UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Sede Amministrativa: Università degli Studi di Padova

Dipartimento di Ingegneria Industriale

SCUOLA DI DOTTORATO DI RICERCA IN: Scienze, Tecnologie e Misure Spaziali

INDIRIZZO: Scienze e Tecnologie per Applicazioni Satellitari e Aeronautiche

XXIX CICLO

ADAPTIVE GRID REFINEMENT AND SCALING TECHNIQUES APPLIED TO PERIDYNAMICS

Direttore della scuola: Ch.mo Prof. Giampiero Naletto

Coordinatore d'indirizzo: Ch.mo Prof. Giampiero Naletto

Supervisore: Ch.mo Prof. Ugo Galvanetto

Co-Supervisore: Ch.mo Prof. Mirco Zaccariotto

Dottorando: Daniele Dipasquale

To Doretta, Emanuele,

Angelo and everyone

who supported me

Summary

Peridynamics, a recently proposed non-local continuum theory, is particularly suitable to describe fracture phenomena in a wide range of materials. One of most common techniques for its numerical implementation is based on a mesh-free approach, in which the whole body is discretized with a uniform grid and a constant horizon, the latter related to the length-scale of the material and/or of the phenomenon analysed. As a consequence of that, computational resources may not be used efficiently. The present work proposes adaptive refinement/scaling algorithms for 2D and 3D peridynamic grids, to reduce the computational cost of peridynamic based software. Adaptive refinement/scaling is here applied to the study of dynamic crack propagation in brittle materials. Refinement is activated by using a new trigger concept based on the damage state of the material, coupled with the more traditional energy based trigger, already proposed in the literature. The use of a varying horizon and grid spacing over the grid may introduce some anomalies on the numerical peridynamic solution, such anomalies are investigated in detail through static and dynamic analyses. Moreover, while the scientific community is working to assess the full potential of peridynamics, few researchers have observed indirectly that the evolution of crack paths can follow, in an unphysical way, the axes of symmetry of the grid. The main parameter affecting such a numerical phenomenon seems to be the value of the m ratio, namely the ratio between the horizon and the grid spacing. The dependence of the crack path on the grid orientation would be a serious drawback for peridynamic based software since it would undermine what is believed to be one of its most important advantages over other computational methods, i.e. its capability to simulate (multiple) crack nucleation, propagation, branching and interaction in solids in a simple way. Finally, in order to show the effectiveness of the proposed approach, several examples of crack propagation in both 2D and 3D problems are presented. Then, the results obtained are compared with those obtained with other numerical methods and with experimental data.

Sommario

La Peridynamica, una teoria non locale del continuo proposta recentemente, è particolarmente adatta a descrivere fenomeni di frattura in una vasta gamma di materiali. Una delle tecniche più comuni per la sua implementazione numerica è basata su un approccio senza *mesh*, in cui l'intero corpo viene discretizzato con una griglia uniforme e un orizzonte costante, essendo quest'ultimo in relazione con la lunghezza di scala del materiale e/o del fenomeno analizzato. Di conseguenza le risorse computazionali possono non essere utilizzate in modo efficiente. Il presente lavoro si propone di sviluppare gli algoritmi per l'implementazione dell'*adaptive grid refinement and scaling* per griglie peridinamiche 2D e 3D, con lo scopo di ridurre il costo computazionale dei software basati sulla peridynamica. Questo approccio viene applicato allo studio della propagazione dinamica di cricche in materiali fragili. Il *refinement* viene attivato utilizzando un nuovo concetto di "innesco" che si basa sullo stato di danneggiamento del materiale, accoppiato con il più tradizionale innesco basato su un criterio energetico, già proposto in letteratura. L' utilizzo di un orizzonte e di un passo di griglia variabile può introdurre nella soluzione numerica della peridynamica alcune anomalie, che vengono analizzate dettagliatamente tramite analisi statiche e dinamiche. Inoltre, mentre la maggior parte della comunità scientifica sta lavorando per valutare a pieno le potenzialità della peridynamica, solo alcuni ricercatori hanno osservato indirettamente come il percorso della cricca possa seguire, in modo chiaramente non realistico, gli assi di simmetria della griglia. Il principale parametro che influisce su tale comportamento sembra essere il valore assunto dal rapporto *m*, definito come il rapporto tra l'orizzonte e il passo della griglia. La dipendenza del percorso della cricca dall'orientamento della griglia sarebbe un grave ostacolo per lo sviluppo di un software basato sulla peridynamica, poiché ciò porterebbe a pregiudicare quella che si ritiene essere uno dei suoi vantaggi più importanti rispetto ad altri metodi di calcolo, ossia la sua capacità di simulare la nucleazione (anche multipla), la propagazione, la ramificazione e l'interazione di cricche in materiali solidi in modo semplice. Successivamente, al fine di dimostrare l'efficacia del metodo proposto, vengono presentati alcuni esempi di propagazione di cricche per problemi 2D e 3D. Infine,

i risultati ottenuti sono confrontati con quelli ottenuti con altri metodi numerici e con dati sperimentali.

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor Prof. Ugo Galvanetto and co-supervisor Prof. Mirco Zaccariotto for their constant support during my research, stimulating me to work hard. It was a pleasure to work with them, giving useful contributions to my background knowledge and my life as well.

I am deeply grateful to Prof. Erkan Oterkus for giving me the possibility to stay in Glasgow, Scotland. I thank him as well as his wife Selda Oterkus for their kindness and availability.

Special thanks to Prof. Florin Bobaru for his useful pieces of advice during his stay here in Italy.

I would like to thank all my friends and colleagues for their support and for their useful discussions on the problems I had in my research, and especially for all the memorable moments we have shared together. I found new friends who are like brothers and sisters to me, such as Giulia, Siamak, Teo and Arman. In particular, I want to express my special gratitude to Giulia who always helps me with patience, even at this very moment. I want also to thank other friends and colleagues: Edoardo, Guido, Duzzo, Maz, Soheil, Chiodo and Curling.

I particularly want to thank Ilaria who has always supported me with her love and friendship. I wish the best for her life hoping that our relationship will always remain solid as until now.

A special thanks to my newest best friend and little sister Wilasinee, we had a great time together and hopefully other funny days will come soon for us.

Above all, I am deeply indebted to my mother, father and brother for their endless love, since they have regularly encouraged me to believe in myself. In addition, thanks to all my relatives.

I want to apologize if I missed acknowledging someone, it is only due to my lack of memory.

Publications

Articles

- Dipasquale D., Zaccariotto M. and Galvanetto U., *Crack propagation with adaptive grid refinement in 2D peridynamics*. Int. J Fract., pp. (22), 2014, DOI 10.1007/s10704-014-9970-4.
- Dipasquale D., Sarego G., Zaccariotto M., Galvanetto U., *Dependence of crack paths on the orientation of regular peridynamic grids*, Eng. Fract. Mech., Vol. 160, pp. 248-263, 2016, DOI 10.1016/ j.engfracmech.2016.03.022.

Proceedings

- Dipasquale D., Sarego G., Zaccariotto M., Galvanetto U., *Peridynamics with adaptive grid refinement*, Proceeding of the 11th World Congress on Computational Mechanics (WCCM XI), Spain, Vol. IV: pp. 4009-4020, 2014.
- Dipasquale D., Zaccariotto M., Sarego G., Duzzi M., Galvanetto U., *Peridynamics computations with variable grid size*, Proceeding of the 27th Nordic Seminar on Computational Mechanics (NSCM-27), Sweden, pp. (4), 2014.
- Zaccariotto M., Sarego G., Dipasquale D., Shojaei A., Mudric T., Duzzi M., Galvanetto U., *Discontinuos mechanical problems studied with a peridynamics-based approach*, Proceeding of the 23rd Conference of the Italian Association of Aeronautics and Astronautics (AIDAA2015), pp. (19), Turin, 2015.
- Zaccariotto M., Sarego G., Dipasquale D, Galvanetto U., *Strategies for fatigue damage modeling with peridynamics*, Proceeding of the 8th International Congress of Croatian Society of Mechanics, Opatija, Croatia, 2015.
- Dipasquale D., Oterkus E., Sarego G., Zaccariotto M., Galvanetto U., *Refinement and scaling effects on peridynamic numerical solutions*, Proceeding of the ASME 2016 International Mechanical Engineering Congress and Exposition, Phoenix, Arizona (USA), 11-17 November 2016.

Abstracts

- Galvanetto U., Zaccariotto M., Dipasquale D., Sarego G., Duzzi M., *Grid refinement in peridynamic computational applications*, Contribution of International CAE Conference, Pacengo del Garda, Verona, 2014.
- Dipasquale D., Sarego G., Shojaei A., Zaccariotto M., Galvanetto U., *Addresing grid sensitivity in Peridynamics: an adaptive refinement approach*, Abstract In: International Conference on Computational Modelling of Fracture and Failure, pp. 248-249, Rennes: CCSd (Centre pour la Communication Scientifique Directe), Cachan, France, 3-5 June 2015.
- Zaccariotto M., Sarego G., Dipasquale D., Galvanetto U., *Remarks on constitutive laws and influence functions used in the Peridynamic theory*, Abstract In: International Conference on Computational Modelling of Fracture and Failure, pp. 248-249, Rennes: CCSd (Centre pour la Communication Scientifique Directe), Cachan, France, 3-5 June 2015.
- Zaccariotto M., Sarego G., Dipasquale D., Galvanetto U., *Alternative thin plate formulation using a peridynamic approach*. Contribution In: SPB 2015 International Conference on Shells, Plates and Beams, pp. 55-56, Bologna, 2015.

Posters

- Duzzi M., Zaccariotto M., Dipasquale D., Galvanetto U., *A Concurrent Multiscale Model to Predict Crack Propagation in Nanocomposite Materials with the Peridynamic Theory*, Poster presented at International Nano-techItaly2014 Conference, Venezia, 2014.
- Dipasquale D., Zaccariotto M., Duzzi M., Sarego G., Galvanetto U., *Dynamic and static simulations with peridynamic approach using finite element analysis*. Poster presented at International CAE Conference, Pacengo del Garda, Verona, 2014.

Awards

- Author winner of 2015 A.I.D.A.A. "*Premio in memoria del Prof. Teodoro Merlini*" prize for best paper in XXIII AIDAA Conference in Turin, Italy, 17-19 November 2015.

Contents

List of figures

List of tables

1. Introduction

1.1. Motivation and objectives

Failure of engineering structures is a phenomenon which is often caused by a decrease in structure strength due to the presence of cracks. Cracks can be originated in different ways, such as localized damage caused by accidental loads or defects resulting from manufacturing fabrication. Such cracks may grow for different causes, stress-corrosion, thermal-stresses, fatigue, impacts and so on; depending on the type of material and of structure, the fracture can rapidly lead to the collapse of the whole structure. The study of quasi-static fracture in linear elastic materials is usually based on the work of Griffith [1]. The dynamics of crack propagation in brittle materials, referred as "dynamic" for the important role that inertial effects have on crack propagation, is still a challenge [2]. Several theories and numerical methods have been proposed and developed in the last century with the target to predict the dynamics of crack propagation. Existing methods within the classical theory of mechanics are based on partial differential equations, which suffer from the inherent limitation that spatial derivatives are not defined at discontinuities. Supplemental kinetic relations that dictate crack growth and ad-hoc damage criteria have to be adopted in order to overcome this limitation. The new non-local theory of continuum called "Peridynamics" [3] removes such a drawback by replacing in the equation of motion the spatial derivatives with an integral operator. Peridynamics is non-local because each material point of a body described with it does interact with all surrounding points within a spherical finite volume, the radius of which is called *horizon*. This internal length can be related to the length-scale of the material or phenomena under investigation [4]. By doing so, the presence of every type of discontinuities, i.e. cracks or voids, can be easily treated with no need of additional equations and criteria to decide if the crack should nucleate and in which direction should propagate. If on the one hand peridynamics is showing to be very suitable for dealing with dynamic crack propagation, on the other hand, its numerical implementation is computationally expensive, especially for cases of large-scale 3D simulations. An interesting technique referred as "Adaptive Grid Refinement and Scaling" (AGRS), seems to be a promising candidate to increase its computational efficiency by reducing both the grid spacing of the grid and the

horizon in the region where the crack should propagate. In particular, with the term “refinement” we mean the reduction of the grid spacing, while the term “scaling” means to change the size of the horizon. Therefore, we developed the algorithms to implement the AGRS with the following objectives:

- Developing a robust algorithm by the introduction of an additional trigger for the AGRS with respect to that energy based proposed in the literature. The new trigger is damage based and allows to manage in a more efficient way the AGRS activation in the considered domain.
- Investigating the anomalies in the numerical peridynamic solutions introduced by the use of a varying grid spacing and horizon length over the domain. This is done by performing static and dynamic analyses.
- Investigating the possible dependence of crack propagation on grid orientation introduced by the type of discretization currently adopted by the peridynamic community.
- Applying to 2D/3D problems of dynamic crack propagation in brittle materials, the AGRS algorithm autonomously activates the grid refinement at the crack tip during crack propagation and in the regions where crack nucleation is likely.

1.2. Organization of the Thesis

This thesis is organized in seven chapters and one appendix. After the present first chapter, which gives a brief introduction to fracture mechanics and motivation for this research, in chapter two an overview of peridynamic fundamentals is given. In chapter three we introduce the working principle of the adaptive grid refinement and scaling algorithm applied to peridynamic theory. The use of a varying grid spacing and horizon length leads to introduce some anomalies in the numerical peridynamic solution: hence in chapter four its effects on peridynamic solutions are evaluated by carrying out specific static and dynamic analyses followed by a discussion of the results. Further, in chapter five the issue regarding the dependence of crack propagation on grid orientation is addressed, which would be a serious drawback for the numerical peridynamic solution. Finally, for verifying the effectiveness of the proposed technique, in chapter six several benchmark problems are treated and compared with the results

given in the literature, followed by a summary of the main conclusions derived from this work in chapter seven. Appendix provides extensions and technical clarifications of the work presented in this dissertation.

1.3. Fracture mechanics

1.3.1. Brief overview of historical perspective

Looking back through the history of the engineering structures, it is easy to realize how many injuries and financial loss have been caused by fracture failure. The Industrial Revolution led to change the design of the structures since iron and steel materials allowed to remove the earlier restrictions on design. In fact, unlike the brick and mortar adopted in pre-Industrial Revolution architectures, which work mainly in compression, such relatively ductile materials can carry tensile stresses. One of the most famous failures caused by the unexpected fracture failure regards the "Boston Molasses Disaster" occurred on January 15, 1919, in which the rupture of a molasses tank took a wave of molasses to rush through the streets killing and injuring many people (see Fig. 1.1). Although the designers adopted safety factors of 10 or more with respect to the tensile strength of the material, the structure collapsed. An explanation of the cause that led to that disaster can be given taking into account of the qualitative studies performed by Leonardo da Vinci several centuries earlier, who pointed out that the strength of a wire made of iron varies inversely with its length. This implies that a longer wire has a higher probability of containing flaws with respect of a shorter one. A quantitative study on the relationship between flaws and fracture stresses was first performed by Griffith in 1920 [1], showing through a stress analysis of an elliptical hole that the fracture failure can occur as a consequence of the unstable growth of the flaw. He invoked the first principle of thermodynamics to formulate a fracture theory which is based on an energy balance, unfortunately, this model can capture the relationship between the fracture strength and flaws of ideally brittle materials only. Only later in 1948, Irwin (leader of the fracture mechanics research group of Naval Research Laboratory in Washington, DC) gave the first significant extension of Griffith's model to the fracture of metals.

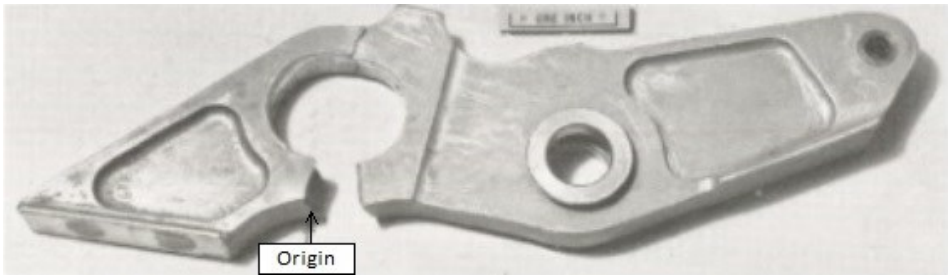
Fig. 1.1: Photograph of the aftermath of "Boston Molasses Disaster" (Photograph taken from Wikipedia).

In more recent years, the "Challenger Space Shuttle disaster" occurred on January 28, 1986, in which after 73 seconds of the take-off, Shuttle exploded killing all crew members. The embrittlement of the O-ring seal in one of the solid rocket boosters caused the disaster, since it was not designed to fly under unusually cold weather. In particular, due to the low temperature the O-ring seal stiffened, leading to the fracture of the rocket joint in consequence of the external walls expansion of its chamber. Its fracture failure caused a breach in the booster joint, taking the pressurized burning gas generated by the solid fuel outside and then leading to damage the external tank of liquid fuel. Although the engineers at NASA advised their managers to postpone the date of the launch, the available data was not enough to support their position. Almost 17 years after the "Challenger Space Shuttle disaster", the Space Shuttle Columbia burst during the re-entry phase in consequence of the impact of a piece of foam insulation which struck the left wing of Shuttle. This incident caused a fracture damage of the thermal wing insulation making the orbiter vulnerable for the re-entry in which high temperature can be reached (1650°C).

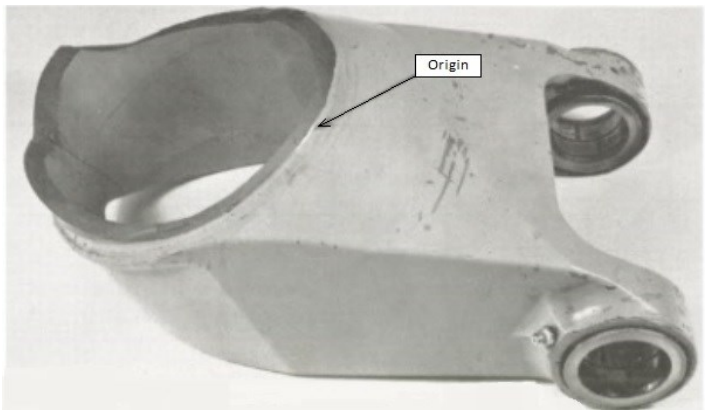
Several reasons may lead to a catastrophic fracture failure of an engineering structure. In the field of aeronautical and aerospace structures, the fatigue mechanism has been recognised as the most powerful threat to the reliability and safety of the aircraft. Fatigue is the weakening of the material caused by the action of loads applied repeatedly, if the loads are above a certain threshold, microscopic cracks may begin to form in the areas of stress concentration.

Eventually, if the cracks reache a critical size, they will propagate dynamically leading to the collapse of the structure.

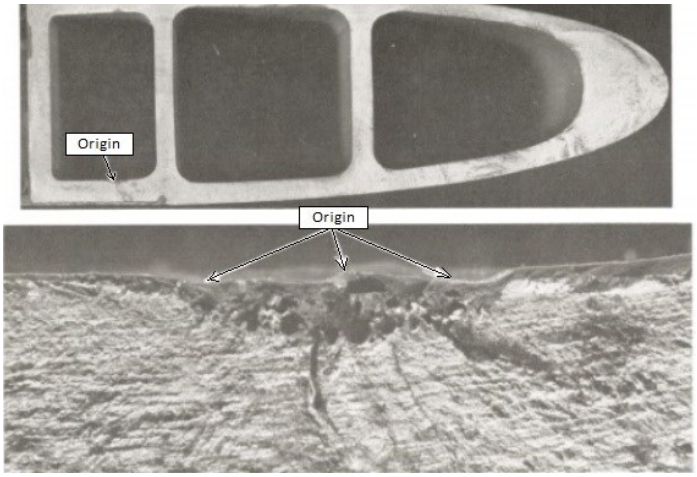
a)



b)



c)



d)

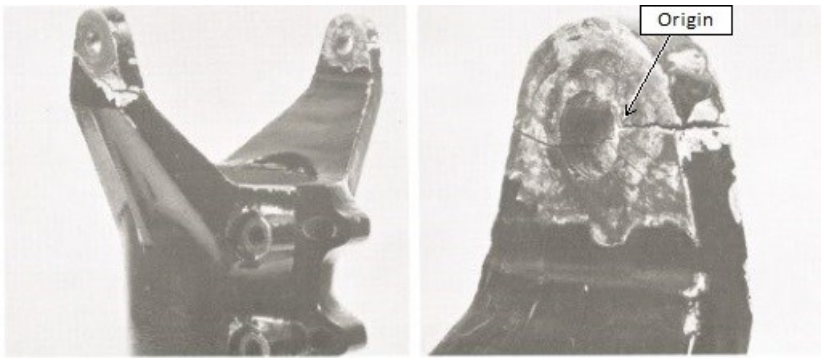


Fig. 1.2: Photographs of in-service failure components of aircraft, a) landing gear door up-lock operation lever, b) pivot bracket, c) helicopter rotor blade extrusion, d) helicopter rotor drive yoke (Images taken from [5]).

In the last decades, several procedures and methods have been developed for the design, testing and routine inspections in order to prevent failure, especially when cracks are found in the primary structure of the airframe. In any case, nowadays the routine inspections made during the aircraft service remain the main procedure for detecting the presence of cracks and keeping their growth under control. Fig. 1.2 shows few photographs of some components failed in service, due to the fatigue phenomena, which have caused the complete loss of an aircraft or serious damage to the main structure. The photographs clearly show that the damage is always originated from areas characterized by stress concentrations, like holes or rivet holes, bolts, cut-outs, and from flaws in the material and imperfections of welded joints.

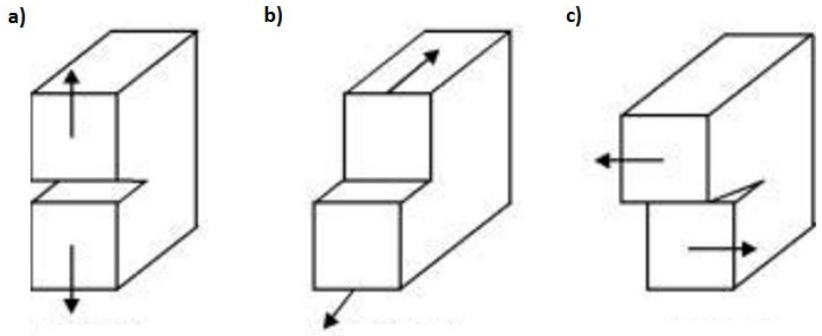


Fig. 1.3: The three fracture modes, a) Opening mode (Mode 1), b) Sliding mode (Mode 2), c) Tearing mode (Mode 3).

Basically, as shown in Fig. 1.3, we can identify three fracture modes depending on the way of applying the force at the crack: Opening mode (Mode 1) in which

the load is applied normal to the crack plane, Sliding mode (Mode 2) in which an in-plane shear load leads one crack face to slide with respect to the other one and Tearing mode (Mode 3) in which shear load acting parallel to the plane of the crack and parallel to the crack front. Fracture mechanics can be divided in three main families (see Fig. 1.4):

1. Linear Elastic Fracture Mechanics (LEFM)
2. elastic-plastic fracture mechanics
3. dynamic, viscoelastic and viscoplastic fracture mechanics.

While the LEFM is mainly applicable to linear elastic materials under quasi-static conditions (even if recent advances in research allowed to incorporate the behaviour of other materials), the elastic-plastic theory also includes plastic deformation (non-linear behaviour). Both are time-independent since time variable is not included. While dynamic, viscoelastic and viscoplastic fracture mechanics include time as a variable. The dashed line in Fig. 1.4 indicates the connection between the LEFM and the Dynamic Fracture Mechanics (DFM), in particular, the dynamic version of LEFM is termed "elastodynamic fracture mechanics", as it will be discussed in Section 1.3.3.

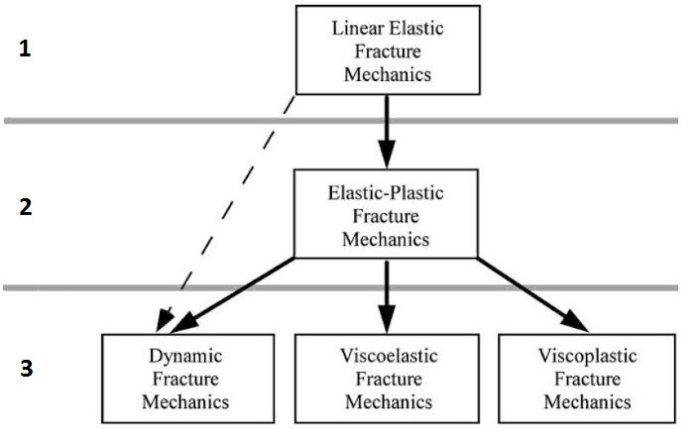


Fig. 1.4: Main families of fracture mechanics (Image taken from [6]).

1.3.2. Linear Elastic Fracture Mechanics

The effort made by the pioneers Inglis [7] and Griffith [1] paved the way for the fundamentals of fracture mechanics through the introduction of LEFM. Although LEFM is mainly applicable to materials that obey Hooke's law, its

formulation can be extended to deal with crack propagation characterized by a small-scale plasticity deformation at the crack tip of structures with a global linear elastic behaviour. This theory introduced the concepts of energy release rate and stress intensity parameters. According to the first law of thermodynamic, a crack under a static load condition propagates if the energy available for crack growth is sufficient to overcome the resistance of the material. In particular, the critical stress (fracture stress) is defined at the condition with no net change in total energy. Let us consider a case regarding the fracture mechanics of Mode 1, an infinitely wide plate of thickness B with a crack of length $2a$ and subjected to a remote uniform tensile stress σ (see Fig. 1.5).

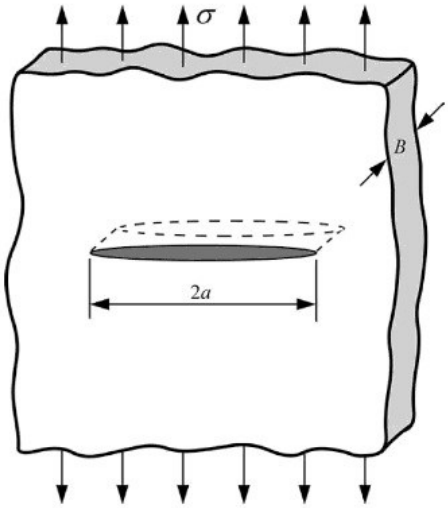


Fig. 1.5: Infinitely wide plate with a crack subjected to a remote tensile stress (Image taken from [6]).

The incremental increase in the crack area dA, under the condition of equilibrium, can be expressed as:

$$\frac{dE}{dA} = \frac{d\Pi}{dA} + \frac{dW_s}{dA} = 0 \tag{1.1}$$

where E is the *total energy*, Π is the *potential energy* supplied by the external forces and the internal strain energy and W_s the *work* required to create new surfaces. Regarding the particular case shown in Fig. 1.5, the potential energy can be expressed as:

$$\Pi - \Pi_0 = -\frac{\pi\sigma^2 a^2 B}{E} \tag{1.2}$$

with Π_0 the potential energy of an uncracked plate. The work required for the creation of two new surfaces is given by $W_s = 4aB\gamma_s$, where γ_s is the *surface energy* of the material, thus:

$$\frac{d\Pi}{dA} = -\frac{\pi\sigma^2 a}{E} \tag{1.3}$$

and

$$\frac{dW_s}{dA} = 2\gamma_s \tag{1.4}$$

Therefore, substituting Eq. (1.3) and Eq. (1.4) in Eq. (1.1) and solving for the stress gives:

$$\sigma_f = \left(\frac{2E\gamma_s}{\pi a}\right)^{1/2} \tag{1.5}$$

with σ_f defined *fracture stress*, then a tensile $\sigma \geq \sigma_f$ will take the crack to propagate. Irwin's approach, which is more convenient for solving engineering problems, defines the concept of *energy release rate G* as a measure of the rate of change in potential energy with the crack area, even referred as the *crack driving force*. With reference to the specific case of Fig. 1.5, *G* is defined as follows:

$$-\frac{d\Pi}{dA} = \frac{\pi\sigma^2 a}{E} \tag{1.6}$$

Hence, the propagation of the crack occurs if the energy release rate reaches the critical value G_c defined as:

$$G_c = \frac{dW_s}{dA} = 2\gamma_s \tag{1.7}$$

which it is a measure of the *fracture toughness* of the material.

1.3.3. Elastodynamic Fracture Mechanics

The dynamic version of LEFM is termed *elastodynamic fracture mechanics*, it considers the inertia forces and the reflected stress waves neglecting the nonlinear response of the material. Failure criteria for dynamic problems of brittle materials can be considered as an extension of Griffith's ideas in which, as suggested by Mott [8], the kinetic energy must be incorporated in the energy balance. Let us consider a crack in an unbounded 2D body which is propagating at constant speed *v* (see Fig. 1.6), namely steady-state crack growth.

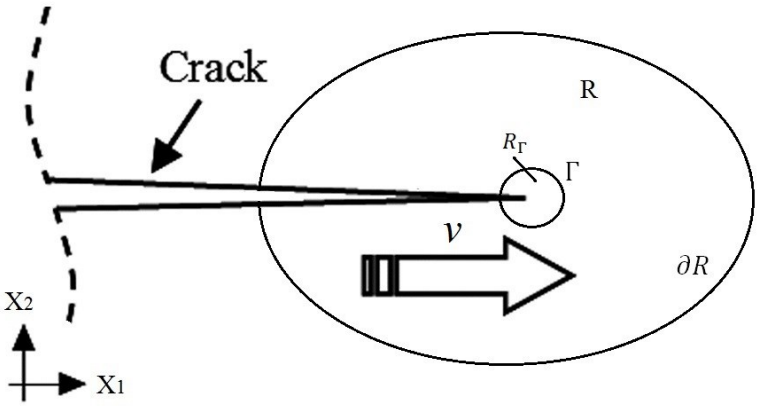


Fig. 1.6: Crack tip contour for evaluating the energy balance of a crack which is propagating at constant velocity *v*.

Let $\dot{U}$ define the rate of change of the stored elastic energy and $\dot{K}$ the rate of change in the kinetic energy, then:

$$\dot{U} + \dot{T} = \int_{R-\mathrm{R}_\Gamma} \frac{1}{2}\left[\sigma_{\alpha\beta}\varepsilon_{\alpha\beta} + \rho\ddot{u}_\alpha\ddot{u}_\beta\right]dA \tag{1.8}$$

with $\alpha = 1{,}2$, $\beta = 1{,}2$ indicate the directions of a Cartesian reference system (X_1, X_2), u the displacement component, σ the stress component, ε the strain component, dA the infinitesimal area. R is the region near the crack tip bounded by the curve ∂R and R_Γ the small region enclosed by the contour Γ. The small region R_Γ excludes the singularity of the stress and strain fields at the crack tip. Let us consider that the contour Γ translates with the same speed of crack propagation; invoking Reynolds transport theorem of Eq. (1.8) to the limit $\Gamma \to 0$:

$$\dot{U} + \dot{T} = \lim_{\Gamma \to 0} \int_{R - R_\Gamma} \left[\sigma_{\alpha\beta} \dot{u}_{\alpha,\beta} + \rho \dot{u}_\alpha \ddot{u}_\alpha\right] dA + \lim_{\Gamma \to 0} \int_\Gamma \frac{1}{2} \left[\rho \dot{u}_\alpha \dot{u}_\alpha + \sigma_{\alpha\beta} u_{\alpha,\beta}\right] v n_1 ds \tag{1.9}$$

with n_1 the component of the unit vector outward normal to the contour Γ aligned in the direction of crack propagation and *ds* the infinitesimal length. The subscripts *α*, *β* indicate the partial differentiation of the *α* component with respect to the coordinate identified by the index *β*. The application of divergence theorem to the first integral of Eq. (1.9) yields:

$$\lim_{\Gamma \to 0} \int_{R - R_\Gamma} \left[\sigma_{\alpha\beta} \dot{u}_{\alpha,\beta} + \rho \dot{u}_\alpha \ddot{u}_\alpha\right] dA = \int_{\partial R} \sigma_{\alpha\beta} \dot{u}_\alpha n_\alpha ds + \lim_{\Gamma \to 0} \int_\Gamma \sigma_{\alpha\beta} \dot{u}_\alpha n_\alpha ds \tag{1.10}$$

Being the newly created crack surfaces traction free, the first term of Eq. (1.10) represents the power *P* of the external forces, then:

$$P - \dot{U} + \dot{T} = \lim_{\Gamma \to 0} \int_\Gamma \left[\sigma_{\alpha\beta} \dot{u}_\alpha n_\alpha + \frac{1}{2} \left(\sigma_{\alpha\beta} \varepsilon_{\alpha\beta} + \rho \dot{u}_\alpha \dot{u}_\beta\right) v n_1\right] ds \tag{1.11}$$

The integral of Eq. (1.11) is defined as *energy flux integral* at the crack tip and represents the amount of energy flowing out of the region R and into the crack tip through the contour Γ. Then, the energy flux integral is defined as:

$$F = \int_\Gamma \left[\sigma_{\alpha\beta} \dot{u}_\alpha n_\alpha + \frac{1}{2} \left(\sigma_{\alpha\beta} \varepsilon_{\alpha\beta} + \rho \dot{u}_\alpha \dot{u}_\beta\right) v n_1\right] ds \tag{1.12}$$

Defining γ as the fracture energy per unit extension of the crack, and D as the total dissipation of energy in the fracture process, then the rate of change of D is defined as:

$$\frac{dD}{dt} = v \frac{dD}{da} \tag{1.13}$$

with *a* the crack length, hence in analogy with the LEFM we can define the dynamic energy release *G* as follows:

$$G = v^{-1}F = \gamma \tag{1.14}$$

Note that fracture energy γ can include the surface energy γ_s introduced by Griffith (see Section 1.3.2) as well as the dissipated energy due to all the dissipative phenomena which can be observed in dynamic crack propagation. Eq. (1.14) highlights that the dynamic release energy rate is the energy released per unit area of crack extension into the region around the crack tip and has to be equal to the dissipated energy. As opposed to quasi-static fracture in which the physical quantity G_c finds application as fracture criterion for defining the condition of crack initiation of the crack growth [6], G can provide the crack growth law in dynamic problems.

1.4. Theories for treating dynamic fracture

1.4.1. Methods based on classic mechanics

The primary assumption of the classic theory of mechanics is the locality of the interaction between material points since the interaction is restricted to neighbours. Therefore, the stress tensor $[\boldsymbol{\sigma}]$ at any material point of the body $\boldsymbol{x}$ depends on the deformation tensor $[\boldsymbol{\varepsilon}]$ at that point only. Classical mechanics assumes the hypothesis that a body has to remain continuous even after its deformation, hence the equation of motion at any material point is defined by means of spatial partial differential equations, as follows:

$$\rho\ddot{u}_k = \frac{\partial\sigma_{kX}}{\partial X} + \frac{\partial\sigma_{kY}}{\partial Y} + \frac{\partial\sigma_{kZ}}{\partial Z} + b_k \tag{1.15}$$

where $k = (X, Y, Z)$ indicates the directions of the cartesian reference system, ρ the mass density and $\ddot{u}_k$ the kth-component of the acceleration. It is known that the spatial partial differential terms in Eq. (1.15) are not defined at discontinuities, hence the governing equations of classic theory lose their meaning due to the presence of discontinuities, such as cracks or voids. The *Finite Element Method* (FEM) is one of the most robust numerical methods based on classic theory, it allows to model complex geometries calculating stress fields due to the application of general boundary conditions with a good level of approximation. However, when a problem with cracks has to be modelled, FEM requires the use

of special types of elements to capture correctly the singularities at the crack tip; redefining the body when the crack propagates since the crack surface is treated as a new boundary. In addition to this shortcoming, supplementary kinetic equations have to be employed in order to decide the evolution of the crack growth, which are concerned mainly with the propagation of existing cracks rather than their nucleation. During the last century, numerous studies have been performed aiming to overcome the shortcomings of the FEM, as a first study, the concept of cohesive zone introduced by Dugdale [9] and Barenblatt [10] paved the way to develop the *Cohesive Zone Elements* (CZE). Basically, methods based on CZE [11, 12, 13] adopt interface elements placed along the elements boundaries and equipped with a traction-separation law for which the tractions are zero when the relative displacements overcome a critical value. Usually, such elements are inserted along the potential crack path, in addition, the crack growth is sensitive to the texture and alignment of the regular elements of the mesh. A remeshing technique was proposed by Camacho and Ortiz [14] to address problems in which the crack paths are unknown a priori, but it still suffers from a certain dependence of crack propagation direction on local mesh orientation, since the crack is not completely free to propagate in the medium [15]. In order to remove these limitations, the *eXtended Finite Element Method* (XFEM) [16] was introduced to allow the crack to propagate within the regular elements of the mesh, in this way the crack can grow in any direction regardless of the texture of the mesh. The basic idea originally introduced in [17] is based on the use of the partition of unity finite elements, using an enrichment of discontinuous displacement functions to add degrees of freedom to the standard finite element approximation. Nevertheless, 'the formulation becomes cumbersome with increasing number of cracks and crack branches' [18]. Moreover, XFEM suffers (in practice) from the absence of reliable crack branching criteria [18]. Reference [19] proposed an approximation scheme for variational models of Griffith's theory [20] based on the notion of eigendeformation which are extensively used in mechanics as approach to describe the deformation modes that cost no local energy. Basically, such an approach assumes two fields, the displacement field and an eigendeformation field that describes discontinuities, such as cracks, which may be present in the body. By doing so, the displacement field can develop jumps at no cost in local elastic energy. Based on the work proposed in [19], the

eigenerosion approach [21, 22] has been originally developed for dealing with brittle fracture. The main idea is to restrict the eingendeformations in a binary sense: they assume a value equal to zero if the local behavior is elastic or a value equal to the local displacement gradient, in which case the corresponding material neighborhood is eroded. This scheme can be combined with different methods. For instance, combining it with the FEM, depending on the value assumed by the eingendeformation, the regular elements of the mesh can have the usual elastic behavior or be eroded, in the sense that they do not have load bearing capacity anymore. Recently, alternative methods for the numerical simulation of fracture problems have emerged such as the *phase-field method* [23, 24, 25]. Whereas other methods require the use of complex algorithms to track the fracture surfaces since discontinuities are introduced into the solid, the phase-field method treats the discontinuities approximating the fracture surface by a phase-field, smoothing the boundary of the crack over a small region. In this way, the evolution of the phase field is used to capture the trajectory of the crack. However, this new approach treats mainly brittle fracture, in addition, it shows to be sensitive to the phase field parameters, hence they have to be chosen very carefully.

1.4.2. Nonlocal Theories and Methods

Let us consider that the understanding and prediction of material failure process is governed by the presence of a wide variety of mechanics associated with dislocations, micro-flaws, grain boundaries which play an important role at specific length-scales. Methods based on classic local theory of mechanics suffer from the lack of an internal length parameter. The difficulties encountered in such methods utilizing classical continuum mechanics can be overcome by adopting numerical methods based on non-local approaches like the atomistic model [26], describing fracture as a rupture of inter-atomic forces. They are certainly able to capture in a realistic way the mechanisms which occur in fracture phenomena, giving important contribution to the understanding of the physical processes that lead to the nucleation and propagation of cracks. On the other side, because of the large computational resources that such methods require, their application is limited to the qualitative understanding of fracture rather than to the prediction of it. As an example, the largest-scale atomistic simulation performed in [27] used

320 billion atoms to model a copper cube with an edge length of 1.56 μm, not enough for dealing with real-life problems. This drawback can be overcome by coupling atomistic models with other methods computationally more efficient such as FEM. The coupling has to make use of multi-scale approaches [28], which need rather complex strategies for a correct transfer of the forces between models of different scale. Besides, the coupling between local and non-local models often generates the artifact called *spurious forces or ghost forces* [29] that strongly affect equilibrium in the transition zone between the two different models. The connection between the classic theory of continuum and the atomistic models is established by the nonlocal theories of the continuum. Unlike the classic theory of continuum, in the nonlocal theories the state of material points is influenced by material point enclosed inside a region of finite length. Therefore, the stress at a given point does not only depend on that local strain but also on the strain of the points enclosed in its neighbourhood [18]. [30] suggests several reasons why to use a nonlocal theory rather than a local one, i.e. capturing the effects of microstructural heterogeneities or the influence of microcracking growth in the proximity of the crack tip on crack propagation. Several different types of non-local theories have been proposed in the past, as the *nonlocal elasticity theory* introduced by Eringen and Edelen [31] which defines the equation of motion by means of a nonlocal balance of thermodynamic statements. Another type is the *gradient enhanced models* [32], referred as weakly nonlocal models since based on differential equations, it assumes higher order spatial derivatives in the formulation allowing the material point to account for the field in its vicinity. However, all these nonlocal theories suffer from the difficulty to deal with discontinuities in the domain since are based on spatial derivatives. Another type of nonlocal theory that allowed to circumvent this difficulty has been introduced by Kunin [33, 34] and Rogula [35] using displacement fields rather than their derivatives. Some shortcomings of this theory can be identified in the lack of a formulation to address 2D/3D models, damage included in the material response and a nonlinear material response with respect to displacements.

1.4.3. Peridynamics

Recently, in order to overcome the limitations related to the differential formulation of methods commonly based on classic theory of continuum as well on the nonlocal theories developed in the past, the nonlocal *PeriDynamic* theory (PD) has been proposed by Silling [3] and Silling et al. [36]. PD is an alternative nonlocal formulation of continuum mechanics which defines the equation of motion by substituting the divergence of the stress tensor, that is the partial differential term of Eq. (1.15), with an integral operator. In this way, the equation of motion does not suffer from the presence of discontinuities over the domain. The main concept introduced by Silling is based on the assumption that material points can interact with each other if located inside a region of finite length. That is the reason why the Greek roots of the words *near* and *force* were proposed to call such a formulation "peridynamics". The interaction between material points is defined *bond*, which contains all of the constitutive law for defining the response of the material. In addition, the damage model is included in the constitutive law so that an unambiguous definition of local damage at a material point of the continuum domain can be defined. Then, the peridynamic governing equations can be defined at crack surfaces without the need to employ special techniques such as the use of interface elements and enrichment discontinuous functions in order to treat the fracture. These attributes make peridynamics suitable to deal with complex problems characterized by simultaneous multi-crack propagations and interactions by simple integration of equation of motion. Another important attribute of peridynamics is the length scale referred as horizon δ introduced by its formulation. The horizon defines the size of the region of nonlocal interaction and can be linked to the characteristic length-scale of the material and/or of the considered phenomenon [4, 37]. Then it is possible to capture the material response on different length scales, from the macroscale to the nanoscale, since the horizon can be controlled.

Two main formulations for peridynamic models have been introduced: the first which is the original form of peridynamic model is termed Bond-Based PeriDynamic (BB-PD) [3], while the second one is termed State-Based PeriDynamic (SB-PD) [36]. BB-PD assumes that any pair of particles enclosed

inside the nonlocal region interact only through a central force potential [38], namely the interaction between a couple of nodes is completely independent of all the others. As already demonstrated in [38], under this assumption, Poisson's ratio is constrained to assume the value of 1/4 for 3D and 2D plane strain cases and 1/3 for 2D plane stress cases for isotropic materials. Also, it cannot distinguish the volumetric deformation from the distortional one, therefore it is not suitable to capture the response of materials with different mechanical properties, i.e. materials with plastic incompressibility response. Reference [39] proposed a strategy called *micropolar peridynamic model* to remove the limitation of Poisson's ratio, but it is not clear if such an approach can capture the plastic behaviour. Therefore, in order to remove such a drawback Silling et. al. [36] introduced the new general formulation referred as State-Based-PD, defining the concept of "state" as infinite dimensional arrays containing information regarding the bonds. In particular, there are two different formulations of SB-PD: the ordinary SB-PD and the non-ordinary SB-PD. In ordinary SB-PD the forces between two material points act along the vector connecting the points in the deformed configuration, while in non-ordinary SB-PD the forces can act in different directions with respect to that of the vector connecting the points. In spite of the notable advantages introduced by PD, some drawbacks arise out of the integro-differential peridynamic formulation even in the SB versions. First, the application of the boundary conditions is completely different from that of the classical theory, since point and surface boundary conditions have to be applied over a volume [40]. In addition, due to the nonlocality of the theory, the material stiffness near the free surfaces and the interfaces between different material regions can be over- or under-estimated, leading to an error in the PD solution. Other shortcomings of PD are related to its numerical discretization, as presented in the following paragraph.

The PD solution can be obtained through the implementation of different numerical methods [41], however, one of the most popular methods for implementing PD theory is based on a meshfree approach [42], in which the whole body is discretized with a uniform structured grid of nodes and a constant horizon. A wide variety of problems can be solved with a peridynamic approach for the description of material failure and damage: for predicting crack

propagation and branching in brittle materials [43, 44, 45], damage in concrete [46, 47], fracture in composite materials [48, 49, 50], fatigue damage [51, 52], corrosion damage and stress corrosion cracking [53, 54], among others. Although PD shows to capture damage and fracture phenomena successfully, its application to real-life structures still requires a considerable amount of computational resources, especially if it is compared with methods such as FEM. Numerous approaches have been proposed with the goal to restrict the use of peridynamics to regions where crack should take place, while adopting models based on classic theory in domains characterized by smooth displacement fields, for instance [55, 56] have proposed the coupling between PD models and FEM. Whereas staying within the PD framework, the authors in [57, 58] proposed the use of the technique based on adaptive refinement/coarsening and scaling as a strategy to reduce both sizes of grid spacing and horizon length in the proximity of the crack tip. Only the strategy based on adaptive refinement and scaling has been applied in [59, 60] to capture the dynamic crack propagation in brittle materials. The development of such a strategy could allow in the future to pave the way for an efficient concurrent multi-scale modelling in the framework of PD. In fact, by this strategy, it will be possible to reduce the horizon length to the wanted length-scale near the crack tip, by adopting a size convenient for the computational efficiency far from the crack tip.

2. Overview of peridynamics

2.1. Fundamentals

Peridynamics is a recently proposed non-local theory of continuum [3, 36, 61] which assumes that each material point of a given body $\mathcal{B}$ can interact with all surrounding points within a finite distance δ named *horizon*. The equation of motion at any instant in time t for a material point of coordinate $\boldsymbol{x}_i$ (referred as *source node*) is defined by means of the following integro-differential expression:

$$\rho(\boldsymbol{x}_i)\ddot{\mathbf{u}}(\boldsymbol{x}_i,t) = \int_{H(\boldsymbol{x})} \left\{\underline{\mathbf{T}}[\boldsymbol{x}_i,t]\langle \boldsymbol{x}_j - \boldsymbol{x}_i\rangle - \underline{\mathbf{T}}[\boldsymbol{x}_j,t]\langle \boldsymbol{x}_i - \boldsymbol{x}_j\rangle\right\} dV_{\boldsymbol{x}_j} + \boldsymbol{b}(\boldsymbol{x}_i,t), \quad \forall \boldsymbol{x}_j \in H(\boldsymbol{x}_i) \tag{2.1}$$

with $\rho(\boldsymbol{x}_i)$ the *mass density* field at the initial configuration of the body, $\boldsymbol{u}$ the *displacement* field, $\underline{\boldsymbol{T}}[\boldsymbol{x}_i,t]\langle \boldsymbol{x}_j - \boldsymbol{x}_i\rangle$ the *force vector state* field which represents the force density vector that a point $\boldsymbol{x}_j$ (referred as *family node*) exerts on point $\boldsymbol{x}_i$, $\boldsymbol{b}$ the *body density force* field and the integration domain $H(\boldsymbol{x}_j) \in \mathcal{B}$ referred as neighborhood of the point $\boldsymbol{x}_i$ which is defined as:

$$H(\boldsymbol{x}_i) := \{\boldsymbol{x}_j \in \mathbb{R}^D : \|\boldsymbol{x}_j - \boldsymbol{x}_i\| \leq \delta\} \tag{2.2}$$

with D = 1, 2, 3 the dimension of the treated problem. Therefore, the neighbourhood can have the geometric shape of a line segment, a disk or a sphere of radius/length δ centred at point $\boldsymbol{x}_i$. The interaction between any $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$point is called *bond*. Let $\boldsymbol{\xi} = \boldsymbol{x}_j - \boldsymbol{x}_i$ define the notion of relative position vector in the reference configuration of a bond connected to the point $\boldsymbol{x}_j$, then the vector state is an object which allows mapping vectors (bonds) into vectors and a scalar state to map vectors (bonds) into scalars; the angle brackets $\langle \cdot \rangle$ enclose the bond vector. As an example, consider the case of vector state, let $\boldsymbol{t}_{i,j} = \underline{\boldsymbol{T}}[\boldsymbol{x}_i,t]\langle \boldsymbol{x}_j - \boldsymbol{x}_i\rangle$ define the force density vector acting on $\boldsymbol{x}_i$ as a consequence of all the collective deformations of the bonds enclosed in its neighbourhood $H(\boldsymbol{x}_i)$, and $\boldsymbol{t}_{j,i} = \underline{\boldsymbol{T}}[\boldsymbol{x}_j,t]\langle \boldsymbol{x}_i - \boldsymbol{x}_j\rangle$ the force density vector acting on $\boldsymbol{x}_j$ as a consequence of all the collective deformations of the bonds enclosed in its neighbourhood $H(\boldsymbol{x}_j)$. By

using the concept of vector state, the force density can be stored in the vectors states (see Eq. (2.1)) as follows:

$$\underline{\boldsymbol{T}}[\boldsymbol{x}_i, t] = \begin{Bmatrix} \vdots \\ \boldsymbol{t}_{i,j} \\ \vdots \end{Bmatrix} \quad \text{and} \quad \underline{\boldsymbol{T}}[\boldsymbol{x}_j, t] = \begin{Bmatrix} \vdots \\ \boldsymbol{t}_{j,i} \\ \vdots \end{Bmatrix} \tag{2.3}$$

If we express with $\boldsymbol{y}(\boldsymbol{x}_i, t) = \boldsymbol{x}_i + \boldsymbol{u}(\boldsymbol{x}_i, t)$ the current position at time t of the material point $\boldsymbol{x}_i$, the *deformation vector state* is defined as:

$$\underline{\boldsymbol{Y}}[\boldsymbol{x}_i, t]\langle\boldsymbol{\xi}\rangle = \boldsymbol{y}(\boldsymbol{x}_j, t) - \boldsymbol{y}(\boldsymbol{x}_i, t) \tag{2.4}$$

which maps each bond connected with the material point $\boldsymbol{x}_i$ into its deformed image.

The following hypotheses are assumed:

1. the force state depends only on the deformation state (the material is referred as simple)
2. the material is ordinary, which means that the force density vector is parallel to the direction of the deformed bond (see Fig. 2.1a)
3. the material is homogeneous.

Under these hypotheses, peridynamics is referred as Ordinary State-Based PeriDynamic (OSB-PD), hence the force vector state takes the form:

$$\underline{\boldsymbol{T}}[\boldsymbol{x}_i, t]\langle\boldsymbol{\xi}\rangle = \underline{t}[\boldsymbol{x}_i, t]\langle\boldsymbol{\xi}\rangle \frac{\underline{\boldsymbol{Y}}[\boldsymbol{x}_i, t]\langle\boldsymbol{\xi}\rangle}{\left\|\underline{\boldsymbol{Y}}[\boldsymbol{x}_i, t]\langle\boldsymbol{\xi}\rangle\right\|} \tag{2.5}$$

where $\underline{t}[\boldsymbol{x}_i, t]\langle\cdot\rangle: \mathbb{R}^D \to \mathbb{R}$ is the *force scalar state* and depends on the constitutive law of the material. As demonstrated in [61], the form of Eq. (2.5) leads to satisfy for any bounded body $\mathcal{B}$ both the balance of linear and angular momentum. In particular, as Fig. 2.1a shows, it is evident that for an ordinary material the balance of angular momentum is always satisfied due to the alignment of the forces along the direction of the deformed bonds. By assuming the constitutive model referred as *Linear Peridynamic Solid* (LPS) [61], a macro-elastic potential density energy function $W(\theta, \underline{e}^d)$ can be defined as follows:

$$W(\theta, \underline{e}^d) = \frac{k'\theta^2}{2} + \frac{\alpha}{2}(\underline{\omega e}^d) \bullet \underline{e}^d \tag{2.6}$$

where θ is the volume dilatation of the neighborhood $H(\boldsymbol{x}_i)$ of the material point $\boldsymbol{x}_i$, $\underline{e}^d = \underline{e} - \frac{\theta \underline{x}}{3}$ the scalar deviatoric state component of the single bond elongation $\underline{e}\langle\boldsymbol{\xi}\rangle$ defined as:

$$\underline{e}\langle\boldsymbol{\xi}\rangle = \|\boldsymbol{y}(\boldsymbol{x}_j, t) - \boldsymbol{y}(\boldsymbol{x}_i, t)\| - \|\boldsymbol{\xi}\| \tag{2.7}$$

where $\underline{x}$ is the scalar state defined by $\underline{x}\langle\boldsymbol{\xi}\rangle = \|\boldsymbol{\xi}\|$. $\underline{\omega}$ is the scalar state of the influence function which associates a weight to each bond. If the influence function is spherical, then the material is isotropic; this assumption will be considered in the whole dissertation. The constants k' and α are positive scalars to be related to the material properties [36, 62], which are defined by:

$$k' = k \quad , \quad \alpha = \frac{15\mu}{q} \qquad \text{3D cases} \tag{2.8a}$$

$$k' = k + \frac{\mu}{9}\frac{(\nu+1)^2}{(2\nu-1)^2}, \quad \alpha = \frac{8\mu}{q} \qquad \text{2D plane stress cases} \tag{2.8b}$$

$$k' = k + \frac{\mu}{9} \quad , \quad \alpha = \frac{8\mu}{q} \qquad \text{2D plane strain cases} \tag{2.8c}$$

with k and μ the bulk and shear modulus of the medium, respectively. The weighted volume q is a scalar defined by:

$$q = (\underline{\omega x}) \bullet \underline{x} = \int_H \underline{\omega}\langle\|\boldsymbol{\xi}\|\rangle \|\boldsymbol{\xi}\|^2 dV \tag{2.9}$$

The *force scalar state* is consequently calculated by means of *Fréchet* [36] derivative of the potential density energy function as:

$$\underline{t}[\boldsymbol{x}_i] = \frac{3k'\theta}{q}\underline{\omega x} + \alpha\underline{\omega e}^d \qquad \text{3D cases} \tag{2.10a}$$

$$\underline{t}[\boldsymbol{x}_i] = \frac{2(2\nu-1)}{\nu-1}\left[k'\theta + \frac{\alpha}{3}(\underline{\omega e}^d) \bullet \underline{x}\right]\frac{\underline{\omega x}}{q} + \alpha\underline{\omega e}^d \qquad \text{2D plane stress cases} \tag{2.10b}$$

$$\underline{t}[\boldsymbol{x}_i] = 2\left[k'\theta - \frac{\alpha}{3}(\underline{\omega e}^d) \bullet \underline{x}\right]\frac{\underline{\omega x}}{q} + \alpha\underline{\omega e}^d \qquad \text{2D plane strain cases} \tag{2.10c}$$

As stated in Section 1.4.3, a special version of the formulation given by OSB-PD is referred as BB-PD [3] since the interaction is restricted between couple of material points. In such cases the behaviour of a bond is completely independent of all the others.

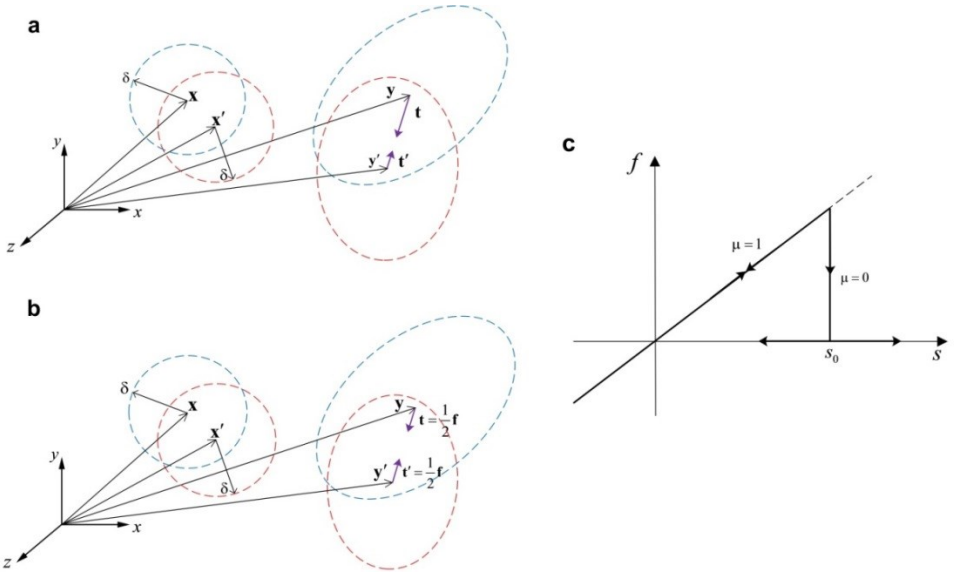


Fig. 2.1: a) Deformation of the bond between points $\boldsymbol{x}$ and $\boldsymbol{x}'$ and the vector forces developed between them in the OSB-PD formulation. b) Deformation of the bond between points $\boldsymbol{x}$ and $\boldsymbol{x}'$ and the vector forces developed between them in the BB-PD formulation. c) linear-failure constitutive law of a bond.

This assumption implies that the force density that a point $\boldsymbol{x}_j$ exerts on point $\boldsymbol{x}_i$ and that is exerted by $\boldsymbol{x}_i$ on $\boldsymbol{x}_j$ are parallel to the direction of the deformed bond and equal in magnitude (see Fig. 2.1b). Therefore, the *force scalar state* takes the form:

$$\underline{t}[\boldsymbol{x}_i] = \frac{9k''}{q}\underline{\omega e} \tag{2.11}$$

with k''constant to be related to the material properties. Let s the stretch of the bond be defined as $s = \underline{e}\langle\boldsymbol{\xi}\rangle/\|\boldsymbol{\xi}\|$, Eq. (2.11) can be written for the image of a bond as:

$$\underline{t}[\boldsymbol{x}_i]\langle\boldsymbol{\xi}\rangle = f(\boldsymbol{\eta},\boldsymbol{\xi}) = c\underline{\omega}\langle\boldsymbol{\xi}\rangle s \tag{2.12}$$

with $f(\boldsymbol{\eta},\boldsymbol{\xi})$ called *pairwise force function* and c is the micromodulus (stiffness of the bond). Assuming $\underline{\omega}\langle\boldsymbol{\xi}\rangle = 1$, the micromodulus derived by equating the density

of potential energy of classical mechanics with that of peridynamics [3, 42, 46] takes the form:

$$c = \frac{12E}{\pi\delta^4} \qquad \text{3D cases} \qquad (2.13a)$$

$$c = \frac{9E}{\pi\delta^3 t_b} \qquad \text{2D plane stress cases} \qquad (2.13b)$$

$$c = \frac{48E}{5\pi\delta^3} \qquad \text{2D plane strain cases} \qquad (2.13c)$$

$$c = \frac{3E}{\delta^2} \qquad \text{1D cases} \qquad (2.13d)$$

with t_b the thickness of the body. With reference to the BB-PD, Eq. (2.6) of the density of potential energy at each point $\boldsymbol{x}_i$ and time t can be rewritten as:

$$W(\boldsymbol{x}_i, t) = \frac{1}{2}\int_{H_{x_i}} w(\boldsymbol{\eta}, \boldsymbol{\xi}, t)\, dV_{x_j} = \frac{1}{2}\int_{H_{x_i}} \frac{c\underline{\omega}\langle\boldsymbol{\xi}\rangle s(t)^2 \xi}{2}\, dV_{x_j} \qquad (2.14)$$

where for simplicity's sake $\xi = \|\boldsymbol{\xi}\|$, $w(\boldsymbol{\eta}, \boldsymbol{\xi}, t)$ is referred as *micropotential energy* which is the energy in a single bond (dimensions of energy per unit volume squared). The concept of local damage as degradation of the mechanical response of a particle of material of the body can easily be introduced in peridynamics by using the constitutive law referred as *Prototype Microelastic Brittle* (PMB) model. According to this model, every bond breaks if its stretch overcomes a given limit value s_0 called *critical stretch*, which is a function of the critical fracture energy release rate of the material G_0 [42]. The critical stretch s_0 is given, for different cases, by the following expressions:

$$s_0 = \sqrt{\frac{5G_0}{6E\delta}} \qquad \text{3D cases} \qquad (2.15a)$$

$$s_0 = \sqrt{\frac{4\pi G_0}{9E\delta}} \qquad \text{2D plane stress cases} \qquad (2.15b)$$

$$s_0 = \sqrt{\frac{5\pi G_0}{12E\delta}} \qquad \text{2D plane strain cases} \qquad (2.15c)$$

Moreover, the rupture of the bond is irreversible so that the constitutive model is history-dependent, as shown in Fig. 2.1c, therefore, a binary scalar-valued function $\mu(\xi)$ has to be introduced to take into account the structural health state of the bond; $\mu(\xi) = 1$ means that the bond is active and $\mu(\xi) = 0$ that it is broken. Then, the force scalar state assumes the form as:

$$\underline{t}[\boldsymbol{x}_i]\langle\boldsymbol{\xi}\rangle = c\underline{\omega}\langle\boldsymbol{\xi}\rangle\mu(\xi)s \tag{2.16}$$

The local damage level ϕ at a given material point $\boldsymbol{x}_i$ at time *t* can then be defined as:

$$\phi(\boldsymbol{x}_i, t) = 1 - \frac{\int_{H_{x_i}} \mu(t,\xi) dV_{x_j}}{\int_{H_x} dV_{x_j}} \tag{2.17}$$

where $0 \leq \phi \leq 1$, 0 represents the pristine state of the material and 1 the complete separation of the material point from all surrounding points within its horizon. As the classic theory of mechanics, the equation of motion of peridynamics can be linearized when the hypothesis of small deformations are assumed. Under this assumption, Eq.(2.1) reduces to a linear integral equation [61]. Let ℓ be defined as:

$$\ell = \sup_{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|} \|\boldsymbol{u}(\boldsymbol{x}_j, t) - \boldsymbol{u}(\boldsymbol{x}_i, t)\| \tag{2.18}$$

then the displacement field is said to be small if the condition $\ell \ll \delta$ is satisfied. We want to underline that, unlike the classic theory, the definition of small displacements in peridynamics does not restrict rigid translations of the body, while does restrict rigid body rotations to small angles [61]. Moreover, it allows for possible small discontinuities in the displacement field, which is one of the key difference between PD and the classic theory. Therefore, the linearized version of Eq. (2.1) assumes the form:

$$\begin{aligned} \rho(\boldsymbol{x}_i)\ddot{\mathbf{u}}(\boldsymbol{x}_i, t) = &\int_{H_{x_i}} \boldsymbol{C}_0(\boldsymbol{x}_i, \boldsymbol{x}_j)(\boldsymbol{u}(\boldsymbol{x}_j, t) - \boldsymbol{u}(\boldsymbol{x}_i, t))\, dV_{x_j} \\ &+ \boldsymbol{b}(\boldsymbol{x}_i, t), \qquad \forall \boldsymbol{x}_j \in H(\boldsymbol{x}_i) \end{aligned} \tag{2.19}$$

with $\boldsymbol{C}_0$ is called *micromodulus tensor* field and is defined for the case of BB-PD as follows:

$$\boldsymbol{C}_0 = \frac{c\underline{\omega}\langle\boldsymbol{\xi}\rangle}{\xi^2}\boldsymbol{\xi}\otimes\boldsymbol{\xi} \tag{2.20}$$

2.2. Surface Effect

The non-locality of peridynamic formulation leads to the called "*surface effect*", which is also known with the term "*skin effect*". Basically, peridynamic material parameters, such as the micromodulus of the bonds (see Eq. (2.13)), are derived by assuming that the neighborhood $H(\boldsymbol{x}_i)$ of a point $\boldsymbol{x}_i$ is fully embedded in the domain. Such an assumption is not valid if point $\boldsymbol{x}_i$is close to the boundary of the domain, since the neighborhood $H(\boldsymbol{x}_i)$ has a truncated shape, as shown in Fig. 2.2.

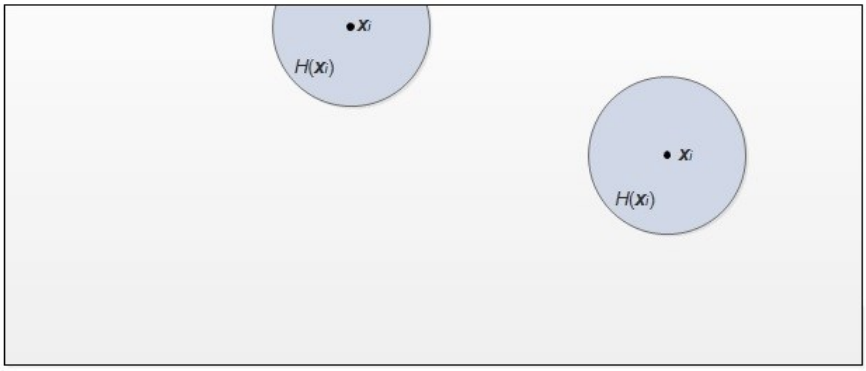


Fig. 2.2: Material point $\boldsymbol{x}_i$ in a 2D domain with a truncated neighbourhood if located near the external surface and with the full disk shape if located in the bulk.

For instance, if the domain shown in Fig. 2.2 is stretched with a constant strain, the potential energy density (see Eq. 2.14) of a material point near the external surface is lower than in the bulk. This results in a reduction of the material stiffness near the boundaries of the domain of interest, leading to the introduction of an error in the PD solution. Numerous methods have been applied to reduce the surface effect, i.e. see [55, 63, 64, 65], the method adopted in this dissertation (see Section 4.1) is based on the potential energy approach [66].

2.3. Refinement and scaling

The idea of introducing the adaptive grid refinement and scaling as a strategy to reduce automatically both the grid spacing and horizon in a peridynamic model was first introduced in [57, 58], with the goal to overcome the drawbacks shown

by this theory when discretized with a uniform grid of nodes and a constant horizon [42]. In fact, it is clear that the adoption of such a grid does not allow optimizing its computational efficiency, which is an essential feature if real-life structures have to be analysed. Reference [37] has underlined how the horizon δ can be linked to the characteristic length-scale of the material or of the phenomena under investigation, especially with reference to damage and fracture phenomena related to the crack propagation [4]. As shown in [59], in the future it will be possible to use the AGRS to capture the desired length-scale into the regions in which the crack propagation should take place, by adopting an horizon size convenient for computational efficiency far from those regions. By doing so, the potentialities shown by peridynamics could be adopted in order to obtain a concurrent multi-scale model in a unique framework. Of course, the use of a varying horizon in the peridynamic domain needs to be addressed since, as stated in [58]: "*The original formulation of peridynamics* [3] *and the subsequent state-based formulation* [36] *are based on the implicit assumption that the horizon is constant over the domain*". In order to perform this mixed description of a material using two horizons with a different length, scaling concept in peridynamics has been introduced [57, 67]. Let δ and ε define two different horizon radii, the scaling requires the change of the micropotential function defined by means of the following expressions:

$$w_\varepsilon(\boldsymbol{\eta},\boldsymbol{\xi}) = \gamma^3 w_\delta(\gamma\boldsymbol{\eta},\gamma\boldsymbol{\xi}) \qquad \text{3D cases} \qquad (2.21a)$$

$$w_\varepsilon(\boldsymbol{\eta},\boldsymbol{\xi}) = \gamma^2 w_\delta(\gamma\boldsymbol{\eta},\gamma\boldsymbol{\xi}) \qquad \text{2D cases} \qquad (2.21b)$$

$$w_\varepsilon(\boldsymbol{\eta},\boldsymbol{\xi}) = \gamma\, w_\delta(\gamma\boldsymbol{\eta},\gamma\boldsymbol{\xi}) \qquad \text{1D cases} \qquad (2.21c)$$

where $\gamma = \delta/\varepsilon$ is defined the geometrical scaling of the horizon size. In this way the deformation energy of a body characterized by a uniform stretch has to be constant in the whole domain regardless the changing of the horizon length, as demonstrated, i.e. for the 2D case, by the insertion of Eq. (2.21b) in Eq. (2.14):

$$\begin{aligned} W_\delta(\boldsymbol{x}_i) &= \frac{t_b}{2}\int_{H_\delta} w_\delta(\gamma\boldsymbol{\eta},\gamma\boldsymbol{\xi})\, d(\gamma^2 A) \\ &= \frac{t_b}{2}\int_{H_\varepsilon} \frac{1}{\gamma^2} w_\varepsilon(\boldsymbol{\eta},\boldsymbol{\xi})\gamma^2 dA = W_\varepsilon(\boldsymbol{x}_i) \end{aligned} \qquad (2.22)$$

in which *dA* represents the infinitesimal area of the disk centered at point $\boldsymbol{x}_i$. According to the scaling concept, the change of the horizon induces the change of the micromodulus (see Eqs. (2.13)), moreover, by this strategy all the bonds inside the horizon of the source node have the same micromodulus calculated by using its horizon in Eq. (2.13). As done with the micromodulus, the same strategy of scaling suggested in [57] can be extended for the critical bond stretch calculated by using Eq. (2.15). Another issue arises when the horizon size varies over the domain; Fig. 2.3 clearly shows that the change of the horizon size leads to loose some peridynamic bonds. Let us consider the case showed in Fig. 2.3a: if the family point *j* is located inside the neighbourhood of the point *i*, Eq. (2.1) defined on source node *i* includes the bond between the source node and its family node *j*. Whereas, considering the reverse case in which *j* is the source node, it is clear from Fig. 2.3b that Eq. (2.1) does not include the bond anymore, then the bond is basically "lost".

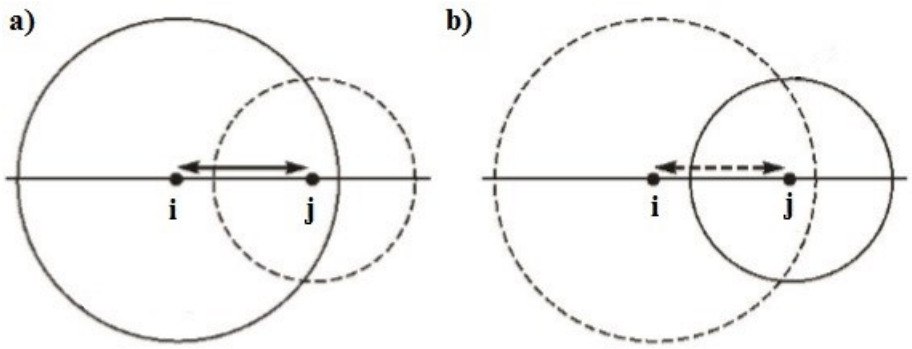


Fig. 2.3: Two points *i* and *j* with a different horizon length, the double narrow segment line indicates the bond (interaction) between the two points, a) the interaction exists (continuous line) when the point *j* is inside the horizon of the point *i*, b) the interaction is lost (dashed line) when point *i* is not inside the horizon of the point *j*.

The loss of bonds leads to the presence of the ghost forces [67], which introduces anomalies in the displacement field of the PD solution. Such an artifact is originated by the disequilibrium of the equation of motion imposed at the nodes with a variable horizon [60]. Reference [60] shows that the use of Eq. (2.1) originally introduced by peridynamics does not satisfy the conservation of linear momentum [60]. The authors in [67] point out that even if the equilibrium is not satisfied, there is not a net force on the body since the ghost forces are self-equilibrated. Moreover, they show that such an artifact in the displacement field may be neglected when the horizon length varies smoothly in the medium.

Otherwise, in applications where we have discontinuous jumps of the horizon size which may lead to have big errors, two methods [67] referred as "*partial stress*" and "*splice*" are proposed to eliminate the presence of ghost forces. Due to the complexity of such strategies, an alternative approach has been proposed in [60] introducing the concept of dual-horizon, as discussed in next Section 2.4.

2.4. Dual-horizon concept

As stated in the Section 2.3, an alternative strategy introduced with the aim of eliminating the ghost forces is given in [60]. When a varying horizon size is adopted in the domain, we shall address the issue if the conservation of linear and angular momentum are still satisfied by using the original equation of motion of peridynamics (see Eq. (2.1)). Reference [60] demonstrated that the balance of linear momentum can be respected by introducing the concept of dual-horizon, which is defined for a point of coordinate $\boldsymbol{x}_i$ as a set of points $\boldsymbol{x}_j$ whose horizons include it, denoted as:

$$H'(\boldsymbol{x}_i) = \{\boldsymbol{x}_j : \boldsymbol{x}_i \in H(\boldsymbol{x}_j)\} \tag{2.23}$$

with the superscript prime to indicate *dual*. As a results to introducing the dual-horizon concept, Eq.(2.1) can be rearranged as follows:

$$\begin{aligned}\rho(\boldsymbol{x}_i)\ddot{\mathbf{u}}(\boldsymbol{x}_i,t) = &\int_{H'(\boldsymbol{x}_i)} \underline{\mathbf{T}}[\boldsymbol{x}_i,t]\langle \boldsymbol{x}_j - \boldsymbol{x}_i\rangle \, dV_{\boldsymbol{x}_j} - \\ &- \int_{H(\boldsymbol{x}_i)} \underline{\mathbf{T}}[\boldsymbol{x}_j,t]\langle \boldsymbol{x}_i - \boldsymbol{x}_j\rangle dV_{\boldsymbol{x}_j} + \boldsymbol{b}(\boldsymbol{x}_i,t)\,, \\ &\boldsymbol{x}_j \in H(\boldsymbol{x}_i), \quad \boldsymbol{x}_j \in H'(\boldsymbol{x}_i)\end{aligned} \tag{2.24}$$

We can observe from Eq.(2.24) as the original Eq.(2.1) has been split into two distinct integral parts, the first one corresponding to the region of integration $H'(\boldsymbol{x}_i)$ and the second one to the region of integration $H(\boldsymbol{x}_i)$. The former can be called the integral of the active forces since the integrand represents the force density exerted by the points $\boldsymbol{x}_j \in H'(\boldsymbol{x}_i)$ on point $\boldsymbol{x}_i$, while the latter can be called the integral of passive forces since represents the undertaken reaction force density of the active forces exerted by $\boldsymbol{x}_i$ on the family nodes $\boldsymbol{x}_j \in H(\boldsymbol{x}_i)$, as a consequence of Newton's third law. It is clear that when the horizons are set constant, namely $H(\boldsymbol{x}_i) = H'(\boldsymbol{x}_i)$, Eq. (2.24) reduces to Eq. (2.1). Moreover, let

us emphasize the fact that the horizon has a physical meaning which allows relating the micromodulus to the mechanical properties of the material (see Eq. (2.13)), whereas the dual-horizon is only a set of points which allows satisfying Newton's third law in the '*overlap*' regions characterized by a varying horizon. Unlike the conservation of the linear momentum, the balance of the angular momentum is satisfied even if the horizon varies over the domain, as a consequence the dual-horizon is not involved. Having always in mind the BB-PD, let $c(\delta_{\boldsymbol{x}_i})$ and $c(\delta_{\boldsymbol{x}_j})$ denote the micromoduli based on $\delta_{\boldsymbol{x}_i}$ and $\delta_{\boldsymbol{x}_j}$ calculated from Eq. (2.13), then the pairwise force functions are computed by the following expressions:

$$\boldsymbol{f}_{\boldsymbol{x}_i\,\boldsymbol{x}_j} = \underline{\mathbf{T}}[\boldsymbol{x}_i, t]\langle \boldsymbol{x}_j - \boldsymbol{x}_i \rangle = \\ = \frac{c(\delta_{\boldsymbol{x}_j})}{2}\, s\, \frac{\boldsymbol{y}(\boldsymbol{x}_j, t) - \boldsymbol{y}(\boldsymbol{x}_i, t)}{\left\| \boldsymbol{y}(\boldsymbol{x}_j, t) - \boldsymbol{y}(\boldsymbol{x}_i, t) \right\|}, \qquad \forall \boldsymbol{x}_j \in H'(\boldsymbol{x}_i) \tag{2.25a}$$

$$\boldsymbol{f}_{\boldsymbol{x}_j\,\boldsymbol{x}_i} = \underline{\mathbf{T}}\big[\boldsymbol{x}_j, t\big]\langle \boldsymbol{x}_i - \boldsymbol{x}_j \rangle = \\ = \frac{c(\delta_{\boldsymbol{x}_i})}{2}\, s\, \frac{\boldsymbol{y}(\boldsymbol{x}_i, t) - \boldsymbol{y}(\boldsymbol{x}_j, t)}{\left\| \boldsymbol{y}(\boldsymbol{x}_j, t) - \boldsymbol{y}(\boldsymbol{x}_i, t) \right\|}, \qquad \forall\, \boldsymbol{x}_j \in H(\boldsymbol{x}_i) \tag{2.25b}$$

in which the subscripts $\boldsymbol{x}_i\boldsymbol{x}_j$ of the pairwise force function indicate the active force exerted by $\boldsymbol{x}_j$ on $\boldsymbol{x}_i$ and $\boldsymbol{x}_j\boldsymbol{x}_i$ the reaction force exerted by $\boldsymbol{x}_i$ on $\boldsymbol{x}_j$. The same considerations can be applied to the more general formulation of peridynamics, such as OSB-PD and non-ordinary SB-PD (see [60] for further details). The concepts of dual-horizon presented in this section will be used in chapter 4.1 in order to investigate the possible benefits on the numerical PD solution when the ghost forces are completely removed from the analyzed domain.

3. Numerical discretization and algorithms

In the following, the methods employed in this work to implement numerically peridynamics and the adaptive grid refinement and scaling approach are introduced. As a first step, the numerical method adopted when the domain is discretized with a structured uniform grid of nodes will be shown, then the algorithms developed for the application of the adaptive grid refinement and scaling in the peridynamic model will be presented and discussed as well. All the codes developed in this work have been written by using the software Matlab®, especially the structure of the code has been designed with the target to optimize the time of the simulations.

3.1. Mesh-free method

Even if peridynamic theory can be implemented with different numerical methods [41], one of the most popular and widely used, due basically to its implementation simplicity, is based on a mesh-free approach [42], in which the whole domain is discretized with a uniform regular grid of nodes and a constant horizon. The nodes are equally spaced in both the *X, Y* and *Z* direction, therefore the grid spacing is uniform so that $\Delta X = \Delta Y = \Delta Z$. To each node a square cell of volume $\Delta V = \Delta X^3$ is assigned, such that the node is located at the center of its cell. These nodes are referred to as *source nodes* (or *source cell* with reference to its corresponding cells), while the cells whose nodes interact with its horizon and dual-horizon will be referred as *family cells* and *dual-family cells*, respectively. By adopting the one-point Gauss quadrature rule for the space integration, the discretized form of Eq. (2.1) is expressed as:

$$\rho\ddot{\boldsymbol{u}}_i^n = \sum_j \left\{\underline{\boldsymbol{T}}[\boldsymbol{x}_i^n]\langle \boldsymbol{x}_j^n - \boldsymbol{x}_i^n\rangle - \underline{\boldsymbol{T}}[\boldsymbol{x}_j^n]\langle \boldsymbol{x}_i^n - \boldsymbol{x}_j^n\rangle\right\}\beta_j \Delta V_j + \boldsymbol{b}_i^n \qquad \forall\, \boldsymbol{x}_j \in H_{x_i} \tag{3.1a}$$

OSB-PD

$$\rho\ddot{\boldsymbol{u}}_i^n = \sum_j \boldsymbol{f}(\boldsymbol{u}_j^n - \boldsymbol{u}_i^n, \boldsymbol{x}_j - \boldsymbol{x}_i)\beta_j \Delta V_j + \boldsymbol{b}_i^n \qquad \forall\, \boldsymbol{x}_j \in H_{x_i} \tag{3.1b}$$

BB-PD

for the ordinary state-based and bond-based, respectively. While, for the case of bond-based Dual-Horizon PeriDynamics (DHPD), the discretized form of Eq.(2.24) is expressed as:

$$\rho \ddot{\boldsymbol{u}}_i^n = \sum_k \boldsymbol{f}(\boldsymbol{u}_k^n - \boldsymbol{u}_i^n, \boldsymbol{x}_k - \boldsymbol{x}_i') \, \beta_i \Delta V_k + \\ - \sum_j \boldsymbol{f}(\boldsymbol{u}_j^n - \boldsymbol{u}_i^n, \boldsymbol{x}_j - \boldsymbol{x}_i) \beta_j \Delta V_j + \boldsymbol{b}_i^n, \quad \forall \boldsymbol{x}_k \in H'_{\boldsymbol{x}_i}, \forall \boldsymbol{x}_j \in H_{\boldsymbol{x}_i} \qquad \text{BB-DHPD (3.1c)}$$

In Eqs. (3.1) n is the time step number, $\boldsymbol{x}_i$ the coordinate of the source node, $\boldsymbol{x}_j$ the centroid coordinate of the family cell, $\boldsymbol{x}_k$ the central coordinate of the dual-family cell, and $\boldsymbol{x}_i'$ the centroid coordinate of the source cell seen by the horizon of its dual-family cell. The weighting factors (also called volume correction factors) β_i and β_j are multiplied by the corresponding volume cell ΔV in order to define the effective volume included in the neighborhood $H_{\boldsymbol{x}_i}$. Several numerical algorithms have been introduced in order to calculate both the locations of Gauss points of the cells and their corresponding weighting factors, as presented in [68]. In particular, two algorithms will be employed in this work: the first one is referred as Partial Area - PDLAMMPS algorithm (PA-PDLAMMPS) based on the commonly used PD software PDLAMMPS [69], while the second is called IPA-Hybrid (Improvement Partial Area) introduced by Seleson in [68]. According to the PA-PDLAMMPS algorithm the source node interacts with the nodes located inside the horizon, the Gauss quadrature points coincide with the geometrical center of cells, even if the cell is intersected by the horizon of the source node, and the weighting volume of such cells is calculated by means of a linear interpolation expression (see Fig. 3.1a). Whereas, the IPA-Hybrid extends the number of interactions between the source and the family nodes even to those nodes outside the horizon whose cells are partially located inside the neighborhood of the source node. This algorithm is based on analytical calculations of the portion of volume of cells partially covered by the neighborhood of the source node, while the corresponding geometrical centroids of such portions are estimated by means of a linear interpolation expression (see Fig. 3.1b).

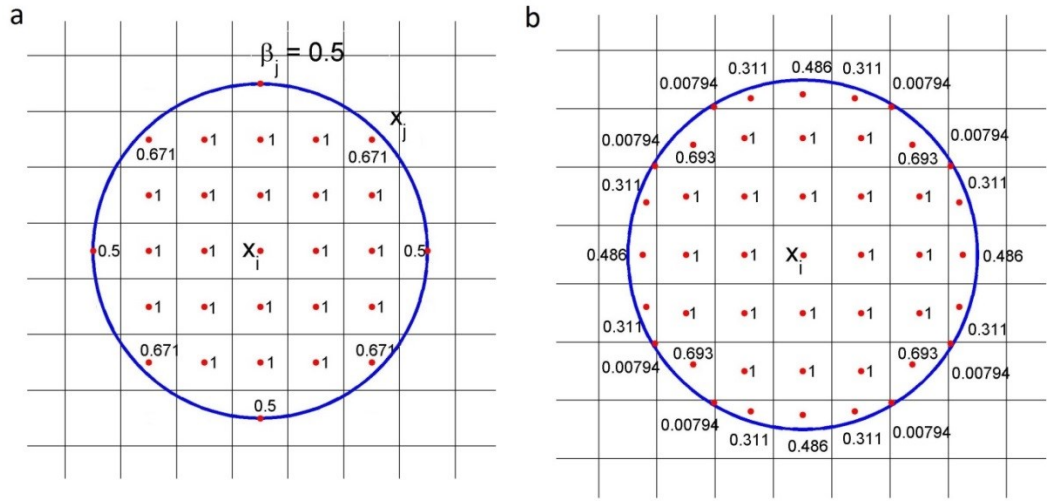


Fig. 3.1: Example of application of different algorithms in order to calculate Gauss quadrature points and the corresponding coefficient correction factors in a uniform grid of nodes: a) PA-PDLAMMPS algorithm [69], b) IPA-Hybrid [68].

We remark as the IPA-Hybrid algorithm has only been developed for 2D grids. The numerical discretization of peridynamics introduces a parameter called *m ratio* defined as the ratio between the horizon and the grid spacing. This parameter is usually chosen as a tradeoff between accuracy and computational efficiency of the solution, even if it shall be carefully chosen when we are dealing with a crack propagation problem, as discussed in Chapter 5. The three parameters horizon δ, grid spacing ΔX and ratio m play a crucial role in determining the type of convergence of the peridynamic numerical solution either to the exact peridynamic solution or to that of the classical elasticity theory [44]. Two types of basic numerical convergence can be identified:

- the δ-convergence: $\delta \to 0$ and m ratio is kept constant or is increased at a slower rate with respect to the decreasing of δ
- the m-convergence: δ is kept constant and m ratio $\to \infty$.

By the combination of these numerical convergences we can obtain the δm-convergence, namely $\delta \to 0$ while the m ratio is increased faster than the decreasing of δ. Eqs. (3.1) are solved in time by means of an explicit solver as the Velocity-Verlet scheme [70], which is robust, reliable and simple:

$$\dot{\boldsymbol{u}}_{n+\frac{1}{2}} = \dot{\boldsymbol{u}}_n + \frac{\Delta t}{2}\ddot{\boldsymbol{u}}_n \tag{3.2}$$

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \Delta t \dot{\boldsymbol{u}}_{n+\frac{1}{2}}$$

$$\dot{\boldsymbol{u}}_{n+1} = \dot{\boldsymbol{u}}_{n+\frac{1}{2}} + \frac{\Delta t}{2} \ddot{\boldsymbol{u}}_{n+1}$$

where Δt is the constant time step which is chosen smaller than the critical time step $\Delta t_c = \xi_{min}/c_m$, being ξ_{min} the smallest bond length (namely ΔX for a regular grid) and c_m the speed of the longitudinal wave in the medium.

3.2. Adaptive grid refinement and scaling algorithms

Several examples of application of remeshing techniques can be found in the literature. For instance, in FE analyses [71] such a technique is used to automatically activate the size reduction of the elements of the mesh and stopped when the approximation error of a particular physical quantity of the field analysed is below a given threshold. One of the most popular algorithms to generate new elements in FEM applications is based on a recursive decomposition using *quadtrees* in two dimensions or *octrees* in three dimensions [72, 73], as shown in Fig. 3.2 in the case of quadrilateral elements. Such a technique has generally been adopted in FEM applications in order to split the selected elements of the mesh, called parent elements, in more sub-elements called child elements. Then, by a recursive splitting of the generated elements different levels of refinement can be reached, as shown in Fig. 3.2. As done in FEM applications, in the mesh-free method we apply a similar approach generating new nodes (called child nodes) from the nodes belonging to the initial grid (called parent nodes) .

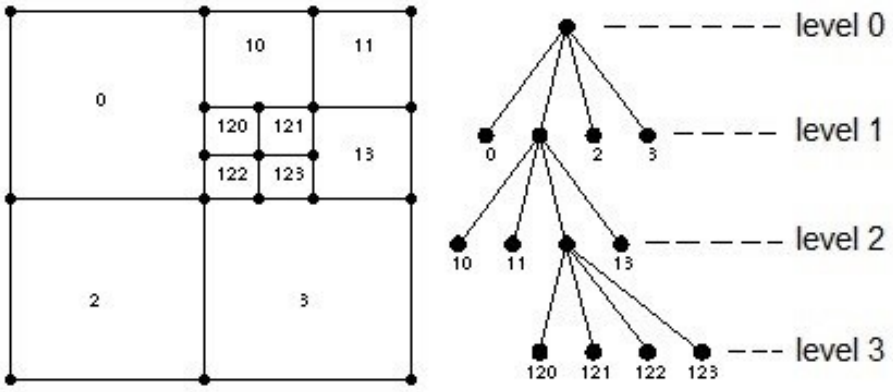


Fig. 3.2: Quadtree structure for square elements.

In FE analyses such a technique is used to automatically activate the size reduction of the elements of the mesh and it is stopped when the approximation error is below a given threshold [71, 74]; In the present work grid refinement is activated if the value of a physical quantity (trigger) associated to the nodes overcomes a given threshold value. Moreover, the refinement level is predetermined by the user. In the following we present the triggers used to activate the refinement and scaling, while in Appendix A the pseudo-codes of the employed AGRS algorithm are given.

3.2.1. Trigger based on potential energy

In [58] a trigger based on the density of potential energy has been proposed, see Eq.(2.14), so that a node of the initial grid $\boldsymbol{x}_i$ is selected for the activation of the refinement if $W(\boldsymbol{x}_i) \geq W_{thres}$, the suggested threshold value is chosen as $W_{thres} = 0.4W_{max}$, where W_{max} is the maximum value of the density of potential energy in the grid at that instant in time. Such a threshold value is considered a good compromise between accuracy of the numerical solution and efficiency of the computational procedure. However, the refinement in [58] was applied only to linear elastic analyses, in the present work the AGRS is applied to dynamic crack propagation in brittle materials. In this case a trigger based only on the density of potential energy of the nodes exhibits two important weaknesses:

- It is not immediately possible to indicate a unique optimal threshold value since W_{thres} depends on the type and the complexity of the analysis
- If in the domain several cracks are simultaneously present the energy density trigger cannot guarantee an efficient activation of the AGRS, the refined regions may be too large, and cover the entire domain, or too small.

In certain cases very high levels of energy are concentrated in a limited zone of the domain in such a way that the energy density trigger misses other zones of the grid which would require a refinement, as shown in Sect. 6.2 about the case of the benchmark problem of the Kalthoff–Winkler experiment. The problem can be overcome by adopting lower threshold values, but the risk would be to have refined regions far too large and a consequent loss of efficiency. Fig. 3.3 shows the flow chart of the working principle of energy based trigger, in Appendix A the pseudo-code employed to implement the algorithm is given (see Algorithm A.1).

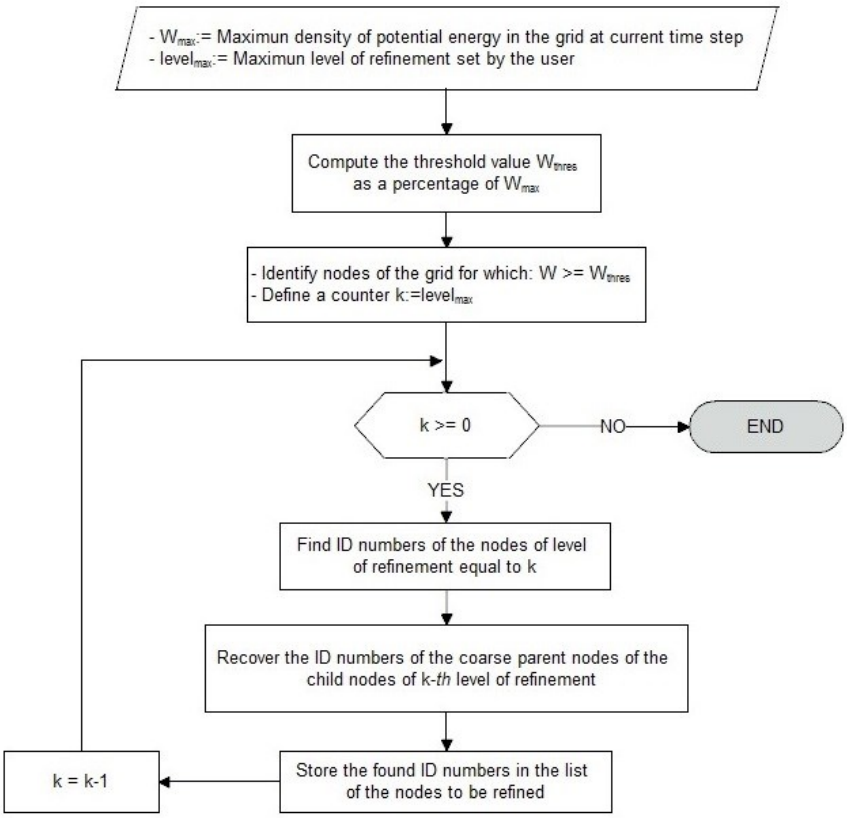


Fig. 3.3: Flow-chart of the energy based trigger.

3.2.2 Trigger based on damage

In the present work a new approach is proposed, two activation criteria are used simultaneously, a new trigger based on the damage state of the nodes is coupled to the energy density trigger: the nodes are chosen for the AGRS if $\Delta\phi = \phi - \phi_0 > 0$, where ϕ_0 is the initial damage state of the nodes (see Eq. (2.17) for the definition of damage). In Appendix A the pseudo-code employed to implement the damage based trigger is given (see Algorithm A.2). Fig. 3.4 shows how the two triggers work in order to activate the AGRS. The joint use of the two criteria makes more robust the AGRS activation, in particular the use of the damage trigger improves two important aspects in the numerical solution:

- It keeps the refinement active at the tip of the crack during its propagation in a region refined in previous instants of time.
- It activates the refinement in a coarse region where a crack is going to nucleate whenever the energy trigger fails: it is important to observe that

in the adopted approach all bond failures should happen in a refined region.

The working principle of the algorithm developed by employing the energy and damage based triggers is given as follows: at each time step of the analysis, the damage based trigger verifies if any of the coarse nodes has an incremented damage level and, if so, the analysis is repeated after having refined the coarse region around the identified node at the previous time step.

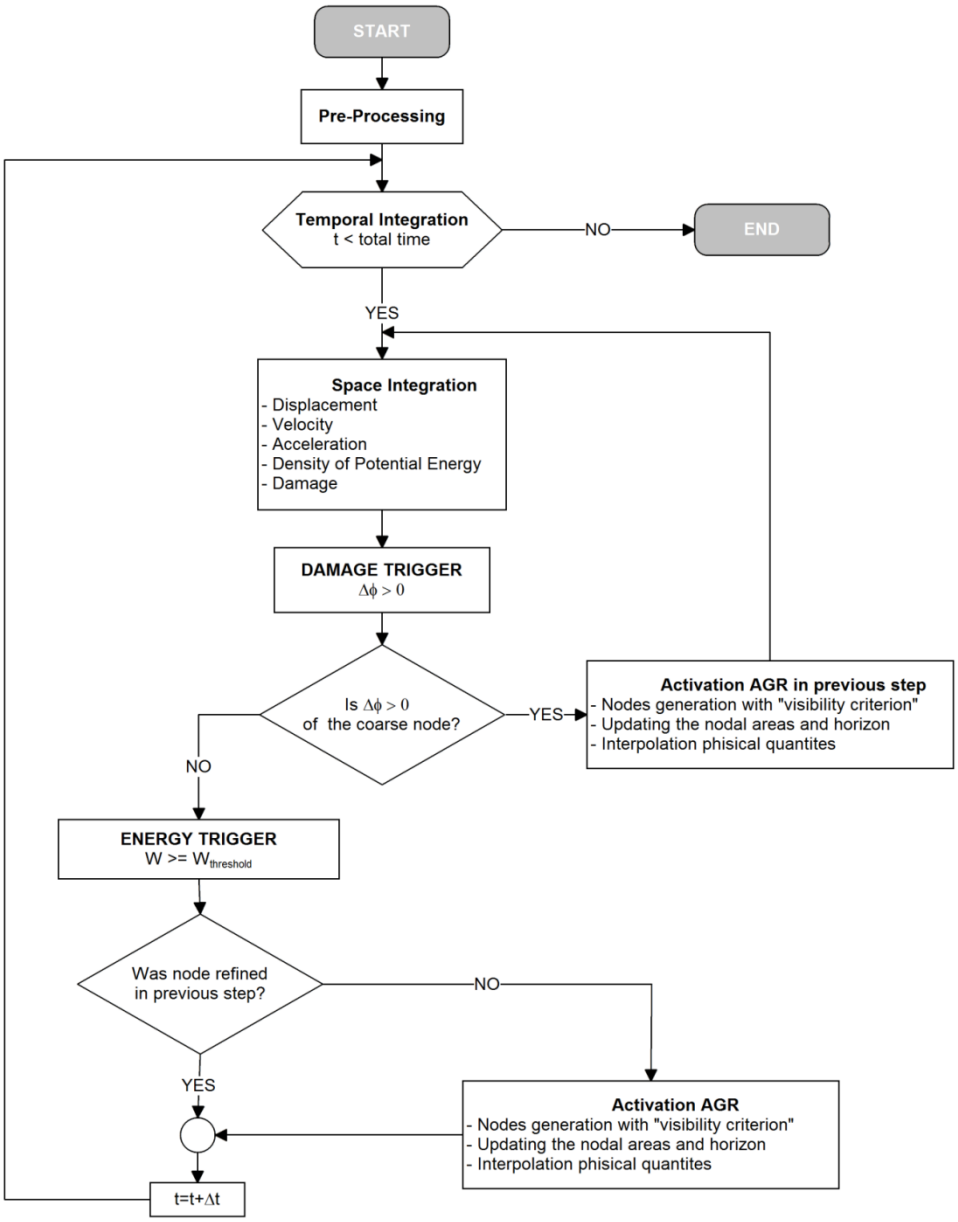


Fig. 3.4: Structure of the algorithm to implement the AGRS.

If no incremental damage is detected, then the analysis proceeds according to the flow chart shown in Fig. 3.4. If the identified node belongs to the original grid then it is added to the list of nodes to be refined, otherwise a search function is activated to find the coarse nodes close to the identified refined node, which had been generated by a previous refinement. It is clear that when we are dealing with a dynamic analysis, the AGRS cannot be started at the beginning of the simulation, since the energy density associated to each grid point is zero and therefore the whole grid would be refined. For this reason a starting time t_{st} has to be determined by carrying out a preliminary analysis on the coarse grid, long enough to generate an inhomogeneous distribution of energy. The simulation duration of the preliminary analysis t_{pr} can be simply determined by evaluating analytically the time that elastic waves take to propagate in the whole domain. An upper limit to t_{pr} can be estimated by using the expression d_{max}/c_m, with d_{max} the 'maximum distance' in the discretised domain (i.e. the diagonal in a rectangle domain). As a result of using this simple strategy, the user does not have to run the whole analysis, which may take too much time. Finally, t_{st} is determined empirically by the user who examines the solution of the preliminary analysis.

3.2.3 Node generation and updating properties

The generation of the new nodes can be obtained by adopting different techniques, in the following the strategies adopted in this work are reported, with their advantages and disadvantages:

A. The new nodes are inserted at the midpoints of the lines connecting adjacent nodes that have the same '*degree of kinship'*, namely between nodes belonging to the same level of refinement; i.e. with reference to 2D cases, a single parent node leads to generate nine child nodes included the parent node which switches to child node, as shown in Fig. 3.5a. This approach is fast, simple and easy to implement since the parent nodes do not have to be removed from the list of nodes of the grid, hence when the grid matrix (see Eq. (A.1) in Appendix A) is updated as a consequence of adding new nodes in the grid, the node ID number will not have to be changed. In this way, search operations of the nodes for which the new ID number should be reassigned can be avoided, reducing the computational

time of the analysis. Nevertheless, this approach requires the area modification of the nodes at interface with the nodes belonging to a different level of refinement, as shown in Fig. 3.5. In fact, this approach leads to modify the shape and the size of the area associated to the interface node. Moreover, some of the nodes are no more at the centre of their volume; that causes a loss of accuracy of the spatial integration since the motion equations are solved with a mid-point integration in space (see Eqs. (3.1)). Let *level* = 0,1,…*i* identify the refinement level of the nodes; then the initial grid spacing ΔX_0 will be split in half as shown by the following expression:

$$\Delta X_{level} = \frac{\Delta X_0}{(2^{level})} \tag{3.3}$$

then the volume $V_{level} = t_b \Delta X_{level}^2$ associated to each node has to be corrected according to the following expressions:

$$V_{level,int} = \begin{cases} V_{level} - V_{(level+1)}/4 & , \quad j = 1 \\ V_{level} - 3V_{(level+1)}/4 & , \quad j = 2 \\ V_{level} - V_{(level+1)} & , \quad j = 3 \\ V_{level} - 3\,V_{(level+1)}/2 & , \quad j = 4 \\ V_{level} - 7\,V_{(level+1)}/4 & , \quad j = 5 \end{cases} \tag{3.4}$$

with *j* the number of nodes of the *i*th level adjacent to the nodes of the *i*th-1 level, as shown in Eqs. (3.4) *j* can vary between 1 and 5 depending on the number of nodes of the *i*th level the distance of which is not bigger than the diagonal $d = \sqrt{2}\Delta x_{(level-1)}$ of the grid spacing of the nodes with the lower level of refinement.

B. The new nodes are generated by splitting the volume assigned to the node of the same '*degree of kinship*' in more sub-volumes, i.e. in four sub-volumes for 2D cases as shown in Fig. 3.5b. Then, the new nodes are located at the center of the corresponding sub-volumes. Unlike the previous approach, the parent node has to be removed from the grid because of its null contribution to the analysis. As a result of doing this, the current list of nodes (see Eq. (A.1)) of the grid should be updated at each step in which new nodes are added in the grid. Anyway, this drawback can be overcome by keeping the

removed nodes from the grid stored in the data structure of Eq. (A.1); we suggest as a strategy to detect such nodes by setting to zero their volume as well as material ID number. Besides, another useful strategy is to set their ID number used for detecting the level of refinement as $ID = -(level_{child} - 1)$ of the child nodes generated, in this way the removed parent nodes can be easily identify.

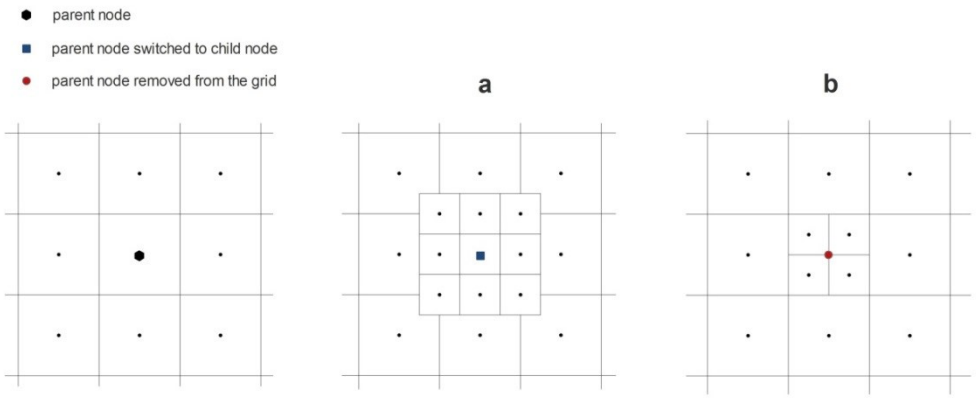


Fig. 3.5: Parent node with the assigned area: a) child nodes generated by using the A approach, b) child nodes generated by using the B approach.

The size of the area to be refined around each node identified by the triggers is determined according to the visibility criterion introduced in [57] which suggests that the refined zone has to be such that the horizon of the coarsest interface nodes has not to include the identified node. Therefore, all the family nodes within the horizon of the identified source node are refined, i.e. Fig. 3.6 shows both the 1st and 3rd level of refinement produced by the activation of the AGRS of a single node in the case of m = 3. One can notice from Fig. 3.6c that successive refinements are 'self-similar' since the same procedure is used to generate 1st level nodes from 0 level nodes, 2nd level nodes from 1st level nodes and so on. In particular, Fig. 3.6 shows the case of refinement in which the δ-convergence strategy is adopted, being both the initial grid spacing and horizon lengths divided by two for each level of refinement and scaling applied. We can notice that the grid spacing gradually varies from the coarsest region to the most refined region, this approach is in accord with what highlighted in [75, 76] for FEM and [57] for BB-PD about the fact that spurious wave reflections can be mitigated by only

adopting such a variation of refinement. Fig. 3.7 shows that the shape of the refined area depends on the value of the parameter *m*.

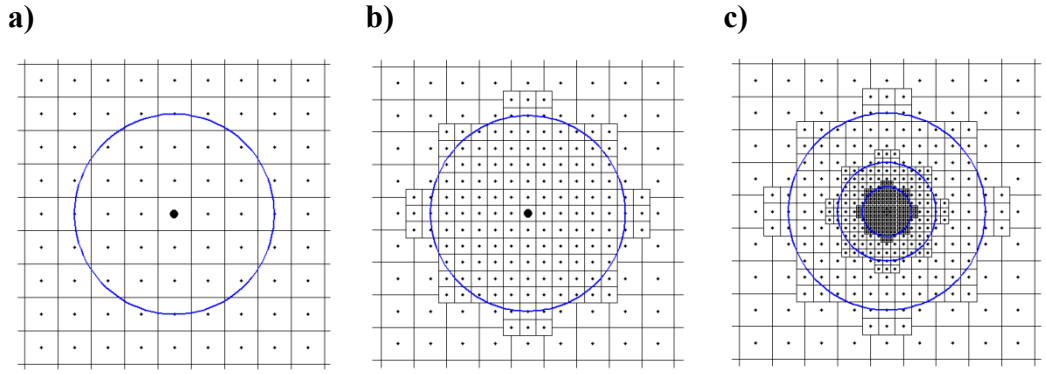


Fig. 3.6: Application of the visibility criterion for the grid refinement around a single node identified by the trigger when the δ-convergence strategy is applied (with m =3): a) coarse grid, identified node with its horizon, b) 1st level of refinement, c) 3rd level of refinement.

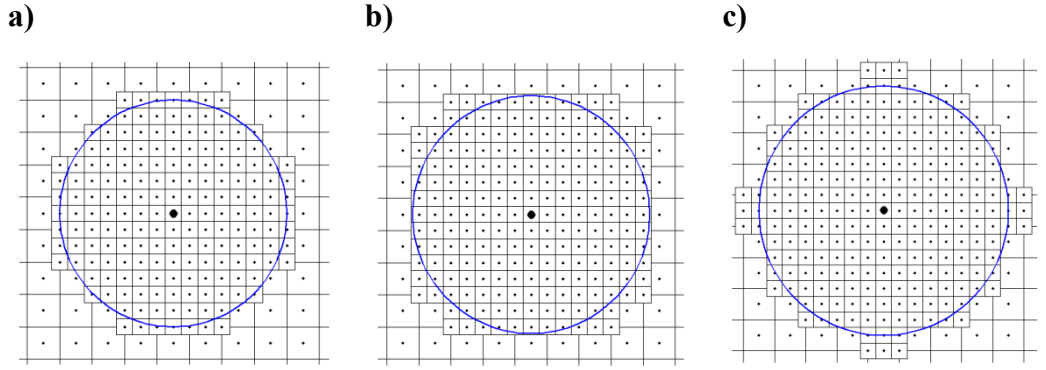


Fig. 3.7: Shape of the refined zone generated around a single node for various values of *m*: a) m =3.5, b) m =3.7, c) m =4.

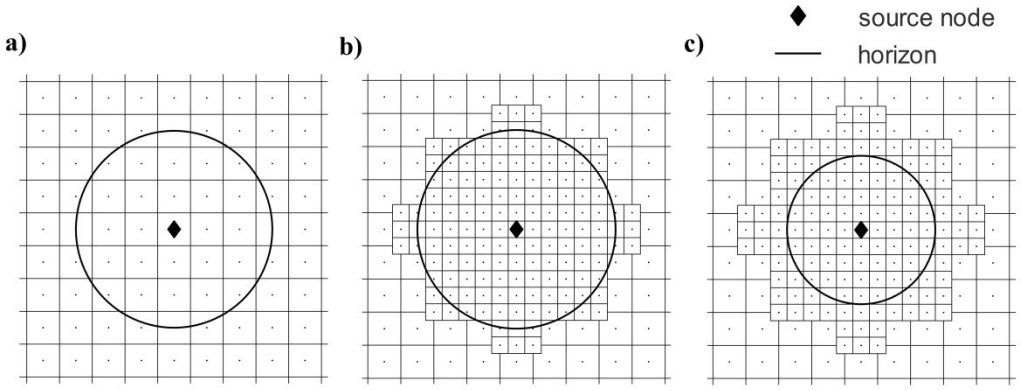


Fig. 3.8: Examples of application of different refinement types: a) node selected by trigger, b) refinement of 1st level obtained by keeping constant the horizon length, c) refinement obtained with simultaneous variation of horizon and *m*.

As stated in Sect. 3.1, different types of numerical convergences of peridynamic theory to the classic theory of mechanics have been originally identified in [57],

which can be implemented by reducing opportunely the horizon length with respect to that one of grid spacing, as shown in Fig. 3.8 for the case of *m*-convergence as well as δm-convergence. As it will be discussed in Sect. 5.3.2, such numerical convergences may be really incisive as well as useful to solve problems related to the grid sensitivity shown by peridynamics when discretized with a uniform grid of nodes. Appendix A gives the skeleton of the pseudo-codes employed to generate the new nodes with the approach B (see Algorithm A.3), a similar algorithm structure is adopted when the approach A is implemented. Fig. 3.9 shows the flow chart of the working principle of Algorithm A.3.

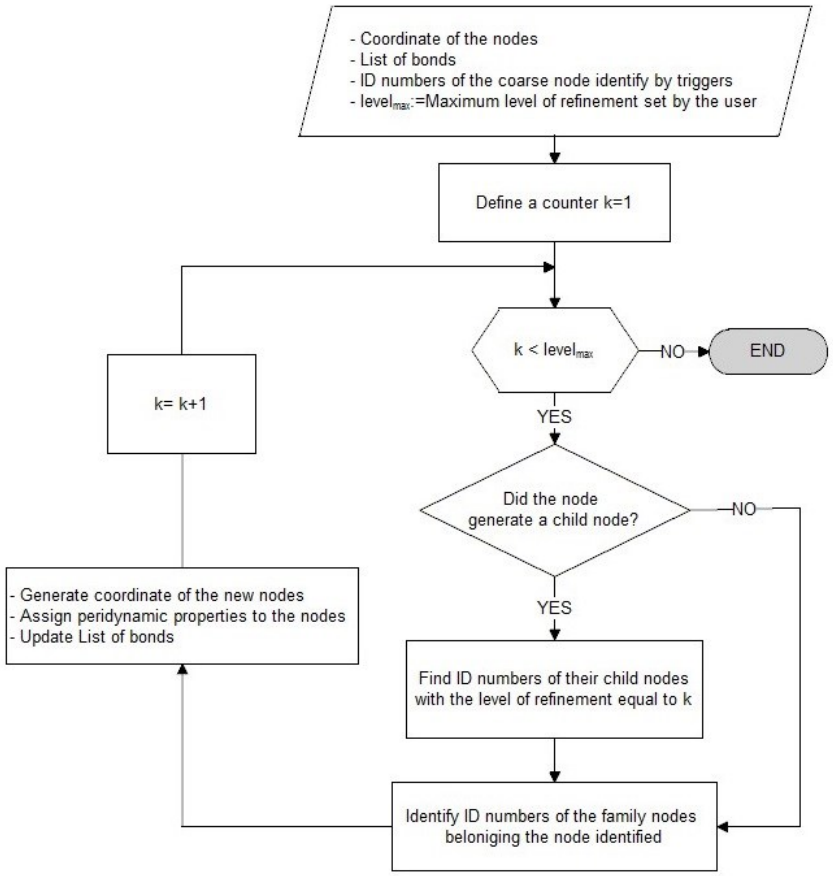


Fig. 3.9: Flow-chart of the algorithm to generate the refinement.

The addition of new nodes in the grid requires the updating of data such as the list of bonds of the model (see Eq. (A.3) in Appendix A) and the volume correction factor matrix (Eq. (A.5) in Appendix A). The simplest way to update them, but inefficient from the computational point of view, it is to apply the required algorithm on the whole grid, updating properties of regions of the grid in which no refinement has been applied. Another approach, which surely allows using the

computational resources in an efficient way, is to restrict the updating of the properties to a window of the just the refined area. Anyway, the former approach is adopted in this work since easier to implement. As it has been introduced in Sect. 3.1, the volume correction factor algorithms allow to compute the approximated weighted volumes and integration point coordinates to be inserted in Eqs. (3.1) for improving the accuracy of the space integration. Such algorithms have been originally developed for a uniform arrangement of the nodes, hence some adaptations have to be applied when a non-uniform grid is adopted. Let us consider the PA-PDLAMMPS algorithm [68], in which only the nodes enclosed inside the horizon define the set of family nodes, it is possible to notice from Fig. 3.10a that some of the interface nodes belonging to the refined region are affected by a loss of volume computed inside the horizon. We suggest as a strategy to reduce this loss of volume defining the horizon length of such nodes as a multiple of the distance between the node and the closest node with the lower level of refinement. It is apparent that such a distance can be different from any grid spacing present in the model, (see Fig. 3.10b), the multiplying factor is the adopted value of m of the coarse grid. Such correction implies an increased horizon for some interface nodes (see Fig. 3.10b), and therefore an increased volume taking part in the computation of the force acting on the node itself. In this way it is possible to drastically reduce distortion phenomena of the energy flux going through a zone with non-uniform grid, this will be described in Sect.4.2.

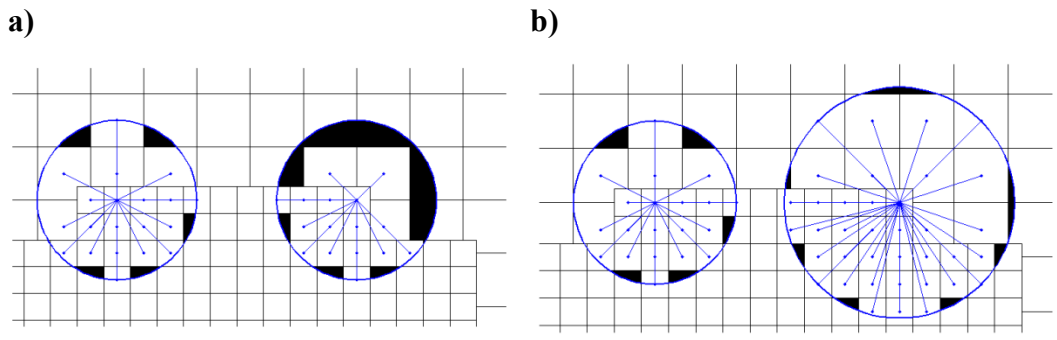


Fig. 3.10: Volume losses affecting interface nodes, the black colour indicates the lost volumes :a) examples of volume losses for constant m, b) examples of volume losses for variable m.

Unlike the PA-PDLAMMPS algorithm, the IPA-Hybrid algorithm does not require such a strategy for reducing the losses of volume since the exact

computation of the partial volume of the family cells is calculated by means of analytical expressions [68].

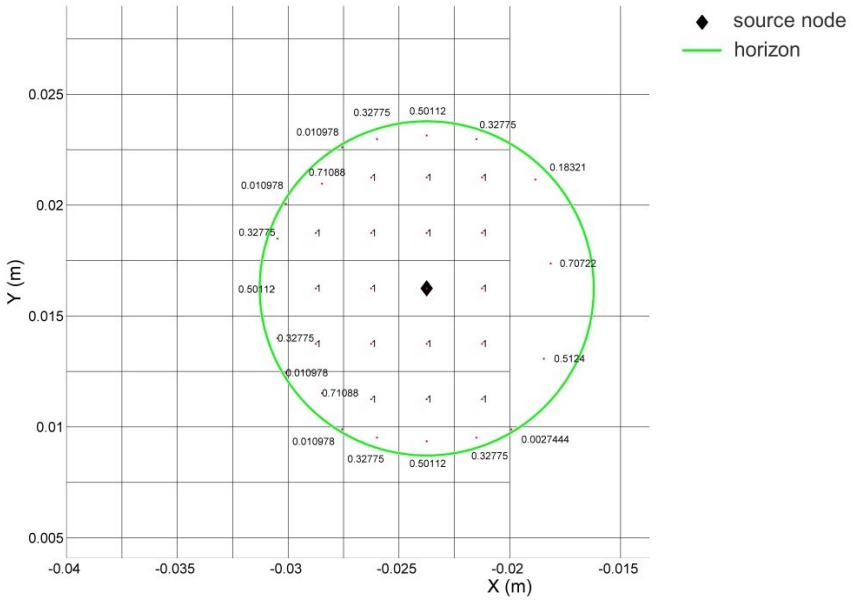


Fig. 3.11: Example of computation of the integration points of family cells with the assigned volume correction factors when the IPA-Hybrid algorithm is adopted in the refined interface regions.

Although the analytical expressions have been devised to be applied in cases in which the nodes are uniformly arranged with a constant grid spacing, they can be easily applied in the regions characterized by a varying grid spacing. This is done by using like parameters in the equations given in [68] the horizon of the source node and then the grid spacing associated to the family nodes taken into account for the computation of its volume correction factor. Fig. 3.11 shows an example about the estimation of the integration point of the family cells and corresponding volume correction factors. Anyway, let us consider that the PA-PDLAMMPS algorithm is computationally more efficient than the IPA-Hybrid algorithm.

3.2.4. Interpolation of physical quantities

After the updating of the properties of new nodes inserted into the grid, physical quantities such as the displacement, velocity and acceleration have to be assigned by the interpolation of the quantities of the nodes belonging to the previous grid. Which nodes should be involved to interpolate such quantities depends on the method of interpolation employed. Matlab software offers several useful functions for interpolating values of a function at specific query points.

Different interpolation methods can been chosen, in particular we adopt the linear interpolation for which four nodes for the 2D case (eight nodes for the 3D one) closest to the query point are used. Knowing the field of the relevant quantities at each time step of the simulation, it is easy to obtain the interpolated values by inserting only the coordinates of the new nodes generated in the input variable of the function. Although this approach is easy and fast to be applied, we point out that such a interpolation offered by Matlab has some limitations when applied in this context. First, its use is restricted to a regular and structured arrangement of the sampling nodes, hence only the initial nodes of the grid can be adopted for interpolating the values. Therefore, in case of an expanding refined region, the interpolated values of the new nodes cannot get the benefits from the higher accuracy of the refined function field close to them. It is clear that such a interpolation can only be applied when the method A of generation of the new nodes is employed, being that the parent nodes of the grid are not removed from the analysis (see Sect. 3.2.3). Another function offered by Matlab which can remove this limitation performs interpolation on scattered data set by using an approach based on the Delaunay triangulation. However, when such a function is used in problems in which symmetrical results are supposed to be obtained, it may be a source of asymmetries on the trend of the interpolated functions due to the random generation of the Delaunay triangulation. Finally, let us consider the fact that when we are treating problems dominated by displacement jumps due to the presence of initial cracks within the domain, we shall ask questions about which set of nodes should be used to approximate the function value when new nodes are added in the proximity of the pre-crack tip. In fact, sometimes it can happen that using the default set of nodes adopted by Matlab functions, some bonds in the proximity of the pre-crack tip may break as a consequence of an uncorrected interpolation, leading the crack to a premature propagation. Such cases could be avoided by introducing a new data structure to select the right set of nodes, which may lead to reduce the computational efficiency.

An interesting interpolation method that can remove all the drawbacks of the interpolation methods offered by Matlab is known as the PIM (Point Interpolation Method), such a method is frequently used for the creation of the shape functions adopted in mesh-free methods [77]. Let $u(\boldsymbol{x})$ be an unknown component of the

displacement field at any point $\boldsymbol{x}$ within the problem domain, then its function value can be expressed as:

$$u(\boldsymbol{x}) = \sum_{i=i}^{k} \phi_i(\boldsymbol{x}) u_i = \boldsymbol{\Phi}(\boldsymbol{x}) \boldsymbol{U}_s \tag{3.5}$$

with k the number of nodes included in a "*local domain*" of the point $\boldsymbol{x}$ defined by the user, $u_i$the known component of the displacement at the ith node in the local domain and $\phi_i(\boldsymbol{x})$ the shape function of the ith node determined using all the nodes included in the local domain. Both the shape functions and the components of the displacement can be collected in the array form $\boldsymbol{\Phi}(\boldsymbol{x})$ and $\boldsymbol{U}_s$, respectively. The number of nodes involved to approximate the function value at the point $\boldsymbol{x}$ is determined from the size of the local domain. Several local domain shapes can be used, i.e. circular and rectangular are most often used, and its dimension can change for different nodes. One of the advantages to use such a method in peridynamics derives from the possibility to adopt the horizon for defining the size of the local domain of interpolation. As a result, the local domain of interpolation coincides with the non-local region $H(\boldsymbol{x})$ introduced by peridynamics. Moreover, as previously stated, in order to avoid the premature failure of the bonds in the proximity of the crack tip, we propose to adopt for simplicity of implementation the truncated local domain, known as visibility criterion in the context of mesh free methods [78]. Basically, the nodes included in the local domain coincide with the current list of family nodes belonging at that point $\boldsymbol{x}$, this information can be easily obtained from the bond history list (see Appendix A). The shape functions $\phi_i(\boldsymbol{x})$ in Eq. (3.5) are built by adopting the PIM method based on polynomial basis functions [77], according to the proposed strategy its formulation starts with the following finite series expression:

$$u^h(\boldsymbol{x}, \boldsymbol{x}_q) = \sum_{i=1}^{k} p_i(\boldsymbol{x}) a_i(\boldsymbol{x}_q) = \boldsymbol{p}^t(\boldsymbol{x}) \boldsymbol{a}(\boldsymbol{x}_q) \tag{3.6}$$

where $u^h(\boldsymbol{x}, \boldsymbol{x}_q)$ represents the approximation of the function $u(\boldsymbol{x})$ using the values of the family nodes enclosed inside the horizon $H(\boldsymbol{x}_q)$ of the point $\boldsymbol{x}_q$, k is the number of family nodes, $a_i(\boldsymbol{x}_q)$ is the coefficient associated with the

monomial $p_i(\boldsymbol{x})$ corresponding to the point $\boldsymbol{x}_q$. Assuming the hypothesis of linear interpolation, the vector $\boldsymbol{p}\,(\boldsymbol{x})$ of the basis function of monomials in a 2D domain is expressed as:

$$\boldsymbol{p}^{\,t}(X,Y) = [1, X, Y] \tag{3.7}$$

Introducing the matrix form of the basis functions $\boldsymbol{P}_Q$called *moment matrix*:

$$\boldsymbol{P}_Q = \begin{bmatrix} 1 & X_1 & Y_1 \\ 1 & X_2 & Y_2 \\ \vdots & \vdots & \vdots \\ 1 & X_k & Y_k \end{bmatrix} \tag{3.8}$$

and collecting the $u\,(\boldsymbol{x}_i)$ of a component of the displacement field in the vector form $\boldsymbol{U}_s$ of all the family nodes:

$$\boldsymbol{U}_S = \begin{bmatrix} u(\boldsymbol{x}_1) \\ u(\boldsymbol{x}_2) \\ \vdots \\ u(\boldsymbol{x}_k) \end{bmatrix} \tag{3.9}$$

then the coefficients collected in the vector form $\boldsymbol{a}$ are calculated by enforcing that Eq. (3.14) be satisfied at the *k* nodes, as follows:

$$\boldsymbol{a} = \boldsymbol{P}_Q{}^{-1}\boldsymbol{U}_s \tag{3.10}$$

assuming that the inverse $\boldsymbol{P}_Q$ exists, see [77] for a detailed explanation of the conditions for which the moment matrix is not invertible. Therefore, if the Eq. (3.10) is substituted into Eq. (3.6), we obtain:

$$u^h(\boldsymbol{x}, \boldsymbol{x}_q) = \boldsymbol{p}^t(\boldsymbol{x})\boldsymbol{P}_Q{}^{-1}\boldsymbol{U}_s = \boldsymbol{\Phi}(\boldsymbol{x})\boldsymbol{U}_s \tag{3.11}$$

The code used for implementing the PIM in order to interpolate the nodal displacement values in the case of a 2D domain is given in Appendix A (see Algorithm A.4). The same procedure is applied to interpolate the velocity and acceleration values.

Finally, we remark that specific tests have not been carried out to check if the interpolated quantities satisfy the conservation of quantities such as the kinetic energy, linear and angular momentum. Such tests shall be performed in the future application of AGRS.

4. Tests to evaluate the performance of a non-uniform grid

As stated in Sections 2.3-2.4, the use of a varying horizon size and grid spacing may introduce some anomalies in the displacement field of the PD numerical solution. The possible reasons of such anomalies can be essentially identified in the following points:

- the numerical noise introduced by the change of grid spacing over the discretized domain as a result of the change of space integration accuracy
- effects on PD solution introduced by the loss of bonds through the transition region where the horizon varies (see Section 2.3), then the presence of ghost forces which affect the local equilibrium through that region (see Section 2.4)
- the change of dispersion response of the material due to the change of non-locality (change of the horizon size) over the domain.

It has been well known for a long time that when elastic waves cross regions discretized with elements of different size in FEM meshes [75, 76], spurious reflections can occur due to the numerical errors and noise which affect the transition zone. Such a phenomenon was also observed in non-uniform peridynamic grids in 1D/2D problems [57, 58]. Let us consider that, unlike the methods based on classical mechanics, which are devoid of an intrinsic length-scale, the change of the non-locality over the domain introduces an additional error in numerical PD solutions, especially in the case of dynamic simulations. Taking into account that the dispersion of the material is related to the size of the horizon length [3], the change of the horizon length leads to have regions characterized by a different dispersion law of elastic waves [79, 80]. Strategies to mitigate the spurious wave reflections introduced by the change of non-locality over the domain are proposed in [81]. This chapter aims to provide an estimation of the numerical errors introduced by the use of a non-uniform and scaling grid in PD solutions. In particular, we are interested to answer the following questions:

1. how much do the ghost forces influence the error in numerical PD solution?
2. in the absence of ghost forces, is the solution still affected by the presence of anomalies in the displacement field?
3. do such anomalies affect the PD solution in the regions far from the transition zone?
4. How much do the refinement and scaling affect the propagation of elastic waves?

For answering these issues, static analyses are first performed by comparing the 2D numerical PD solution with the analytical solution of an elastic linear problem. By static analyses, the additional effects introduced by dispersion of the model can be avoided, allowing to better understand the origin of such anomalies introduced by the refinement and scaling. Then, dynamic analyses concerning the propagation of a continuum elastic wave in a 2D plate are performed in order to evaluate the spurious waves and wave distortions generated by the refinement and scaling.

4.1. Static analyses

A 2D static example is addressed through the comparison of the numerical peridynamic solution with the analytical solution of a linear elastic problem. The same problem has been proposed in [66], using a uniform grid of nodes and constant horizon. The model (see Fig. 4.1) is a rectangular plate of size 1x0.5 m^2 and thickness t = 0.01 m subjected to a uniform uniaxial tension of magnitude p = 200 MPa. The mechanical properties of the material are: E = 200 GPa, ν = 1/3 (plane stress case). The numerical solutions are obtained by using the linearized formulation of the equation of motion of BB-PD (see Eq. (2.19)), while the static solution is numerically solved by means of the stiffness matrix, see in [82] for a detailed explanation of the method adopted for its assembly. The spatial integration is carried out using the IPA-Hybrid algorithm (see Section 3.1) for computing the volume correction factor. In order to investigate the anomalies introduced by the use of a non-uniform grid of nodes and a varying horizon size in numerical peridynamic solution, we analyze several models which differ in the configuration of the refined region. As a first step, only the left side of the model

(see Fig. 4.1) is refined by keeping constant the m ratio and reducing the horizon according to the δ-convergence strategy (see Section 3.1), the grid spacing is reduced according to Eq. (3.3),

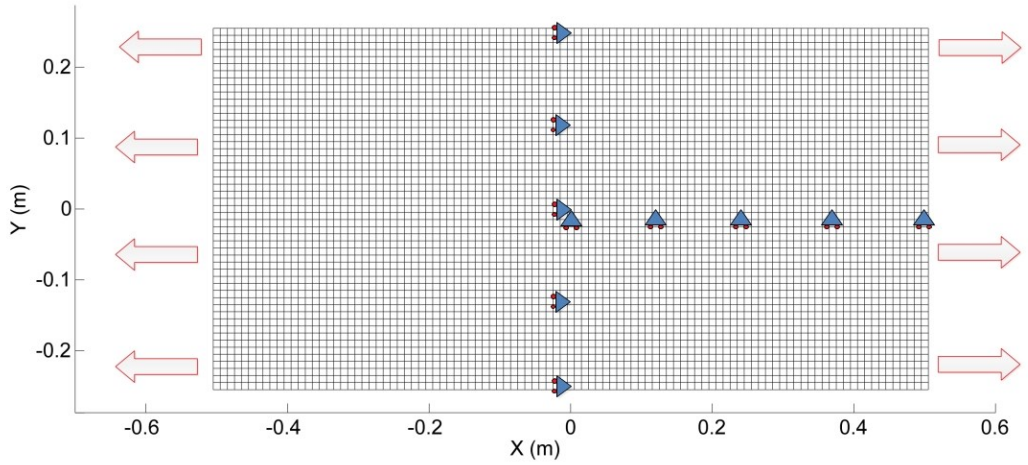


Fig. 4.1: Setup of the problem, uniform model of the plate subjected to a uniform tension, the triangle markers indicates the support constraints.

The load is applied to one only row of nodes, while Fig. 4.1 shows how the boundary conditions are set along the symmetry axes of the uniform coarse model. In particolar, the support constraints are set along the horizontal line to restrain u_Y and along the vertical line to restrain u_X. We can notice from Fig. 4.1 that only the X axis on the right side of the model is constrained due to the lack of nodes on the refined side. This set of constraints has been chosen in order to keep the ill-conditioning of the stiffness matrix to a minimum. The method B (see Section 3.2.3) is adopted as a method of generation of the nodes.

The initial uniform grid has the following peridynamic parameters: ΔX_0 = 10 mm, δ_0 = 30.15 mm and consequently m = 3.015. Three models are implemented, for sake of clarity they will be called model A, B and C, in which the first, second and third level of refinement and scaling are implemented, respectively. In Fig. 4.2 the models are shown, while Table 4.1 provides the values of the main peridynamics parameters adopted. The refined models have been obtained applying the visibility criterion introduced in [57] (as explained in Sect.3.2.3), as shown from the close-up view of the transition zone in Fig. 4.2c.

Model	Level of refinement	$\Delta X = \Delta Y$ (mm)	δ (mm)
A	1st	5.00	15.075
B	2nd	2.50	7.537
C	3rd	1.25	3.768

Table 4.1: Values of the peridynamic parameters regarding the model A, B and C.

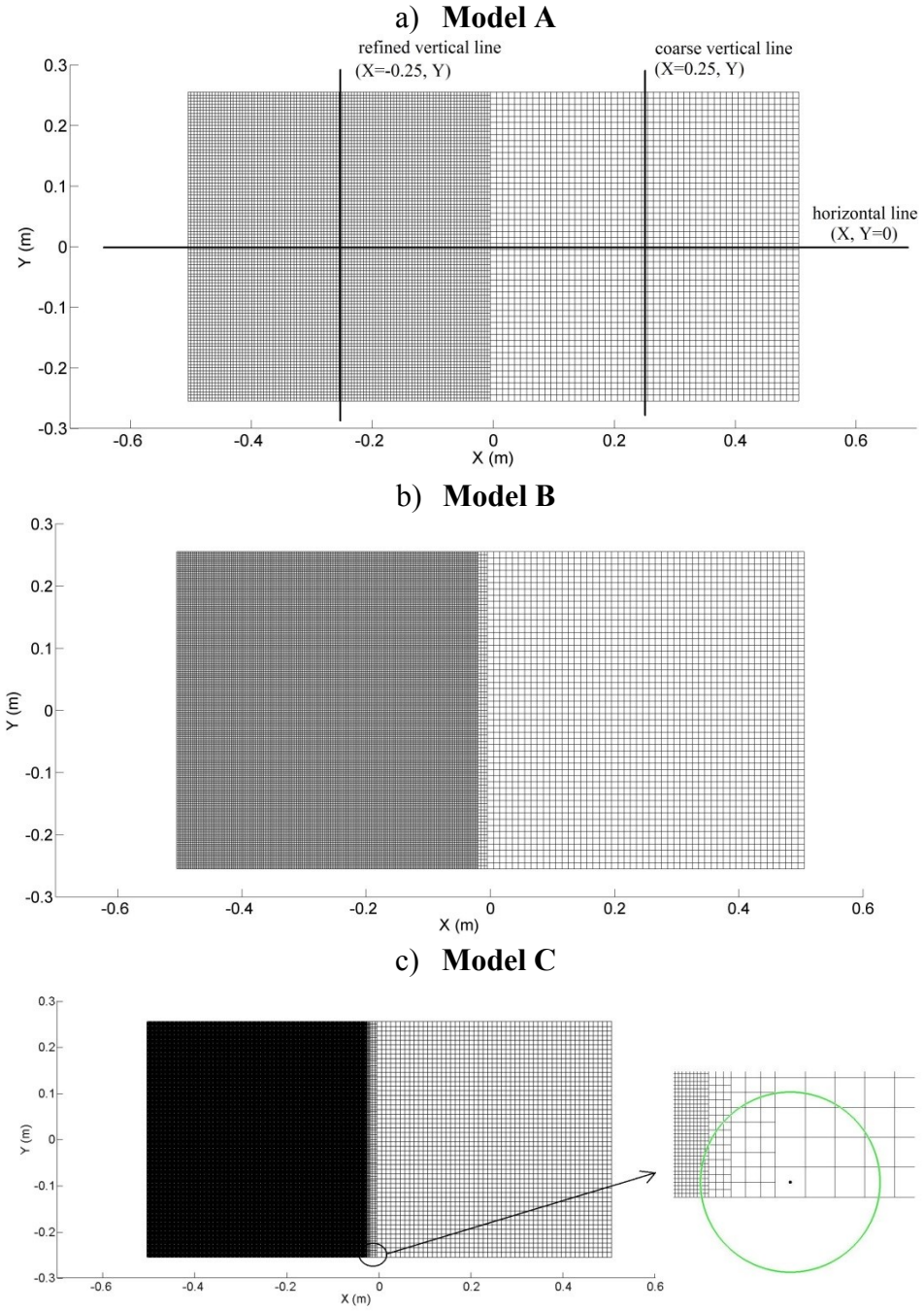


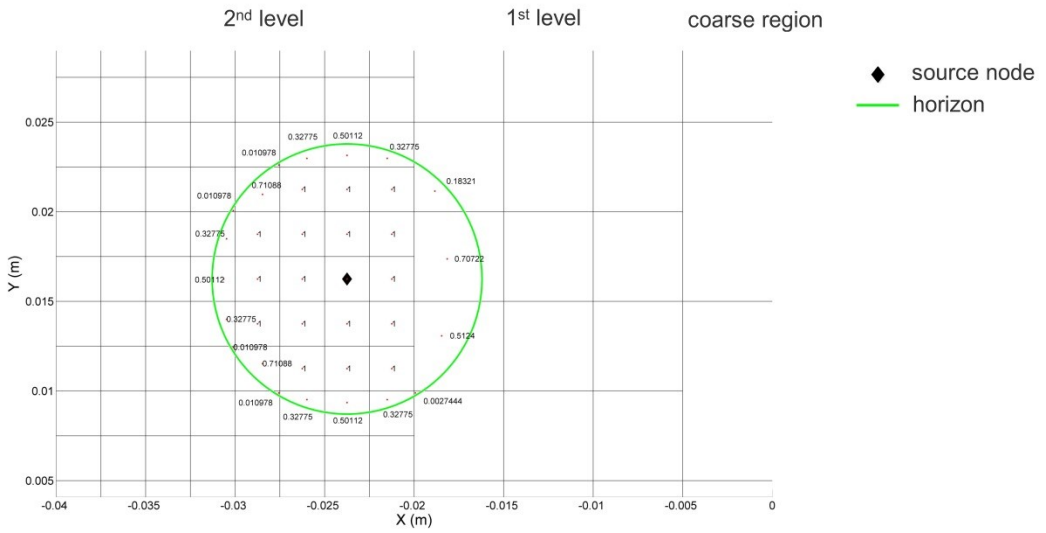
Fig. 4.2: Non-uniform grids obtained by applying the refinement and scaling on the left half of the uniform model: a) Model A with the 1st level of refinement, b) Model B with the 2nd level of refinement, c) Model C with the 3rd level of refinement and a close-up view of the interface zone to show the effect of the visibility criterion.

As a consequence, the coarse particles can interact with the particles belonging to the most refined region. As a first step, we are interested to compare the numerical solutions obtained by adopting both the scaling & DH (see Eq.(3.1c)) and scaling PD formulations (see Eq. (3.1b)) with the analytical solution given by means of the following expressions:

$$u_X(X,Y) = \frac{p}{E}X \tag{4.1a}$$

$$u_y(X,Y) = -\upsilon\frac{p}{E}Y \tag{4.1b}$$

a)



b)

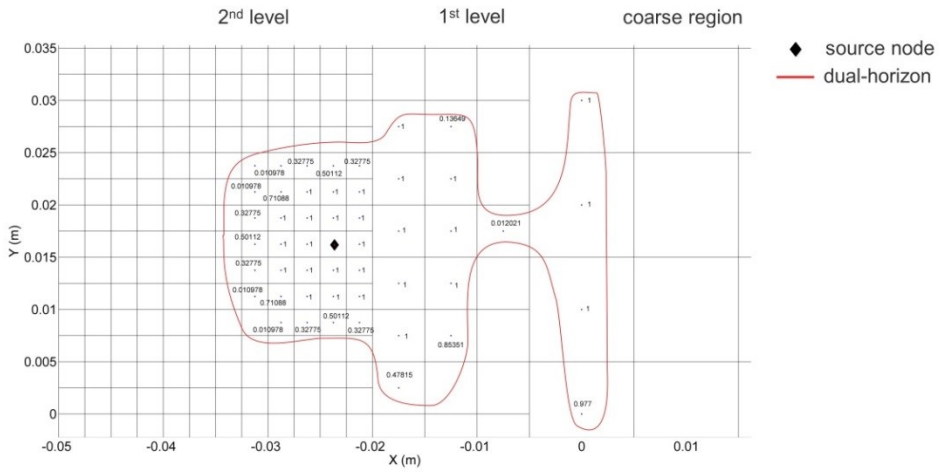


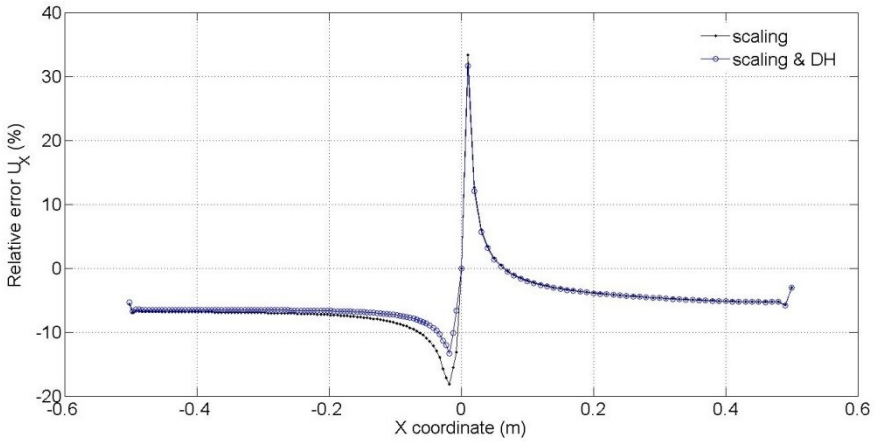
Fig. 4.3: Close-up views of a portion of Model B: a node located in the most refined region and close to the interface zone, a) nodes belonging to its horizon, b) nodes belonging to its dual-horizon. (The numbers close to the particles indicate the volume correction factors).

As stated in Section 2.4, we want to highlight that unlike the non-local region $H(\boldsymbol{x})$ defined with a specific shape and dimension, the dual-horizon is only a set of nodes with no predefined shape whose dimension depends on the number of nodes enclosed in it, i.e. Fig. 4.3 shows the sets of nodes belonging to the horizon and dual-horizon of a refined node with the associated volume correction factors. As shown in Eqs. (3.1c), the volume correction factors are those associated to the portion of source cell covered by the horizon of its dual-family nodes.

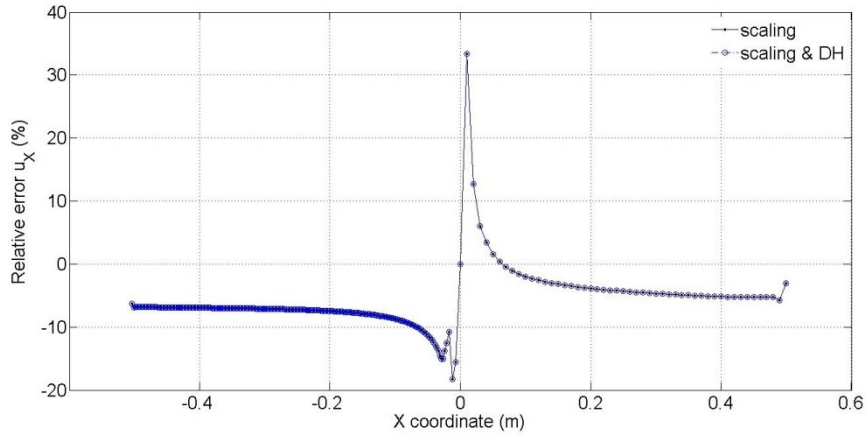
Fig. 4.4 and Fig**.** 4.5 show the graphs of the relative error of the displacements of the PD solution evaluated along the lines displayed in Fig. 4.2a. It can be noticed from figures that the error has a null value at coordinate X=Y=0, being that the displacements are clamped (see Fig. 4.1). The solutions shown in the graphs reveal important aspects that have to be discussed: although Fig. 4.4a (model A) shows that the application of the scaling & DH formulation leads to slightly improve the accuracy of the numerical solution, the graphs of Fig. 4.4b and Fig. 4.4c show a completely opposite trend. In fact, the curves appear to be almost overlapped. Such a behaviour suggests that the anomalies on the displacement field introduced by varying the size of the horizon may not be due to the presence of ghost forces but to an improper scaling formulation. Consider that the scaling approach, initially proposed in [57] for the BB-PD and then formulated later in [67] for the general peridynamic formulation, computes the micromodulus of the bonds using Eq. 2.13. This equation has been originally formulated assuming implicitly that the horizon is constant over the domain. It is clear that such an issue should be further investigated.

Comparing the trend of the displacement field u_X of the models obtained with the scaling & DH formulation (Fig. 4.4), it is possible to notice that the anomalies introduced by the refinement and scaling seem to be confined into the interface region. In particular, the highest numerical relative error of about 31.5 % takes place in the coarse interface zone regardless of the level of refinement applied. Such a trend suggests that the numerical error in the interface region is not so sensitive to the level of refinement and scaling applied.

a) **Model A**



b) **Model B**



c) **Model C**

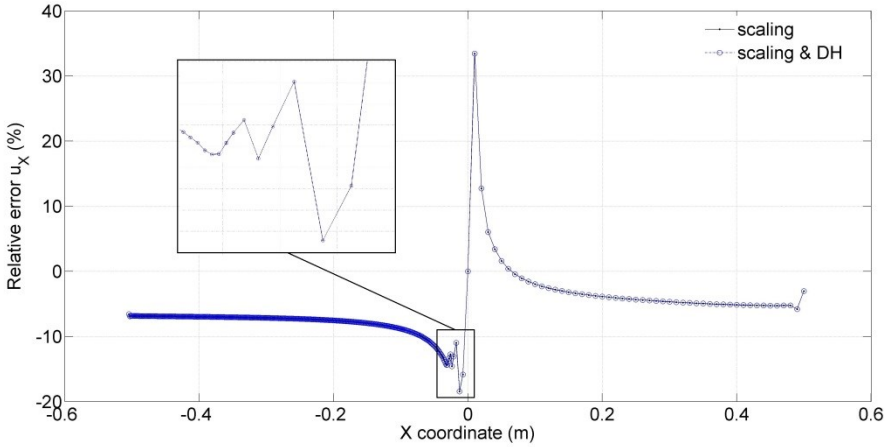
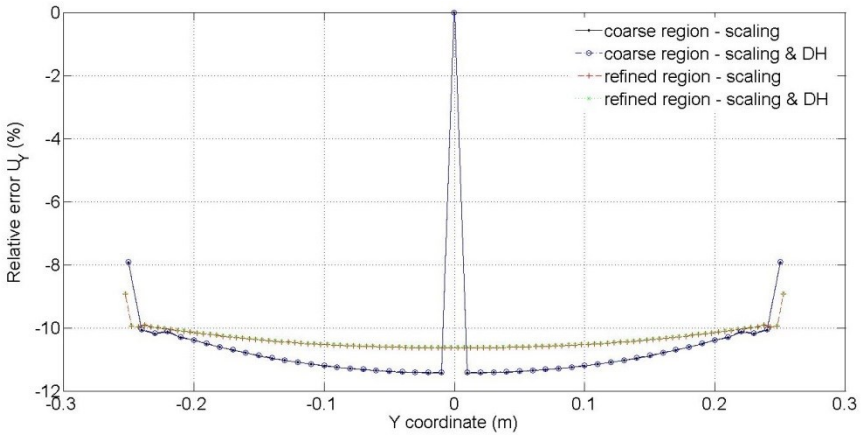
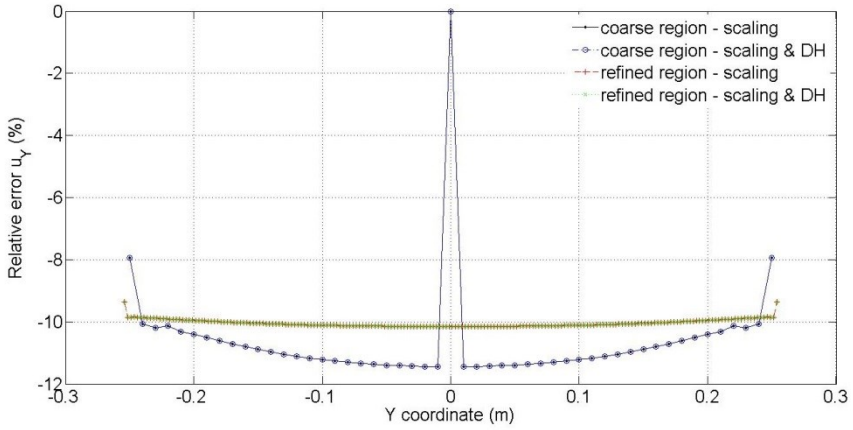


Fig. 4.4: Relative error of the displacements evaluated along the horizontal line: a) model A with the 1st level of refinement, b) model B with the 2nd level of refinement, c) model C with the 3rd level of refinement.

a) **Model A**



b) **Model B**



c) **Model C**

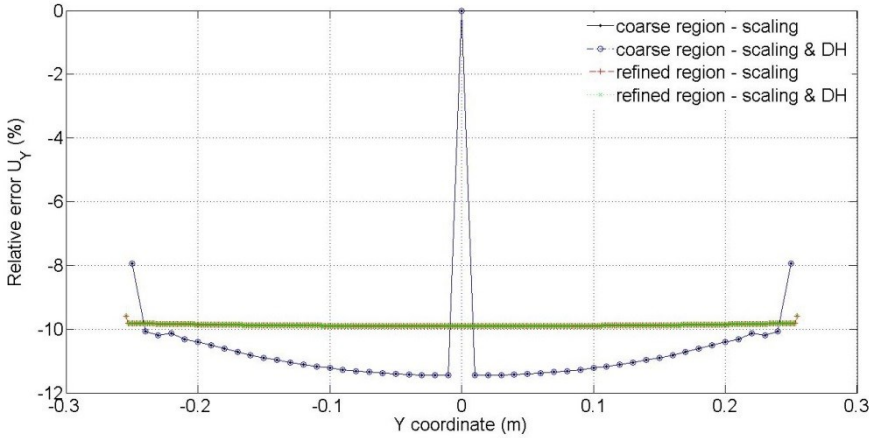


Fig. 4.5: Relative error of the displacements evaluated along the vertical lines: a) model A with the 1st level of refinement, b) model B with the 2nd level of refinement, c) model C with the 3rd level of refinement.

Another issue concerns the results evaluated far from the the boundary of the different peridynamic regions. The solution in the refined region should be supposed to have a higher level of accuracy than the solution of the coarse region. For instance, regarding u_X, the average relative error of the plateau trend on the refined side is about 7%, which is higher than the avarage error of about 5% on the coarse side.

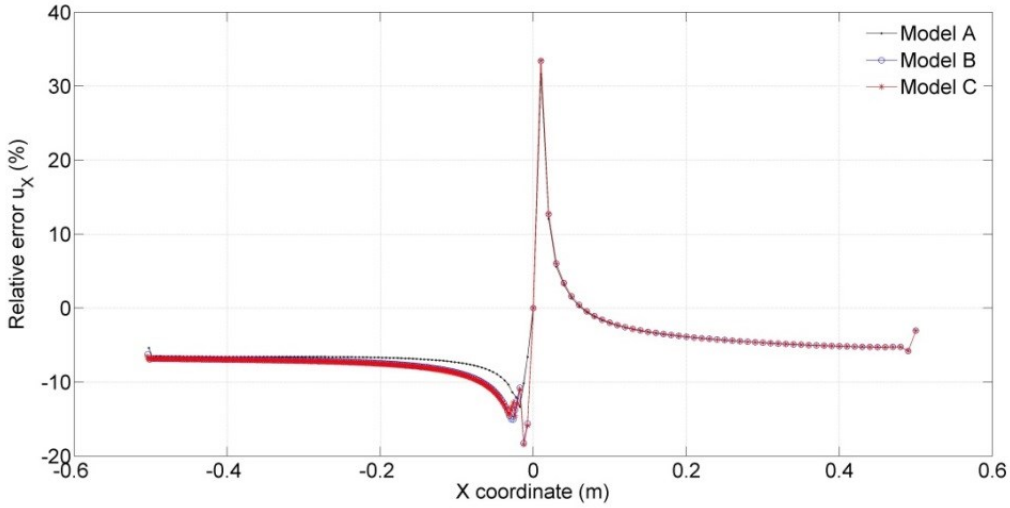


Fig. 4.6: Comparison of the relative error of the u_x displacements component with the scaling & DH formulation evaluated along the horizontal line of the three models.

The reason of that can be explained as follows: PD grids suffer from surface effect (see Section 2.2) which leads to have material softening in the regions close to the boundaries of the domain, the skin effect is worse if the horizon size is bigger. As shown in [58], the skin effect does not only affect the deformation in the proximity of the boundaries, but also in the bulk material. Taking into account that the forces between the refined and coarse sides are spread through the central vertical line (where u_X are clamped) as a consequence of the non-local nature of PD theory, the deformations in the different regions are mutually affected by the skin effect. As shown in Fig. 4.7, this effect can be reduced by the application of the surface correction factor. By adopting the surface correction factor, the average error on the refined side is 1.5%, lower than 2% on the coarse region.

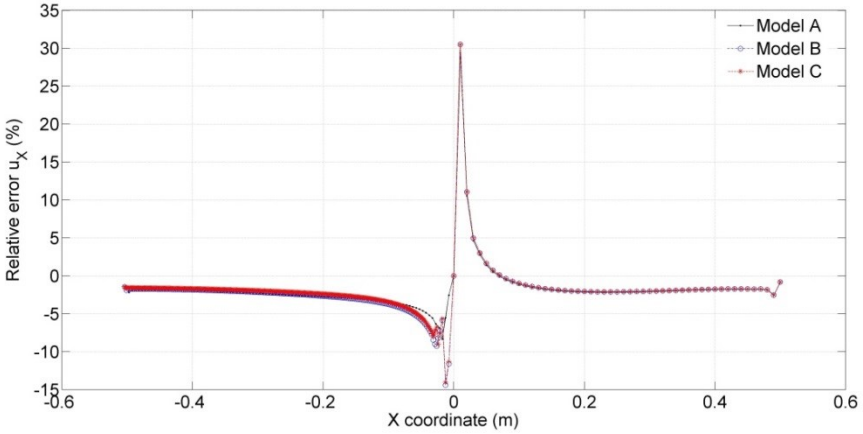


Fig. 4.7: Comparison of the relative error of the u_x displacements component evaluated with the scaling & DH formulation and surface correction factor along the central horizontal line of the three models.

In order to evaluate the sensitivity of PD solutions with respect to the *m* ratio, other analyses are performed by applying the *m*-convergence strategy (see Section 3.1). The models are built refining the grid and keeping constant the horizon lengths of the models implemented with *m* =3.015 (see Table 4.1 for the horizon lengths). As a result, the *m*-ratio increase from 1.515 to 5.015, Fig. 4.8 and Fig. 4.10 show that the increase of the *m*-ratio takes the numerical PD solution to converge to the analytical solution given in Eq. (4.1). However, the anomalies in the proximity of the interface region still persist. The L_2 error norm of the displacement components is computed to measure the average difference between the PD and the analytical solutions, the general expression is given as:

$$L_2\ error = \frac{1}{q}\frac{\sqrt{\sum_{i=1}^{j}(u_i^{PD} - u_i^{analytical})^2}}{\sqrt{\sum_{i=1}^{j}(u_i^{analytical})^2}} \tag{4.2}$$

where *j*=1,2 (2D cases) indicates the component *X* and *Y* of the displacement field and *q* the number of nodes involved to calculate it; Fig. 4.11 compares the convergence rates of L_2 error of the displacement components evaluated along some rows of nodes located in both the refined and coarse regions (as specified in Fig. 4.11). Although the average error of u_X in the coarse region is lower than the error in the refined region, we can note that the rate of convergence of the error in the refined region is $2.2 \cdot 10^{-2}$ which is one order of magnitude higher than the rate $0.3 \cdot 10^{-2}$ in the coarse region. Besides, Fig. 4.11 shows that unlike the linear rate of convergence of u_X, the rate of convergence of u_Y is non-linear.

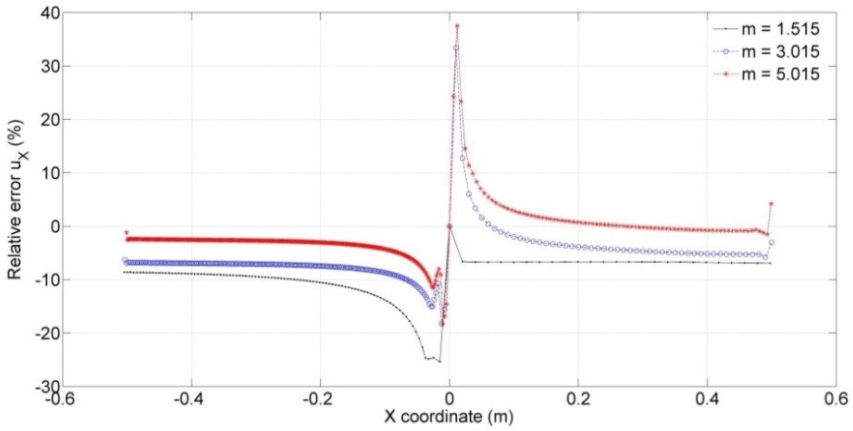


Fig. 4.8: Sensitivity study on *m* ratio of the relative error of u_x evaluated along the horizontal line for the model B.

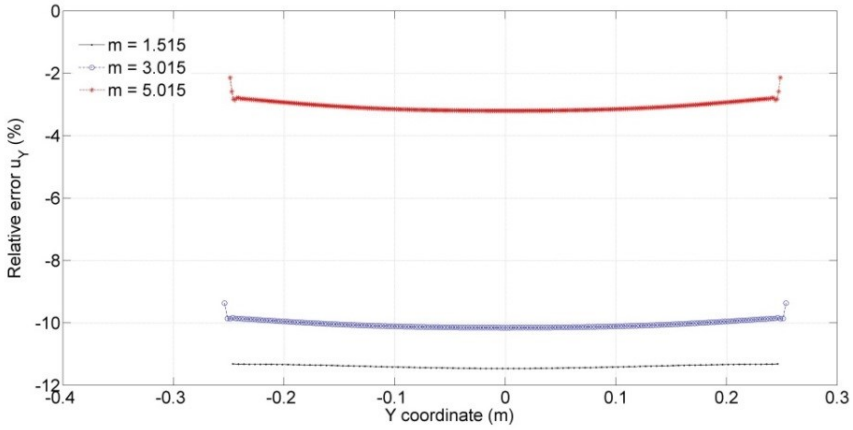


Fig. 4.9: Sensitivity study on *m* ratio of the relative error of u_y evaluated along the refined vertical line of the model B.

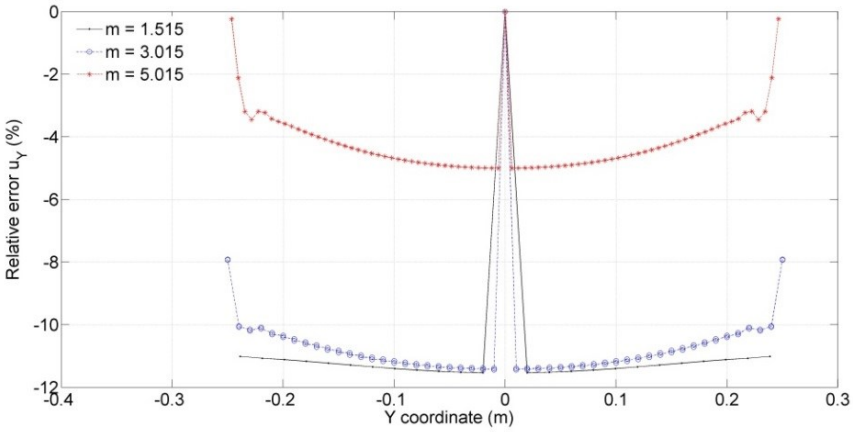


Fig. 4.10: Sensitivity study on *m* ratio of the relative error of u_y evaluated along the coarse vertical line of the model B.

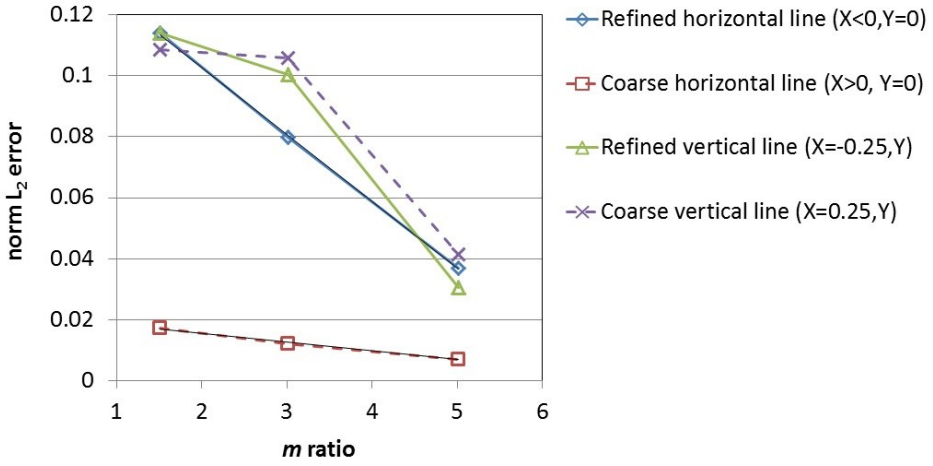


Fig. 4.11: Convergence rates of L_2 error between analytical solution and PD solution for u_X and u_Y evaluated along the specified sample lines.

So far we have analysed the case of refinement and scaling applied on the left side of the coarse model in order to investigate the effect of the mutual influence between the coarse and refined regions on the PD solution. Another configuration that may give us a useful understanding of the influence of refinement and scaling on PD solutions concerns its application on the upper side of the plate. As shown in Fig. 4.13, three models called models D, E and F have been built using the same set of peridynamic parameters adopted in the cases analysed previously (see Table 4.1). The boundary conditions are always set along the symmetry axes of the coarse region with the same strategy proposed in the previous model, see Fig. 4.12.

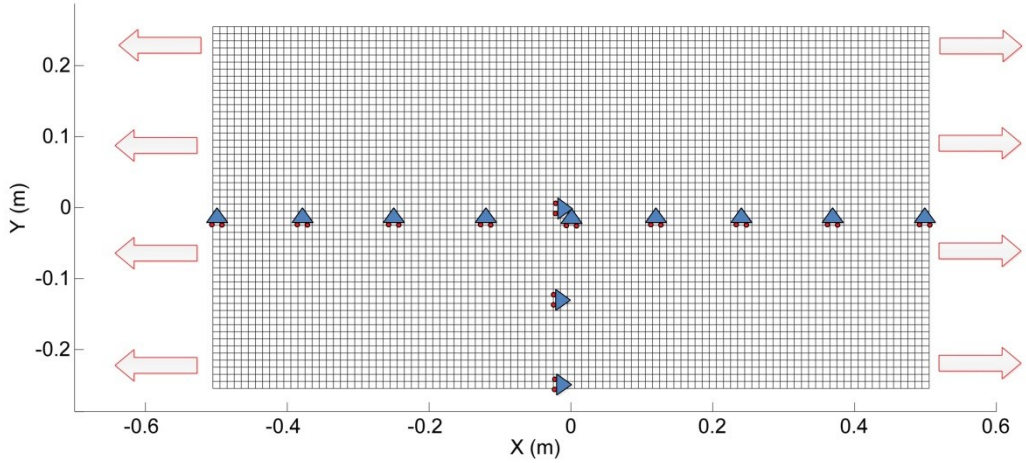


Fig. 4.12: Setup of the problem, uniform model of the plate subjected to a uniform tension, the triangle markers indicates the support constraints.

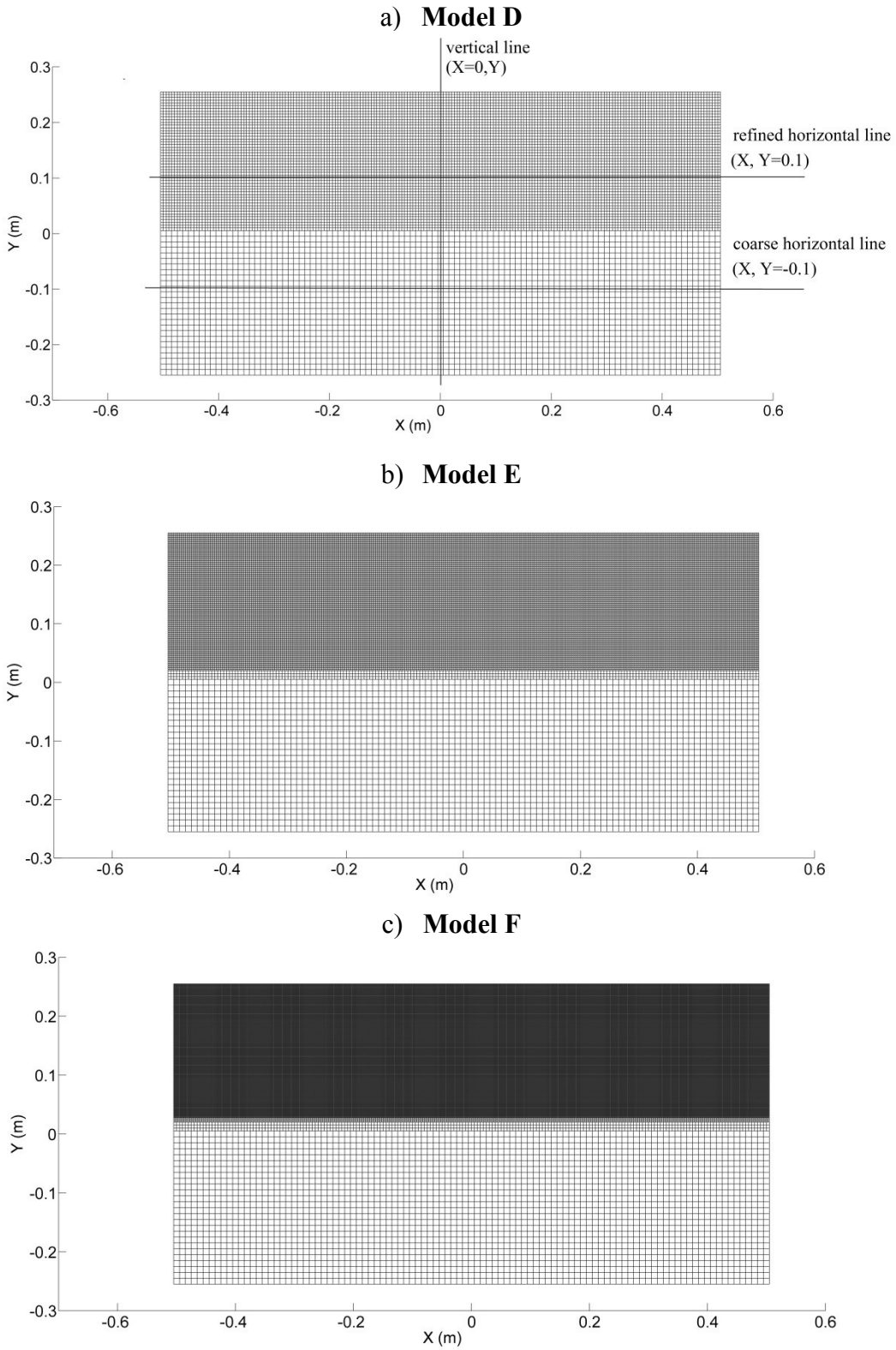


Fig. 4.13: Non-uniform grids obtained by applying the refinement and scaling on the upper half of the model: a) model D with the 1st level of refinement, b) model E with the 2nd level of refinement, c) model F with the 3rd level of refinement.

The PD solutions are compared with the analytical solutions given in Eqs. (4.1): Fig. 4.14, Fig. 4.15 and Fig. 4.16 display the relative error of u_X and u_Y evaluated along the lines shown in Fig. 4.13a. Fig. 4.16 shows that the anomalies in the displacement field still persist, with a maximum relative error of about 15% observed on the side of the refined region.

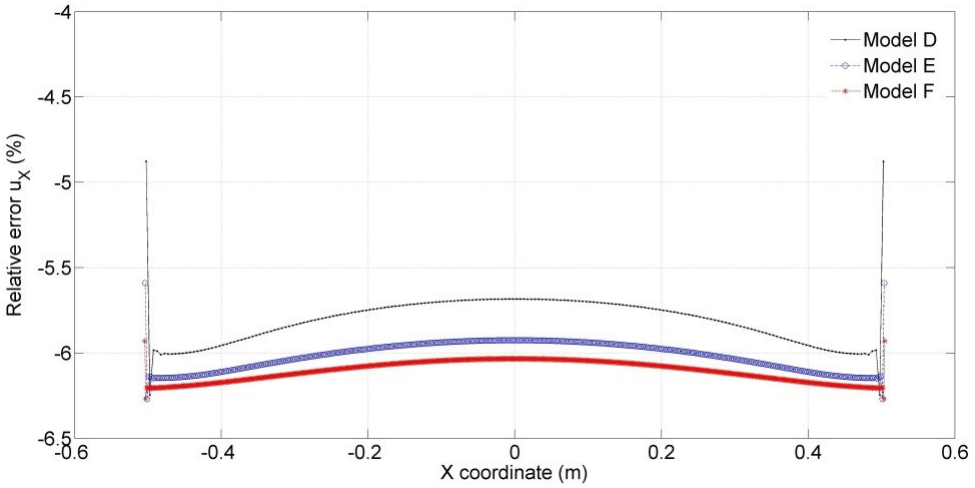


Fig. 4.14: Relative error of the u_X displacement component evaluated along the refined horizontal line.

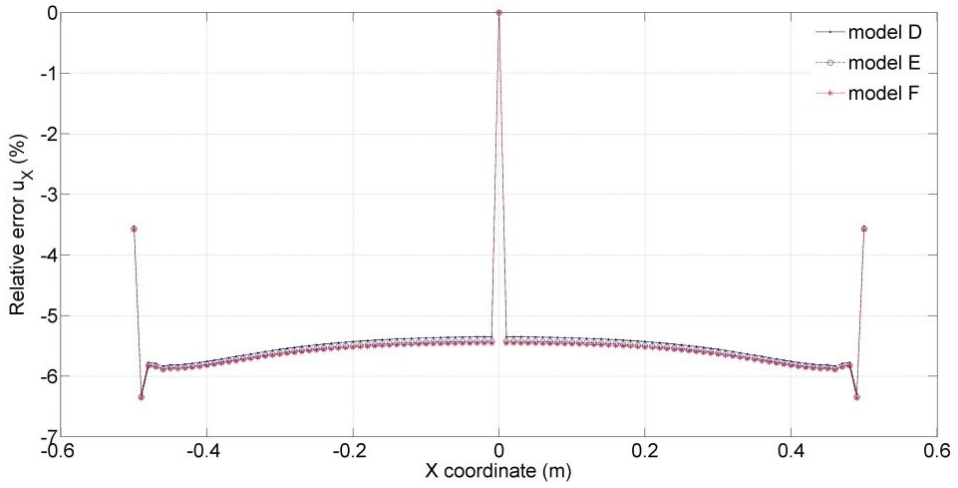


Fig. 4.15: Relative error of the u_X displacement component evaluated along the coarse horizontal line.

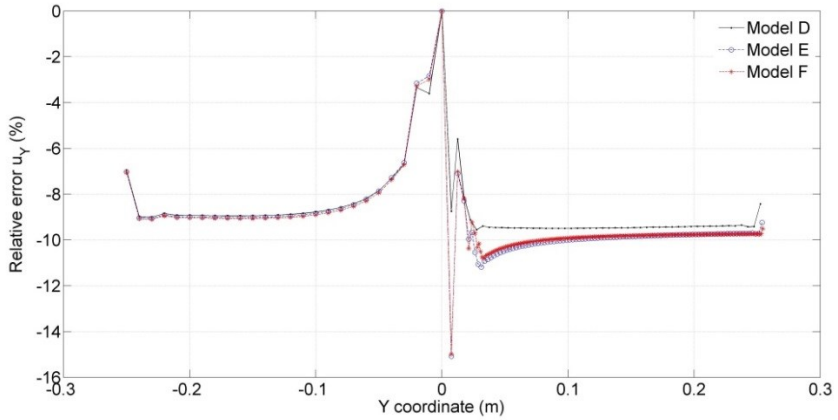


Fig. 4.16: Relative error of the u_y displacement component evaluated along the vertical line.

By carrying out a convergence study on *m*-ratio parameter with the same strategy adopted in the previous case, see Fig. 4.19, we can observe that the maximum error of the u_Y changes from a value of 23% with m = 1.515 to a value of 31% with m = 5.015. In addition, the side of the grid in which there is the maximum error changes depending on the m parameter. It is clear that this strange behaviour should be investigated further. Anyway, the trends of the convergence rates shown in Fig. 4.20 reveal that the PD solution in the refined region converges to the analytical solution faster than in the coarse region, having the former a rate of $3.6 \cdot 10^{-2}$ in comparison with the rate of $5.8 \cdot 10^{-3}$. Finally, Fig. 4.21 shows u_X displacements of the models along a vertical line of coordinate (X=0.20, Y). We can observe that, with this model configuration, the mutual interaction between the two regions of the grid results in a small in-plane bending of the plate, being the value of u_X in the coarse region higher than that in the refined region.

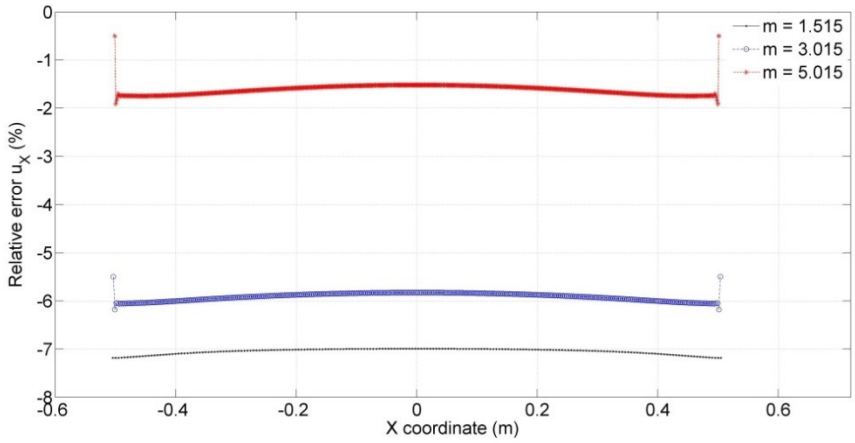


Fig. 4.17: Sensitivity study on *m* ratio of the relative error of u_x displacements evaluated along the refined horizontal line for the model E.

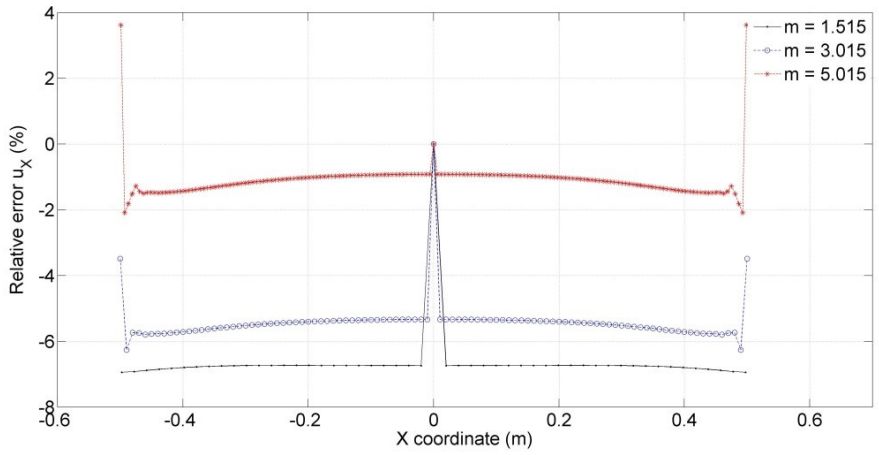


Fig. 4.18: Sensitivity study on *m* ratio of the relative error of u_x displacements evaluated along the coarse horizontal for the model E.

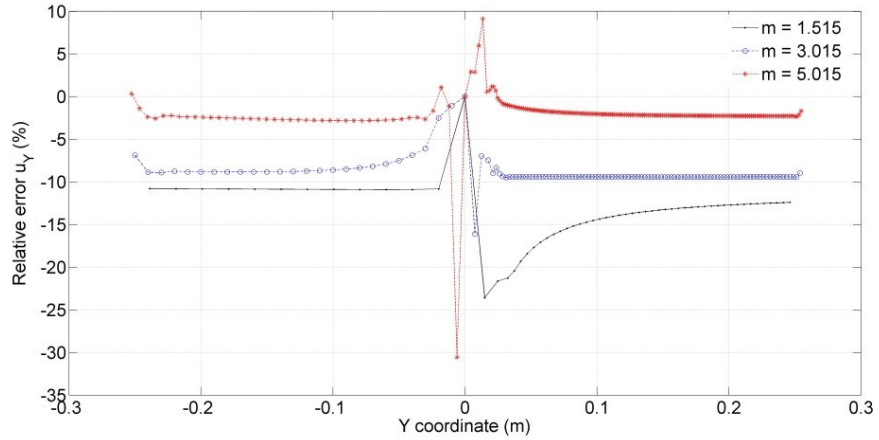


Fig. 4.19: Sensitivity study on *m* ratio of the relative error of u_Y displacements evaluated along the vertical line for the model E.

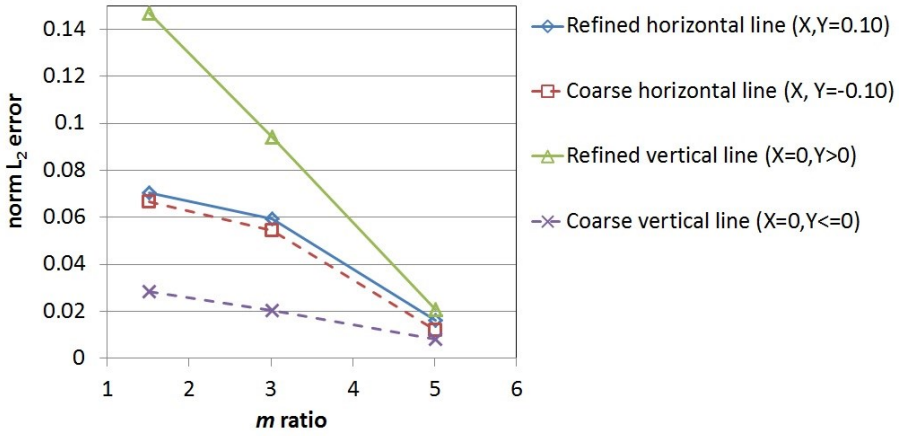


Fig. 4.20: Convergence rates of L_2 error for u_X and u_Y evaluated along the specified sample lines.

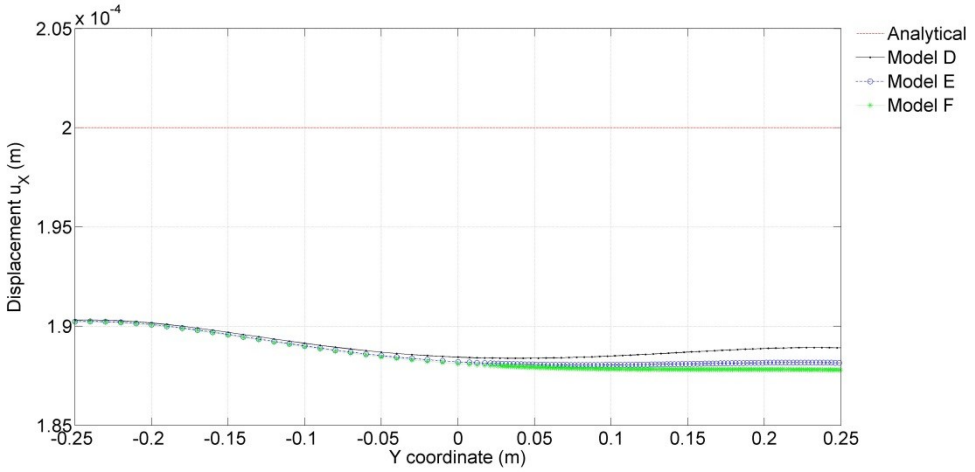


Fig. 4.21: Comparison between the analytical and numerical u_X displacements along the vertical line of coordinates ($X = 0.2$, Y) for the implemented models, $m = 3.015$.

The results of the static simulations carried out in this section reveal important aspects that should be further investigated with other numerical examples. The first interesting aspect concerns the effect of the ghost forces in PD solutions. The same order of magnitude of the error in the displacement field can be observed, regardless of the presence of the ghost forces in the interface zone. Let us remind that the ghost forces are originated by the disequilibrium of the equation of motion imposed at the nodes with a variable horizon [60, 67]. These results suggest that such anomalies are likely due to the scaling formulation, namely the peridynamic properties assigned to the bonds such as the micromodulus. These properties are calculated by means of equations (see Eq. (2.13)) derived from the continuum formulation of peridynamic which implicitly assumes that the horizon is constant over the domain.

Another interesting aspect concerns the magnitude of the error on the coarse side of the interface zone, which seems to be independent of the level of refinement adopted in the grid. Finally, although the results show that the anomalies introduced by refinement and scaling are confined into the interface region, the mutual interaction between regions characterized by a different surface effect can lead to unexpected results in the refined region. The problem can be solved by adopting strategies based on the surface correction factor, such as that adopted in this section (see [66] for further details of this strategy).

4.2. Dynamic analyses

We investigate the effects of using a non-uniform grid on the propagation of a (initially plane) Gauss wave which propagates in a 2D model of a plate. As a first step, we treat the problem by only considering the refinement of first level applied to a model of rectangular plate with size 0.1×0.04 m^2 (see Fig. 4.22a) in which all edges are free, while the assigned mechanical properties are: Young modulus E = 1 and mass density ρ = 1, the micromodulus is computed according to Eq. (2.13b) (υ = 1/3 plane stress case). The solution is obtained by means of Eq. (3.1b) of the linearized Eq. (2.19) solved in time with the Velocity-Verlet scheme (see Eq. (3.2)), having used the PA-PDLAMMPS as algorithm for the space integration. The initial shape of the wave is given by the following expression:

$$u_0(X,Y) = 0.2exp\left[-\left(\frac{X}{0.01}\right)^2\right] \quad , \quad v_0(X,Y) = 0 \tag{4.1}$$

u is the displacement in the X direction and v in the Y direction. In order to reduce the non-local dispersion, the wave length adopted is much larger than the horizon used in the models [57]. Three different models (see Fig. 4.23) are considered with the aim of evaluating the effect of the non-uniform grid on the propagation of the wave, Table 4.2 provides the values of grid spacing and horizon: model A has a constant horizon, model B a constant m ratio, in model C the horizon of interface nodes can have different dimensions (see Sect. 3.2.3). The original grid is characterised by m = 3, in the central zone of the plate a limited area is refined as shown in Fig. 4.22. Solutions obtained with the three different models are compared with the solution provided by the model with uniform coarse grid. The time integration is sufficient for the wave to cross the refined zone, but it is interrupted before reflected waves are produced by the free edges. A first comparison is carried out between the longitudinal profiles of the potential energy density, see Eq. 2.14, at time t = 0.0582 s when the pick of potential energy is in the refined zone, as shown in Fig. 4.24. Fig. 4.22b shows the lines along which the profiles of the potential energy density are recorded.

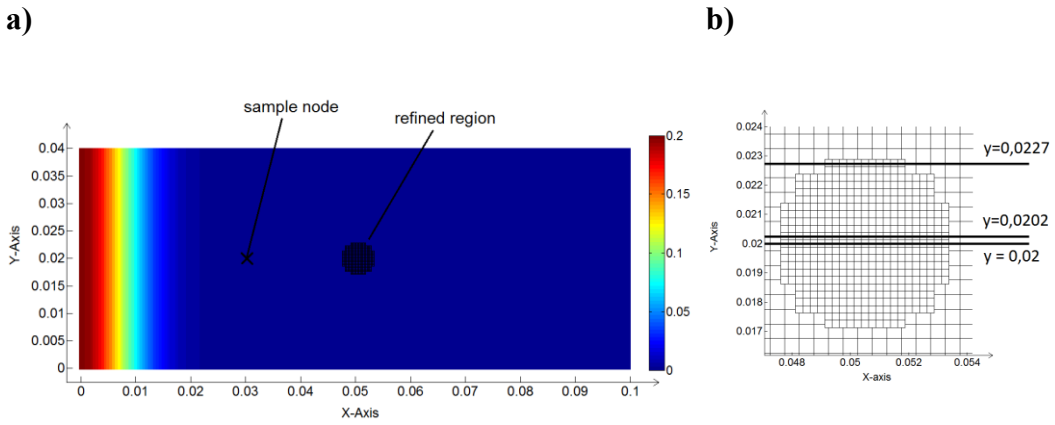


Fig. 4.22: 2D plate model in which a contour plot of the initial displacement field with indication of the refined region is shown.

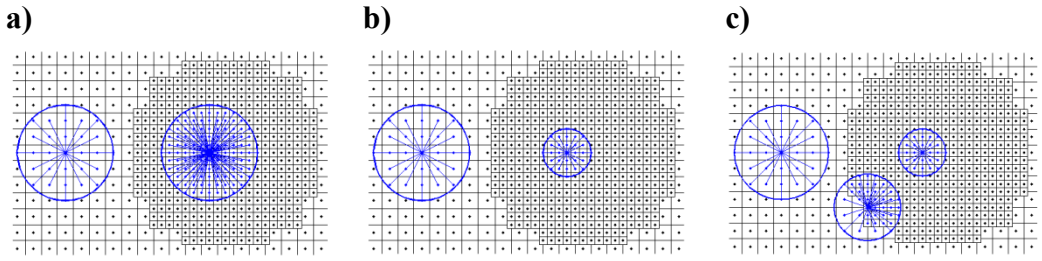


Fig. 4.23: a) Model A: same horizon for all nodes, b) model B: horizon $\delta = m \cdot \Delta X$ for all nodes, c) model C: horizon $\delta = m \cdot \Delta X'$ where $\Delta X'$ can be different from the grid spacing ΔX, for interface nodes, whereas for all other nodes $\Delta X' = \Delta X$.

Model	$\Delta X = \Delta Y$	δ ($\times 10^{-3}$)
A	0.0005 coarse 0.00025 refined	1.5
B		1.5/0.75
C		1.5/1.06/0.75

Table 4.2: Values of the peridynamic parameters for the three models.

The results show that model A is characterized by a considerable loss in accuracy of the numerical integration which generates an energy reduction, bigger at the border of the refined zone. Model B exhibits an even bigger energy reduction, this is probably due to an excessive loss of volume in the application of Eq. (3.1b), as shown in Fig. 4.24. In this case, the numerical errors are so big that they generate a distortion of the energy distribution even in the nodes of the coarse zone near the refined region (see Fig. 4.24b). Results of model C show that the strategy suggested in Sect. 3.2.3 significantly reduces the energy flux distortion, moreover, differently from models A and B, the external nodes exhibit a small increase in energy. Fig. 4.25 presents the relative reduction of the density of potential energy

in the refined zone during the whole simulation, while Table 4.3 summarizes maximum, ΔW_{max}, ΔK_{max}, and minimum, ΔW_{min}, ΔK_{min}, values of the reduction of the density of potential and kinetic energy for a single node. Always in Table 4.3, the maximum relative reduction of the total energy ΔE_{tot} for the whole grid of the three models is also given. In model A the area of the refined region affected by high energy reductions is rather large, on the contrary, it is much smaller in model B, in which however, energy losses reach values of 71.7%. In model C such losses are lower, up to 21.6% and they mainly affect only a few nodes, as shown in Fig. 4.25c.

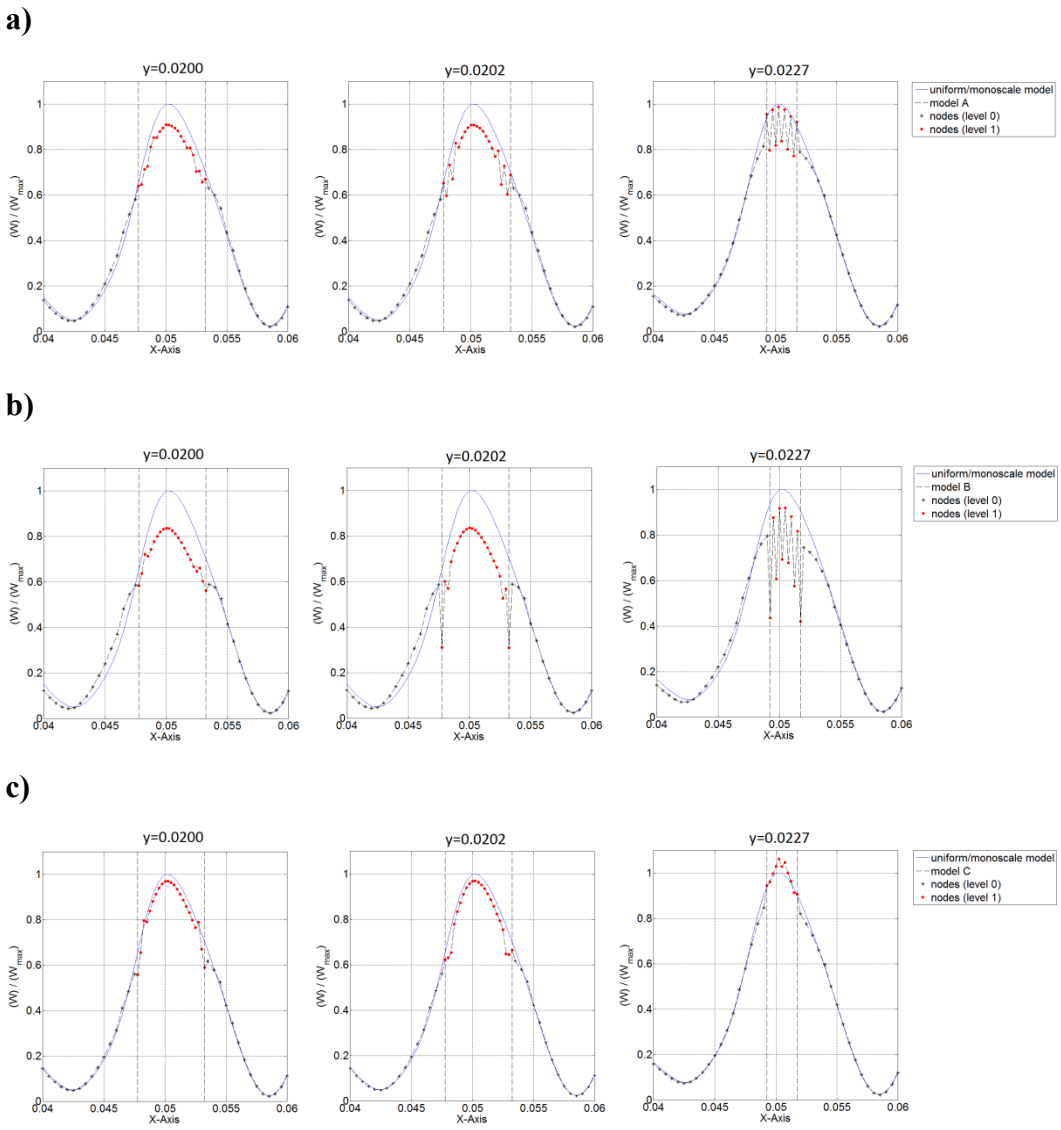


Fig. 4.24: Longitudinal profiles of the potential energy density, numerical values are normalised with respect to the maximum value of the uniform grid solution: a) model A, b) model B, c) model C.

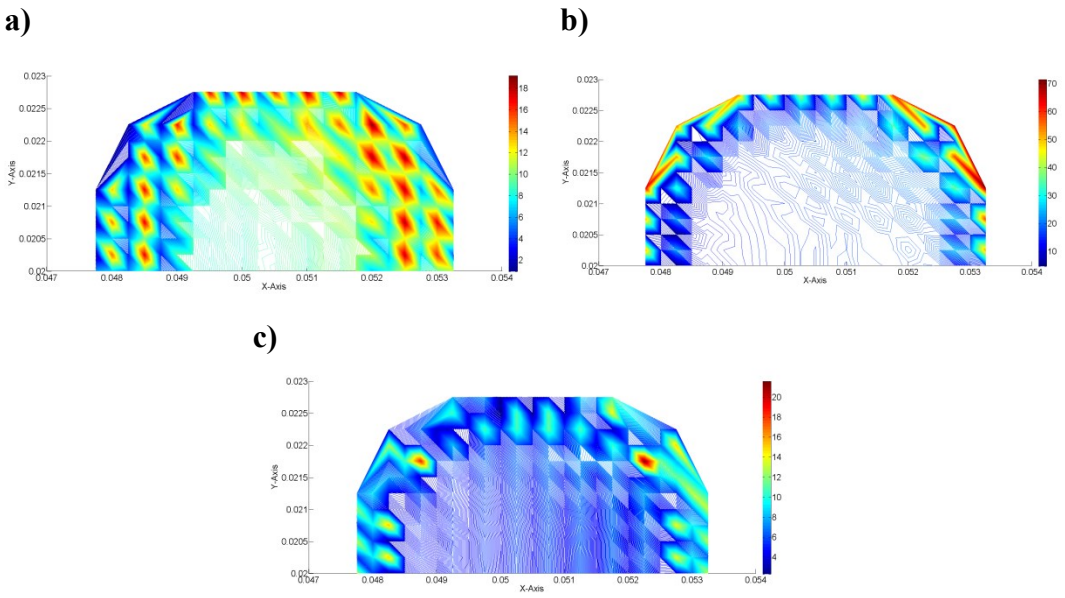


Fig. 4.25: Contour plot of the reduction of the density of potential energy in the nodes of the refined zone: a) model A, b) model B, c) model C.

Model	ΔW_{max} (%)	ΔW_{min} (%)	ΔK_{max} (%)	ΔK_{min} (%)	ΔE_{tot} (%)	Δu (%)
A	19.21	0.89	12.40	1.26	0.12	0.76
B	71.67	4.35	21.91	1.74	0.44	1.64
C	21.62	2.26	8.68	1.66	0.05	0.44

Table 4.3: Maximum and minimum values of the variations of various quantities with respect to the same value of the uniform grid solution (ΔE_{tot} is the total energy of the model, while Δu is the amplitude of the reflected wave with respect to that of the incident wave).

Numerical results presented in Chapter 6 seem to suggest that such localized losses have a negligible effect on crack propagation. The displacement of a sample node in the coarse part of the grid, of coordinates $X = 0.034$ and $Y = 0.02$ (see Fig. 4.22a), is monitored in time. The point is located near the central part of the refined zone. The displacement values in time in the cases of uniform and non-uniform grids are compared. All models show that the refined portion of the grid generates a reflected wave with a maximum pick at time $t = 0.072$s. The amplitude of the reflected wave with respect to that of the incident wave is given as a percent value Δu in Table 4.3. However, the reflected wave is small, in particular in model C and its effects are in general negligible. The order of magnitude of our results is similar to that shown in [57] where a Gaussian wave propagates in a 1D beam model. After having analysed the effects of the first level of refinement on the

propagation of a continuous wave by using strategies of numerical convergence such as the *m*-convergence and δ-convergence, as a second step, a similar problem is investigated by adopting higher levels of refinement as well as the more general $m\delta$-convergence. Fig. 4.26 shows the model of a rectangular plate with size 0.3×0.04 m^2 and all edges are free; the contour plot of the wavefront of total energy density (potential plus kinetic) generated of the initial half wave (it is normalized with respect to half of its maximum value) is displayed as well. The problem is numerically solved by adopting the same method applied in the previous problem, the same is done for the mechanical properties of the model. The initial displacements field is defined by the following expression:

$$u_0(X,Y) = 0.2exp\left[-\left(\frac{X}{0.03}\right)^2\right] \quad , \quad v_0(X,Y) = 0 \tag{4.2}$$

Table 4.4 and Table 4.5 show the parameters of the implemented models, obtained changing the refinement level and the type of numerical convergence; the coarse grid spacing is $\Delta X_0 = 0.001$. As far as the obtained models are concerned adopting the δ-convergence in which, as previously seen, the interface nodes show a significant loss of integrating volume in the calculation of Eq. (3.1b); therefore, as done in the previous example, in order to minimize such loss the horizon length is increased by an appropriate amount.

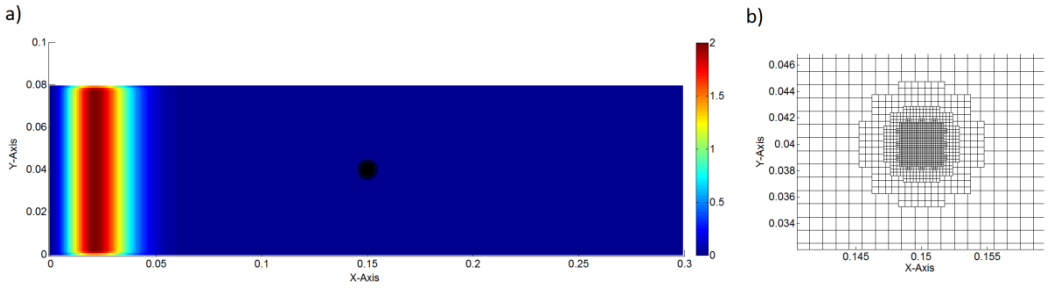


Fig. 4.26: Grid employed for analysis: a) contour plot of total energy density flux, b) example of refined region of 3rd level.

Level Refinement	δ ($\times10^{-3}$)	*m*
0	3.0	3
1	1.5 - **2.1**	3 - **4.2**
2	0.75 - **1.06**	3 - **4.2**
3	0.375 - **0.53**	3 - **4.2**

Table 4.4**:** Characteristic parameters for the models with δ-convergence, the values in bold correspond to the interface nodes with modified horizon.

In Fig. 4.27 and Fig. 4.28, the longitudinal contours (they are taken at the coordinate X=0.1 and Y=0.04) of the wavefront of the total energy density flux are compared at the moment in which the maximum energy peak is located in the refined region. It is evident how the energy flux is subjected to a distortion while crossing the refined region; such distortion is more noticeable for those models in which the δm-convergence is applied. To evaluate the distortion, in Table 4.6 the maximum (ΔE_{max}) and the minimum (ΔE_{min}) percentage variations of the total energy density in the refined region when it is crossed by the energy flux are listed.

Level Refinement	δ ($\times 10^{-3}$)	m
0	3.0	3
1	2.0	4
2	1.25	5
3	0.75	6

Table 4.5: Characteristic parameters for the models with δm-convergence

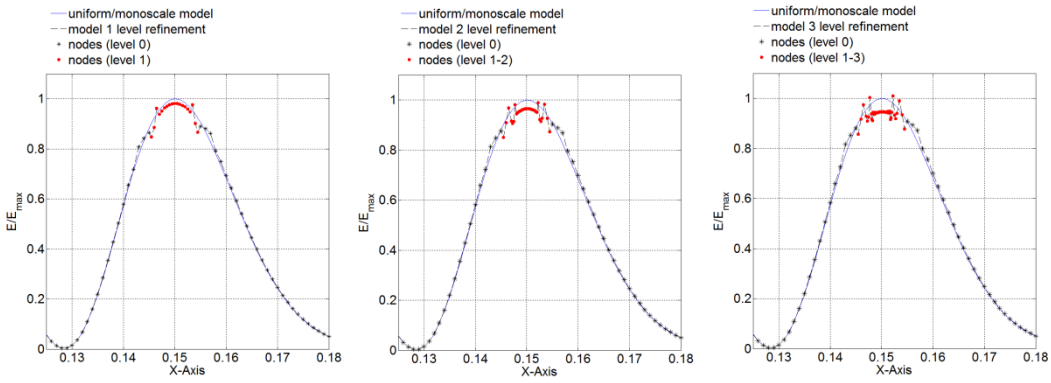


Fig. 4.27: Wavefront of the total energy density for the non-uniform/multiscale models when a δ-convergence is adopted.

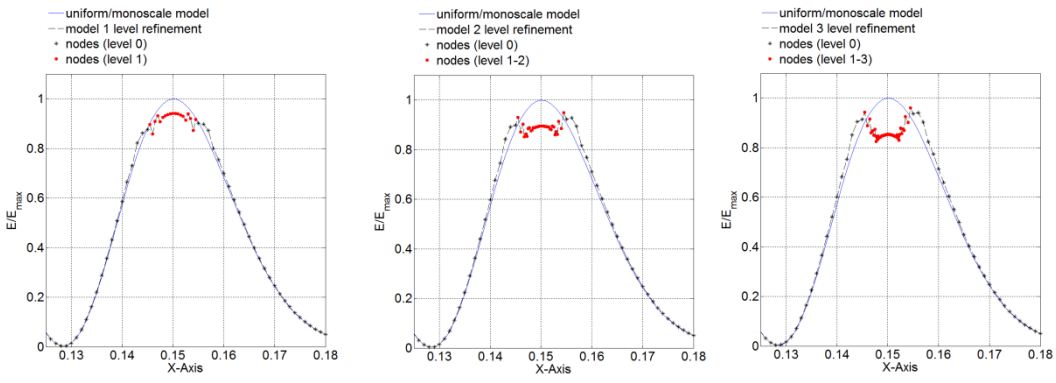


Fig. 4.28: Wavefront of the total energy density for the non-uniform/multiscale models when a δm-convergence is adopted.

Such variations are computed with respect to the solution obtained with the uniform/monoscale model; in Fig. 4.29a the energy reduction contours are plotted for one of the analyzed models (which is representative of the behavior of the other models as well). It is highlighted that the interface nodes show a conspicuous energy decrease due to the volume loss in the calculation of Eqs. (3.1b); this effect can be reduced by modifying the horizon length as it was done for the interface nodes of the δ-convergence models. In Fig. 4.29b the longitudinal contour of wavefront is plotted for the δ-convergence model without modification of the horizon length of interface nodes, these manifest energy reductions of the 37.4%.

Convergence	*Level Refinement*	ΔE_{min} [%]	ΔE_{max} [%]	Δu [%]
δ	1	9.83	5.86	0.24
	2	11.00	5.23	0.31
	3	11.63	5.88	0.35
δm	1	17.10	4.38	0.58
	2	18.96	6.25	0.87
	3	20.81	7.27	1.03

Table 4.6: Total energy density variations of the refined region's nodes and maximum displacement variation of node of coordinates $X = 0.1$, $Y = 0.04$.

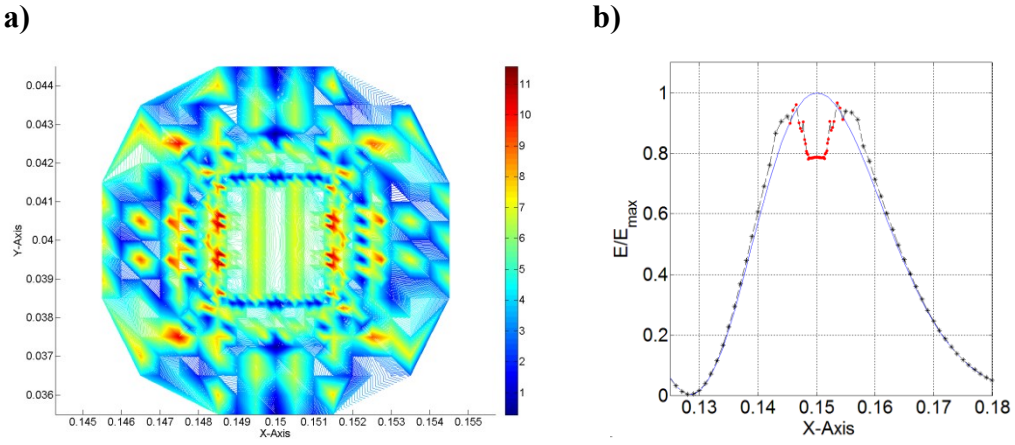


Fig. 4.29: a) Total energy density percentage reduction in the 3rd level refined region within a δ-convergence model, b) Wavefront's distortion without modification of the horizon length of interface nodes.

It is shown that the size of the energy decrease is bigger than the energy increase and that the amount increases as the refinement level goes up. Strategies in order to investigate how to minimize such distortions and how to properly transmit stress waves between regions with different grid spacing and length scale are being developed. This is important since the crack path is determined by the

interaction of the stress wave on the crack tip during its propagation. As far as spurious reflections are concerned, the maximum displacement magnitude of the node located at X=0.1 and Y=0.04 is reported in Table 4.6 (it is expressed as a percentage of the amplitude of the Gaussian wave). The amplitude of the reflected wave increases as the refinement level increases, but this reflection amplitude is of small entity (1% of the Gaussian wave), hence it is negligible.

These examples highlight two important aspects which have to be taken into account when dealing with dynamic analyses. First, when a continuum elastic wave is moving through the refined and scaling region, significant distortions of the elastic waves can be detected in the interface regions. Such distortions are mainly due to the inaccurate calculation of the volume enclosed in the horizon belonging to the interface nodes. This problem can be mitigated by changing the size of their horizon, so that the volume losses are drastically reduced. Of course, consider that the simulations are carried out by using the PA-PDLAMMPS algorithm (see Section 3.1), which does not include the volume of the cells partially located inside the neighborhood of the source node. This shortcoming leads to an underestimation of the volume, limitation that can be overcome by using the IPA-Hybrid algorithm (see Section 3.1) which calculates the exact volume enclosed in the neighborhood. Lastly, the results show that spurious reflections arise when the elastic wave crosses the boundary between the coarse and the refined regions. However, we can conclude that the amplitude of the reflections can be neglected since they have no significant effect on the crack propagation. This aspect will be clear from the benchmark problems treated in Chapter 6.

5. Dependence of the crack paths on grid orientation

Few researchers have observed indirectly [69, 80, 83] that the evolution of crack paths can follow, in an unphysical way, the axes of symmetry of the grid. The main parameter affecting such a numerical phenomenon seems to be the value of the *m*-ratio. The dependence of the crack path on the grid orientation would be a serious drawback for peridynamic based software since it would undermine what is believed to be one of its most important advantages over other computational methods, i.e. its capability to simulate multiple, mutually interacting dynamic fractures with branching in 2D as well as in 3D solids in a simple way. As a natural consequence of such a description it is shown that this deficiency can be drastically reduced by adopting higher *m* ratio values in the whole discretized domain. Such a strategy implies a considerable computational cost, therefore it is a natural candidate for the use of the AGRS to manage with more efficiency the simulations. It is remarked that all the results presented in the following have been obtained by using the IPA-Hybrid algorithm (see Sect.3.1) for the computation of the volume correction factors and the PIM algorithm (see Sect. 3.2.4) for interpolating the physical quantities.

5.1. Description of the problem

As a first step, a problem of dynamic crack propagation in 2D will be presented to make clear that in peridynamics the direction of crack propagation is affected by the grid orientation. In particular, a plate model with a size of 0.25x0.25m^2 and an initial horizontal crack of length 0.05m subjected to a tensile load is investigated, as shown in Fig. 5.1a. In the following, the results obtained when the load is applied dynamically and quasi-statically are presented

5.1.1. Dynamic load case

We assume plane stress conditions and two different materials for the BB-PD and OSB-PD model: Young's modulus E = 73.4 GPa, Poisson's ratio ν = 0.33 and mass density ρ = 2440 kg/m^3 for the BB-PD, Young's modulus E = 100 GPa, Poisson's ratio ν = 0.15 and mass density ρ = 4000 kg/m^3 for the OSB-PD. The load is applied suddenly to the upper and lower edges of the plate, uniform along the edges and kept constant in time. Its intensity is 0.8 MPa for the BB-PD case

and of 0.7 MPa for the OSB-PD model. The plate and its load are symmetric with respect to the horizontal line along which the initial crack is located so that, for symmetry reasons, the crack has to grow along the same line or branch from it in a symmetric way. This behavior is correctly captured by a peridynamic model of the plate in which the material points of the grid are aligned along the vertical and the horizontal directions, as shown in Fig. 5.1b, this grid is the 0° grid. The direction of the crack in the numerical model does not change if important grid parameters such as the grid spacing, the horizon and the *m*-ratio are varied.

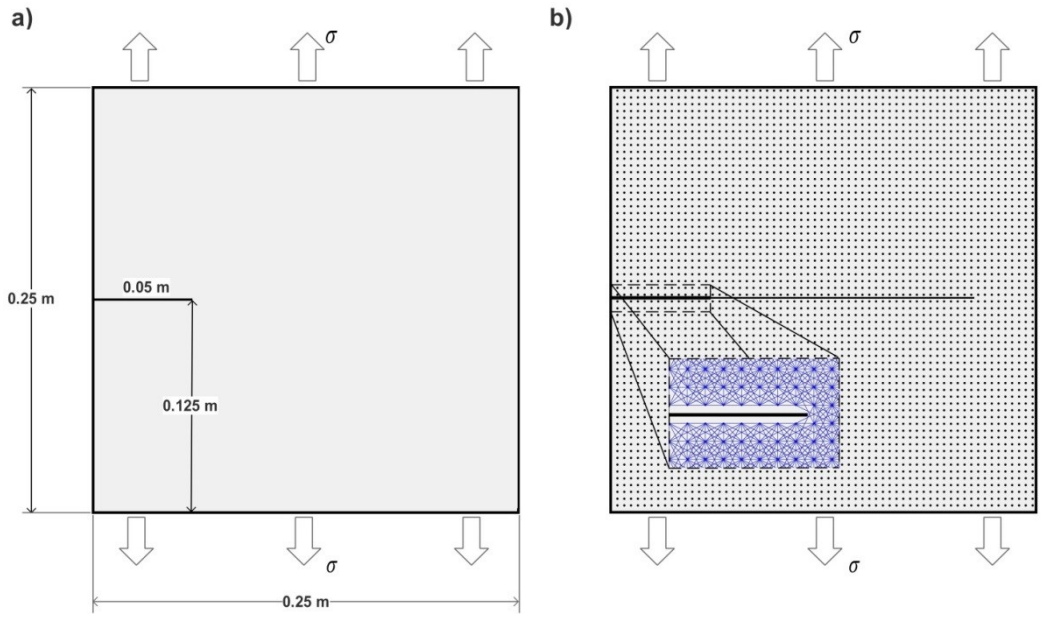


Fig. 5.1: a) Plate, load condition and initial crack. b) Propagation of the crack in the regular 0° grid. The insert shows how the initial crack is described in the discretization by removing all bonds that would intersect it.

The aim of this section is to compare the results obtained when the plate is modelled with a uniform grid rotated with respect to the direction of the initial crack; various grids with an inclination of 5°, 10°, 20°, 30° and 40° with respect to the 0° grid, have been employed. We can see in Fig. 5.2b the influence of a rotated grid on the modelling of the initial crack, which is obtained by removing all bonds which cross the crack line. It appears slightly irregular, but such an irregularity has a negligible influence on the direction of crack propagation. This issue will be further clarified in Appendix B. A grid spacing $\Delta X = 1$ *mm* has been chosen which leads to have an average number of 62,500 nodes for all the simulated models**.** The simulations have been carried out assuming an *m* ratio of 3, this choice is in accordance to the value often used in the existing literature as a good tradeoff between the computational efficiency and the accuracy of the

solution. The failure stretches used are $s_{03} = 2.1070 \cdot 10^{-4}$ for the BB-PD and $s_{03} = 10^{-4}$ for the OSB-PD, in which the subscript 3 expresses that the failure stretch is associated to the grid characterized by $m = 3$; the time step of $\Delta t = 100$ ns has been adopted being lower than the critical time step [84]. The total simulation time is 560 μs and 400 μs for the BB-PD and OSB-PD, respectively.

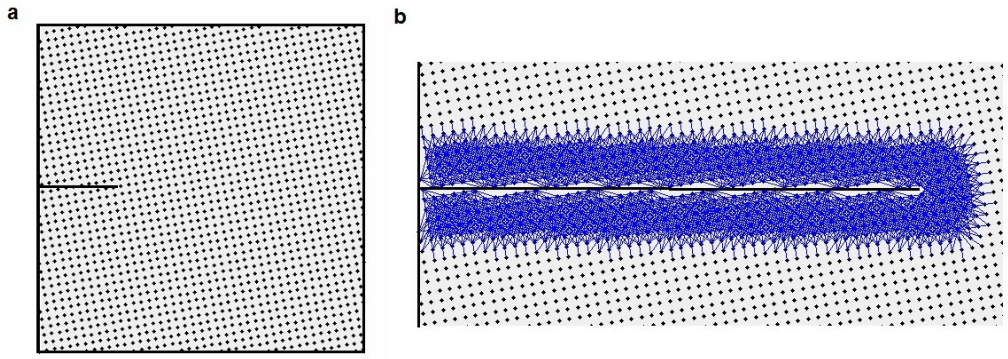


Fig. 5.2: a) Example of plate discretised with a rotated grid. b) Example of the initial crack modelled by removing all bonds which cross the crack line.

Fig. 5.3 shows four examples of cracks propagated in rotated grids with the relevant grid orientation. Two of the examples were obtained with a BB-PD formulation and the other two with a OSB-PD formulation. It is apparent that the grid orientation considerably affects the direction in which cracks propagate, which is an undesirable feature for a computational method such as the one adopted in the present work. Moreover, Fig. 5.3 shows that the results obtained with BB-PD and OSB-PD are very similar regardless of the material properties used and the peridynamic formulation adopted. For this reason, all following results have been obtained by using the BB-PD formulation, which requires lower computing times with respect to the one based on the OSB-PD. Fig. 5.4 represents the crack paths for all considered BB-PD grid orientations. It is worth to highlight that if the plate is discretized with a uniform grid characterized by an orientation different from the horizontal one, as shown in Fig. 5.2a, in general the edges of the plate will not be exactly described by the grid. The irregularity of the boundaries and that of the initial crack profile makes the problem asymmetric so that some form of asymmetry of the solution has to be accepted. In particular, the elastic waves due to the sudden application of the load can affect, in part, the direction of the crack propagation. However all these 'noisy' perturbations should not excessively affect the results produced by a robust computational technique.

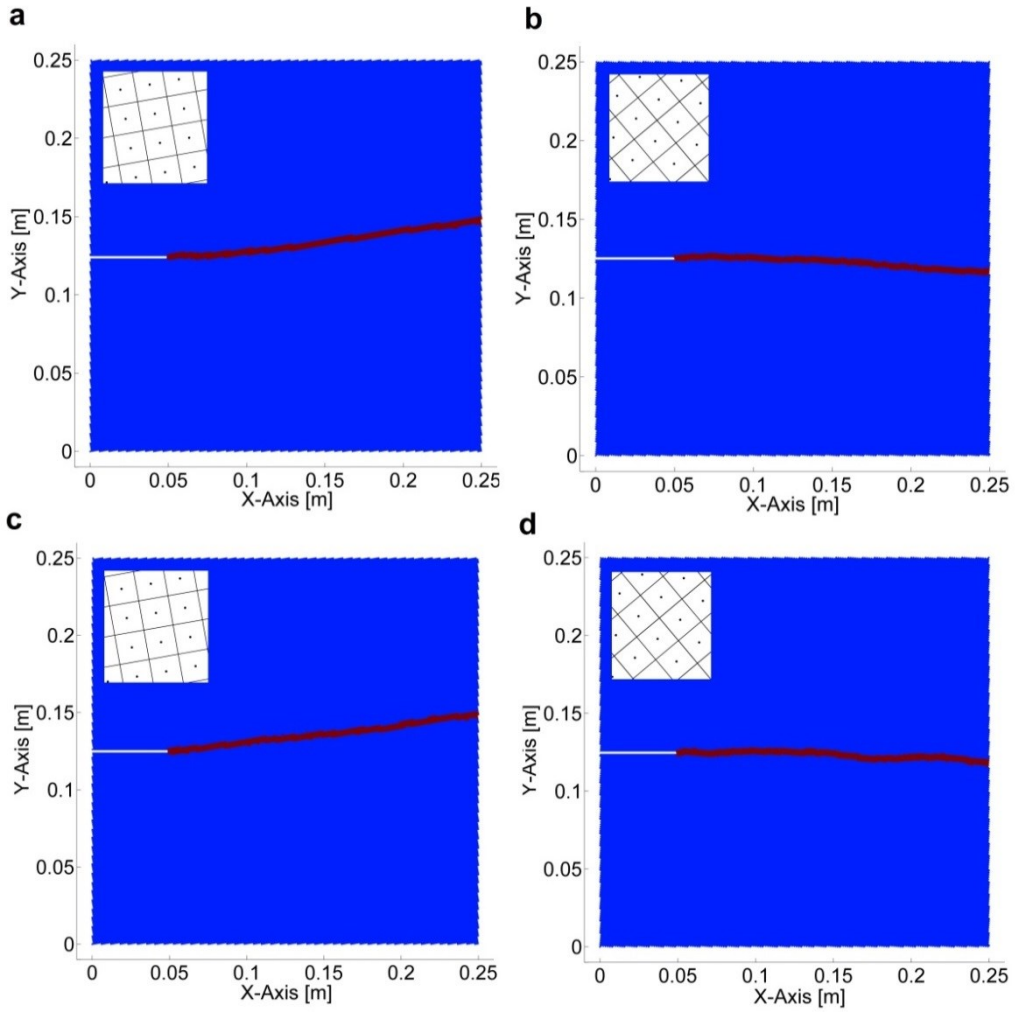


Fig. 5.3: Crack paths obtained with different grid orientations and m=3. a) Model with the grid at 10° implemented with the BB-PD. b) Model with the grid at 40° implemented with the BB-PD. c) Model with the grid at 10° implemented with the OSB-PD. d) Model with the grid at 40° implemented with the OSB-PD.

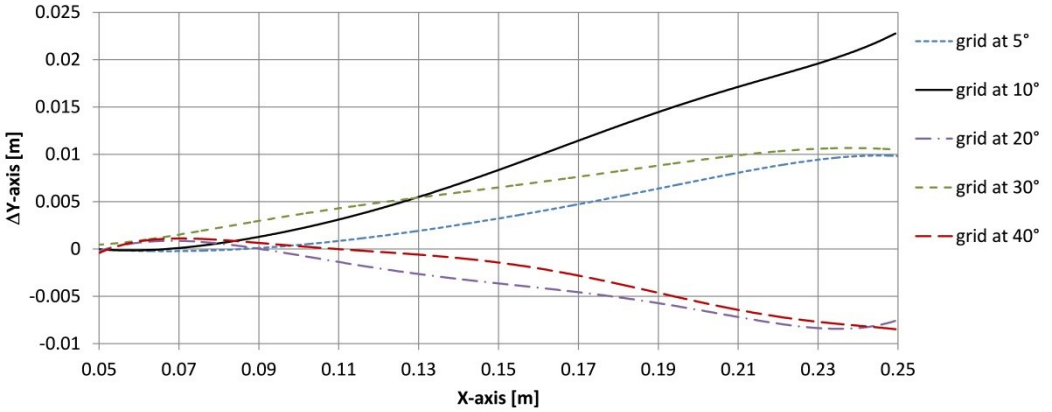


Fig. 5.4: Crack paths obtained with different rotated grids with m = 3.

One could be induced to think that the dependence of the crack direction on the grid orientation is a problem of the grid spacing. To clarify this aspect the example is proposed again by using a uniform more refined grid, characterized by

ΔX = 0.5 mm. The problem is simulated using m=3 and m=5. The relative failure elongation s_{03} = 2.9795·10^{-4} has been used for the case of m=3 while Eq. (2.15b) has used to evaluate s_{05}. The time step is Δt = 50 ns while an average number of 250,000 nodes for all the models simulated has been adopted. It is clear from the results shown in Fig. 5.5 and Fig. 5.6 that the refinement of the grid with unchanged m does not improve the direction of the cracks. However, in the next section 5.1.2. the case of the plate with quasi-static load is considered to assess the relevance of the waves generated by the sudden application of the load on the crack direction.

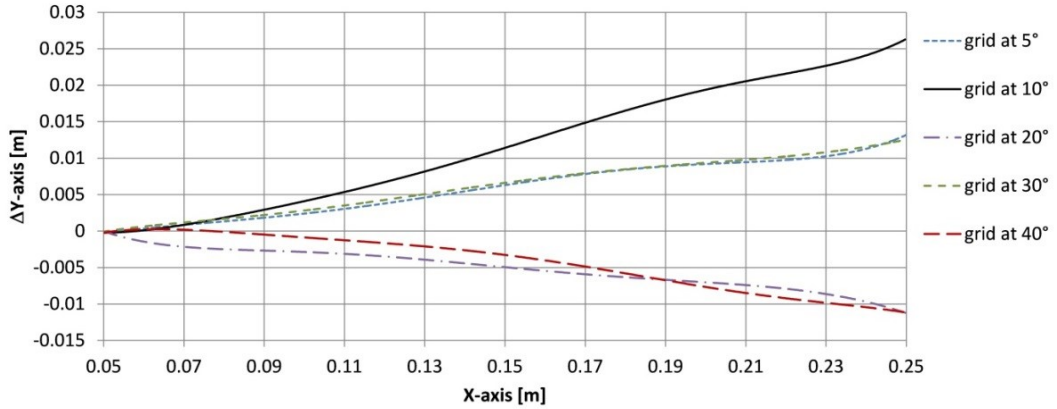


Fig. 5.5: Crack paths obtained with different rotated grids with m = 3 and reduced ΔX.

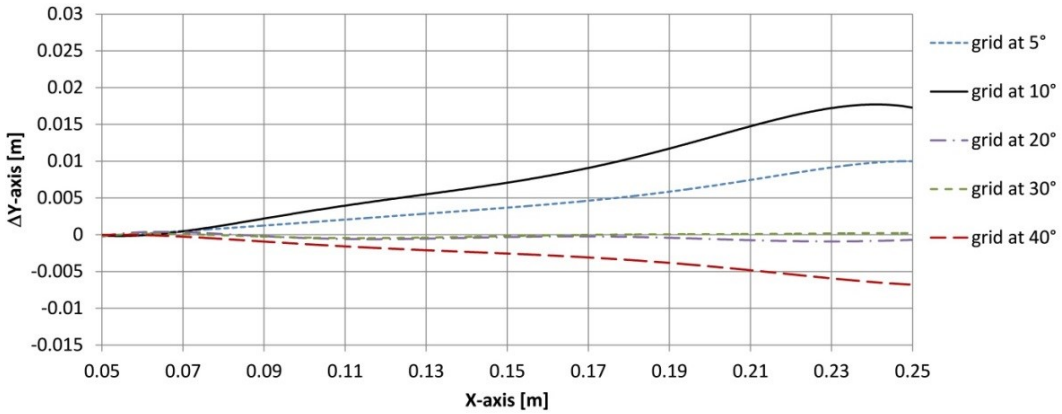


Fig. 5.6: Crack paths obtained with different rotated grids with m = 5 and reduced ΔX.

5.1.2. Quasi-static load case

The quasi-static application of the load is considered in the present section. The applied load is increased linearly in time in a way to guarantee that the inertial forces and kinetic energy effects on the solution are negligible, the load has been

increased from zero to its maximum value in a time interval of 3*ms*. In particular, the maximum magnitude of the load is enough to activate crack propagation. The analysis are carried out using *m*=3; Fig. 5.7 shows the crack path for different grid orientations. It is possible to observe as the deviation of crack path from the horizontal direction still persists although, as expected, the crack paths are different from the ones obtained in the dynamic load case.

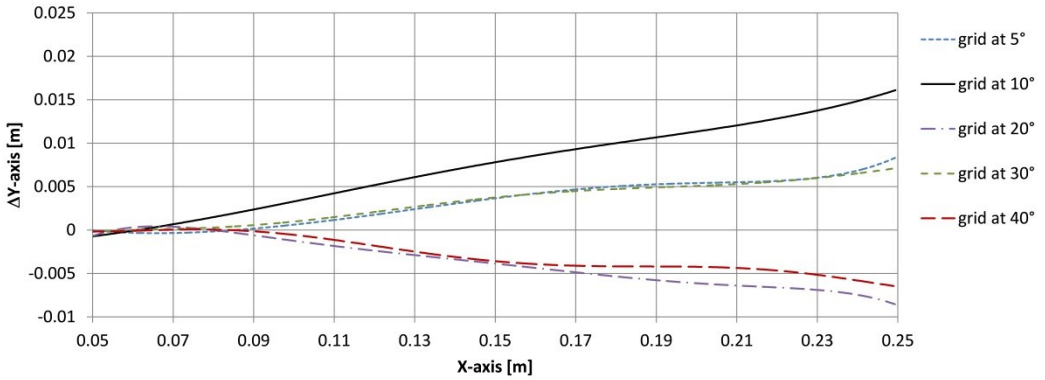


Fig. 5.7: Crack paths obtained with different rotated grids with *m* = 3.

5.2. Understanding of the problem

In order to explain with clarity the cause of the dependence of crack path on grid orientation, Fig. 5.8 provides an example of a source node connected with all its family nodes for a grid characterized by *m*=2.51: continuous lines represent active bonds, dotted lines represent broken bonds, dashed lines represent possible directions of crack propagation. From the Fig. 5.8 it is clear that a crack propagating in the horizontal direction (crack path 1) intersects, and therefore has to break, a given set of bonds (eleven of the bonds connected to the source node), whereas a crack propagating in a slightly inclined direction (crack path 2) intersects, and therefore has to break, an increasing set of bonds (all previous eleven bonds plus three more horizontal bonds to the left of the source node). Therefore, it is easier for the crack in the peridynamic grid to propagate in the horizontal direction than in other directions close to it. A similar reasoning applies to all directions parallel to the bonds: for example a crack propagating at 45° has to break twelve bonds whereas if it propagates at (45°±ε^{o}) (with ε^{o} small) it has to break (12+2) bonds.

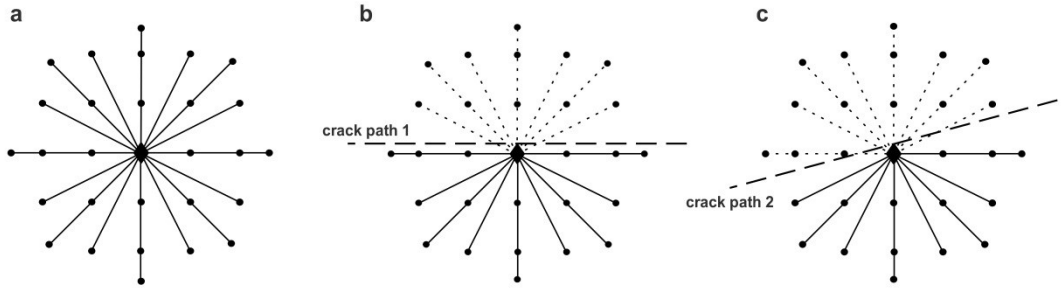


Fig. 5.8: a) Source node connected to its family nodes for m=2.51. A crack path at 0° (b) 'breaks' less bonds than a crack path slightly inclined (c). 0° is a weaker direction with respect to neighboring directions.

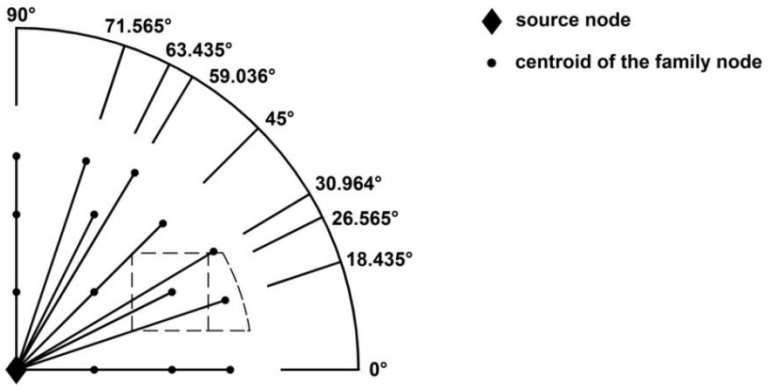


Fig. 5.9: Bond directions in the plane of a grid with m = 3. The figure shows as well the area A_j associated to the family nodes on the bonds at 26.565° and 18.435°.

So we can conclude that the discrete distribution of bonds in the peridynamic grid introduces a set of directions (the bond directions called as well *weaker directions* below) in which the crack can propagate by breaking a smaller number of bonds with respect to the surrounding directions. More precisely, in order to propagate cracks have to break fourteen bonds, for m=2.51, in all directions of the plane except in the directions parallel to the bonds, for which a smaller number of bonds has to be broken. We observe that in the discretized continuum the value of the damage would correspond to the theoretical value of 0.5 for cracks propagating in all directions except the bond directions. With reference to the case addressed in Section 5.1.1 with m=3, the weaker directions are distributed as shown in Fig. 5.9 and they correspond to the following angles: 0°,18.435°, $\tan^{-1}(0.5)$ = 26.565°, 30.964°, 45°, 59.036°, $\tan^{-1}(2.0)$ = 63.435, 71.565°, 90°.

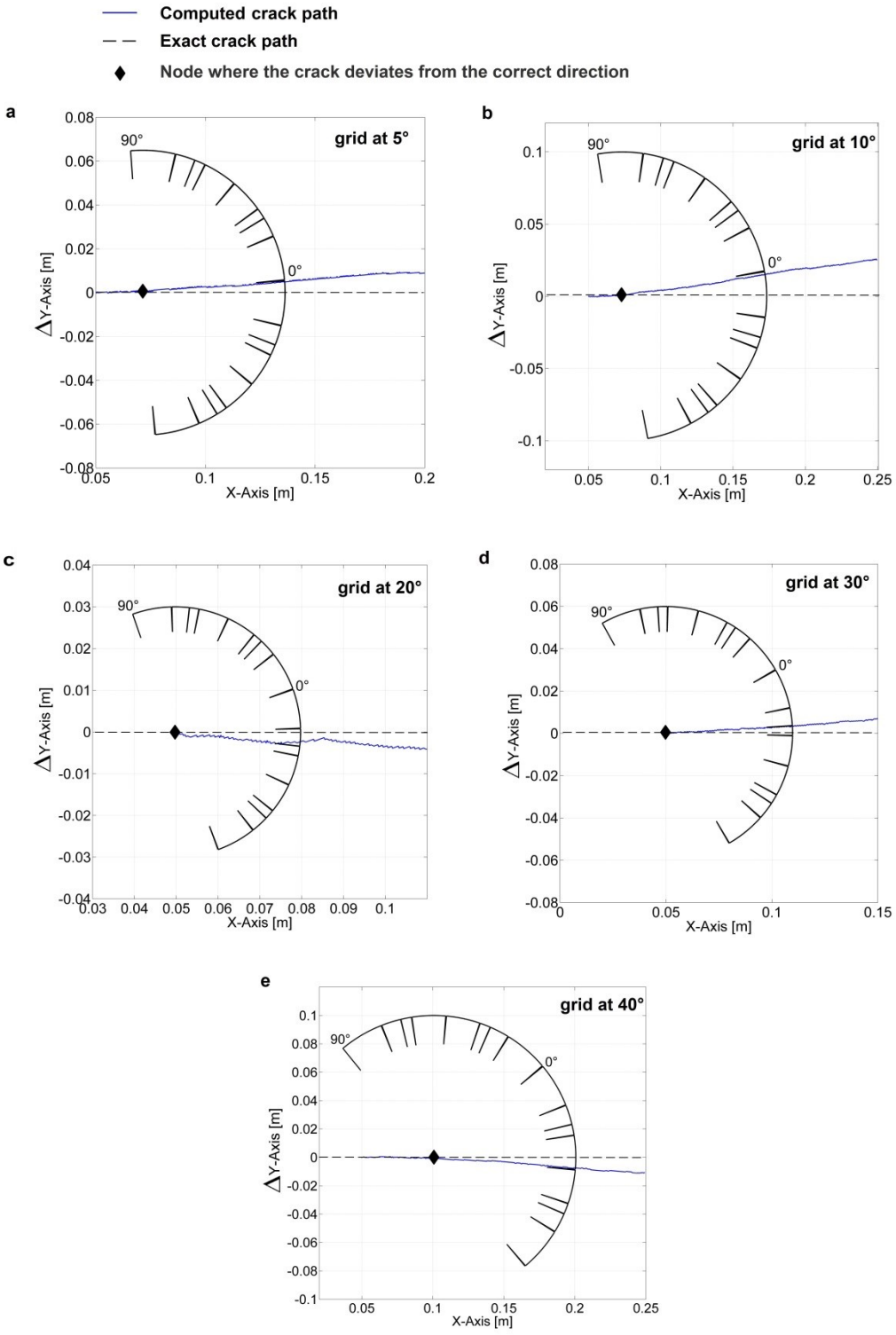


Fig. 5.10: Comparison of the crack propagation paths with the weaker directions for m=3 and different grid inclinations. a) Grid at 5°. b) Grid at 10°. c) Grid at 20°. d) Grid at 30°. e) Grid at 40°.

Fig. 5.10 compares the crack paths shown in Fig. 5.4 with the weaker directions of a grid with m=3. Fig. 5.10 suggests that often cracks propagate along the weaker direction closest to the correct direction but the same figure shows as well that exceptions can take place. In particular, in Fig. 5.10c the crack is aligned with the

bonds at -26.565° rather than with those at -18.435°, which would be closer to the correct direction. This is probably due to the fact that the crack at -26.565° does not break the bond with the same inclination but the one at -18.435°, since the latter requires a smaller failure force which is proportional to the volume *Vj* assigned to the family node of the bond (see Fig. 5.9). A similar remark applies as well to the case of Fig. 5.10d.

5.3. Solution to the problem

A possible way to reduce the problem of the dependence of crack direction on grid orientation is to increase the number of weaker directions, in this way cracks will be attracted most probably by a bond direction sufficiently close to the correct direction of crack propagation. From what was said in the previous section it is clear that can be achieved by increasing the number of bond directions. Fig. 5.11 shows that the number of bond directions can be increased by increasing the number *m*.

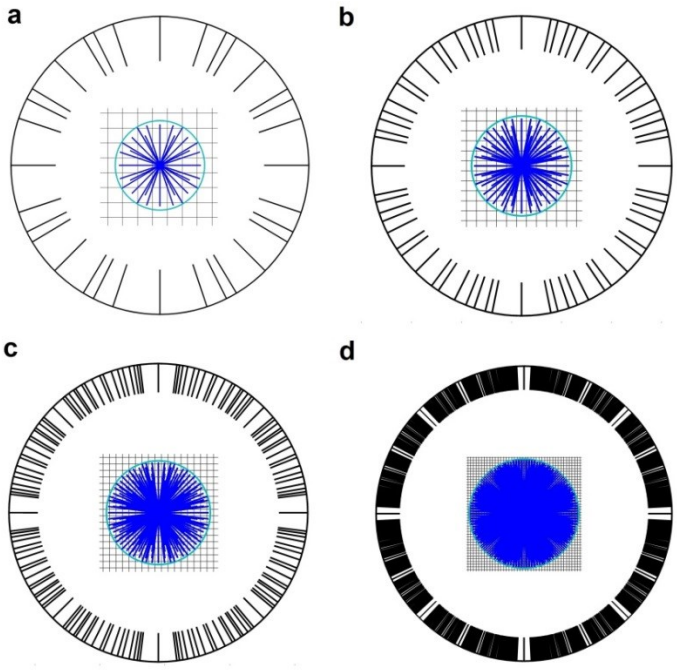


Fig. 5.11: Bond directions for different values of the *m* ratio, a) *m* = 3, b) *m* = 5, c) *m* = 8, d) *m* = 20.

Fig. 5.11 clearly shows that increasing the bonds in a uniform grid does not increase the number of bond directions uniformly in the plane. There exist angles in the plane, in particular close to 0°, 45° and 90° where bond directions remain

'rare' and therefore where the error on crack direction remains somehow larger than in other zones of the plane.

5.3.1. Increasing the m-ratio with a uniform grid

The simplest approach is to adopt a uniform grid with constant value for *m*. Fig. 5.12 and Fig. 5.13, which concern the worst case of grid at 10°, show how the direction of the crack varies as the value of *m* is increased, while keeping constant the grid spacing.

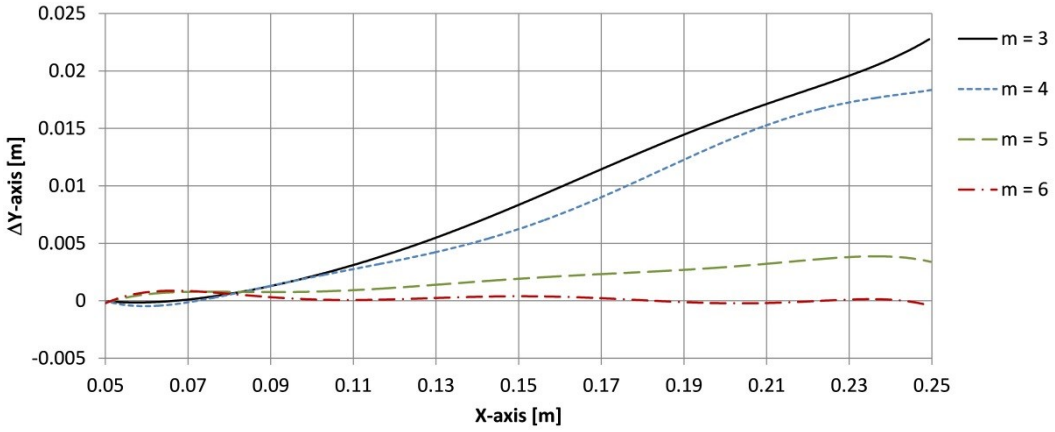


Fig. 5.12: Comparison of the crack paths when the load is applied dynamically, grid at10° for different values of *m* ratio.

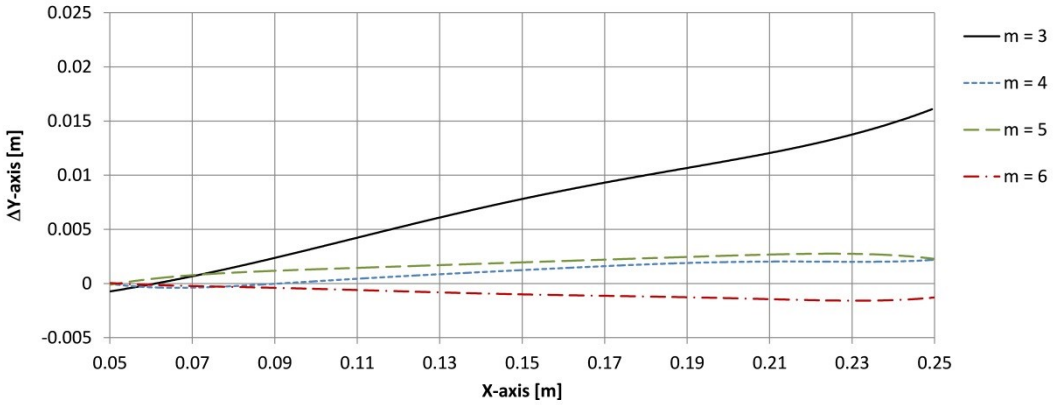


Fig. 5.13: Comparison of the crack paths when the load is applied quasi-statically, grid at10° for different values of *m* ratio.

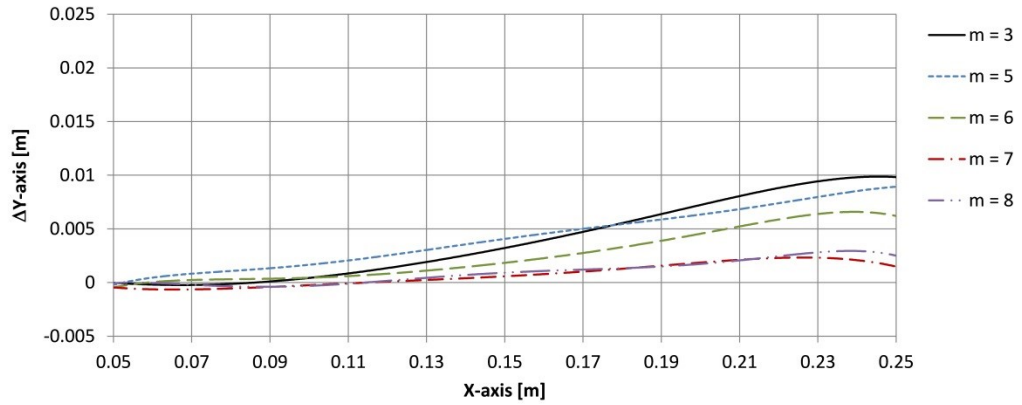


Fig. 5.14: Comparison of the crack paths when the load is applied dynamically, grid at 5° for different values of *m* ratio.

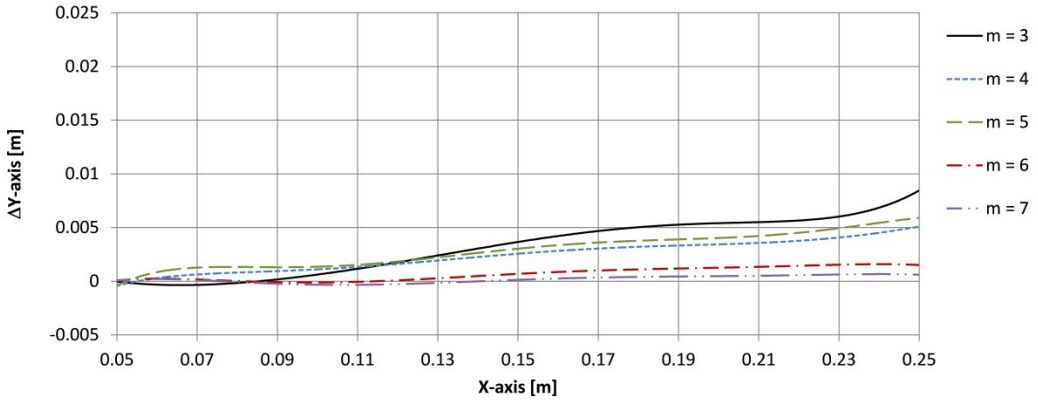


Fig. 5.15: Comparison of the crack paths when the load is applied quasi-statically, grid at 5° for different values of *m* ratio.

The corresponding critical stretches s_{0m} are obtained by applying the following expression derived by using Eq. (2.15)**.** Fig. 5.14 and Fig. 5.15 show the evolution of crack paths for increasing values of *m* when the inclination of the grid is 5°. The results show that higher values of *m* are required in this case than when the grid is inclined at 10°. Such an indication is consistent with what shown in Fig. 5.11: larger values of *m* are necessary to correct wrong crack paths if the grid inclination is small, i.e. close to 0° where the density of 'weaker' directions remains low. It is apparent that the increase of the *m*-ratio improves the accuracy of the direction of crack propagation but, at the same time, greatly increases the computational cost of the simulations (see Table 5.2 reported in Sect. 5.3.2). In order to reduce such a drawback in an efficient way the AGRS algorithm can be adopted.

5.3.2. *Increasing the m-ratio with the AGRS*

As stated at the introduction of this Chapter, the AGRS is a good candidate to eliminate the dependence of crack propagation on the grid orientation. The idea is to apply the AGRS in order to increase automatically the *m*-ratio only in the proximity of the crack tip by keeping a low value of the *m*-ratio far from the region where the crack may propagate, by doing so the computing resources will be managed more efficiently. This result can be achieved through the application of different strategies, i.e. by using the numerical *m*-convergence for increasing the *m*-ratio and keeping constant the horizon length, otherwise, when the problem requires to reach the desired length scale [4], the δm-convergence for increasing the *m*-ratio shrinking the horizon length may be a good choice. Both strategies are applied in the following.

Refined Model	ΔX_0 [mm]	m_0 ratio	ΔX_{mr} [mm]	m_{mr} ratio
grid at 5°	2.0	3.5	1.0	7.0
grid at 10°	1.0	3.0	0.5	6.0
grid at 10°	2.0	3.0	0.5	6.0

Table 5.1: Peridynamic parameters used in the adaptive refined models, the subscript "*mr"* indicates the parameters defined on the most refined region.

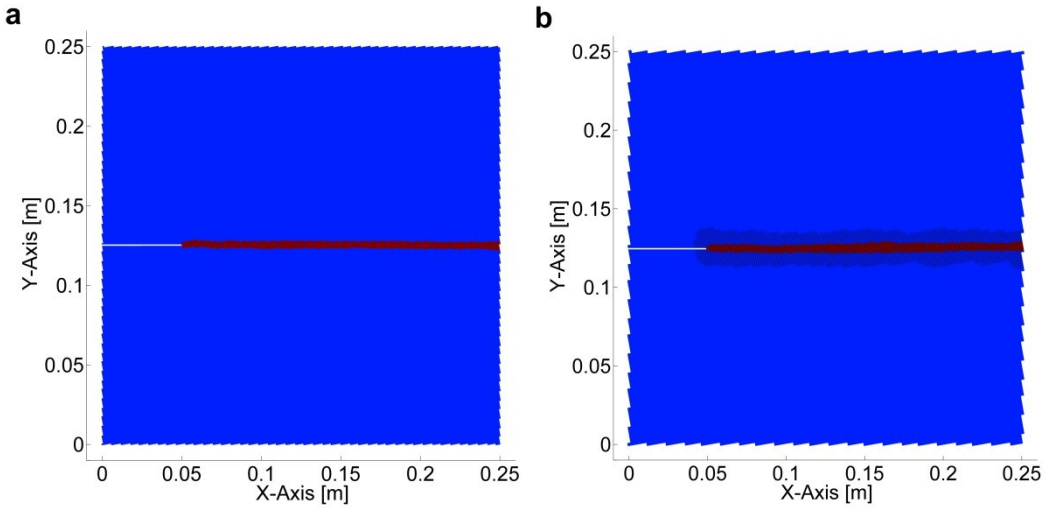


Fig. 5.16: Comparison of damage state of the plate with grid rotated of 10°: a) refined uniform model with $\Delta x_0 = 0.001m$ (62,502 nodes) and $m_0 = 6$, b) adaptively refined model (15,600 nodes at the beginning of the simulation, 19,062 nodes at the end).

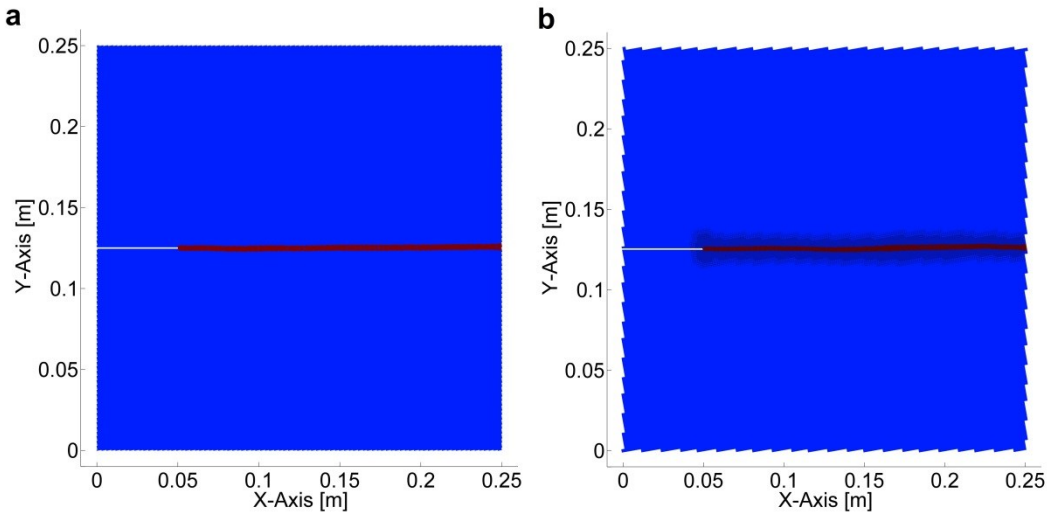


Fig. 5.17: Comparison of damage state of the plate with grid rotated of 10°: a) refined uniform model with $\Delta X_0 = 0.0005m$ (250,002 nodes) and $m = 6$, b) adaptively refined model (15,600 nodes at the beginning of the simulation, 26,778 nodes at the end).

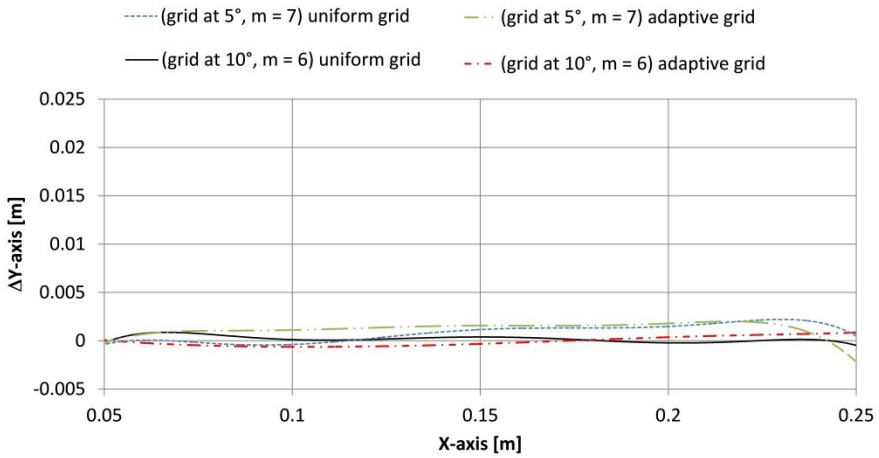


Fig. 5.18: Comparison of the crack paths obtained with the uniform and adaptively refined grids at different inclinations.

We remark that the aim of applying such strategies is to increase the m-ratio in the region where the crack propagation takes place rather than to reach the convergence of the numerical solution to that analytical one (see Sect. 3.1). In Table 5.1 the main peridynamic parameters used for the adaptive models are given. With respect to the case of grid rotated of 10°, Fig. 5.16 shows the contour plot of the final crack path obtained by adopting the m-convergence strategy, while Fig. 5.17 shows that obtained with the δm-convergence strategy. The results are more clearly shown in Fig. 5.18 where some of the crack paths obtained with uniform grid are compared to those obtained with adaptively refined grid. It can be noted that the dependence of crack propagation on grid orientation can be

eliminated through the use of the adaptive refined grid. Table 5.2 presents the CPU time of the simulations carried out for the analyzed cases, it shows that the use of the AGRS algorithm is crucial to reduce the computational cost of the simulations.

Model	ΔX_0 [mm]	*m*-ratio	CPU-time [*hour*]	Crack direction
Uniform (grid at 5°)	1	3	0.83	deviated
		7	2.90	horizontal
Uniform (grid at 10°)	0.5	3	7.08	deviated
		6	16.09	horizontal
Adaptive single level refinement (grid at 5°)	2	3.5 → 7	2.20	horizontal
Adaptive double level refinement (grid at 10°)	2	3 → 6	3.52	horizontal

Table 5.2: Comparison of CPU-times required by uniform and adaptive models for the grid rotated of 5° and 10°.

5.4. Conclusions about grid sensitivity in peridynamics

After having clearly stated that regular peridynamic grids suffer from dependence of crack paths on grid orientation, the present chapter has shown that for 2D examples such dependence is due to the existence in the discrete grid of directions along which the peridynamic material offers preferential paths for crack propagation. The 'weaker' directions coincide with those of the bonds, both in regular bond based and ordinary state based grids. In particular, cracks often propagate along the weaker direction closest to the correct direction. Exceptions to that have been discussed. It has been shown that such dependence persists regardless of the nature, dynamic or quasi-static, of the load. In general, the deviation of the crack path from the correct direction is larger for the case of dynamic load application. That seems to be mainly due to the interaction between the waves generated by the dynamic load and the growing crack. A possible way to reduce the dependence of crack path on grid orientation is to increase the number of weaker directions by increasing the value of the *m*-ratio. Values of *m* as high as 6-7 seem to provide reasonably accurate solutions. Higher values of *m* are

required to accurately describe cracks propagating at angles close to 0°, 45° and 90° with respect to the grid orientation. Adopting such a high values of *m* in the whole grid would greatly increase the computational cost of peridynamic based simulations and therefore the adoption of suitable algorithms like the AGRS developed in this work does considerably improve the computational efficiency by limiting the use of the high *m* values only to the regions where cracks are propagating. We believe that the issues discussed in this work are relevant as well for 3D cases. The bonds connected to a central node will most probably define in space planes along which cracks can propagate by intersecting, and therefore breaking, a given set of bonds, whereas a crack propagating in a slightly inclined direction would intersect a larger set of bonds. Therefore, it is easier for the crack in the regular 3D peridynamic grid to propagate along a weaker plane than in other directions close to it. It is reasonable to assume that increasing the value of *m* in a 3D regular grid would reduce the dependence of the crack pattern on the grid orientation.

6. Numerical examples

It is well known that the study and simulation of dynamic brittle fracture is still a challenging problem due to the complexity of all the mechanisms which involve a wide range of length scales of the medium. In order to test the effectiveness of the proposed AGRS to deal with such phenomena, in the present sections three crack propagation cases will be treated, which represent benchmarks commonly used in the scientific literature about brittle dynamic fracture [85, 86, 87]. They represent all 2D samples with an initial crack, the material is in all cases homogeneous and isotropic, while the adopted constitutive model is the PMB (see Section 2.1). We remark that all the simulations have been performed with in-house Matlab codes by using the double-precision computation. Moreover, all simulations have been carried out through the application of the BB-PD formulation, while the space integration is performed by means of the PA-PDLAMMPS algorithm (see Section 3.1). The strategy proposed in Section 3.2.3 is adopted in order to mitigate the volume losses affecting interface nodes (see Fig. 3.10).

6.1. Crack branching

The first example presents a crack branching in a plate with an initial crack and an applied step load as shown in Fig. 6.1. The load is kept constant for the whole duration of the simulation. A similar case has been simulated in [43, 44] with BB-PD and in [88] with XFEM. In both cases a uniform grid was used. The same problem has been solved with different grids with the aim to verify the effectiveness of the proposed AGRS method. Three grids were used; two uniform grids, one with grid spacing of level zero refinement (ΔX = 1mm) and the other with the 1st level of refinement of the whole grid (ΔX = 0.5mm). The third grid is obtained by applying the proposed AGRS method and therefore its grid spacing is not uniform (cfr. Table 6.1). The mechanical properties of the material of the plate (*Soda-Lime Glass*) are E = 72 GPa, ρ = 2,440 kg/m^3 and G_0 =135 J/m^2 is the fracture energy at branching [43]. In Table 6.1 the relevant data for the peridynamic grid for the two refinement levels are reported. The plane stress condition of the cases imposes a Poisson coefficient υ = 1/3. The tensile stress is applied only to the nodes of the upper and lower borders, as shown in Fig. 6.1.

The initial crack is modelled by removing the broken bonds. In all simulations the applied stress is 22MPa, for a duration of t = 50μs, the time step of the numerical integration is Δt =50ns; Fig. 6.2 shows a few images of the temporal evolution of the damage state of the adaptive model. One can notice that the refined region is generated in front of the crack which develops from the tip of the initial crack, since this region is characterised by a high level of elastic energy density. The refinement is activated with the energy trigger $W \geq 0.7 W_{max}$ applied from time t_{st} = 5.7μs.

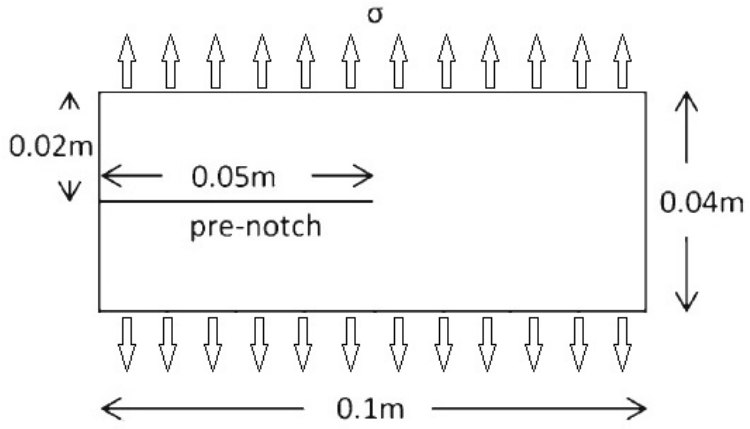


Fig. 6.1: Setup of the pre-cracked plate under traction load.

Level	ΔX (mm)	δ (mm)	c_o $[10^{18} \cdot N/m^6]$	$10^{-3} \cdot s_o$
0	1.00	3.00	7.64	0.93
1	0.50	1.50 – **2.10**	61.10 – **21.60**	1.32 – **1.11**

Table 6.1: Peridynamic grid parameters (values in bold character are those of the interface nodes with modified horizon).

The grid refinement generates 2,293 additional nodes. Fig. 6.3 compares the contour plots of the damage levels obtained by the three models, in Table 6.2 some of the main features of the crack morphologies are compared as well as a few data on the timing of the crack phenomena: the length of the crack before branching l_b, the inclination of the initial and final branches of the crack θ_i and θ_f , the time instants at which the crack starts propagating t_i and branching t_b (see Fig. 6.3b). Fig. 6.3 and Table 6.2 show that the model with the adaptive grid provides results very similar to those of the homogeneous fine grid. Another important

information which BB-PP can provide is an estimate of the propagation speed of the crack.

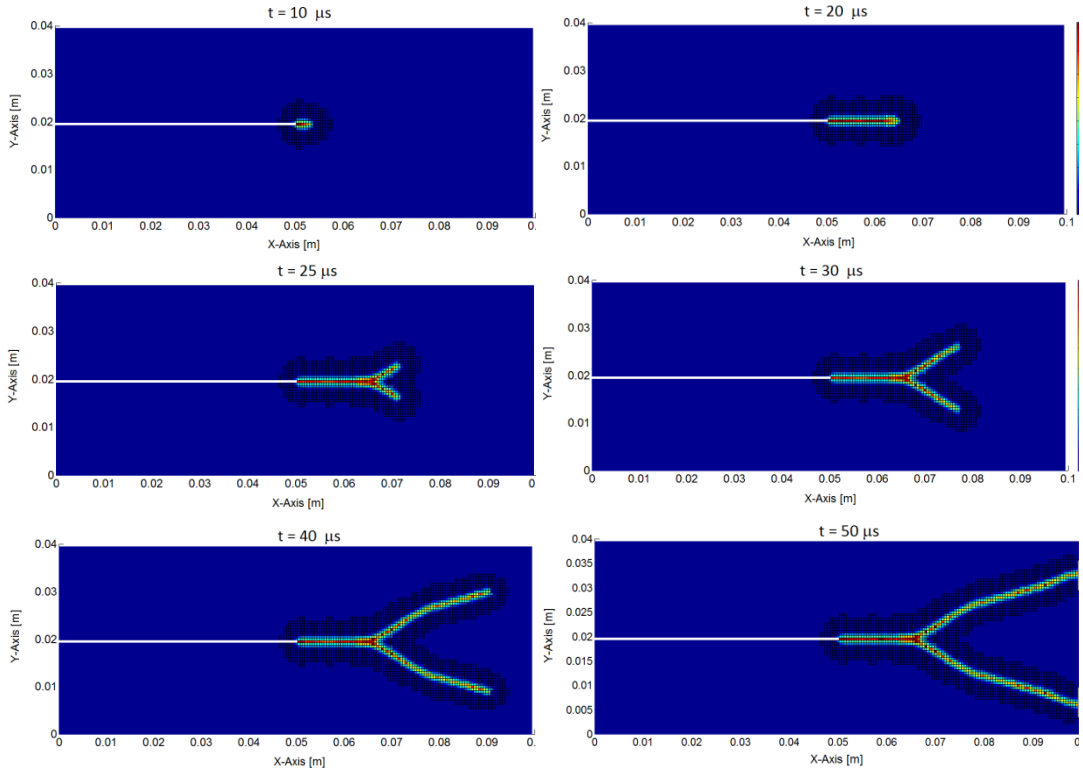


Fig. 6.2: Six snap-shots showing the temporal evolution of the damage level in the adaptive model. The darker region around the crack is the refined zone.

As suggested in [43], the time history of the crack tip position can be identified by monitoring at each time step the position of the farthest node with a damage level above 0.35. The speed estimate is given by the following expression:

$$V_{l-1/2} = \frac{\|\boldsymbol{x}_l - \boldsymbol{x}_{l-1}\|}{t_l - t_{l-1}} \tag{6.1}$$

where $\boldsymbol{x}_l$ e $\boldsymbol{x}_{l-1}$ are the positions of the crack tip at times t_l and t_{l-1} respectively; due to the discrete nature of both the special and temporal solutions, the evaluation of the crack speed at each time step is characterised by wide oscillations; this is the reason why the crack propagation speed is computed by taking into account a time interval of $2\mu s$. Fig. 6.4 shows a comparison between the experimental measurement of the crack propagation speed [85] and the three numerical evaluations, the three average values of the computed speed are given in Table 6.3. It is interesting to notice that all models seem capable to capture what has been experimentally observed in [86] on the trend of the crack

propagation speed just before branching. One can notice in Fig. 6.4 that the maximum speed is reached immediately before the start of the crack branching (≈22μs, see Table 6.2), whereas a speed reduction is observed during the crack bifurcation. The efficiency of the AGRS is demonstrated in Table 6.4 where the computational times required by the three different models are given. The hardware and software features of the computer are:

- Intel® Core™ i7-3770 CPU @ 3.40 GHz
- RAM: 32 GB
- OS: Windows 8 Pro 64 bit

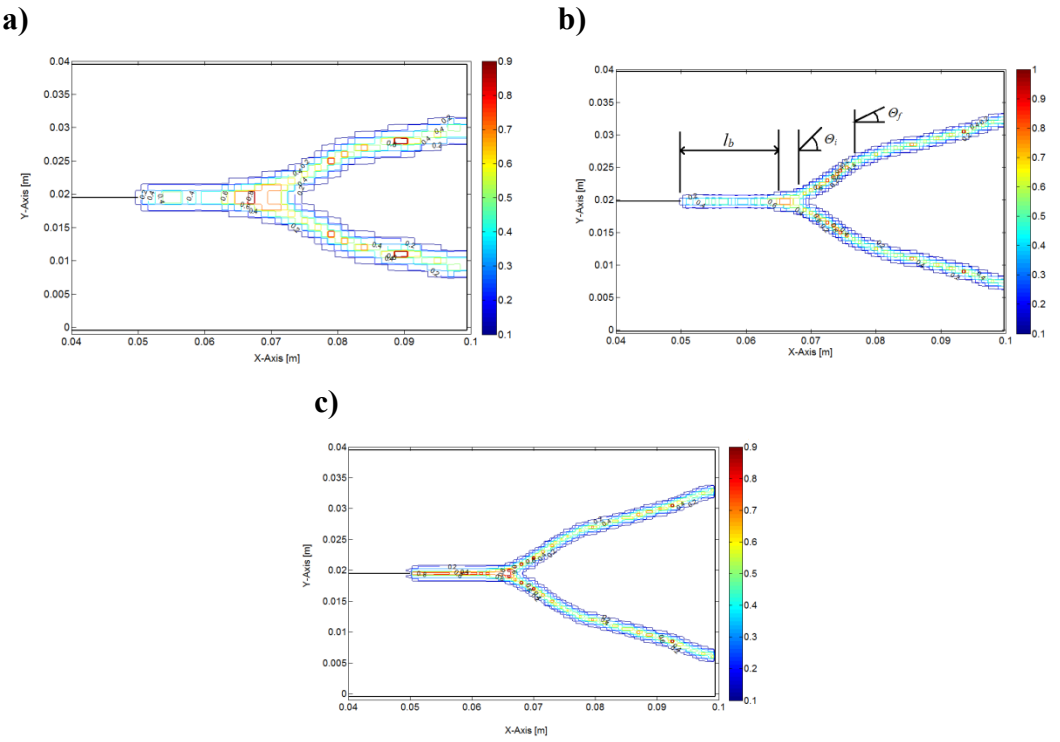


Fig. 6.3: Comparison of the crack shape at time t = 50μs: a) uniform coarse grid (grid size 0 level), b) uniform refined grid (grid size 1st level), c) adaptive grid.

Model	$N°$ nodes	t_i (μs)	t_b (μs)	l_b (mm)	Θ_i [°]	Θ_f [°]
Uniform	4,000 (0 level)	6.05	21.85	17.0	26.0÷27.0	11.0÷12.0
Adaptive	4,000 ÷ 6,293	6.00	21.83	16.5	31.0÷32.0	15.0÷16.0
Uniform	16,000 (1st level)	6.00	20.85	15.0	33.0÷34.0	15.0÷16.0

Table 6.2: Parameter values of the crack for the implemented models.

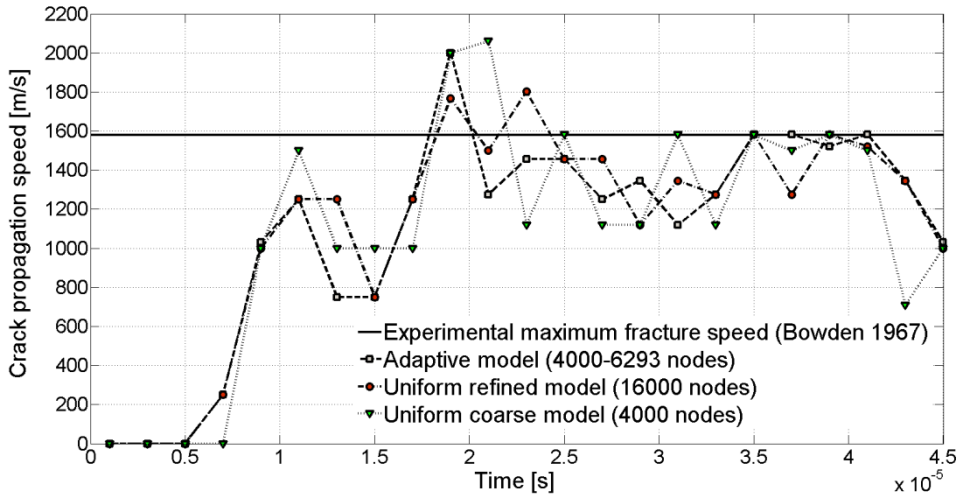


Fig. 6.4: Comparison of the crack tip propagation speed estimated with the implemented models, 1,580m/s is the maximum speed measured experimentally by Bowden et al. **[85]**.

Model	$V_{mean}\,[m/s]$
Uniform (4,000 nodes)	1,319
Adaptive (4,000 - 6,293 nodes)	1,308
Uniform (16,000 nodes)	1,344

Table 6.3: Average crack tip propagation speed estimated with the three models.

Model	*CPU-time* [hours]
Uniform (4,000 nodes)	5,294
Adaptive (4,000 - 6,293 nodes)	12,233
Uniform (16,000 nodes)	34,090

Table 6.4: Comparison of the running times of the three different models.

We want to remark that the focus of the present work is not on computational efficiency but on the effects of refining the grid and reducing the horizon length on the numerical solution of the problem. In spite of that, the use of the AGRS reduces by 64.1% the solution time with respect to a fine grid. So far the problem proposed in this section has been treated by using 2D models, therefore, in order to check the effectiveness of the AGRS to capture crack branching in 3D domains, its application is extended in a 3D grid.

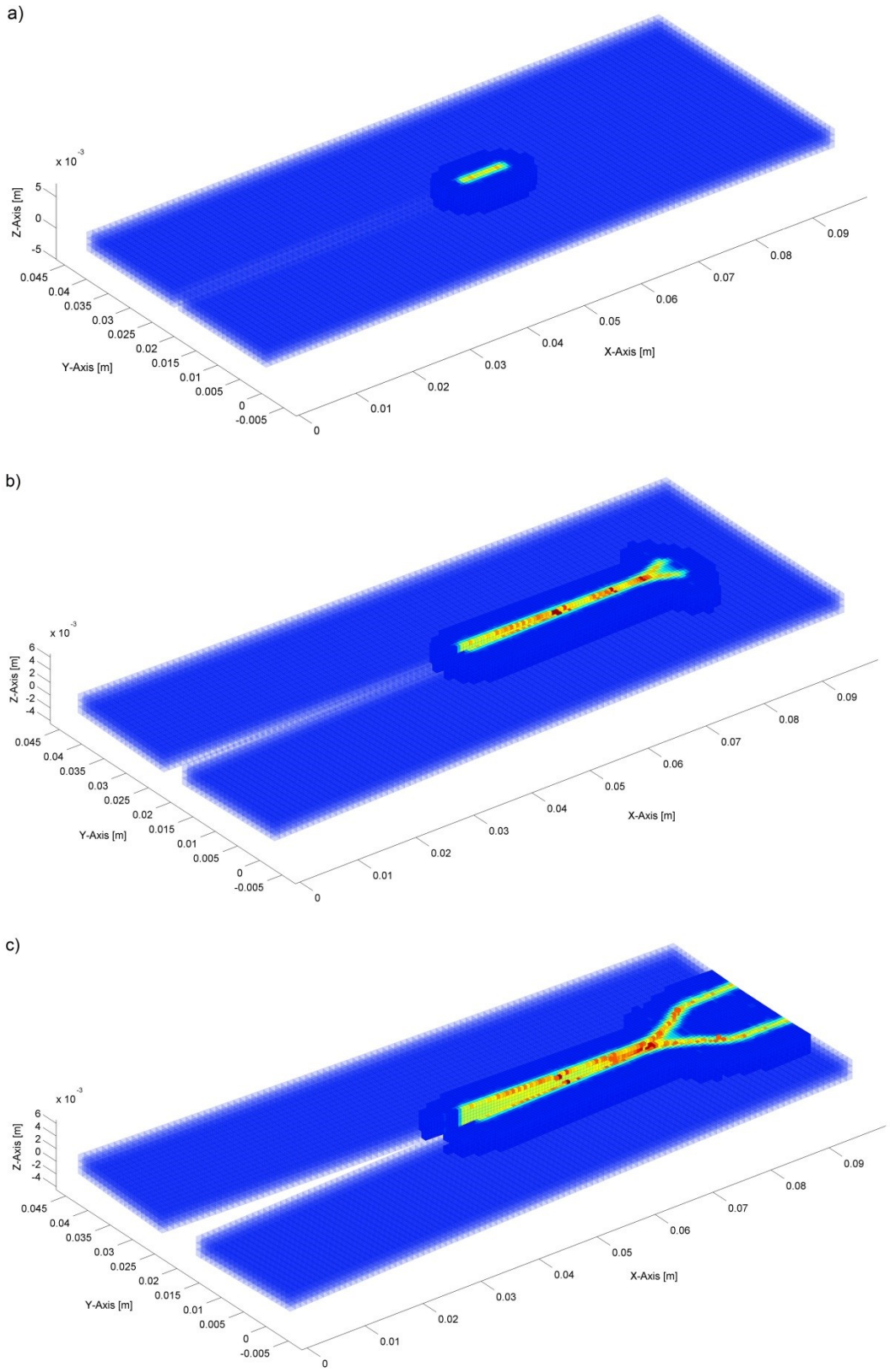


Fig. 6.5: Three snap-shots showing the temporal evolution of the damage level in the adaptive 3D model, a) at time t=25µs, b) at time t=50µs, c) a time t=70µs.

The mechanical properties of the material of the plate (*Duran 50 glass*) are E = 65 GPa, ρ = 2,235 kg/m^3 and G_0 =204 J/m^2 is the fracture energy at branching [43]. The plate has a thickness of t_b = 3mm and subjected to a step load of 6MPa. The initial coarse grid has a uniform grid spacing of $\Delta X = \Delta Y = \Delta Z$ = 1mm, resulting in an initial grid of 12000 nodes; the m-ratio is set equal to 3. The total time t of simulation is 70μs with a time step of Δt =50ns. Let us consider that the low number of nodes through the thickness of the plate surely leads to increase the surface effect on the crack propagation. Regardless, we remark that we are interested in verifying if AGRS can capture the crack branching. The refinement and scaling of 1st level, which is activated with the energy trigger $W \geq 0.7W_{\max}$ at time t_{st} = 7.5μs, generates 14,644 new nodes. Besides, the δ-convergence strategy is adopted to reduce the size of the horizon length. In Fig. 6.5 three snap-shots of the temporal evolution of the crack propagation are shown, demonstrating as the proposed AGRS algorithm can be easily applied to 3D models.

6.2. Kalthoff –Winkler's experiment

Another well know study of fracture dynamics is the Kalthoff–Winkler experiment [89] in which a pre-cracked steel plate is hit laterally from a projectile as shown in Fig. 6.6; such an experiment shows that the fracture type depends on the impact speed.

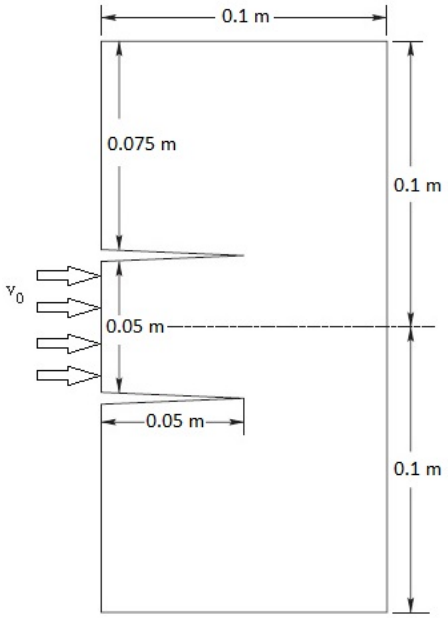


Fig. 6.6: Kalthoff–Winkler's experimental setup.

If the plate is made of steel 18Ni1900, an impact speed of v_0 =32m/s causes a brittle fracture mainly in mode I [89] leading the crack fracture to propagate with an angle of approximately 70° with respect to the horizontal direction. A coarse model with 5,000 nodes is the starting point for the AGRS simulation, the input parameters are given in Table 6.5. As a first step, only the AGRS of the first level is applied. The problem is symmetric so that symmetry conditions could be used to speed up the numerical solution [88, 90], however, our model represents the entire plate. The properties of the material are E = 190 GPa, ρ = 8,000 kg/m^3 and G_0 = 22,170 J/m^2, plane strain conditions are assumed so that Poisson's ratio is υ =0.25. The impact is simulated by imposing to the nodes on the left surface of the sample between the two cracks an initial speed of v_i = 16.5 m/s in the horizontal direction, in this way we assume that the materials of the projectile and of the plate have the same elastic impedance, all other borders are free. The simulation duration is t = 90μs and a time step Δt =70 ns is chosen. The refinement is activated with both energy trigger $W \geq 0.7\, W_{max}$ and damage trigger $\Delta\phi > 0$ applied from time t_{st} = 15.3μs. Fig. 6.7 shows a sequence of frames of the damaged plate, the use of AGR generates 4,453 new nodes. The primary fracture starts propagating at about time t_i = 24μs along an almost straight line inclined of 68° with respect to the horizontal axis, a value very close to that determined experimentally. Two fracture paths are visible: a primary fracture which originates from the pre-existing cracks, and a secondary fracture that is generated at the side of the plate opposite to the impacted side. A similar morphology was observed as well in references [88, 90], as shown in Fig. 6.8a,b. The last frame of Fig. 6.7 shows that some damage appears as well on the impacted side of the sample, similar results were found as well in [88, 91] (see Fig. 6.8b-c). However, there was no reported evidence of a secondary crack in the experiment [89]; reference [88] points out that for finer grids such a fracture is not observed. In the adaptive model some limited asymmetries are present, as a non-perfectly symmetric distribution of the refined zones. The loss of symmetry of peridynamic solutions has already been observed, for example in [44] and it is probably caused by a variation in truncation errors due to the sequential order with which operations are carried out for the computation of the integral in Eq. (3.1b). Another important quantity which has to be checked is the crack propagation speed.

Level node	$\Delta X\,(m)$	δ (mm)	$c_o\ (10^{18}\cdot N/m^6)$	$10^{-3}\cdot s_o$
0	2.00	6.00	2.69	5.05
1	1.00	3.00 – **4.20**	21.50 – **7.60**	7.13 – **6.00**

Table 6.5: Peridynamic grid parameters for the nodes of the grid (values in bold character are those of the interface nodes with modified horizon).

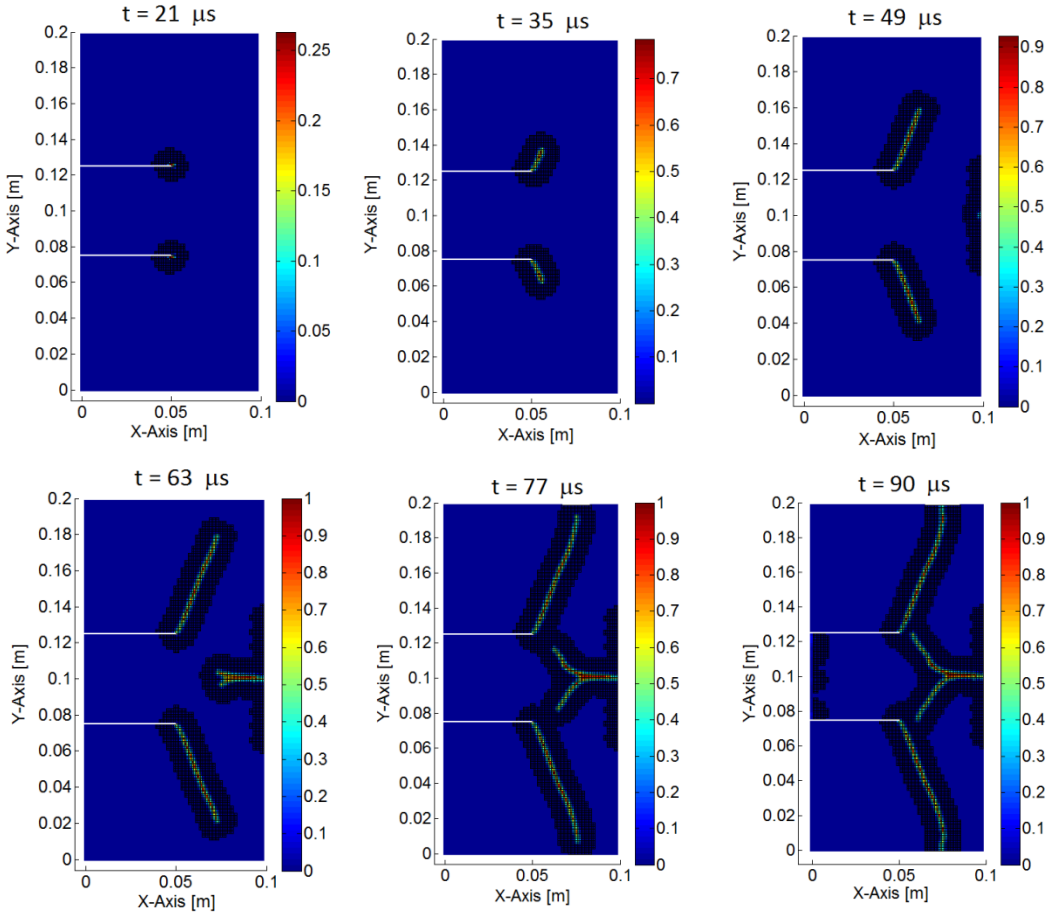


Fig. 6.7: Six snap-shots showing the temporal evolution of the damage level in the adaptive mode with the first level of refinement and scaling.

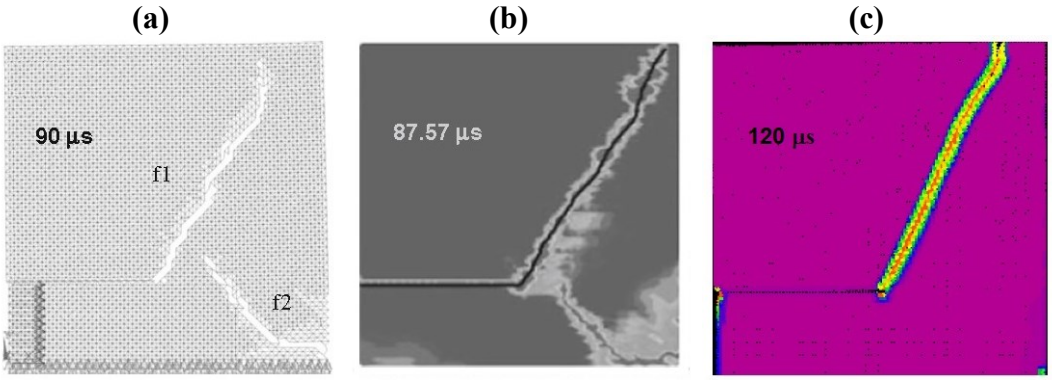


Fig. 6.8: Crack paths estimated by: a) truss-like discreet element method 2D [90], b) XFEM with loss of hyperbolicity criterion 2D [88], c) BBP with uniform/monoscale grid 3D and material X2 NiCoMo 18 9 5 [91].

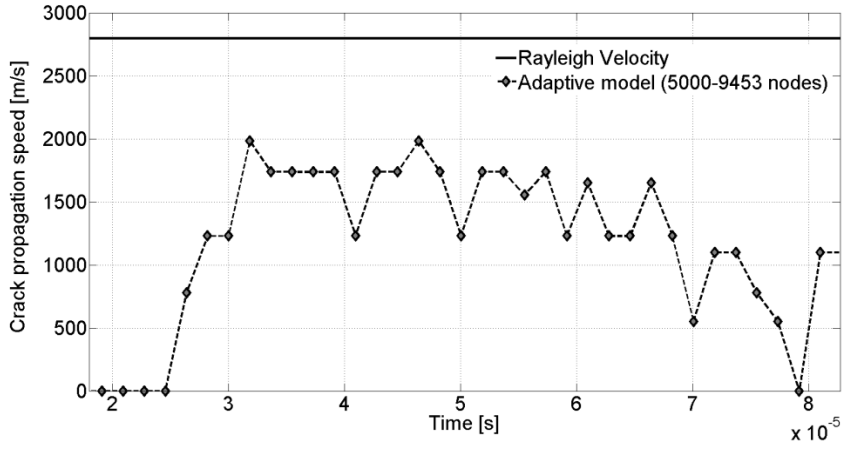


Fig. 6.9: Trend of the crack propagation speed of the primary crack.

Reference [92] states that it cannot be higher than Rayleigh's speed, which for the material under exam is c_r = 2,799.2 m/s. Fig. 6.9 shows the evolution of the crack propagation speed, computed with Eq. (6.1) with an interval of 1.82μs. The maximum computed crack propagation speed is v_{max} = 1,981m/s, equal to approximately 71% of Rayleigh's speed, and very close to the value of 75% given in [88]. So far only the first level of refinement and scaling has been applied, hence in order to verify if an higher level of refinement and scaling of the AGRS is able to capture the angle experimentally observed in Kalthoff-Winkler's benchmark, another adaptive model is implemented. The initial grid always has the same peridynamic parameters given in Table 6.5, while keeping constant the *m*-ratio the horizon has been shrunk to 1.5 mm. The simulation duration is t = 54μs and a time step Δt =35 ns is chosen. Both the damage ($\Delta\phi > 0$) and energy ($W \geq 0.7\, W_{max}$) based triggers are activated from time t_{st} = 15.3μs leading to generate 6,902 new nodes at the end of the simulation. Fig. 6.10 shows how the crack still propagates straight and with an inclination of approximately 68°, which is close to that experimentally observed. We underline that the use of a refinement trigger based only on the deformation energy would require threshold values very low in order to activate the AGRS procedure at the crack tip, especially in the presence of high deformations due, for example to the impact. The refined region would be too large and consequently, the numerical procedure would be inefficient. Fig. 6.11 shows the results obtained by carrying out the previous simulations with only the energy based trigger set to $W_{thres} = 0.3 W_{max}$. As it can be clearly seen from the solution obtained, the only energy criterion would lead to

generate a very large refined zone, not interested by cracks, and, at the same time, would miss the activation of the refinement in the area where the secondary crack is going to be generated. To capture the secondary crack an even lower value of W_{thres} would be necessary, which would further increase the size of the refined zone.

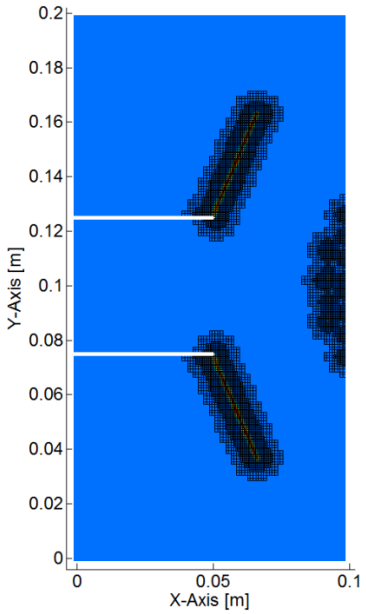


Fig. 6.10: Snap-shot of damage state of the model with applied the 2nd level of AGRS at time 54 µs.

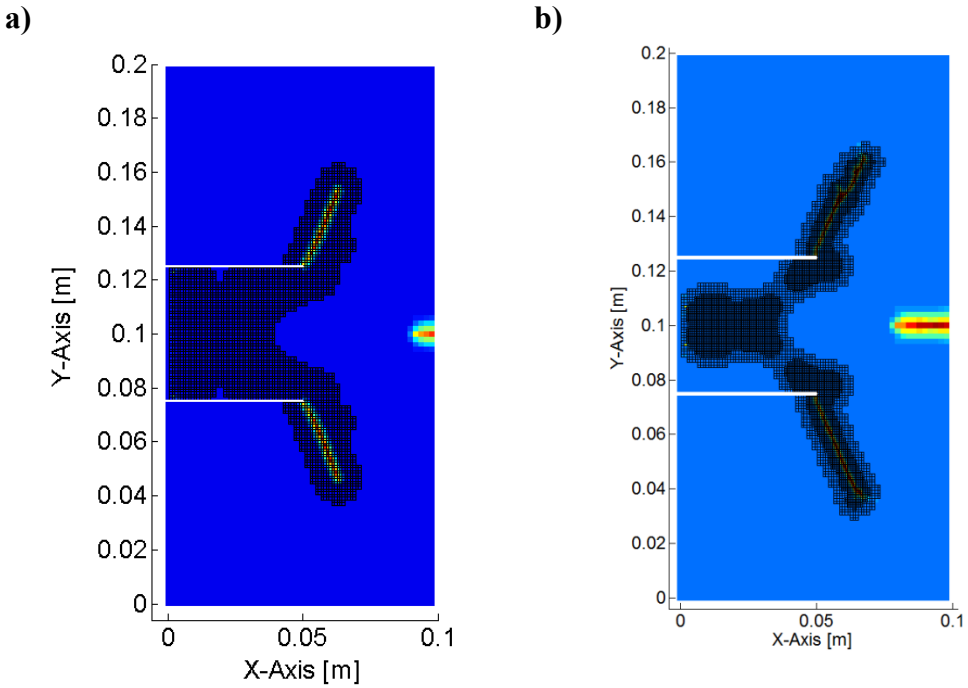


Fig. 6.11: Refined zone of the adaptive models in which only the energy-based trigger is used, a) model with the 1st level of refinement/scaling, b) model with the 2nd level of refinement/scaling.

6.3. Traction of pre-cracked plate with a hole

The case under examination has been considered by various authors: reference [93] used the FEM based method called Arbitrary Local Mesh Replacement, reference [94] used the explicit FEM code DYNA3D, whereas [90] used the Truss-Like Discrete Element Method. The problem is about a rectangular plate with a lateral pre-crack and a circular hole, as shown in Fig. 6.12a. The bottom side of the plate is clamped and on the opposite upper side a distributed traction is applied. The aim of the simulations is to examine how the crack path is affected by the presence of the hole as the distance between pre-crack and hole is varied. Such distance is measured by the quantity h shown in Fig. 6.12a. Three different configurations are analysed: h = 0.015m (model A), h = 0.01m (model B) and h = 0.005m (model C). The BB-PD models contain a 2D grid of 1,238 nodes (see Fig. 6.12b). The mechanical properties of the material are E = 71.4GPa, ρ = 2,700 kg/m^3 and G_0 = 1,000 J/m^2, while other input parameters for the two refinement levels are given in Table 6.6. Plain strain conditions are assumed [93] and the external load is applied as a load step kept constant during the whole simulation duration.

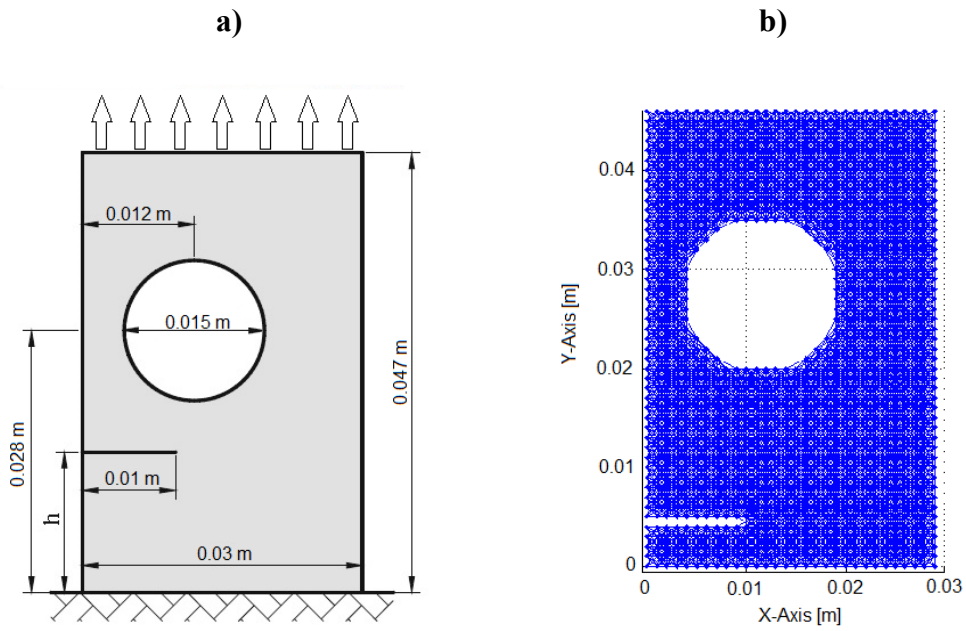


Fig. 6.12: a) Setup of the pre-cracked plate with a hole, b) visualisation of the bonds of model C.

Level	ΔX (mm)	δ (mm)	c_o $[10^{18} \cdot N/m^6]$	$10^{-3} \cdot s_o$
0	1.00	3.00	8.08	2.47
1	0.50	1.50 – **2.1**	64.7 – **22.90**	3.50–**2.94**

Table 6.6: Peridynamic grid parameters for the nodes of the grid (values in bold character are those of the interface nodes with modified horizon).

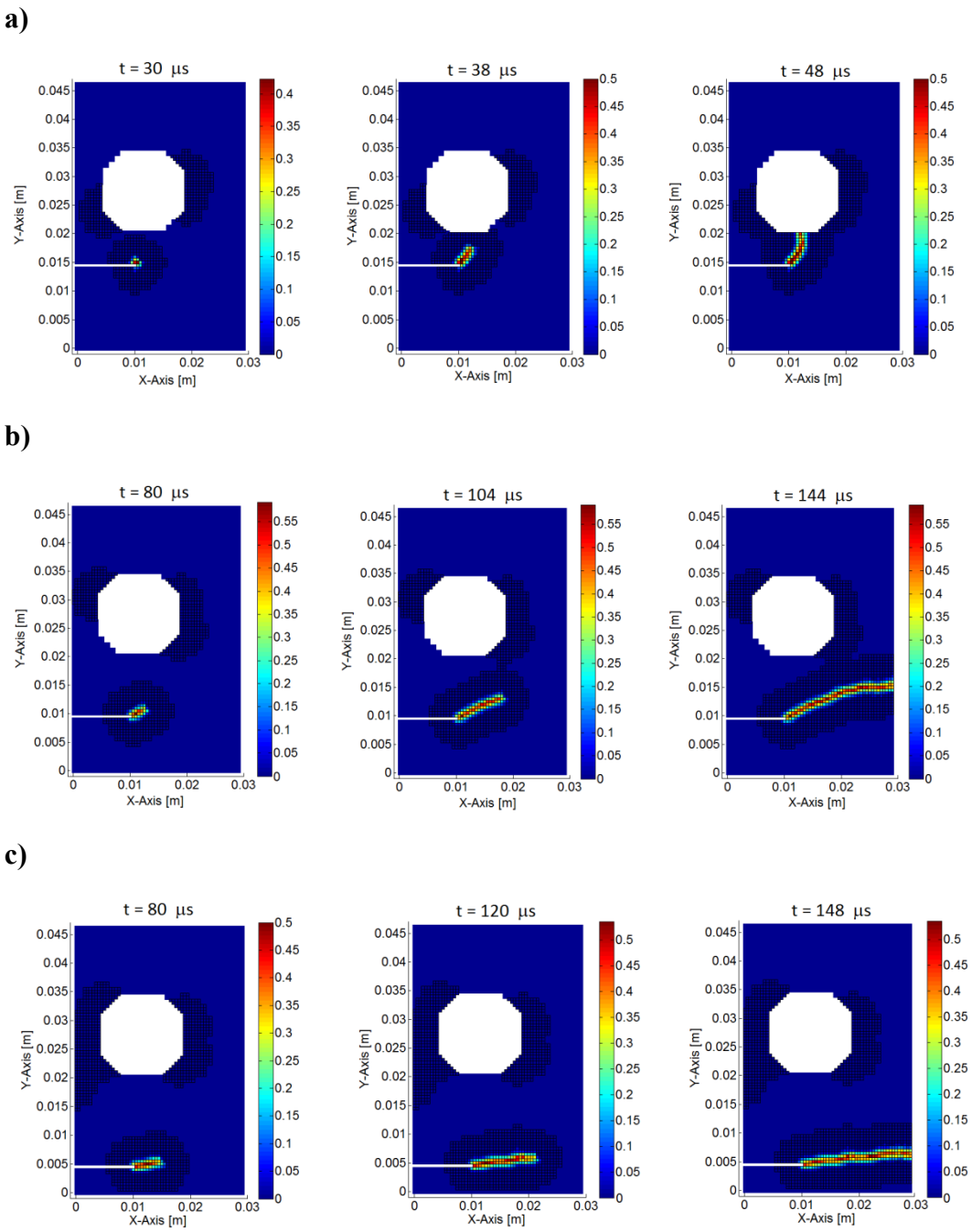


Fig. 6.13: Three snap-shots showing the temporal evolution of the damage level in the adaptive models: a) model A with 1,238–1,860 nodes (h =0.015m), b) model B with 1,238–2,340 nodes (h =0.010m), c) model C with 1,238–2,522 nodes (h =0.005m).

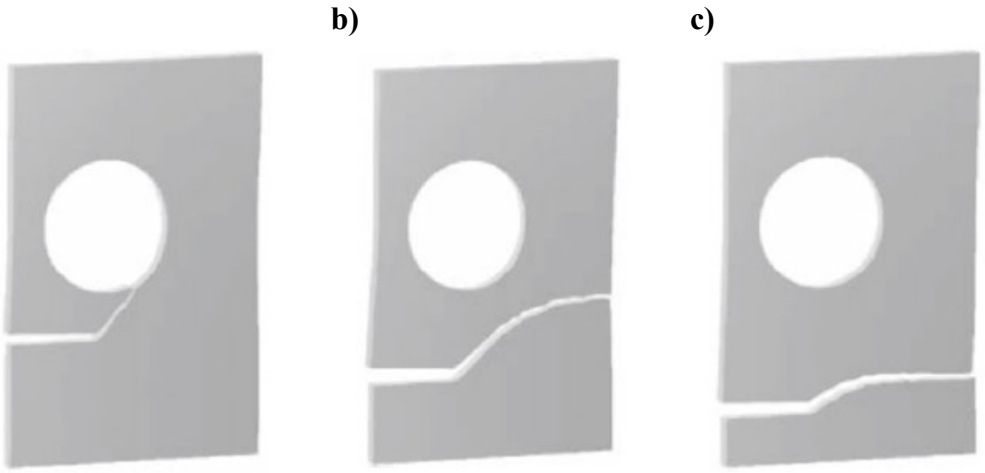


Fig. 6.14: The crack path according to [94]: a) model A (h =0.015m), b) model B (h =0.010m), c) model C (h =0.005m).

Due to the lack of available data about the load used in this numerical example, the tractions applied to the upper side have been chosen in order to induce crack propagation in the three models: 22MPa for case A, 26MPa for case B and 28MPa for case C. The time step Δt = 40ns is chosen, the refinement is activated with both energy trigger $W \geq 0.5W_{max}$ and damage trigger $\Delta\phi > 0$ applied from time t_{st} = 1.92μs for the models A–C and t_{st} =1.80μs for the model B. As mentioned in Sect. 3.2.2, t_{st} is determined by carrying out a preliminary analysis on the coarse grid. In this numerical example the preliminary analysis has been run for a duration of t_{pr} =9.9μs, a small fraction of the solution time duration given in Fig. 6.13 which shows a few images of the progression of the damage in the plate for the three cases. The figures show that the AGRS is activated precisely where the strain gradient is bigger, around the hole, and at the tip of the crack which is followed by the refined zone along its path. Fig. 6.14 shows the results obtained in [94] the overall features of Fig. 6.13 are similar to those of Fig. 6.14 although some differences can be noted. However, it has to be observed that in our models the hole is not represented accurately because the grid is rather coarse. As a consequence of that the distribution of the 'stress' concentration in the grid might be slightly inaccurate. A better description of the hole without a dramatic increase of the node number would require an irregular grid which is not a topic of the present work.

7. Conclusions

The application of adaptive grid refinement and scaling to Peridynamic to analyse dynamic crack propagation in brittle materials shows that this technique allows for a considerable reduction in the use of computational resources with respect to those required by uniform grids. This aspect is of fundamental importance for peridynamics, because, being nonlocal, it requires much more resources than the discretized implementations of the classical theory, such as the Finite Element Method.

The adaptive refinement is activated in the regions where energy concentrations occur and the damage is likely to increase. A new refinement trigger based on the damage state of the coarse grid is coupled with the energy based trigger, previously proposed in literature. The energy trigger plays a fundamental role in the activation of the refinement in regions where damage does not take place but high gradients of strain energy are present. However, the introduction of the damage based trigger enables the activation of the refinement process whenever the energy trigger fails to identify a region as a reasonable area for the crack propagation. This situation may happen, for example, when several cracks are simultaneously present in the analysed domain and different energy concentrations are associated to each crack.

One of the proposed numerical examples (see Section 6.2) shows that the coupled triggers help reducing the size of the refined region and therefore improve the efficiency of the method. The examples presented in the dissertation (see Section 6.1) show that the adaptive refinement and scaling correctly capture complex phenomena such as crack branching in both 2D and 3D models, and reduce the CPU time with respect to that required by a uniform grid providing a similar accuracy. Results are in good agreement with those obtained with other computational methods or in experimental tests.

Problems arising from the use of the refinement/scaling technique are addressed. They are related to the anomalies introduced in the numerical peridynamic solution due to the varying horizon and grid spacing over the domain. Moreover the use of AGRS can improve the efficiency of the numerical

procedures proposed to reduce the dependence of crack propagation on grid orientation.

Regarding the numerical errors, static analyses have been performed and the numerical peridynamic solutions have been compared to the linear elastic solution (see Section 4.1). These analyses reveal an important aspect concerning the role played by ghost forces in PD solutions. In fact, the results suggest that the anomalies in the displacement field are not strictly related to the presence of the ghost forces. They may be introduced by the formulation used to derive the material peridynamic parameters associated to a varying horizon over the domain. This topic requires further investigations. In addition, dynamic analyses regarding the propagation of continuum waves in 2D plates have been carried out (see Section 4.2). The analyses reveal the presence of spurious reflections and distortions of the elastic waves exhibited when they pass through the boundary region between areas of the discretized grid with different grid spacing and horizon. Some ideas have been presented on how to reduce such a spurious reflections and distortions, for instance, assigning a variable horizon to the interface nodes.

The second problem concerns the dependence of crack propagation on grid orientation, which could be a strong shortcoming for a software based on peridynamics (see Chapter 5). The problem has been investigated by means of a benchmark 2D example of crack propagation, showing that regular peridynamic grids suffer from dependence of crack paths on grid orientation. Such dependence is due to the existence in the discretized grid of directions along which the peridynamic material shows preferred paths for crack propagation. In particular, cracks often propagate along the weaker direction closest to the expected direction. It has been shown that such dependence persists regardless of the nature, either dynamic or quasi-static, of the load. In general, the deviation of the crack path from the correct direction is bigger for dynamic load applications. This seems to be mainly due to the interaction between the waves generated by the dynamic load and the growing crack. A possible way to reduce the dependence of crack path on grid orientation is to adopt an m-ratio of 6-7, which seems to provide reasonably accurate solutions for the analysed systems. The use of the

adaptive grid refinement and scaling algorithm considerably improves the computational efficiency by limiting the use of high m values only to regions where cracks are actually propagating.

7.1. Future developments

It must be pointed out that several future activities should be carried out in order to:

- improve the robustness and the efficiency of the adaptive grid refinement and scaling algorithm;
- investigate the problems which arise when a non-uniform grid with a varying horizon is employed for numerically implementing peridynamics.

It would be useful to develop an algorithm based on a refinement and *coarsening* technique. In fact, the refinement strategy reduces the grid spacing in the regions of interest by adding permanently new nodes as soon as the refinement is activated during the simulation. As a result of doing that, the areas of the grid behind the crack tip are kept refined throughout the simulation, leading to an increase in the computational cost of the overall simulation. On the contrary, the refinement and coarsening technique is based on the concept of refining the region around the crack tip, removing the nodes left behind at every update of the grid. Intuitively, due to the non-locality of the peridynamic formulation, several issues arise when the coarse nodes near the crack path are recovered in the grid. As an example, we should take into account the influence of the change of the horizon size close to the boundary of the crack path on the crack propagation: for example, an increase of the horizon size leads to an increase in the softening of the material due to the skin effect. Therefore, it would be interesting to evaluate the effect of the interaction between the elastic waves and the coarse boundaries of the crack on peridynamic solution. However, before removing the refined nodes from the region in which the crack takes place, a specific function to keep track of the refined crack path should be adopted. Refinement and coarsening are both affected by the need to define displacements, velocities, masses and other physical quantities in previously non-existing nodes. Interpolation techniques, which play a crucial role in that, should be further investigated.

Some results of static analyses presented in this dissertation highlight that the use of a varying horizon over the domain introduces anomalies in the displacement field, regardless of whether the local equilibrium is satisfied in the region in which the horizon size varies (see Section 4.1). Hence, further investigations concerning the origin of such anomalies should be carried out. It should be noticed that the scaling formulation proposed in the literature defines the peridynamic properties to be assigned to the bonds by means of expressions derived in the original peridynamic formulation, which implicitly assumes that the horizon is constant over the domain. Therefore, specific mathematical relations should be introduced in order to reduce or eliminate such anomalies.

One of the potentialities offered by peridynamics is the possibility to capture the material response at different length-scales of the material or of the analysed phenomena. The adaptive grid refinement and scaling pave the way to concurrent multi-scale models in a unique framework by automatically reducing the length-scale in those regions where the crack propagation takes place. For instance, this strategy would allow to capture microscopic processes that control dynamic fracture in brittle materials, such as microcracking phenomena. This process is one of the main mechanisms of energy dissipation in the zones near the crack tip observed experimentally which controls dynamic crack propagation. Using a uniform grid with a constant horizon would require a big amount of computational resources.

Another problem that should be examined thoroughly concerns the dependence of crack propagation on grid orientation in 3D models. As stated previously, the issues arose by carrying out 2D analysis are expected to be relevant for 3D cases as well. As a matter of fact, in 3D the bonds connected to a central node will most probably define planes along which cracks can propagate by intersecting, and therefore breaking, a given set of bonds, whereas a crack propagating in a slightly inclined direction would intersect a larger set of bonds. Therefore, it is easier for the crack in the regular 3D peridynamic grid to propagate along a weaker plane than in other directions close to it. It is reasonable to assume that increasing the value of m in a 3D regular grid would reduce the dependence of the crack pattern on the grid orientation. Anyway, further 3D benchmark problems regarding both

static and dynamic fracture should be carried out in order to validate this hypothesis.

Finally, another possible future development regards the extension of the adaptive refinement and scaling technique to the application of non-uniform unstructured grids, which are more suitable to model complex geometries.

References

[1] A. A. Griffith, *The Phenomena of Rupture and Flow in Solids*, Philosophical Transactions of the Royal Society of London: Series A, vol. 221, pp. 163-198, 1920.

[2] K. Ravi-Chandar, *Dynamic fracture*, Boston: Elsevier, 2004.

[3] S. A. Silling, *Reformulation of elasticity theory for discontinuities and long-range forces*, J Mech Phys Solids, vol. 48, pp. 175-209, 2000.

[4] F. Bobaru and W. Hu, *The meaning, selection, and use of the peridynamic horizon and its relation to crack branching in brittle materials*, Int J Fract, vol. 176, pp. 215-222, 2012.

[5] H. Liebowitz, *Fracture Mechanics of Aircraft Structures*, T. E. a. R. Ltd, Ed., London: North Atlantic Treaty Organization, 1974.

[6] T. Anderson, *Fracture Mechanics*, Third Edition ed., T. &. F. Group, Ed., Boca Raton: CRC Press, 2005.

[7] C. E. Inglis, *Stresses in a plate due to the presence of cracks and sharp corners*, Transactions of the Institute of Naval Architects, vol. 55, pp. 219-241, 1913.

[8] N. F. Mott, *Brittle fracture in Mild Steel Plates*, Engineering, vol. 165, pp. 16-18, 1948.

[9] D. S. Dugdale, *Yielding of steel sheets containing slits*, J Mech Phys Solids, vol. 8, no. 2, pp. 100-104, 1960.

[10] G. I. Barenblatt, *The mathematical theory of equilibrium cracks in brittle fracture*, Adv Appl Mech, vol. 7, pp. 55-129, 1962.

[11] Y. Mi, M. A. Crisfield, G. A. O. Davies and H. B. Hellweg, *Progressive delamination using interface elements*, J Compos Mater, vol. 32, no. 14, pp. 1246-1272, 1998.

[12] J. J. Muñoz, U. Galvanetto and P. Robinson, *On the numerical simulation of fatigue driven delamination with interface elements*, Int J Fatigue, vol. 28, no. 10, pp. 1136-1146, 2006.

[13] B. Yang, S. Mall and K. Ravi-Chandar, *A cohesive zone model for fatigue crack growth in quasibrittle materials*, Int J Solids Struct, vol. 38, no. 22-23,

pp. 3927-3944, 2001.

[14] G. T. Camacho and M. Ortiz, *Computational modelling of impact damage in brittle materials*, Int J Solids Struct, vol. 33, no. 20-22, pp. 2899-2938, 1996.

[15] J. H. Song, H. Wang and T. Belytschko, *A comparative study on finite element methods for dynamic fracture*, Comput Mech, vol. 42, no. 2, pp. 239-250, 2008.

[16] T. Belytschko and T. Black, *Elastic crack growth in finite elements with minimal remeshing*, Int J Numer Meth Eng, vol. 45, no. 5, pp. 601-620, 1999.

[17] J. Melenk and I. Babuška, *The Partition of Unity Finite Element Method: Basic Theory and Applications*, Comput Meth Appl Mech Eng, vol. 139, pp. 289-314, 1996.

[18] T. Rabczuk, *Computational Methods for Fracture in Brittle and Quasi-Brittle Solids: State-of-the-Art Review and Future Perspectives*, ISRN App Math, vol. 2013, p. 38 pages, 2013.

[19] B. Schmidt, F. Fraternali and M. Ortiz, *Eigenfracture: An Eigendeformation Approach to Variational Fracture*, Multiscale Model Simul, vol. 7, no. 3, p. 1237–1266, 2009.

[20] B. Bourdin, G. A. Francfort and J. J. Marigo, *The Variational Approach to Fracture*, J Elasticity, vol. 91, pp. 5-148, 2008.

[21] A. Pandolfi and M. Ortiz, *An eigenerosion approach to brittle fracture*, Int J Numer Meth Engng, vol. 92, no. 8, p. 694–714, 2012.

[22] B. Li, A. Pandolfi and M. Ortiz, *Material-point erosion simulation of dynamic fragmentation of metals*, Mechanics of Materials, vol. 80, p. 288–297, January 2015.

[23] M. J. Borden, C. V. Verhoosel, M. A. Scott, T. J. R. Hughes and C. M. Landis, *A phase-field description of dynamic brittle fracture*, Comput Methods Appl Mech Eng, Vols. 217-210, p. 77–95, 2012.

[24] M. F. Wheeler, T. Wick and W. Wollner, *An augmented-Lagrangian method for the phase-field approach for pressurized fractures*, Comput Methods Appl Mech Eng, vol. 271, p. 69–85, 2014.

[25] C. Miehe, M. Hofacker and F. Welschinger, *A phase field model for rate-independent crack propagation:Robust algorithmic implementation based on operator splits*, Comput Methods Appl Mech Eng, vol. 199, no. 45-48, p.

2765–2778, 2010.

[26] S. J. Zhou, P. S. Lomdahl, R. Thomson and B. L. Holian, *Dynamic crack processes via molecular dynamics*, Phys Rev Lett, vol. 76, no. 13, p. 2318–2321, 1996.

[27] K. Kadau, T. C. Germann and P. S. Lomdahl, *Molecular dynamics comes of age: 320 billion atom simulation on BlueGene/L*, Int J Mod Phys, vol. 17, no. 12, pp. 1755-1761, 2006.

[28] J. D. Lee, X. Q. Wang and Y. P. Chen, *Multiscale material modeling and its application to a dynamic crack propagation problem*, Theor Appl Fract Mech, vol. 51, no. 1, p. 33–40, 2009.

[29] V. B. Shenoy, R. Miller, E. B. Tadmor, D. Rodney, R. Phillips and M. Ortiz, *An adaptive finite element approach to atomic scale mechanics: the quasi continuum method*, J Mech Phys Solids, vol. 47, no. 3, p. 611–642, 1999.

[30] Z. P. Bažant and M. Jirásek, *Nonlocal integral formulations of plasticty and damage: survey of progress*, J Eng Mech, vol. 128, no. 11, pp. 1119-1149, 2002.

[31] A. C. Eringen and D. G. B. Edelen, *On nonlocal elasticity*, Int J Eng Sci, vol. 10, no. 3, pp. 233-248, 1972.

[32] R. H. J. Peerlings, R. De Borst, W. A. M. Brekelmans and J. H. P. De Vree, *Gradient enhanced damage for quasi-brittle materials*, Intl J Num Meth Engin, vol. 39, no. 19, p. 3391–3403, 1996.

[33] I. A. Kunin, *Elastic media with microstructure I: one dimensional models, Berlin*: Springer, 1982.

[34] I. A. Kunin, *Elastic media with microstructure II: three-dimensional models*, Berlin: Springer, 1983.

[35] D. Rogula, *Nonlocal theory of material media*, Berlin: Springer, 1982, pp. 137-243.

[36] S. A. Silling, M. Epton, O. Weckner, J. Xu and E. Askari, *Peridynamic states and constitutive modeling*, J Elast, vol. 88, no. 2, p. 151–184, 2007.

[37] E. Askari, F. Bobaru, R. B. Lehoucq, M. L. Parks, S. A. Silling and O. Weckner, *Peridynamics for multiscale materials modeling*, J Phys Conf Ser, vol. 125, no. 1, p. 1–11, 2008.

[38] A. E. H. Love, *Mathematical Theory of Elasticity*, New York: Dover, pp. 616-627.

[39] W. Gerstle, S. A. Silling, D. Read, V. Tewary and R. Lehoucq, *Peridynamic simulation of electromigration*, Comput Mater Continua, vol. 8, no. 2, pp. 75-92, 2008.

[40] B. Kilic, *Peridynamic theory for progressive failure prediction in homogeneous and heterogeneous materials*, PhD Dissertation, Department of Aerospace and Mechanical Engineering, 2008.

[41] E. Emmrich and O. Weckner, *The peridynamic equation and its spatial discretisation*, Math. Model. Anal., vol. 12 , no. 1, p. 17–27, 2007.

[42] S. A. Silling and E. Askari, *A meshfree method based on the peridynamic model of solid mechanics*, Comput Struct, vol. 83, no. 17-18, pp. 1526-1535, 2005.

[43] Y. D. Ha and F. Bobaru, *Studies of dynamic crack propagation and crack branching with peridynamics*, Int J Fract, vol. 162, no. 1, p. 229– 244, 2010.

[44] Y. D. Ha and F. Bobaru, *Characteristics of dynamic brittle fracture captured with peridynamics*, Eng Fract Mech, vol. 78, no. 6, pp. 1156-1168, 2011.

[45] F. Bobaru and G. Zhang, *Why do cracks branch? A peridynamic investigation of dynamic brittle fracture*, Int J Fract, vol. 196, no. 1, p. 59–98, 2015.

[46] W. Gerstle, N. Sau and S. A. Silling, *Peridynamic modeling of plain and reinforced concrete structures*, In: SMiRT 18, Int Conf Struct Mech React Technol, 7-12 August 2005.

[47] W. Gerstle, N. Sau and S. Silling, *Peridynamic modeling of concrete structures*, Nuclear Engineering and Design, vol. 237, no. 12-13, p. 1250–1258, 2007.

[48] E. Oterkus, E. Madenci, O. Weckner, S. Silling, P. Bogert and A. Tessler, *Combined finite element and peridynamic analyses for predicting failure in a stiffened composite curved panel with a central slot*, Compos Struct, vol. 94, no. 3, pp. 839-850, 2012.

[49] E. Oterkus, *Peridynamic Theory For Modeling Three-Dimensional Damage Growth in Metallic and Composite Structures*, Ph.D. Dissertation, The University of Arizona, 2010.

[50] B. Kilic, A. Agwai and E. Madenci, *Peridynamic theory for progressive damage prediction in center-cracked composite laminates*, Compos Struct, vol. 90, no. 2, p. 141–151, 2009.

[51] M. Zaccariotto, F. Luongo, G. Sarego, D. Dipasquale and U. Galvanetto, *Fatigue crack propagation with peridynamics: a sensitivity study of Paris law parameters*, in CEAS 2013 Innov Eur Sweden, 2013.

[52] S. A. Silling and A. Askari, *Peridynamic Model for Fatigue Cracking*, Sandia Report, Albuquerque, New Mexico, 2014.

[53] Z. Chen and F. Bobaru, *Peridynamic modeling of pitting corrosion damage*, J Mech Phys Solids, vol. 78, p. 352–381, May 2015.

[54] D. De Meo, C. Diyaroglu, N. Zhu and E. Oterkus, *Modelling of stress-corrosion cracking by using peridynamics*, Int J Hydrogen Energ, vol. 41, no. 15, p. 6593–6609, April 2016.

[55] R. W. Macek and S. A. Silling, *Peridynamics via finite element analysis*, Finite Elem Anal Des, vol. 43, no. 15, p. 1169–1178, 2007.

[56] U. Galvanetto, T. Mudric, A. Shojaei and M. Zaccariotto, *An effective way to couple FEM meshes and Peridynamics grids for the solution of static equilibrium problems*, Mech Res Commun, vol. 76, pp. 41-47, 2016.

[57] F. Bobaru, M. Yang, L. F. Alves, S. A. Silling, E. Askari and J. Xu, *Convergence, adaptive refinement, and scaling in 1D peridynamics*, Int J Numer Methods Eng, vol. 77, no. 6, p. 852–877, 2009.

[58] F. Bobaru and Y. D. Ha, *Adaptive refinement and multiscale modeling in 2D peridynamics*, J Multiscale Comput Eng, vol. 9, no. 6, pp. 635-659, 2011.

[59] D. Dipasquale, M. Zaccariotto and U. Galvanetto, *Crack propagation with adaptive grid refinement in 2D peridynamics*, Int J Fract, vol. 190, no. 1, pp. 1-22, 2014.

[60] H. Ren, X. Zhuang, Y. Cai and T. Rabczuk, *Dual-horizon Peridynamics*, Int J Numer Methods Eng, pp. 1-25, 2015.

[61] S. A. Silling and R. B. Lehoucq, *Peridynamic Theory of Solid Mechanics*, Adv Appl Mech, vol. 44, p. 73–168, 2010.

[62] Q. V. Le, W. K. Chan and J. Schwartz, *A two-dimensional ordinary, state-based peridynamic model for linearly elastic solids*, Int J Numer Meth Eng, vol. 98, no. 8, pp. 547-561, 2014.

[63] B. Kilic and E. Madenci, *Coupling of Peridynamic Theory and the Finite Element Method*, J Mech Materials Struct, vol. 5, no. 5, pp. 707-733, 2010.

[64] J. A. Mitchell, S. A. Silling and D. J. Littlewood, *A Position-Aware Linear Solid Constitutive Model for Peridynamics*, Journal of Mechanics of Materials and Structures, vol. 10, no. 5, pp. 539-557, 2015.

[65] S. Oterkus, E. Madenci and A. Agwai, Fully coupled peridynamic thermomechanics, *J Mech Phys Solids,* vol. 64, pp. 1-23, 2014.

[66] E. Madenci and E. Oterkus, *Peridynamic Theory and Its Applications*, New York: Springer, 2014.

[67] S. A. Silling, D. J. Littlewood and P. Seleson, *Variable horizon in a Peridynamic medium*, J Mech Materials Struct, vol. 10, no. 5, pp. 591-612, 2015.

[68] P. Seleson, *Improved one-point quadrature algorithms for two-dimensional peridynamic models based on analytical calculations*, Comput Methods Appl Mech Engrg, vol. 282, pp. 184-217, 2014.

[69] M. L. Parks, R. B. Lehoucq, S. J. Plimpton and S. A. Silling, *Implementing peridynamics within a molecular dynamics code*, Comput Phys Commun, vol. 179, no. 11, p. 777–783, 2008.

[70] E. Hairer, C. Lubich and G. Wanner, *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, 2nd ed., Springer Series in Computational Mathematics, 2006.

[71] E. Rank and O. C. Zienkiewicz, *A simple error estimator in the finite element method*, Int J Numer Meth Biomed Eng, vol. 3, no. 3, p. 243–249, 1987.

[72] G. M. Hunter and K. Steiglitz, *Operations on images using quadtrees*, Pattern Anal Mach Intell, vol. 1, no. 2, p. 145–153, 1979.

[73] H. Samet, *An overview of quadtrees, octrees and related hierarchical data structures*, Theor Found Comput Graph CAD, vol. 40, p. 51–68, 1988.

[74] J. T. Oden and S. Prudhomme, *Goal-oriented error estimation and adaptivity for the finite element method*, Comput Math Appl, vol. 41, no. 5-6, p. 735–756, 2001.

[75] Z. P. Bažant, *Spurious reflection of elastic waves in nonuniform finite element grids*, Comput Method Appl M, vol. 16, no. 1, p. 91–100, 1978.

[76] Z. Celep and P. Bažant, *Spurious reflection of elastic waves due to gradually changing finite element size*, Int J Numer Methods Eng, vol. 19, p. 631–646, 1983.

[77] G. R. Liu, *Mesh Free Methods: moving beyond the finite element method*, Boca Raton, London, New York, Washington D.C.: CRC PRESS LLC, 2003.

[78] P. Krysl and T. Belytschko, *Element-Free Galerkin Method: Convergence of the continuos and discontinuous shape functions*, Comput Method Appl M, vol. 148, no. 3-4, pp. 257-277, 1997.

[79] O. Weckner, S. Silling and A. Askari, *Dispersive Wave Propagation in the Nonlocal Peridynamic Theory*, ASME 2008 International Mechanical Engineering Congress and Exposition, Vols. 12: Mechanics of Solids, Structures and Fluids, pp. 503-504, 2008.

[80] P. Seleson and M. L. Parks, *On the Role of the Influence Function in the Peridynamic Theory*, Int J Multiscale Comput Eng, vol. 9, p. 689–706, 2011.

[81] R. Rahman and J. T. Foster, *Onto resolving spurious wave reflection problem with changing nonlocality among various length scales*, Commun Nonlinear Sci Numer Simulat, vol. 34, pp. 86-122, 2016.

[82] G. Sarego, *Structural material damage: novel methods of analysis*, PhD Dissertation, Industrial Department at Univerity of Padua, 2016.

[83] S. F. Henke and S. Shanbhag, *Mesh sensitivity in peridynamic simulations*, Comput Phys Commun, vol. 185, no. 1, p. 181–193, 2014.

[84] T. L. Warren, S. A. Silling, A. Askari, O. Weckner, M. A. Epton and J. Xu, *A non-ordinary state-based peridynamic method to model solid material deformation and fracture*, Int J Solids Struct, vol. 46, no. 5, pp. 1186-1195, 2009.

[85] F. Bowden, H. Brunton, J. E. Field and A. Heyes, *Controlled fracture of brittle solids and interruption of electrical current*, Nature, vol. 216, no. 5110, pp. 38-42, 1967.

[86] J. E. Field, *Brittle fracture:its study and application*, Contemp Phys, vol. 12, no. 1, p. 1–3, 1971.

[87] W. Döll, *Investigation of the crack branching energy*, Int J Fract, vol. 11, p. 184–186, 1975.

[88] T. Belytschko, H. Chen, J. Xu and G. Zi, *Dynamic crack propagation based on loss of hyperbolicity and a new discontinuos enrichment*, Int J Numer

Meth Eng, vol. 58, no. 12, pp. 1873-1905, 2003.

[89] J. F. Kalthoff, *Modes of dynamic shear failure in solids*, Int J Fract, vol. 101, no. 1, pp. 1-31, 2000.

[90] L. Kosteski, B. R. D'Ambra and I. Iturrioz, *Crack propagation in elastic solids using the truss-like discrete element method*, Int J Fract, vol. 174, no. 2, p. 139–161, 2012.

[91] S. A. Silling, *Peridynamic Modeling of the Kalthoff–Winkler experiment*, Submission for the 2001 Sandia Prize in computational science, 2001 Sandia Prize in computational science.

[92] K. Ravi-Chandar, *Dynamic fracture of nominally brittle materials*, Int J Fract, vol. 90, no. 1, pp. 83-102, 1998.

[93] M. M. Rashid, *The arbitrary local mesh replacement method: An alternative to remeshing for crack propagation analysis*, Comput Methods Appl Mech Eng, vol. 154, no. 1-2, p. 133–150, 1998.

[94] A. Tabiei and J. Wu, *Development of the DYNA3D simulation code with automated fracture procedure for brick elements*, Int J Numer Methods Eng, vol. 57, no. 14, p. 1979–2006, 2003.

Appendix A
Adaptive grid refinement and scaling algorithms

In this Appendix, we present the pseudo-codes of the algorithms developed to implement the AGRS. Initially, a brief overview of the main data structure is given in the following:

- The "*Grid matrix*" stores the geometrical properties of the nodes such as their absolute position with respect to the reference system, the assigned volume, the ID number to identify the type of material of the volume assigned to the node, the horizon length and finally the ID number to detect the level of refinement of the node, as shown in the following for 2D cases:

$$grid = \begin{array}{c} 1 \\ 2 \\ \vdots \\ \vdots \\ N-1 \\ N \end{array} \begin{bmatrix} X_1 & Y_1 & \Delta V_1 & ID_{mat,1} & \delta_1 & ID_{ref,1} \\ X_2 & Y_2 & \Delta V_2 & ID_{mat,2} & \delta_2 & ID_{ref,2} \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ X_{N-1} & Y_{N-1} & \Delta V_{N-1} & ID_{mat,N-1} & \delta_{N-1} & ID_{ref,N-1} \\ X_N & Y_N & \Delta V_N & ID_{mat,N} & \delta_N & ID_{ref,N} \end{bmatrix} \quad \text{(A.1)}$$

with N the number of nodes of the grid. We can note that there is no need to store the ID number of the nodes since they are identified by the index of their corresponding row, in this way it is possible to save memory space. The number of columns of the grid matrix may vary between 6 columns and 7 depending on whether we are dealing with 2D or 3D analysis, respectively. In the latter case the Z coordinates of the nodes are added in the 3rd column. Taking into account that the grid matrix has to be updated at each step of the analysis in which new nodes are added to the grid, we propose the following strategy: if N_i represents the initial number of nodes of the grid, and N_f the number of nodes of the grid at the end of the analysis, including activated and deactivated nodes (see Sect. 3.2.3), then the row size N will vary between N_i and N_f. The addition of new nodes involves the updating of the properties of the corresponding parent nodes, such as the volume, the horizon and the level of refinement assigned to them, therefore, at each updating the new properties

in the grid matrix are overwritten on the previous one. As a consequence, three matrixes referred as "*volume updating*", "*horizon updating*" and "*level updating*" are adopted in order to keep track of the history of the properties required to display the results during the post-processing phase. As an example, the volume updating matrix assumes the form:

$$VolUpdating = \begin{bmatrix} \Delta V_1^1 & \cdots & \Delta V_1^{n-1} & \Delta V_1^n \\ \Delta V_2^1 & \cdots & \Delta V_2^{n-1} & \Delta V_2^n \\ \vdots & \vdots & \vdots & \vdots \\ \Delta V_{N_i-1}^1 & \vdots & \vdots & \vdots \\ \Delta V_{N_i}^1 & \vdots & \vdots & \Delta V_{N_f-1}^{n-1} \\ 0 & \vdots & \vdots & \Delta V_{N_f}^n \end{bmatrix} \tag{A.2}$$

- The "*bond matrix*" stores the list of the ID numbers of the family nodes in the initial configuration of the model associated with the corresponding source nodes of the grid, the data structure takes the form:

$$I_{bond} = \begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ \vdots \\ M-1 \\ M \end{array} \begin{array}{c} \overbrace{}^{ID_{source\_node}} \quad \overbrace{}^{ID_{family\_nodes}} \\ \begin{bmatrix} 1 & 2 & \cdots & \cdots & 1 & 1 \\ 3 & 1 & 4 & \cdots & \cdots & 3 \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ N-1 & \cdots & \cdots & \cdots & \cdots & \vdots \\ N & \cdots & \cdots & \cdots & \cdots & \vdots \end{bmatrix} \end{array} \tag{A.3}$$

with M the maximum row index whose value can be different from the maximum row index N of the grid matrix, while the maximum column index is given by the maximum number of bonds of the domain analyzed. Unlike the data structure of the grid matrix, in the bond matrix the source ID numbers are stored in the 1st column, the reason of that is clear by reading Sect. 3.2.3. It can be noted that the null elements are replaced with the source ID numbers in those rows in which the number of the family nodes is lower than the maximum column index of the matrix. As explained in [82], such a structure allows to take advantage of the matrix operations offered by Matlab rather than managing single sequences of operations like "*if*" conditions.

- The "*bond history matrix*", which will be called H_{bond} in the following, keeps track of the history of the health state of the bonds storing the time step for which the bonds breaks. The data structure is the same as that of the bond matrix, with the difference that the number ID of the nodes are replaced with 0 for unbroken bonds and with the time step number for the broken bonds.
- The "*Bond length component matrix*" stores the components with respect to the reference system of the relative distance between the family nodes and the corresponding source nodes. Two or three matrixes are separately built to store (X,Y) components for 2D cases and (X,Y,Z) components for 3D cases, respectively. For instance, with reference to 2D cases, the matrixes assume the following form:

$$X_{rel} = \begin{matrix} 1 \\ 2 \\ \vdots \\ \vdots \\ M-1 \\ M \end{matrix} \begin{bmatrix} \xi_{x(1,1)} & \xi_{x(1,2)} & \cdots & \cdots & \xi_{x(1,1)} & \xi_{x(1,1)} \\ \xi_{x(3,3)} & \xi_{x(3,1)} & \xi_{x(3,4)} & \cdots & \cdots & \xi_{x(3,3)} \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \xi_{x(N-1,N-1)} & \cdots & \cdots & \cdots & \cdots & \vdots \\ \xi_{x(N,N)} & \cdots & \cdots & \cdots & \cdots & \vdots \end{bmatrix} \tag{A.4a}$$

$$Y_{rel} = \begin{matrix} 1 \\ 2 \\ \vdots \\ \vdots \\ M-1 \\ M \end{matrix} \begin{bmatrix} \xi_{y(1,1)} & \xi_{y(1,2)} & \cdots & \cdots & \xi_{y(1,1)} & \xi_{y(1,1)} \\ \xi_{y(3,3)} & \xi_{y(3,1)} & \xi_{y(3,4)} & \cdots & \cdots & \xi_{y(3,3)} \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \xi_{y(N-1,N-1)} & \cdots & \cdots & \cdots & \cdots & \vdots \\ \xi_{y(N,N)} & \cdots & \cdots & \cdots & \cdots & \vdots \end{bmatrix} \tag{A.4b}$$

The subscripts (ID_{source},ID_{family}) indicate the relative position between the centroid of the family cell and its source node. Where the family ID number is replaced with the source ID number the components are null, being null the relative distance between them. By comparing Eq. (A.3) with Eqs. (A.4) it is clear to observe that the assigned properties in Eqs. (A.4) are exactly located in the position of the corresponding family and source nodes listed in Eq. (A.3), since the latter is used as a reference matrix. Then the matrixes in Eqs. (A.4) have the same size of the matrix in Eq. (A.3).

- The "*Volume correction factor matrix*" stores the correction factors (see Eq. (3.1) in Section 3.1) to be multiplied to the volume of corresponding family nodes, the data structure assumes the same structure of Eq. (A.3), as follows:

$$Beta = \begin{matrix} 1 \\ 2 \\ \vdots \\ \vdots \\ M-1 \\ M \end{matrix} \begin{bmatrix} \beta_{(1,1)} & \beta_{(1,2)} & \cdots & \cdots & \beta_{(1,1)} & \beta_{(1,1)} \\ \beta_{(3,3)} & \beta_{(3,1)} & \beta_{(3,4)} & \cdots & \cdots & \beta_{(3,3)} \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ \beta_{(N-1,N-1)} & \cdots & \cdots & \cdots & \cdots & \vdots \\ \beta_{(N,N)} & \cdots & \cdots & \cdots & \cdots & \vdots \end{bmatrix} \tag{A.5}$$

  The volume correction factors assigned to the volume of the source nodes are set to zero.

- In order to store the interested quantities such as the displacement space field, the density of potential energy space field and the damage space field at each time step of the analysis, the data structure of the "*displacement field matrix*" assumes for 2D cases the following form:

$$U = \begin{bmatrix} {u^1}_{X,1} & {u^1}_{Y,1} & \cdots & {u^{n-1}}_{X,1} & {u^{n-1}}_{Y,1} & {u^n}_{X,1} & {u^n}_{Y,1} \\ {u^1}_{X,2} & {u^1}_{Y,2} & \cdots & {u^{n-1}}_{X,2} & {u^{n-1}}_{Y,2} & {u^n}_{X,2} & {u^n}_{Y,2} \\ \vdots & & & \vdots & & & \vdots \\ {u^1}_{X,N_i-1} & & & \vdots & & & \vdots \\ {u^1}_{X,N_i} & & & \vdots & & & {u^n}_{Y,N_f-1} \\ 0 & & & \vdots & & & {u^n}_{Y,N_f} \end{bmatrix} \tag{A.6}$$

  while for the "*density of potential energy matrix*":

$$W = \begin{bmatrix} W_1^1 & \cdots & W_1^{n-1} & W_1^n \\ W_2^1 & \cdots & W_2^{n-1} & W_2^n \\ \vdots & \vdots & \vdots & \vdots \\ W_{N_i-1}^1 & \vdots & \vdots & \vdots \\ W_{N_i}^1 & \vdots & \vdots & W_{N_f-1}^{n-1} \\ 0 & \vdots & \vdots & W_{N_f}^n \end{bmatrix} \tag{A.7}$$

The "*damage matrix*" *D* is not given since its structure is the same as the structure of Eq. (A.7).

- The "*Parent and child structure array*" allows to keep track of the relationship between the parent nodes and the generated child nodes. In Matlab language the "*structure array*" is a particular type of array data structure in which multiple fields can be stored, for each field the value input argument can be any data type with different sizes. In the following, an example of a such data structure is given for 2D cases and when the 2nd level of refinement is adopted:

$$PC(1).indexs = \begin{bmatrix} ID_{p,1} & ID_{c,10} & ID_{c,11} & ID_{c,12} & ID_{c,13} \\ ID_{p,2} & ID_{c,20} & ID_{c,21} & ID_{c,22} & ID_{c,23} \\ & & \vdots & & \\ & & \vdots & & \end{bmatrix}$$

$$PC(2).indexs = \begin{bmatrix} ID_{p,10} & ID_{c,100} & ID_{c,101} & ID_{c,102} & ID_{c,103} \\ ID_{p,11} & ID_{c,110} & ID_{c,111} & ID_{c,112} & ID_{c,113} \\ & & \vdots & & \\ & & \vdots & & \\ ID_{p,20} & ID_{c,201} & ID_{c,202} & ID_{c,203} & ID_{c,204} \\ & & \vdots & & \end{bmatrix} \quad \text{(A.8)}$$

The algorithms developed take advantage of some useful built-in functions of Matlab® (see manual of Matlab® for a detailed description), such as:

- *find(A)*: it finds indices and values of nonzero elements of the array A
- *ismember(A,B)*: it returns a logic array containing 1 (true) where the data in A is found in B, elsewhere, it returns 0 (false)
- *sum(A)*: it returns the sum of the elements of A along the first array dimension whose size does not equal 1
- *unique(A)*: it returns the same data as in A, but with no repetitions
- *reshape(A,sz)*: it reshapes the array A using the size vector *sz* by keeping the total number of elements constant
- *numel(A)*: it returns the number of elements of the array A

- "." : operator which allows to compute the element-wise operations on arrays, for instance, A.*B computes the element-wise multiplication between the arrays A and B
- *pinv(A)*: it returns the Penrose pseudo-inverse of matrix A.

Algorithm A.1: Implementation of the energy based trigger, see Section 3.2.1.

1: {Set up the threshold value of the density of potential energy}
2: $W_{thres} = \% \cdot \max(W(:,n))$
3: {Identify the nodes of the coarse grid whose the density of potential energy is higher than the threshold value}
4: $node_{coarse} = find(W(:,n) \geq W_{thres}\ and\ grid(:,6) = 0)$
{Initialize the internal variable to store temporarily the parent nodes belonging to the child nodes}
$node_{parent} = []$
5: {Identify the coarse parent nodes of the grid whose the density of potential energy of their child nodes is higher than the threshold value}
6: **for** $i = \text{level}_{max}: -1: 1$
7: $node_{child,i} = find(W(:,n) \geq W_{thres}\ and\ grid(:,6) = i)$
8: {Add to the list of child nodes identified in row 7 the list of the corresponding parent nodes}
9: $list = [\,node_{child,i}, node_{parent}])$
10: {Find the indexes of the rows in which the parent nodes are stored}
11: $logic_{vector} = ismember(PC(i).indexs(:,2:end), list)$
12: $rows = sum(logic_{vector}, 2) > 0$
13: $node_{parent} = unique(PC(i).indexs(rows, 1))$
14: **end**
15: {Store in the output variable the list of nodes identified}
16: $node_{identify} = [node_{coarse}, node_{parent}]$

Algorithm A.2: Implementation of damage based trigger, see Section 3.2.2

1: {Identify the nodes of the coarse grid whose damage is higher than the initial damage}
2: $node_{coarse} = find(D\,(:,n) \geq D\,(:,1)\ \&\ grid(:,6) == 0)$

{Initialize the internal variable to store temporarily the parent nodes belonging to the child nodes}

$node_{parent} = []$

3: {Identify the coarse parent nodes of the grid whose child nodes are damaged}

4: **for** $i = \text{level}_{max}: -1: 1$

5: $node_{child,i} = find(D(:,n) \geq D(:,1)\ \&\ grid(:,6) == i)$

6: {Add to the list of child nodes identified in the row 7 the list of the corresponding parent nodes}

7: $list = [\,node_{child,i}, node_{parent}])$

8: {Find the indexes of the rows in which the parent nodes are stored}

9: $logic_{vector} = ismember(PC(i).indexs(:,2:end), list)$

10: $rows = sum(logic_{vector}, 2) > 0$

11: $node_{parent} = unique(PC(i).indexs(rows, 1))$

12: **end**

13: {Store in the output variable the list of nodes identified}

14: $node_{identify} = [node_{coarse}, node_{parent}]$

Algorithm A.3: Implementation of the code for generating new nodes to add in the grid by using Method B (2D version), see Section 3.2.3.

1: {Copy the previous geometric matrix, list of bonds matrix, the initial horizon and grid spacing in temporary arrays}

2: $grid_{tem} = grid_{old}$

3: $I_{bond,tem} = I_{bond,old}$

4: $\Delta X_{tem} = \Delta x_0$

5: $\Delta Y_{tem} = \Delta y_0$

6: $\delta_{tem} = \delta_0$

7: {Generate the child nodes for each level of refinement by starting from the lowest level}

8: **for** $i = 1: \text{level}_{max}$

9: {As a first step **,** find the nodes which has to be refined**}**

10: {Proceed in a different way depending on the maximum level of refinement adopted in the analysis}

11: **if** $\text{level}_{max} > 1$

12: $node_{parent} = node_{identify}$

13: {Find the child nodes of *i*-th level belonging to the parent nodes identified by the trigger}

14: **for** $j = 1:(i-1)$

15: $rows = ismember(PC(j).indexs(:,1), node_{identify})$

16: $node_{parent} = unique(reshape(PC(j).indexs(rows, 2:end), 4 * sum(rows), 1)$

17: **end**

18: **else**

19: $nodes = node_{identify}$

20: **end**

21: $rows1 = ismember(I_{bond,tem}(:,1), nodes)$

22: {Select all the family nodes belonging to the identified source parent nodes}

23: $nodes_{parent} = unique(I_{bond}(row1,:))$

24: {As a second step**,** generate the new nodes by computing a loop for each parent node identified**}**

25: {Define counters to increase the row sizes of geometric matrix and parent and child structure array in the loop}

26: $k = numel\big(grid_{tem}(:,1)\big) + 1;$

27: $kk = numel\big(PC.(i)index(:,1)\big);$

28: **for** $j = 1:$ numel $(nodes_{parent})$

29: $XY_{parent} = grid_{tem}(1:2, nodes_{parent}(j))$

30: {Compute the corner coordinates of the area assigned to the parent node}

31: $XY_{corners} = [XY_{parent}(1) - \frac{\Delta X_{tem}}{2}, XY_{parent}(2) - \frac{\Delta Y_{tem}}{2}; \ldots$

$$XY_{parent}(1) + \frac{\Delta X_{tem}}{2}, XY_{parent}(2) - \frac{\Delta Y_{tem}}{2}; \ldots$$

$$XY_{parent}(1) + \frac{\Delta X_{tem}}{2}, XY_{parent}(2) + \frac{\Delta Y_{tem}}{2}; \ldots$$

$$XY_{parent}(1) - \frac{\Delta X_{tem}}{2}, XY_{parent}(2) + \frac{\Delta Y_{tem}}{2}]$$

{Compute the coordinates of the new child nodes}

$XY_{child} = [XY_{parent}(1) + XY_{corners}(:,1)./2, \ldots$

$XY_{parent}(2) + XY_{corners}(:,2)./2]$

{Store all the properties belonging to the new nodes in the geometric matrix}

$grid_{tem}(k:k+3,1:2) = XY_{child}$

$grid_{tem}(k:k+3,3) = \frac{\Delta x_{tem} \cdot \Delta y_{tem}}{4}$

$grid_{tem}(k:k+3,4) = 1$

$grid_{tem}(k:k+3,5) = \delta_{tem}/2$

$grid_{tem}(k:k+3,6) = i$

{Store the ID numbers of both the parent and child nodes in the parent and child array}

$PC.(i)index(kk+j,:) = [\, nodes_{parent}(j), k:k+3]$

end

{Update the properties of the parent nodes eliminated}

$grid_{tem}(nodes_{parent}, 3) = 0$

$grid_{tem}(nodes_{parent}, 4) = 0$

$grid_{tem}(nodes_{parent}, 6) = -i$

{Update temporary variables}

if $i \leq level_{max} - 1$

$\Delta X_{tem} = \Delta x_0/2^i$

$\Delta Y_{tem} = \Delta x_0/2^i$

$\delta_{tem} = \delta_0/2^i$

{Built the list of bonds only for the nodes with the same *'degree kinship'*}

$I_{bond,tem} = f(grid_{tem}, i)$

end

end

$grid_{new} = grid_{tem}$

Algorithm A4: Implementation of the PIM to interpolate the nodal displacement values in a 2D domain, the new nodes are generated by using the method B, see both Section 3.2.3 and Section 3.2.4.

1: {Calculate the difference between the level of refinement of the previous grid with that new one, in order to detect the parent nodes which has just generated new child nodes}

2: $diff = grid_{old}(:,6) - grid_{new}(:,6)$

3: {Compute a loop for each level of refinement applied}

4: **for** $i = 1:\text{level}_{max}$

5: {As a first step, identify the child nodes generated by the parent nodes which are not in the previous grid

6: $logic = (diff == (2i - 1))$

7: {Proceed if previous parent nodes have been found)

8: **if** $sum(logic) \geq 1$

9: $Index_{row} = 1:numel(griglia_{old}(:,1))$

10: $ID_{parent} = Index_{row}(logic)$

11: $logic_1 = ismember(PC(i).index(:,1), ID_{parent})$

12: $Index_{1,row} = 1:numel(PC(i).index(:,1))$

13: $ID_{child} = PC(i).index\big(Index_{1,row}(logic_1), 2:end\big)$

14: {Compute a loop for each identified parent node}

15: **for** $j = 1:\text{numel}(ID_{parent})$

16: $logic_2 = ismember(PC(i).index(:,1), ID_{parent}(j))$

17: {Find the list of the family nodes at the previous time step}

18: $nodes_{family} = I_{bond,old}(logic_2, H_{bond}(logic_2,:) == 0)$

19: {Store into displacement matrix the family nodal displacements}

20: $U_s = U^{n-1}(nodes_{family}, (2n-1):(2n))$

21: {Select the child nodes from the list of child nodes which are still active}

22: $logic_2 = (grid_{new}(\ ID_{child}(j,:), 6) > 0)$

23: $ID_{active,child} = (grid_{new}(\ ID_{child}(j,:), 6) > 0)$

24: {As a second step, which is only applied in the case of i =1, identify the child nodes generated by the parent nodes which are not in the previous grid}

25: {Proceed with the second step if no active child nodes have been found}

26: **if** $any(logic_2 == 0)$ & $(i == 1)$

27: $ID_{inactive,child} = \ ID_{child}(j, logic_2 == 1)$

28: **for** $jj = 2:\text{level}_{max}$

29: $logic_3 = ismember(PC(jj).index(:,1), ID_{inactive,child})$

30: $ID_{nodes} = unique(reshape(PC(jj).index(\ logic_3, 2:end),$

$sum(logic_3) \cdot 4,1)$

31: $logic_4 = (grid_{new}(ID_{nodes}, 6) > 0)$

32: $ID_{active,child} = [ID_{active,child}, ID_{nodes}\ (logic_4)]$

33: $ID_{inactive,child} = ID_{nodes}(logic_4 == 0)$

34: **end**

35: **end**

{Built the moment matrix}

$P_Q = [ones(numel(nodes_{family}), 1), grid_{old}(nodes_{family}, 1{:}\,2)]$

{Compute the inverse of the moment matrix}

$P_{Q,inv} = pinv(P_Q)$

{Built the vector of the basis function of monomials}

$p_{child} = [ones(numel(ID_{active,child}), 1), \ldots$

$grid_{old}(ID_{active,child}, 1{:}\,2)]$

{Interpolate the displacement values of the new nodes }

$U^n\left(ID_{active,child}, (2n-1){:}\,(2n)\right) = (p_{child} * P_{Q,inv}) * U_s$

end

end

end