respectively.

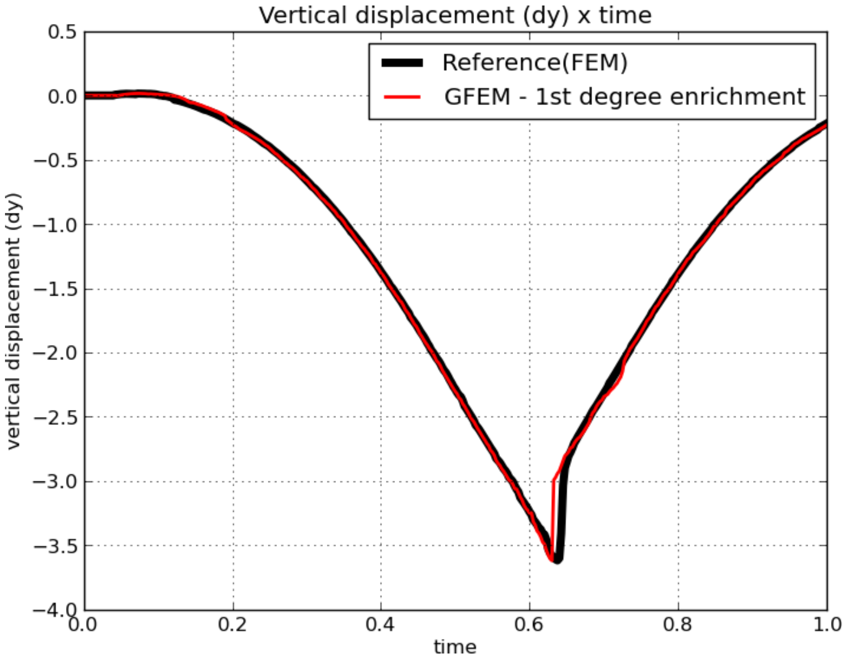


Figure 6.57. Vertical displacement (dy) computed at point A along the nonlinear solution process.

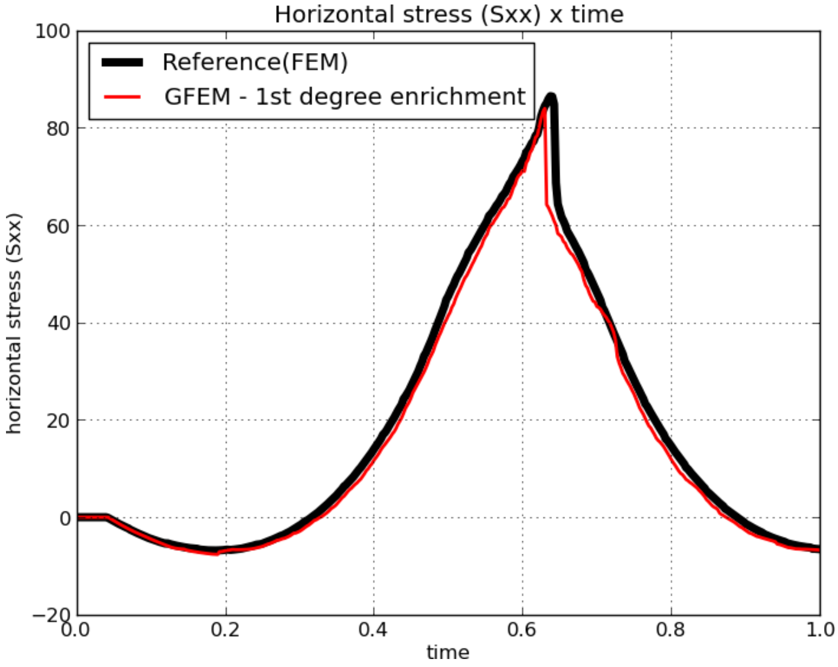


Figure 6.58. Horizontal stress component (Sxx) computed at point A along the nonlinear solution process.

7 - Conclusions

Even though the Generalized Finite Element Method (GFEM) efficiency for solving linear crack problems has already been established in the technical literature, its use for general nonlinear analysis purposes is still an open subject. The present dissertation has studied the method's application for performing nonlinear analyses of Solid Mechanics. The resulting contributions for the theme can be summarized as:

- One has evaluated the shifted and polynomial enrichment functions efficiency in improving the linear partitions of unity interpolation. Despite the fact that such functions have already been used in other researches, some of the presented results bring new information on the subject;
- Regarding the dynamical analyses, one has found that remarkable improvements can be achieved for some of the employed enrichment schemes, for which the total number of DOF was considerably smaller than the ones observed for the FEM models. These results were not found in the literature, and might constitute original contributions for the theme;
- The 2D elastoplastic examples confirmed that the enrichment feature can improve such problem's results for coarse meshes, as it was already found in other researches;
- Likewise for the dynamical models, the kinematically nonlinear examples results have shown that remarkable benefits can be achieved by means of the GFEM enrichment applied for these problems, having enhanced the accuracy of the model at a low computational cost. These results might also constitute original contributions of the present work;
- Probably the most relevant contribution is the formulation, the computational implementation and the numerical assessment of the segment-to-segment frictionless contact elements for the GFEM. The achieved results have shown that, in some cases, the generalized contact elements hereby proposed can provide results even more accurate than the ones delivered by the FEM;
- Finally, an indirect but nonetheless important contribution is the conceiving of an efficient Object-Oriented design for the GFEM programming.

The developed computational code was named SCIEnCE, standing for São Carlos Integrated Environment for Computational Engineering. It is important to mention that presently it is already being used by other doctoral researches. Moreover, a research group having SCIEnCE as its main tool was conceived and registered in CNPq (National Council for Scientific and Technological Development).

Going into more detailed conclusions, regarding the linear elastic model, the performed validation examples have shown that both shifted and non-shifted polynomial enrichments present similarly accurate results, also demonstrating similar efficiency in circumventing errors due to mesh distortion and element side curvatures. Such feature was efficiently explored by mixing different partitions of unity for describing the element's geometry and structural behavior. Actually, the use of the GFEM enrichment framework over linear approximation curved elements resulted in accurate displacement and stress fields, even for very coarse meshes.

Yet regarding the linear elastic model, it is worth mentioning that bilinear (Q4) elements present poor quality results for flexural problems (Wilson et al (1971) apud Brebbia and Connor (1973)). In spite of this fact, the polynomial enrichment functions were able to improve considerably the results in these cases, circumventing such feature.

The displacement boundary conditions enforcement along the whole element side was adopted and it has shown to be especially consistent when the enrichment is applied at boundary nodes. A similar consistency is not observed when the displacement boundary conditions are enforced only at the boundary nodes. In spite of that, such element-wise enforcement is not trivial, and demands quite sophisticated computational routines. Both the Lagrange multipliers and the penalty based formulation allow one to correctly impose such conditions over the whole element side for generic enrichment functions.

In particular, for the Lagrange multipliers based formulation, one can also enrich the Lagrange multipliers field. It is worth to mention that the enrichment of the multipliers fields with the same degree of the enrichment applied to the displacement field provides more accurate results. However, this enrichment feature might be explored carefully since singular system of equations can result, depending on the number of degrees of freedom associated to the enriched Lagrange multiplier field. On the other hand, the penalty based formulation naturally encompasses all the enrichment terms, thus naturally promoting sufficiently accurate results.

Even though just one structural problem was considered in the linear dynamics analyses, different enrichment schemes were tested. One concludes that considerably different dynamical behavior can be observed depending on the type of enrichment employed. For instance, several proposed enrichment schemes failed, mainly due to the loss of convergence during the time integration. Those fails are associated to numerical instabilities, which can be explained by the condition number of the resulting system of equations. On the other hand, for all the tested enrichment functions leading to favorable condition numbers, accurate and stable results schemes were found. Probably, the alternative presenting better compromise among computational cost, numerical stability and accuracy was achieved using first degree shifted enrichment functions associated to a non enriched consistent mass matrix. This alternative successfully passed in tests regarding long time integration models (up to 1800 time steps) and harmonic load problems.

Regarding the nonlinear behavior due to contact, both node-to-segment and segment-to-segment contact elements were derived and implemented for being applied in the GFEM framework. In a general sense, one can affirm that those elements verify and impose the contact conditions in a node-wise and in an element-wise fashion, respectively. Their performance for the FEM was also assessed. In this case, when applied to problems of frictionless contact between a deformable solid and a rigid support (Signorini problem), both node-to-segment and segment-to-segment contact elements provided similarly accurate results.

In the GFEM framework, the node-to-segment contact elements did not verify the impenetrability condition in the regions between nodes, since the contact conditions were imposed in a node-wise fashion, generating stress and strain concentration at the nodes.

On the other hand, the derived enriched segment-to-segment contact elements successfully enforced the impenetrability condition along the entire contact element side. Both Lagrange multipliers and penalty based contact elements have shown to be accurate and stable. The possibility of enriching the Lagrange multipliers field for imposing Dirichlet boundary conditions is present as well.

Differently from the Signorini problem, in problems for which the contact is observed among deformable bodies, the nodal enforcement promoted by the node-to-segment contact elements does not guarantee accurate solutions, even for the traditional FEM. For such problems, the use of segment-to-segment contact elements, based on the so-called mortar contact elements formulation, is recommended. Furthermore, when applied to GFEM models, the segment-to-segment contact elements, referred to here as generalized mortar contact elements, presented accurate results for the proposed problems. In fact, considering a patch test, the vertical stress field provided by the enriched mortar contact elements is even more accurate than the ones obtained with higher order FEM mortar elements.

As it happens for the higher order mortar contact elements, both gap and traction values oscillate along the contact element when higher degrees of enrichment are adopted. Clearly these oscillations can prejudice the stability of the numerical solution and constitute a feature to be improved. Recently, several techniques are being proposed in the literature for recovering stress results in the GFEM, since oscillations in such fields are known effects observed for shifted and polynomial higher order enrichment schemes. Therefore, following this trend, the use of recovering techniques for enhancing the contact tractions is hereby suggested as one of the most natural extensions of the current work.

On the other hand, it is important to bear in mind that one of the advantages of the GFEM is the possibility of enriching just a set of nodes (selective enrichment). Therefore it is totally possible to avoid the enrichment of nodes in regions that can cause instabilities in the model.

Taking all the above comments into account, it is possible to conclude that the present doctoral research has achieved the proposed objective of giving contributions for the solution of nonlinear Solid Mechanics problems by means of the GFEM.

In what follows, one states some suggestions on what we consider subjects that remain unsolved or might constitute natural continuations of the present work:

- The most straightforward continuation of the present work is to advance in the assessment of the accuracy, robustness and efficiency of the GFEM for solving nonlinear problems. Obviously, this aspect demands many more nonlinear analyses for exploring all the nonlinear functionalities implemented;
- At the current stage of development, the conceived computational code presents a nonlinear framework which stands for performing large deformation elastic frictionless contact analysis considering dynamical effects. Therefore, this framework already allows one to perform elastic impact analysis. Enhancing the current computational efficiency is an important demand for solving impact problems, which can be performed exploiting the Python language available tools (libraries). Moreover, including plasticity constitutive models for large deformation problems is a nontrivial but nonetheless natural extension of the research work;
- Taking advantage of the developed OO computational framework, other plasticity models and damage analysis functionalities can be inserted in the code;

- Following the same direction of the previous comment, the well-known crack analyses enrichment features can be implemented in SCIEnCE, aiming to obtain a damage and crack evolution analyses GFEM computation framework;
- The extension of the frictionless contact elements for frictional contact analysis, including the slip and stick behavior, would allow one to assess the performance of the generalized mortar contact element for frictional contact problems. To the author's knowledge, these results are also not yet available in the literature;
- Taking into account the numerical instabilities observed in some of the performed examples, a deeper study on the resulting system of equations condition number and on techniques to overcome such numerical problems is an important matter to improve the method stability and robustness. Regarding this subject, studies towards this direction are currently being performed in another doctoral research using the SCIEnCE computational framework.

Many other suggestions of future trends can be stated here. One believes that the above commented features constitute important additional contributions for efficiently using the Generalized Finite Element Method in nonlinear analyses of Solid Mechanics.

References

ALVES FILHO, J. S. R.; DEVLOO, P. R. B. Object Oriented programming in Scientific computations: the beginning of a new era. **Engineering Computations**, v. 8, Issue 1, p. 81-87, 1991.

ARCHER, G.C. **Object-Oriented Finite Element Analysis**. (Ph.D. dissertation). University of California at Berkeley. Berkeley, 1996.

ASSAN, A.E. **Método dos Elementos Finitos: primeiros passos.** Segunda edição. Editora Unicamp. Campinas, 2003.

BARROS, F.B. **Métodos sem malha e Método dos Elementos Finitos Generalizados em análise não-linear de estruturas.** (Tese de doutorado). Escola de Engenharia de São Carlos. Universidade de São Paulo. São Carlos, 2002.

BATHE, K. J. **Finite Element Procedures.** Prentice Hall. New Jersey, 1996.

BAZANT, Z.P.; CEDOLIN, L. **Stability of Structures: Elastic, Inelastic, Fracture and Damage Theories**. World Scientific Publishing. Singapore, 2010.

BELYTSCHKO, T.; LIU, W.K.; MORAN, B. **Nonlinear Finite Elements for Continua and Structures.** Wiley. New York, 2000.

BELYTSCHKO, T.; GRACIE, R.; VENTURA, G. A review of extended/generalized finite element methods for material modeling. **Modelling and Simulation in Materials and Science and Enginnering.** Volume 17. 24p, 2009.

BERNARDI, C.; MADAY, Y.; PATERA, A. A new conforming approach to domain decomposition: the mortar method. **Nonlinear Partial Differential Equations and their Applications.** John Wiley & Sons. New York, p 13-51, 2001.

BITTENCOURT, M. L. Using C++ templates to implement finite element classes. **Engineering Computations.** v. 17, n. 7, p. 775-788, 2000.

BONET, J.; WOOD, R.D. **Nonlinear Continuum Mechanics for Finite Element Analysis.** 2nd Edition. Cambridge University Press. Cambridge, 2008.

BORDAS, S. P. A., NGUYENS, P. V., DUNANT, C., GUIDOUM, A., NGUENS-DAND, H. An extended finite element library. **International Journal for Numerical Methods in Engineering**, v. 71, p. 703-732, 2007.

BREBBIA, C. A.; CONNOR, J. J. **Fundamentals of Finite Element Techniques for Structural Engineers.** Butterworth Co. (Publishers) ltd. Hungary, 1973.

CROSS, J. T., MASTERS, I., LEWIS, R. W. Why you should consider object-oriented programming techniques for finite element methods. **International Journal of Numerical Methods for Heat & Fluid Flow**, v. 9, n. 3, p. 333-347, 1999.

COWPER, G.R. Gaussian quadrature formulas for triangles. **International Journal for Numerical Methods in Engineering**, v 7, issue 3, p. 405-408, 1972.

DAVIS, T.A. A column pre-ordering strategy for the unsymmetric-pattern multifrontal method. **ACM Transactions on Mathematical Software**, v.30. n 2. p. 165-195, 2004a

DAVIS, T.A. Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method. **ACM Transactions on Mathematical Software**, v. 30, n. 2, p. 196-199, 2004b.

DAVIS, T.A.; DUFF, I.S. An unsymmetric-pattern multifrontal method for sparse LU factorization. **SIAM Journal on Matrix Analysis and Applications**, v. 18., n. 1 p. 140-158, 1997.

DAVIS, T.A., DUFF, I.S. A combined unifrontal/multifrontal method for unsymmetric sparse matrices. **ACM Transactions on Mathematical Software**, v. 25, n. 1, p.1-19, 1999.

DIAS, A.P.C (2013). **Elemento Mortar de Alta Ordem Aplicado à Análise Computacional não Linear de Contato Mecânico Estrutural.** (Dissertação de mestrado) Faculdade de Engenharia Mecânica. Universidade Estadual de Campinas. Campinas, 2013.

DOLBOW, J.E. **An Extended Finite Element Method with Discontiuous Enrichment for applied Mechanics.** (Ph.D. dissertation. Northwestern University) Evaston, 1999.

DUARTE, C.A.M.; ODEN, J.T. An hp adaptative method using clouds. **Numerical Methods in Applied Mechanics and Engineering**, v. 139, p. 237-262, 1996a.

DUARTE, C.A.M.; ODEN, J.T. Hp clouds – an hp meshless method. **Numerical Methods for Partial Differential Equations**, v. 12, p. 673-705, 1996b.

DUARTE, C.A.; BABUŠKA, I.; ODEN, J. Generalized Finite Element Methods for three-dimensional structural mechanics problems. **Computers & Structures**, v. 77, n. 2, p. 215-232, 2000.

DUBOIS-PÉLERIN, Y.; ZIMMERMANN, T.; BOMME, P. Object-oriented finite element programming: II A prototype program in Smalltalk. **Computer Methods in Applied Mechanics and Engineering**, vol. 98, p. 361-397, 1992.

DUBOIS-PÉLERIN, Y.; ZIMMERMANN, T. Object-oriented finite element programming: III. An efficient implementation in C++. **Computer Methods in Applied Mechanics and Engineering**, v 108, p. 165-183, 1993.

FISHER, K.A.; WRIGGERS, P. Frictionless 2D contact formulation for finite deformations based on the mortar method. **Computational Mechanics**, v. 36, p. 226-244, 2005.

FORDE, B.W.R.; FOSCHI, R.O.; STIEMER, S.F. Object-Oriented Finite Element Analysis. **Computer & Structures**, v. 34, n. 3, p. 355-374, 1990.

HEDELAL, O. **Object-Oriented Structuring of Finite Elements.** (Ph.D. dissertation). Aalborg University. Aalborg, Denmark, 1994.

HILL, R. **The Mathematical Theory of Plasticity.** Oxford University Press. London, 1950.

HOLZAPFEL, G.A. **Nonlinear Solid Mechanics: A continuum approach for Engineering.** John Wiley and Sons. Chichester, West Sussex, 2000.

HUGHES, T. J. R. **The Finite Element Method: Linear Static and Dynamic Finite Element Analysis.** Prentice-Hall, Inc. New Jersey, 2000.

HUNTER, J.D. Matplotlib: A 2D graphics environment. **Computing in Science & Engineering.** v 9, p. 90-95, 2007.

JOHNSON, K.L. **Contact Mechanics.** Cambridge University Press. Cambridge, 2003.

KHOEI, A.R.; NIKBAKHT, M. Contact friction modeling modeling with the extended finite element method (X-FEM). **Journal of Materials Processing Technology**, v. 177, p. 58-62, 2006.

KIKUCHI, N.; ODEN, J.T. **Contact problems in elasticity: A study of variational inequalities and finite element methods.** SIAM. Philadelphia, 1988.

KUUTII, J.; KOLARI, K. A local remeshing procedure to simulate crack propagation in quasi-brittle materials. **Engineering Computations: International Journal for Computer-Aided Engineering and Software**, v. 2, n. 2, p. 125-143, 2012.

LANGTANGEN, H. P. **Python scripting for computational science**, Third Edition. Springer-Verlag. Berlin, 2008.

LAURSEN, T.A. **Formulation and Treatment of Frictional Contact Problems Using Finite Elements.** (Ph.D. Dissertation). Department of Mechanical Engineering. Standford University, 1992.

LAURSEN, T. A. **Computational Contact and Impact Mechanics: Fundamentals of Modeling Interfacial Phenomena in Nonlinear Finite Element Analysis.** Springer-Verlag. Berlin, 2002.

LAYMAN, R.; MISSOUM, S.; GEEST, J. V. Simulation and probabilistic failure prediction of grafts for aortic aneurysm. **Engineering Computations: International Journal for Computer- Aided Engineering and Software**, v. 27, n. 1, p. 84-105, 2010.

LUENBERGER, D.G. **Linear and nonlinear programming.** 2nd edition. Springer. Berlin, 2005.

MACKERLE, J. (2000). Object-oriented techniques in FEM and BEM: A bibliography (1996-1999). **Finite Element in Analysis and Design**, v. 36, p. 189-196, 2000.

MACKIE, R.I. An object-oriented approach to calculation control in finite element programs. **Computer and Structures**, v. 77, p. 461-474, 2000.

MARSDEN, J.E.; HUGHES, T.J.R. **Mathematical foundations of Elasticity.** Dover. Mineola, 1994.

MELENK, J.M. **On the Generalized Finite Element Method.** (Ph.D. Thesis). University of Maryland. Maryland, 1995.

MELENK, J.M., BABUŠKA, I. The partition of unity finite element method: basic theory and applications. **Computer Methods in Applied Mechanics and Engineering**, v. 139. p. 289-314, 1996.

MOHAMMADI, S. **Extended finite element method for fracture analysis of structures**. Blackwell Publishing Ltd. Singapore, 2008.

NEWMARK, N.M. A method of computation for structural dynamics. **Journal of the Engineering Mechanics Division.** Proceedings of the American Society of Civil Engineers. p 67-95, 1959.

ODEN, J.T., MARTINS, J.A.C. Models and computational methods for dynamic friction phenomena. **Computer Methods in Applied Mechanics and Engineering**, v. 19, p. 701-725, 1985.

PEREIRA, J. P. A. **Extração de fatores de intensidade de tensão utilizando a solução do Método dos Elementos Finitos Generalizados.** (Dissertação de mestrado). Escola de Engenharia de São Carlos, Universidade de São Paulo. São Carlos, 2004.

PHADKE, M.V. **Application of the GFEM Method to solve Structural Mechanics Problems.** Department of Mechanical, Industrial and Nuclear Engineering of the College of Engineering, University of Cincinnati, 2005.

PIEDADE NETO, D. **Sobre estratégias de resolução numérica de problemas de contato.** Dissertação de Mestrado. Escola de Engenharia de São Carlos. Universidade de São Paulo. São Carlos, 2009.

PIEDADE NETO, D.; FERREIRA, M.D.C.; PROENÇA, S.P.B. An Object-Oriented class design for the Generalized Finite Element Method programming. **Latin American Journal of Solids and Structures**, v. 10, p. 1267-1291, 2013.

PIRES, E.B.; ODEN, J.T. Analysis of contact problems with friction under oscillating loads. **Computer Methods in Applied Mechanics and Engineering**, v.39. p. 337-362, 1983.

PROENÇA, S.P.B. **Análise não-linear de estruturas.** Notas de aula. Escola de Engenharia de São Carlos. Universidade de São Paulo. São Carlos, 2010.

ROSSUM, G. V. **The Python Library Reference** - Release 2.7.2, 2011. available at http://docs.python.org/download.html. (acessed July 2011), 2011.

SAVASSI, W. **Introdução ao método dos elementos finitos em análise linear de estruturas.** Reimpressão. Serviço de publicações da EESC. São Carlos, 2000.

SOUZA NETO, E.A.; PERIĆ, D.; OWEN, D.R.J. **Computational Methods for plasticity: Theory and applications**. John Wiley and Sons. Southern Gate, 2008.

SIMO, J.C.; HUGHES, T.J.R. **Computational Ineslasticity**. Springer. New York, 1998.

SPENCER, A.J.M. **Continuum Mechanics.** Dover Publication. New York, 2004.

STRANG, G. **Linear Algebra and Its Applications**. Academic Press Inc. Orlando, 1980.

THE SCIPY COMMUNITY. **NumPy Reference: NumPy v1.5 Manual (DRAFT).** available at http://docs.scipy.org/doc/numpy-1.5.x/reference/, (acessed 08 April 2011), 2010.

THE SCIPY COMMUNITY (2011). **SciPy: SciPy v0.9 Reference Guide (DRAFT).** available at http://numpy.scipy.org/, (acessed 08 April 2011), 2011.

TIMOSHENKO, S; GOODYEAR, J.N. **Theory of Elasticity.** McGraw-Hill. Singapore, 1970.

TIMOSHENKO, S.P.; GERE, J.M. **Theory of Elastic Stability.** 2nd Edition. McGraw-Hill International Book Company. Singapore, 1985.

TORRES, I.F.R. **Desenvolvimento e aplicações do Método dos Elementos Finitos Generalizados em análise tridimensional não-linear de sólidos.** (Tese de doutorado). Escola de Engenharia de São Carlos. Universidade de São Paulo. São Carlos, 2003.

TUCKER, A. B.; NOONAN, R. E. **Linguagens de programação: princípios e paradigmas.** Segunda Edição, McGraw Hill, 2008.

WARBURTON, G.B. **The dynamical behavior of structures.** Pergamon Press, 1976.

WILSON, E. L.; TAYLOR, R. L.; DOHERTY, W.P.; GHABOUSSI, T. Incompatible displacement models. **Proc. O. N. R. Conf. on Numerical Method.** Urbana, Illinois, 1971.

WRIGGERS, P. **Computational Contact Mechanics.** Springer. Berlin, 2006.

YANG, B.; LAURSEN, T.A.; MENG, X. Two dimensional mortar contact methods for large deformation frictional sliding. **International Jorunal for Numerical Methods in Engineering**, v. 62, p. 1183-1225, 2005.

ZIENKIEWICZ, O.C.; TAYLOR, R.L. **The finite element methods. Vol. I and II**. 5th edition. Butterworth-Heinemann. Oxford, 2000.

ZIMMERMANN, T.; DUBOIS-PÉLERIN, Y.; BOMME, P. Object-oriented finite element programming: I. Governing principles. **Computer Methods in Applied Mechanics and Engineering**, v. 98, p. 291-303, 1992.