UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

MASTER'S THESIS

Master in material science engineering

COMPUTATIONAL MODELING OF ELASTICITY USING METHODS OF PARTICLES

Autor: Vivier Antoine
Director: Caner, Ferhun Cem

Resumen

El método de los elementos finitos es un método numérico bien establecido que se puede utilizar para predecir el comportamiento mecánico en las ciencias de los materiales. Sin embargo, su dependencia de los elementos implica límites y problemas en ciertas aplicaciones de ingeniería. Es por eso por lo que ha surgido el concepto de eliminar la malla o, al menos, limitar la dependencia en los elementos. Así, se han desarrollado los métodos llamados “meshfree” (los métodos sin malla) que se siguen investigados y mejorados hoy en día.

En este trabajo, se estudian los métodos sin malla para aplicaciones de elastoestática, y en particular el “global Radial Point Interpolation Method (RPIM)” y el “Local Radial Point Interpolation Method (LRPIM)”. Él objetive de este trabajo de fin de máster es comprender los procedimientos de solución de los métodos sin malla y los desafíos que se presentan. Por lo tanto, problemas mecánicos sencillos servirán de soportes y de ejemplos para ilustrar los conceptos subyacentes. Además, gracias a los códigos informáticos proporcionados en (G. R. Liu e Y. T. Gu, 2005), se detallan los problemas de implementación debidos al RPIM global y al LRPIM, y se utilizan y modifican los algoritmos para estudiar aplicaciones triviales. Entonces, estos dos métodos se aplican a un problema de mecánica en 2D: un voladizo sometido a una carga parabólica en su extremo derecho, para la que se conoce la solución analítica. Así, se puede cuantificar y estudiar la precisión de la solución.

Abstract

The Finite Element method is a well-established and common numerical method that can be used predict the mechanical behaviour in material sciences. Nevertheless, its strong reliance with the elements implies limits and issues for certain engineering problems. That is why the concept of eliminating the mesh or at least, limiting its dependence has come. Hence, the so-called meshfree methods have been developed and still investigated and improved nowadays.

In this work, meshfree methods are discussed for elasto-static mechanics applications, and in particular the global Radial Point Interpolation Method (RPIM) and the Local Radial Point Interpolation Method (LRPIM). The aim of this master thesis is to understand and go through the meshfree methods solution procedures and challenges. Hence, simple mechanics problems will serve as supports and examples to illustrate the concepts behind such numerical methods. More, thanks to the computer codes provided in (G. R. Liu and Y. T. Gu, 2005), the implementation issues due to the global RPIM and LRPIM are detailed, and the algorithms are used and modified for studies on simple applications. These two methods are applied to a 2D mechanics problem: a cantilever beam subjected to a parabolic load at its right end, for which the analytical solution is known. Thus, the meshfree method parameters that affect the solution accuracy can be highlighted and discussed.

Acknowledgements

I would like to thank **Prof. Ferhun Cem Caner** who introduced me to the meshfree methods field and allowed me to improve my knowledge and interest into numerical methods and simulation.

I sincerely thank **Pedro Vinicius** for its help and guidance during all my master thesis and especially when I encountered difficulties or lack of motivation.

Nomenclature

Abbreviations

CAE	Computer-Aided Engineering
CPU time	Central Processing Unit
FEM	Finite Element Method
LRPIM	Local Radial Point Interpolation Method
PIM	Point Interpolation Method
RBF	Radial Basis Function
RPIM	Radial Point Interpolation Method

Notations

This report uses the following notations for matrices and vectors:

$$\boldsymbol{M} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

$$\boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

$$\boldsymbol{v^T} = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}$$

$$\boldsymbol{M^T} = \begin{bmatrix} M_{11} & M_{21} \\ M_{12} & M_{22} \end{bmatrix}$$

Index

1. Introduction

1.1 Objectives

The aim of this master thesis is to discover and get used with the meshfree numerical methods. The computer codes provided in (G. R. Liu and Y. T. Gu, 2005) are used and modified to understand and investigate different meshfree methods. Simple mechanics problem in two dimensions are treated in order to focus on the numerical method itself. The challenges are going through the meshfree algorithms coded in FORTRAN 90 in order to extracts the influencing parameters, to understand its limits and to try to transpose the code to be functional for a three dimensions problem.

At the beginning of this master thesis the following objectives has been established:

1) Get used with the FORTRAN 90 language

2) Handle the general procedure of the radial point interpolation meshfree shape function construction (see section 3.2).

3) Get used with meshfree general procedures and in particular the Radial Point Interpolation Method and the Local Radial point Interpolation Method.

4) Studying these last two meshfree methods with a numerical example provided in the book.

5) Transpose an FORTRAN algorithm for a 2D problem to be efficient in 3D (Cantilever beam subjected to a parabolic load at its right end

Nevertheless, after many tries the last objective could not be completed. The code led to wrong results and investigations were unable to manage to find the error(s). After that, it has been decided to abandon the transposition of the 2D code, but instead of that, studying the meshfree method parameters and properties that have influence on the solution accuracy.

1.2 Scope of the work

More generally, this master thesis has the role to make aware of the issues in the field of numerical methods. Going through two meshfree methods allow to understand the process behind the meshfree method development and the limits of the established numerical methods, such as the Finite Element Method. Furthermore, this work allows to get used with and gain experience with the processes of problems modelling, results visualizations, numerical simulation, and lastly the analysis of the results. To lead these studies, few articles, but above all two books are taken as support: (G.R. Liu, 2009) and (G. R. Liu and Y. T. Gu, 2005).

2. State of the art

2.1 Meshfree methods

2.1.1 Numerical simulation

By definition, a numerical simulation is executed by a computer in order to obtain a model that can be used to understand or predict physical problems or phenomena. These last can sometimes be described by algebraic, differential, or integral equations. Moreover, to obtain the exact results of such equations by analytical means is extremely rare because the problems are naturally complex. That is why the concept of numerical simulation has come. To approximate the solutions of such problems. In addition, the development of computers and its performance as increased a lot in the last decades. The processors and more and more efficient and so that numerical simulation using computer technologies. With a mathematical approach, a numerical simulation achieves the transformation of a complex problem into a discretized mathematical form. [1] In that sense, after solving the problem, the physical problems can be virtually reproduced. Physical problems in engineering are many and it would be impossible to create simulations for all of them. Historically, the finite element method was first developed for stress analysis but until nowadays, many other physicals problems have been solved with FEM. As examples, common physical problems that can be solved using FEM are mechanics for solids and structures, heat transfer, acoustics, or fluid mechanics. [2] Numerical simulations procedure is summarized in Figure 2.1:

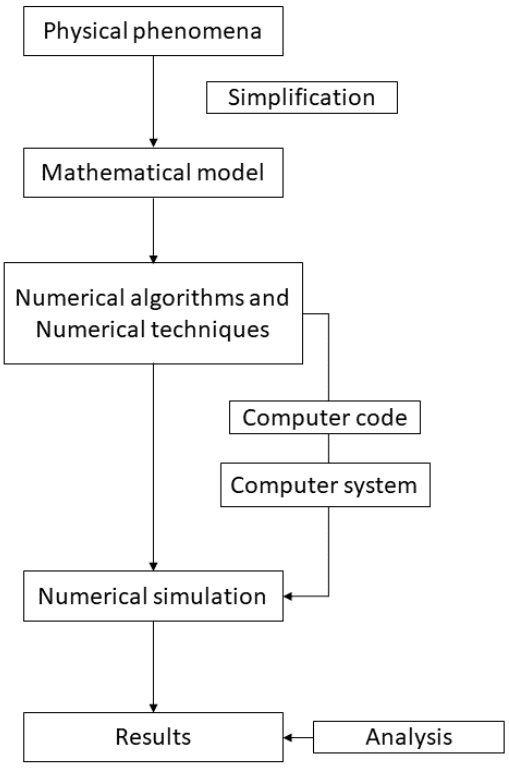

Figure 2.1 Numerical simulation general procedure.

Thus, numerical simulation and especially numerical techniques can represent true physics problems. This is a true engineering challenge to develop numerical methods in order to predict and/or build complicated systems. Until now, many numerical techniques or approximation methods have been established such as the Finite Element Method (FEM), the Finite Difference Method (FDM) or the Finite Volume Method (FVM).

2.1.2 Challenges

The Finite Element Method is nowadays well established and widely used in the numerical methods field. This method first came in the 1950s to solve complex elastic and structural problems in civil and aeronautical engineering. [2] Since its beginning, this method has been largely developed and still a powerful tool for its versatility for complex geometry problems. The main idea of FEM is that it is a numerical method, which is based on elements. Indeed, the geometry of the problem is discretized into elements that are linked together by the nodes in a predefined manner, this procedure is called meshing. And a mesh is defined as any of the continuum open spaces or interstices that is formed by connecting some nodes in a properly manner. Such a procedure requires the use of pre-processors and especially for complex geometry problems.

Nevertheless, the main drawback of such a method is that it is dependent on meshes and elements. Thus, this mesh dependence led to the following limitations:

- **Meshing procedure cost.**

  The meshing procedure is of great importance and meshes must be predefined. A such procedure is accomplished by the help of computers and its resources are termed as CPU time (central processing unit). For problems of complex geometry, this step can require many resources and be time-consuming.

- **Stress accuracy.**

  In FEM, the stresses are often inaccurately predicted. Indeed, they are discontinuous, especially at the interfaces of the elements because FEM formulation relies on a piecewise continuous nature of the displacement field. Thus, special additional requirements can be needed in pre-processing to obtain accurate stress in certain cases.

- **Adaptive analysis.**

  Nowadays, a requirement for numerical methods such as FEM is to obtain a solution with a desired accuracy. In order to fulfil this requirement, adaptive analysis must be performed. In FEM, adaptive analysis involves re-meshing to ensure a proper connectivity. However, this re-meshing step implies the use of complex and computationally expensive pre-processors.

- **Limitations for some analysis.**

  The FEM encounter some limitations for certain type of applications due to the use of mesh. For example, due to the element distortions under large deformations, the accuracy is lost. More, for

problems with cracks growth, phase transformation or breakage the use of FEM in limited. In fact, more generally, non-linear problems lead to errors or misrepresentations because FEM depends on elements that cannot be broken.

All these FEM limitations have the same common point: the mesh dependence. That is why the idea of avoiding the meshing procedure and the element requirement has come in numerical methods. [3]

2.1.3 General solution procedure

2.1.3.1 Definition

The so called meshfree methods are defined as *"a method used to establish system algebraic equations for the whole problem domain without the use of a predefined mesh for the domain discretization".* [1] The geometry of the problem in meshfree methods is represented by a set of nodes called field nodes distributed within the problem domain and on its boundaries. It should be noticed that at this stage the problem domain and its boundaries are represented by the field nodes and not discretized. The term field nodes translate the fat that these nodes will carry the value of the field variables. More, these field nodes do not form a mesh because information on the relationship between each other is not required for the interpolation of the unknown function. This last will be then approximated locally using those sets of field nodes.

The ideal requirement to term a method as meshfree is if no mesh is necessary for all the solving procedure and for a given geometry that is governed by PDEs and with any kinds of boundary conditions. Nevertheless, in practice, the meshfree methods are not always completely meshless. Indeed, some meshfree methods can require a background mesh (composed of background cells) to perform the integration of system matrices over the problem domain. Another example would be methods that required background cells only locally to perform integrations over the problem domain.

Thus, the numerical methods are termed meshfree if the solution do not depend on the quality of a mesh, but one could be used if it is automatically generated and used for some specifics integrations such as integrations. [3]It can be paradoxical for the reader to the fact that many meshfree methods still use a mesh, but it should be mentioned that in these cases the meshes are used in more flexible and free ways. The most important challenges in meshfree methods are to reduce and try to avoid the reliance on the use of meshes that are difficult or expensive for problems with complex geometries. However, automatically created meshes cans still be used if it helps in getting better solutions in more effective ways and if the solution quality does not rely completely on it. [1]

Finally, meshfree methods have been developed since the last decades and have shown good potential to become powerful and widely use numerical tools but are still under development stage. Indeed, in meshfree methods, adaptative analysis can be more easily used thanks to the use of automatically meshing for 2D and 3D complex geometry or non-linear problems that are sufficient to

obtain good solutions. More, these methods provide flexibility to add or delete nodes when and where it is needed, such as stress analysis problem with singularity or stress concentration area. For the study of crack growth problems, nodes can be added wisely around the crack to lead to more accurate solutions.

2.1.3.2 Solution procedure

In this section, the general procedure, and the basics steps in meshfree methods will be explained using solid mechanics problems as example. Also note that in this work only solid mechanics problems will be studied to understand the main idea, operation, and challenges behind meshfree methods. The meshfree methods can also have different procedures and steps in its formulations, but the two methods that will be presented in sections 3.3 and 3.4 are quite similar.

The method like in FEM starts at the stage of geometry creation. In meshfree methods, we speak then about node generation (and possibly background element generation) in contrast with element generation in FEM. Then, shape functions are computed based on selected local nodes. Once again, the difference between the finite element method and the meshfree method in shape function construction is that in FEM the shape functions are computed using elements and are the same for the entire element. However, the meshfree shape functions are generally constructed only for a point of interest and based its close field nodes. These shape functions can be different for different point of interest. Finally, after establishing the global discretized system of equations, the FEM and meshfree procedures stay quite similar and many FEM techniques can be used in the meshfree methods. [3]

The main differences between these two numerical methods can be resumed principally as the meshing, the shape function construction, the global matrices, the enforcements of the EBCs, the computation speed, the solution accuracy, and the state of development. To go further, in FEM the discretized stiffness matrix is banded (sparse matrix where the non-zero values are banded in the diagonal) and symmetric whereas in meshfree methods, the matrix is generally banded by may not be symmetric. More, special treatments are required to enforce the EBCs depending on the meshfree method used. Finally, meshfree methods lead to more accurate results compare to FEM, and it should be mentioned that there are few commercial software packages available.

Step 1: Domain representation/discretization

The first step in the meshfree method procedure is representing the problem domain and its boundary. To perform this step, the geometry is modelled using sets of field nodes distributed. This can be accomplished thanks to CAE (Computer-Aided Engineering) code or pre-processor. The field node density can be changed in order to play on the accuracy, and the distribution may be usually not uniform. For example, a denser field node distribution can be used where the displacement gradient will be larger. Note that the node density should also be chosen depending on the limits of the computer resources available. Then, the BCs and loads are specified for the modelled geometry.

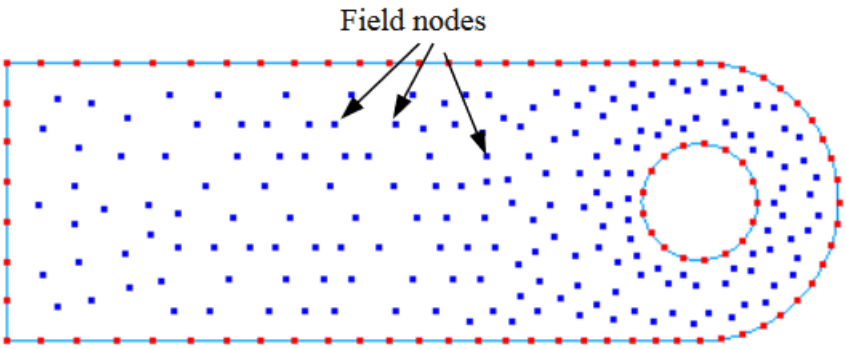

Figure 2.2 Meshfree domain representation. *[3]*

Step 2: Displacement interpolation/approximation

During this second step, the field variable (displacement) at any point at $\boldsymbol{x} = (x, y)$ within the modelled problem domain is approximated/interpolated using the function values (displacements) at the nodes within a small domain called support domain of this point at $\boldsymbol{x}$ (usually a quadrature point). The concept of support domains will be explained later in the section 3.2.1.1. Thus,

2.1

$$u(\boldsymbol{x}) \sum_{i=1}^{n} \phi_i(\boldsymbol{x}) u_i = \boldsymbol{\Phi}^{\boldsymbol{T}} \boldsymbol{U_s}$$

where

n, is the number of nodes included in the local support domain of the point at $\boldsymbol{x}$;

u_i, is the nodal displacement at the i-th node;

$\phi_i(\boldsymbol{x})$, is the shape function for the i-th and it is determined using the n nodes located in the support domain of $\boldsymbol{x}$;

$\boldsymbol{\Phi}$, is the matrix that collects all the shape functions for this n nodes;

$\boldsymbol{U_s}$, is the vector that collects all the displacements at this n nodes;

The support domain of a point at $\boldsymbol{x}$ is used to determine the number of field nodes that should be used for the interpolation of the unknown function at $\boldsymbol{x}$. Such domains can have many various shapes (generally rectangular or circular) and dimensions and these last may be different for different point of interest $\boldsymbol{x}$**.** Moreover, these domains can be weighted using weight function purposely so that vanish on its boundary. However, its use presents some limitations especially for problems where the node density varies too much. In this case, the risk is if a large number of nodes are located in a side and so the shape function constructed will carry errors. In order to overtake this limitation, the influence domains are used. The concepts of influence and support domain will be further detailed in sub-section 3.2.1.1.

Step 3: Formation of system equations

The step 3 consists of formulate the discrete equations for the meshfree method selected using the shape functions and (generally) weak formulations. The obtained equations can be written either in a nodal matrix form and are then assembled into the global system equations in a matrix form. Moreover, these matrices are banded and sparse, and depending on the method selected, may be asymmetric. Note that in the case of a static analysis, the global system equations are a set of algebraic equations.

Step 4: Solving the global meshfree equations

The last step of the procedure is to solve the set of global meshfree equations. For static problems (that will be studied in this work) the displacement at all nodes of the entire problem domain is obtained and then, stress and stain can be retrieved using the relationships that will be stated in the next section. The solving procedure is quite similar to that for FEM, but the used solver may be different depending on the nature of the matrix system. Here some classic solving techniques example: Gauss elimination, iterative methods, or LU decomposition.

2.1.4 Fundamentals for meshfree methods

In the next section, partial differential equations for solid mechanics will be established. Such equations are called strong forms of system equation. In practice, it is very difficult to obtain the exact solution of a strong form of system equations because engineering problems are generally complex. Indeed, the strong form requires a strong continuity on the field variable. Moreover, the approximate unknown function (the displacement u and v in our case) in a strong form formulation should have a sufficient degree of constituency so that it can be differentiated up to the order of the partial differential equation. One example of a method using strong formulation is the finite element method. Nevertheless, due to the strong continuity requirement, this method is used mainly only for simple and regular geometry/BCs problems.

Weak formulations are fundamental for FEM and for meshfree methods. In the FEM, all numerical operations and especially function approximation and integrations of the weak forms of system equations are based on the elements. The difference and advantage of the meshfree methods compare to FEM is that the function approximation and integrations are independent. In other words, meshfree methods allows wider ways to use weak formulations and so can lead to superior methods than FEM. In this report, only weak formulation will be used in order to construct the discretized system equation of two different meshfree weak form methods that will be detailed in another section.

To construct weak forms, various principles can be used. In this report, only two meshfree methods will be studied, the radial point interpolation method and the local radial point interpolation method. These methods are based respectively on the Galerkin weak form and the weighted residual method which are widely used principles to develop meshfree method. The weak forms are often on

integral form and in contrast to strong forms, it requires weaker continuity for the field variables and on the integral operation. This weaker requirement allows forming discretized system equations that lead to good and accurate results and for problems with complex geometry. [1] To sum up, this study will focus on weak formulation and in particular the Galerkin weak form and the weighted residual method in order to create discretized system equations for 2D solid elastostatics problems.

2.2 Mechanics for 2D solids

2.2.1 Introduction

The aim of this second section of the state of the art is to introduce the basics of mechanics applied to two dimensions solids in order to study simple cases with some meshfree methods. This section will go through the classic and established theory of solids subjected to forces and loads. Indeed, the mechanic of solids and structures study the strains and displacements that are generated from stresses and for some given boundary conditions. The challenges in this field are studying these relationships between stresses, strains, displacements, and boundary conditions and in the case of this word studying these relationships through simulations.

The primary issue in this report is to study the meshfree methods from the procedure to the visualization and analysis of the results. That is why the solid mechanics used in the applications will be very trivial and will simply cover statics forces applied on isotropic elastic materials and without increments.

2.2.2 Stress-strain relationship

In solid mechanics, and elastic comportment describe a solid for which the deformations are completely reversible if its unloaded. A material is isotropic if its material properties do not depend on the direction. The two isotropic material constants are the Young's modulus noted E and υ the Poisson's ratio. These two constants describe, respectively for an isotropic elastic material, the relationship between stress and strain and the contraction (or expansion) of the material perpendicularly to the load's direction. [2] [3]

The two types of 2D solid mechanics problems are plane stress solid problems and plane strain solid problems. The first concerned solids for which the thickness in a direction is negligible when compared to the two other directions. If the solid thickness in the z direction is negligible compared with the solid dimensions in x and y directions, the external forces are so only applied on the x-y plane and the stresses in z direction are all zero ($\sigma_{zz} = \sigma_{xz} = \sigma_{yz} = 0$).

The plain strain problems deal with solids whose thickness in a direction is very large compared with the solid dimensions in the two other directions. If the thickness of a solid in z direction is very large

compared with the dimensions of this solid in the x and y directions, the external forces are so only applied uniformly along the z axis and the strain components in z direction are all zero ($\varepsilon_{zz} = \varepsilon_{xz} = \varepsilon_{yz} = 0$). It should be mentioned that for a plane stress problem the strains ε_{xz} and ε_{yz} are zero but ε_{zz} is not. Similarly, for the plane strain problem the stresses σ_{xz} and σ_{yz} are zero but not σ_{zz}.

In two dimensions the stress and strain components are respectively

2.2

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} \qquad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix}$$

(The shear stress component σ_{xy} is also denoted τ_{xy}).

The displacement components vector is composed of u, the displacement in the x direction and v the displacement in the y direction:

2.3

$$\boldsymbol{u} = \begin{bmatrix} u \\ v \end{bmatrix}$$

The strain-displacement relationship in the matrix form is given by

2.4

$$\boldsymbol{\varepsilon} = \boldsymbol{L}\boldsymbol{u}$$

where $\boldsymbol{L}$ is the differential operator matrix obtained from the strain-displacements relationships:

2.5

$$\varepsilon_{xx} = \frac{\partial u}{\partial x}, \varepsilon_{yy} = \frac{\partial v}{\partial y} \; and \; \varepsilon_{xy} = \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}$$

2.6

$$\boldsymbol{L} = \begin{bmatrix} \dfrac{\partial}{\partial_x} & 0 \\ 0 & \dfrac{\partial}{\partial_y} \\ \dfrac{\partial}{\partial_y} & \dfrac{\partial}{\partial_x} \end{bmatrix}$$

The Hook's law for two dimensions elastic solids is written in the matrix form as

2.7

$$\boldsymbol{\sigma} = \boldsymbol{D}\boldsymbol{\varepsilon}$$

where $\boldsymbol{D}$ is the matrix of material constants, which needs to be obtained empirically. The expression of $\boldsymbol{D}$ for isotropic materials in a plane stress problem is

2.8

$$\boldsymbol{D} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 0 & 0 \\ 0 & 0 & \frac{(1-\nu)}{2} \end{bmatrix} \qquad \text{(Plane stress)}$$

For isotropic materials in a plane strain problem, the matrix $\boldsymbol{D}$ can be obtained by replacing E and v respectively with $\frac{E}{1-\nu^2}$ and $\frac{\nu}{1-\nu}$ as

2.9

$$\boldsymbol{D} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)}\begin{bmatrix} 1 & \frac{\nu}{1-\nu} & 0 \\ \frac{\nu}{1-\nu} & 0 & 0 \\ 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix} \qquad \text{(Plane strain)}$$

2.2.3 Equilibrium equations

The equilibrium equation translates the relationship between stress and the external forces. Considering the state of stresses of an infinitely small block of an elastic isotropic solid:

Since in this report dynamics problems will not be treated, the inertial forces of this block will not be considered. The equilibrium equation of the forces in the x direction can be written as

2.10

$$(\sigma_{xx} + d\sigma_{xx})dydz - \sigma_{xx}dydz + \left(\sigma_{yx} + d\sigma_{yx}\right) - \sigma_{yx}dxdz + \left(\sigma_{zx} + d\sigma_{yx}\right) - \sigma_{yx}dxdy + b_x = 0$$

where b_x is the external body force applied at the centre of the block and the vector of body forces is written as

2.11

$$\boldsymbol{b} = \begin{bmatrix} b_x \\ b_z \\ b_z \end{bmatrix}$$

$d\sigma_{xx}$, $d\sigma_{yx}$ and $d\sigma_{zx}$ can be expressed as

2.12

$$d\sigma_{xx} = \frac{\partial \sigma_{xx}}{\partial \mathrm{x}}dx; d\sigma_{xx} = \frac{\partial \sigma_{yx}}{\partial \mathrm{y}}dy \; and \; d\sigma_{xx} = \frac{\partial \sigma_{zx}}{\partial \mathrm{z}}dz$$

Now equation 2.10 can be simplified:

2.13

$$\frac{\partial \sigma_{xx}}{\partial \mathrm{x}} + \frac{\partial \sigma_{yx}}{\partial \mathrm{y}} + \frac{\partial \sigma_{zx}}{\partial \mathrm{z}} + b_x = 0$$

This is the equilibrium equation for x direction. Equations for y and z directions are obtained similarly.

2.14

$$\frac{\partial \sigma_{xy}}{\partial \mathrm{x}} + \frac{\partial \sigma_{yy}}{\partial \mathrm{y}} + \frac{\partial \sigma_{zy}}{\partial \mathrm{z}} + b_y = 0$$

2.15

$$\frac{\partial \sigma_{xz}}{\partial \mathrm{x}} + \frac{\partial \sigma_{yz}}{\partial \mathrm{y}} + \frac{\partial \sigma_{zz}}{\partial \mathrm{z}} + b_z = 0$$

Finally, the static equilibrium equation can be written in the matrix form as

2.16

$$\boldsymbol{L^T \sigma} + \boldsymbol{b} = 0$$

And using equations 2.4 and 2.7

2.17

$$\boldsymbol{L^T D L u} + \boldsymbol{b} = 0$$

This equation is a partial differential equation for three-dimensions elastostatics where $\boldsymbol{u}$ is the vector of the unknown function of field variable. And where $\boldsymbol{L}$ is the differential operator matrix in 3D.

The equation for 2D solids is simply obtained by removing the z coordinate terms and omitting the differential operations related to z. For 2D solids:

2.18

$$\boldsymbol{L^T \sigma} + \boldsymbol{b} = 0$$

or

2.19

$$\boldsymbol{L^T D L u} + \boldsymbol{b} = 0$$

where $\boldsymbol{\sigma}$, $\boldsymbol{b}$ and $\boldsymbol{L}$ are expressed respectively as

2.20

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} \quad \boldsymbol{b} = \begin{bmatrix} b_x \\ b_y \end{bmatrix} \quad \boldsymbol{L} = \begin{bmatrix} \dfrac{\partial}{\partial_x} & 0 \\ 0 & \dfrac{\partial}{\partial_y} \\ \dfrac{\partial}{\partial_y} & \dfrac{\partial}{\partial_x} \end{bmatrix}$$

2.2.4 Boundary conditions

The boundary conditions can be either essential or natural boundary conditions. The notations will be respectively EBCs and NBCs.

The EBC is the displacement BC written as

2.21

$$u_i = \bar{u}_i \ (on \ \Gamma_u)$$

with $\bar{u}_i$, the prescribed displacements on the boundary Γ_u (i and j denote either x or y). The displacement condition is called essential because this condition needs to be enforced and satisfied first before the derivation starts. Otherwise, the resolution will fail, that is why a such condition is "essential". In the case of a meshfree method formulation, this condition is enforced separately, either after or before establishing the final discretized system of equation of the meshfree method used. [2] [3]

The NBCs is the force BC written as

2.22

$$\sigma_{ij} n_j = \bar{t}_i \ (on \ \Gamma_t)$$

with $\bar{t}_i$ the prescribed tractions and n_j a component of the unit vector outward normal on the boundary. In 2D, this vector is noted as

2.23

$$\boldsymbol{n} = \begin{bmatrix} n_x & 0 & n_y \\ 0 & n_y & n_x \end{bmatrix}$$

The force boundary condition is termed as "natural" because it is naturally formulated into the system equation using the weak formulation that will be explained in a different section.

In addition to the boundary conditions, there are the initial conditions that can be displacement or velocity. Since the applications in this report will not depend on time, these conditions will not be explained in more details.

Finally, a solid problem, with the equilibrium (or governing) equation (equation 2.18), the constitutive equation (equation 2.7), the strain-displacement relationship (equation 2.4) and the BCs is called a boundary value problem.

3. Radial Point Interpolation Method

3.1 Introduction

The RIPM is a Meshfree methods based on the point interpolation method (PIM). It is an interpolation procedure developed by GR Liu and his research team. Two specific formulations have been proposed: PIM using radial basis functions and PIM using polynomial basis. The PIM has been developed in order to replace the moving least square approximation to create shape functions. The main advantages of this method are the excellent function fitting accuracy and the Kronecker delta function property which permits simple enforcement of EBCs such as in the FEM. Historically, the radial point interpolation method that is based on PIM has been established from the Element Free Galerkin method by replacing the moving least square (MLS) shape functions with the RPIM shape functions. [1] [3]

3.2 Shape functions construction

There are many construction techniques to establish meshfree shape functions, and this part will focus on the conventional radial point interpolation shape functions. This last are locally supported since a set of field node is used for the construction in a small local domain called support of influence domain. In FEM, shape functions are constructed using interpolation techniques that rely on elements formed by a set of fixed nodes depending on the element topology and this interpolation technique is oppositely termed stationary element-based interpolation. In developing meshfree methods, the construction of efficient shape functions is one of the biggest challenges. Indeed, the difficulty is constructing efficient meshfree shape functions without any topology or predefined relationships between nodes.

3.2.1 Methods and requirements

3.2.1.1 Support domain and influence domain

It is necessary to discuss the concept of support domain and state the difference between influence and support domain. The influence domain of a field node is the domain over which this node has influence. It is used to select nodes for the interpolation, and its centre is the field node. The support domain is an area chosen for the meshfree interpolation for a point of interest (usually a quadrature point) which is also the centre. Generally, to construct the shape functions at the point of interest $\boldsymbol{x_Q}$, a field node i will be involved in the construction if $\boldsymbol{x_Q}$ is located in the influence domain of the node i. This means that if the influence domain covers the point of interest, the node i will be registered and used to take part in the shape function construction. The concepts of support domain and influence domain are illustrated below in Figures 3.1 and 3.2.

The accuracy of the interpolation for a point of interest $\boldsymbol{x_Q}$ depends on the nodes in the support domain. Thus, this last must be specifically chosen to ensure a suitable and efficient approximation. The dimension d_s of such a domain is determined by

3.1

$$d_s = \alpha_s d_c$$

where

α_s is the dimensionless size of the support domain

d_c is the nodal spacing or average nodal spacing near the point of interest. If a set of uniformly distributed field nodes is used to describe the problem, d_c will be the distance between two neighbouring field nodes. If a set of arbitrary distributed field nodes is used the nodal spacing will be an average nodal spacing determined by

3.2

$$d_C = \frac{D_s}{n_{Ds} - 1}$$

with

D_s, an estimated support domain dimension. It must not be accurate but should be estimated following this procedure:

n_{Ds}, the number of nodes within the domain of dimension D_s.

The influence domain of a field node is defined as the domain for which this node has an influence. This domain is centred on the field node, and it is used in the meshfree shape function construction. If the influence domain of a field node includes the sampling point (quadrature point), this field node will be recorded and used in the shape function construction at this sampling point. The use of the concept of influence instead of support domain presents some advantages. [1] [3]

1) Influence domains work better for non-uniform set of nodes.
2) The influence domain is defined for each node of the problem and so can be of different sizes depending on the "importance" of the node. For example, a node in an area with a small density may have a greater influence than a node in a high-density area.
3) Generally, the number of field nodes is much less than the number of points used for the quadrature, such as Gauss points. Thus, if influence domains are used instead of support domains, there will be fewer domains to compute and so the procedure will be more effective.

The shape of a such domain is arbitrary and its dimensions for a 2D problem are expressed as

3.3

$$\begin{cases} d_{ix} = \alpha_{ix} d_{cx} \\ d_{iy} = \alpha_{iy} d_{cy} \end{cases}$$

where d_{ix} and d_{iy} are respectively the size of the influence domain in the x and y directions, α_{ix} and α_{iy} are the dimensionless domain sizes and d_{cx} and d_{cy} are the nodal spacing. The concept of influence domain instead of support domain will be used for the latter numerical applications in this report.

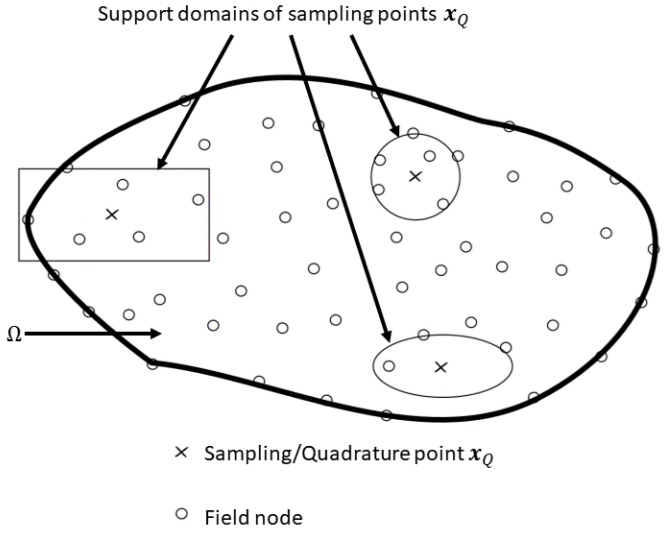

Figure 3.1 Support domains representation.*[3]*

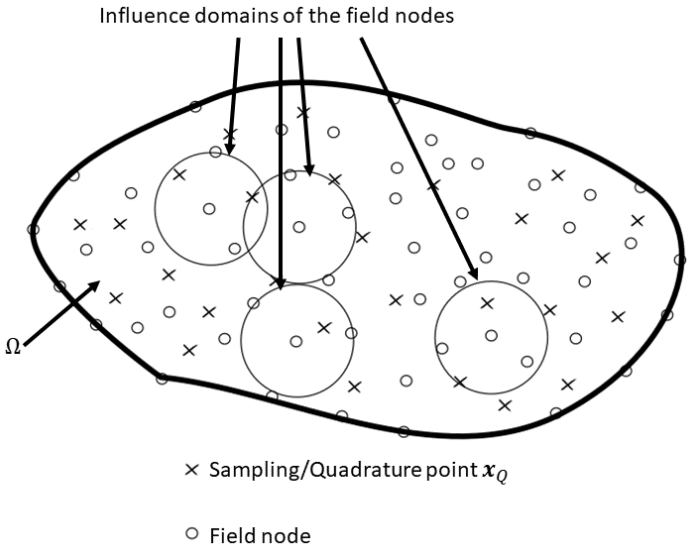

Figure 3.2 Influence domains of field nodes representation. *[3]*

3.2.1.2 Meshfree shape function construction requirements

In the development of a meshfree method, one of the primary steps is meshfree shape function construction. As it has been said previously, the creation of an efficient meshfree shape function is one of the most important challenges in meshfree methods. Indeed, it is necessary to create shape functions that present specific properties using the field nodes scattered in the entire problem domain. Thus, here a non-exhaustive list of the most important requirements for a good meshfree shape function construction:

- The shape function should be sufficiently robust even for arbitrarily and flexible nodal distribution of field nodes. [3]
- It must be numerically stable; in other words, the shape function algorithm must be stable enough for irregularity (within reason) in the nodal distribution. [3]
- It should possess a certain order of consistency, that is to say that the shape function constructed should present the ability to reproduce exactly the polynomials of that order. [3]
- The support (or influence) domain used for the unknown function interpolation should be small enough to include a small number of nodes. More, shape functions should be considered as zero outside the support domain. [3]
- The algorithm constructing shape functions should be computationally efficient (it should be the same order of complexity as the shape function algorithm for FEM). [3]
- Ideally but not necessary, the shape functions would process the Kronecker delta function property. That is to say that the meshfree shape function is unit at the field node and zero at the other nodes of the support domain. The Kronecker delta condition is expressed as

3.4

$$\phi_i(\boldsymbol{x_j}) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

where

$\boldsymbol{x_j}$, is a sampling point;

$i = 1,2,3, \dots, n$;

$j = 1,2,3, \dots, n$;

- The approximated field variable using the meshfree shape function should be compatible throughout either the entire problem domain if a global weak formulation is used and the local quadrature domain if a local weak formulation is employed. [3]

The fulfilment of the requirements above allows to compute robust shape functions that will lead to more accurate numerical solutions.

3.2.2 RPIM Shape functions construction

The conventional RPIM come from the polynomial PIM which uses polynomials as basis function in the interpolation. This is one of the first interpolation schemes and it has been widely used for the numerical methods such as the FEM. However, the polynomial PIM has one main drawback. Indeed, in the polynomial PIM, the moment matrix may be singular. This drawback requires additional stages for the matrix triangulation in the meshfree method algorithm. More, PIM is not robust for non-uniform nodes distribution. That is why the RPIM has been developed, the use of radial basis functions can overcome the

matrix singularity issue. It is also robust and stable for arbitrary distributed set of nodes and since the only variable in the RBF is the distance between the point of interest and the node, RPIM shape function can be easily constructed for 3D numerical applications. [1] [3]

The RPIM shape functions will be used in this thesis for the numerical applications and its procedure comes from (G.R. Liu, 2005). The RPIM interpolation can be written as

3.5

$$\mathrm{u}(\mathbf{x}) = \sum_{i=1}^{n} R_i(\boldsymbol{x}) a_i + \sum_{j=1}^{m} p_j(\boldsymbol{x}) b_j = \boldsymbol{R}^T(\boldsymbol{x})\boldsymbol{a} + \boldsymbol{p}^T(\boldsymbol{x})\boldsymbol{b}$$

where

$R_i(\mathbf{x})$ is an RBF.

n, is the number of RBFS.

$p_j(\boldsymbol{x})$ is monomial in the space coordinates of $\boldsymbol{x}$

m is the number of polynomial basis functions. If $m = 0$, pure RBFs are used and if m is greater than zero, the RBF is augmented with m polynomials basis functions.

a_i and b_j are coefficient to be determined.

As mentioned above, the distance between the point of interest at **x** and the node at $\boldsymbol{x}_j$ is the only variable in the RBF.

For 2D problems this distance is

3.6

$$r = \sqrt{(x - x_i)^2 + (y - y_i)^2}$$

Thera are many RBF and the most often used are

- MQ: Multi-quadratics function.

3.7

$$R_i(x, y) = \left(r_i^2 + (\alpha_c d_c)^2\right)^q$$

Where α_c and q are the shape parameter such as $\alpha_c \geq 0$.

- EXP: the Gaussian exponential function.

3.8

$$R_i(x, y) = e^{-\alpha_c \left(\frac{r_i}{d_c}\right)^2}$$

where α_c is the shape parameter.

- TPS: The thin plate spline function.

3.9

$$R_i(x, y) = r_i^{\eta}$$

where η is the shape parameter.

- The Logarithmic function.

3.10

$$R_i(x, y) = r_i^{\eta} \log r_i$$

where η is the shape parameter.

The most common used are the MQ, EXP, TPS and logarithmic RBFs. In order to use the RBFs, several shape parameters need to be fixed to obtain accurate results. These shape parameters are determined by numerical examination. It has been found for example, that in the MQ RBFs, when the shape parameter $q = 0.98$ or $q = 1.03$ the meshfree method using RPIM shape function leads to good results in the analysis of 2D solid problems.

The polynomial term in equation 3.5 has for role to improve the result's accuracy, reduce the sensitivity of the RBF shape parameters and more, provide a wider choice to determine it. Augmenting RBF with a low degree polynomial can also guarantee the non-singularity of the matrix.

A support domain needs to be formed for the point of interest at $\boldsymbol{x}$, to determine $\boldsymbol{a}$ and $\boldsymbol{b}$ in equation 3.5 with *n* field nodes inside it. Thus, the coefficients a_i and b_i in equation 3.5 will be determined by enforcing this last equation to be satisfied at these *n* nodes within the support domain of point of interest $\boldsymbol{x}$. And so, *n* linear equations will be obtained in the following matrix form:

3.11

$$\boldsymbol{U_s} = \boldsymbol{R}_0\boldsymbol{a} + \boldsymbol{P_m}\boldsymbol{b}$$

With $\boldsymbol{U_s}$ the vector of function values, $\boldsymbol{R}_0$ the moment matrix and $\boldsymbol{P_m}$ the polynomial moment matrix expressed as

3.12

$$\boldsymbol{R}_0 = \begin{bmatrix} R_1(r_1) & R_2(r_1) & \dots & R_n(r_1) \\ R_1(r_2) & R_2(r_2) & \dots & R_n(r_2) \\ \dots & \dots & \dots & \dots \\ R_1(r_n) & R_2(r_n) & \dots & R_n(r_n) \end{bmatrix}_{[n\times m]}$$

3.13

$$\boldsymbol{P_m^T} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & y_n \\ \vdots & \vdots & \ddots & \vdots \\ p_m(\boldsymbol{x_1}) & p_m(\boldsymbol{x_2}) & \dots & p_m(\boldsymbol{x_n}) \end{bmatrix}_{[m\times n]}$$

And with the matrix $\boldsymbol{a}$ and $\boldsymbol{b}$ respectively for the coefficients for the RBFs and the coefficient for polynomial expressed as

3.14

$$\boldsymbol{a}^T = [a_1 \quad a_2 \quad \dots \quad a_n]$$

3.14

$$\boldsymbol{b}^T = [b_1 \quad b_2 \quad \dots \quad b_m]$$

In the moment matrix, the r_k (k ranging from 1 to n) in equation 3.12 are determined using equation 3.6 as follow

3.15

$$r_k = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}$$

Thus, there will be *n + m* variables in equation 3.11. The additional *m* equations that come from the augmenting with polynomials are added following the *m* constraint following conditions.

3.16

$$\sum_{i=1}^{n} p_j(\boldsymbol{x}_i) a_i = \boldsymbol{P}_m^T \boldsymbol{a} = 0, \quad j = 1, 2, 3, \dots, m$$

By combining this last condition with equation 3.11 in the matrix form, a set of equations is obtained

3.17

$$\widetilde{\boldsymbol{U}}_{\boldsymbol{s}} = \begin{bmatrix} \boldsymbol{U_S} \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_0 & \boldsymbol{P_m} \\ \boldsymbol{P_m^T} & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \boldsymbol{G}\boldsymbol{a}_0$$

where

3.18

$$\boldsymbol{a}_0^{\boldsymbol{T}} = [a_1 \quad a_2 \quad \dots \quad a_n \quad b_1 \quad b_2 \quad \dots \quad b_m]_{[n \times m]}$$

3.19

$$\widetilde{\boldsymbol{U}}_{\boldsymbol{s}} = [u_1 \quad u_2 \quad \dots \quad u_n \quad 0 \quad 0 \quad \dots \quad 0]_{[n \times m]}$$

In equation 3.12 the matrix $\boldsymbol{R}_0$ symmetric therefore matrix $\boldsymbol{G}$ is also symmetric, and by solving equation 3.17, the expression of $\boldsymbol{a}_0$ is obtained

3.20

$$\boldsymbol{a}_0 = \begin{bmatrix} a \\ b \end{bmatrix} = \boldsymbol{G}^{-1} \widetilde{\boldsymbol{U}}_{\boldsymbol{s}}$$

The RPIM interpolation in equation 3.5 can then be re-written as follow

3.21

$$\mathrm{u}(\mathbf{x}) = \boldsymbol{R}^T(\boldsymbol{x})\boldsymbol{a} + \boldsymbol{p}^T(\boldsymbol{x})\boldsymbol{b} = [\boldsymbol{R}^{\boldsymbol{T}}(\boldsymbol{x}) \quad \boldsymbol{p}^{\boldsymbol{T}}(\boldsymbol{x})] \begin{bmatrix} a \\ b \end{bmatrix}$$

and using equation 3.21 :

3.22

$$\mathrm{u}(\mathbf{x}) = [\boldsymbol{R}^{\boldsymbol{T}}(\boldsymbol{x}) \quad \boldsymbol{p}^{\boldsymbol{T}}(\boldsymbol{x})] \begin{bmatrix} a \\ b \end{bmatrix} = \widetilde{\boldsymbol{\Phi}}^{\boldsymbol{T}}(\boldsymbol{x})\widetilde{\boldsymbol{U}}_{\boldsymbol{s}}$$

Finally, the RPIM shape functions are expressed as

3.23

$$\widetilde{\boldsymbol{\Phi}}^{\boldsymbol{T}} = [\boldsymbol{R}^{\boldsymbol{T}}(\boldsymbol{x}) \quad \boldsymbol{p}^{\boldsymbol{T}}(\boldsymbol{x})]\boldsymbol{G}^{-1} = [\phi_1(\boldsymbol{x}) \quad \dots \quad \phi_n(\boldsymbol{x}) \quad \phi_{n+1}(\boldsymbol{x}) \quad \dots \quad \phi_{n+m}(\boldsymbol{x})]_{[n+m]}$$

3.24

$$\boldsymbol{\Phi}(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}) \quad \dots \quad \phi_n(\boldsymbol{x})]_{[n+m]}$$

where $\boldsymbol{\Phi}(\boldsymbol{x})$ is the vector of the RPIM shape functions that correspond to the nodal displacement vector. That is the RPIM shape function vector that will be used later in the different Radial Point interpolation meshfree methods.

The RPIM interpolation can be re-written as

3.25

$$\mathrm{u}(\mathbf{x}) = \boldsymbol{\Phi}^{\boldsymbol{T}}(\boldsymbol{x})\boldsymbol{U}_{\boldsymbol{S}} = \sum_{i=1}^{n} \phi_i u_i$$

and thus, the derivatives of $\mathrm{u}(\mathbf{x})$ are

3.26

$$u_{,l}(\mathbf{x}) = \boldsymbol{\Phi}_{,\boldsymbol{l}}^{\boldsymbol{T}}(\boldsymbol{x})\boldsymbol{U}_{\boldsymbol{S}}$$

where l can be either coordinates x or y and the comma indicates a partial differentiation with respect to l.

In general, the order of polynomial used in equation 3.5 is low so there is no singularity problem in the RPIM since a small number of nodes are used in the local support domain of the point of interest at $\boldsymbol{x}$.

The advantages of using RBFs for the RPIM shape functions construction are listed in the table below:

Table 3.1 Radial basis function properties

Advantages	Shortcomings
Avoid the main drawback of the polynomial PIM (singularity problem).	The use of RPIM shape functions is less accurate for solid problems.
Allows stable RPIM shape function for arbitrary distributed set of nodes.	It is necessary to carefully choose the shape parameters in order to not reduce the solution accuracy.
The RPIM shape function construction can be easily transposed for 3D problems by simply changing the distance expression in equation 3.6. Indeed, the only variable in an RBF is the distance between the point of interest and the field node considered, so for 3D problem, the distance along the z-axis is added in the expression.	Concerning the computation, the RPIM shape functions are more expensive because more field nodes are necessary in the local support domain.[3]
Suitable for fluid dynamics problems. [3]	

3.3 Global RPIM

3.3.1 Forewords

The meshfree methods termed as “global RPIM” is based on a global weak form: the Galerkin weak form. Note that a global background cell will be needed in this method to evaluate the integrals of a such weak formulation. As mentioned previously, RPIM was developed from the PIM and in order to overcome the singularity issue of PIM. Indeed, in the polynomial PIM, the moment matrix may be singular. More, it has been shown that the PIM based on Galerkin weak form is not robust for non-uniform node distribution. However, the RPIM show a good stability and robustness for arbitrary nodes distributions. This method is now much more used that PIM and has been successfully used for applications such as solid mechanics (2D and 3D), non-linear problems and plate and shells.

3.3.2 Formulation

The global RPIM formulation comes from (G.R. Liu, 2005). A 2D linearly elastic problem defined over the domain Ω and the boundary Γ is considered. The PDEs and BCs of 2D solid mechanics problems have been defined in the section 2.2.

Equilibrium equation:

3.27

$$\boldsymbol{L^T\sigma} + \boldsymbol{b} = 0$$

NBCs:

3.28

$$\boldsymbol{\sigma n} = \bar{\boldsymbol{t}}$$

EBCs:

3.29

$$\boldsymbol{u} = \bar{\boldsymbol{u}}$$

With

$\boldsymbol{L}$: the differential operator expressed as $\boldsymbol{L} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial_y} \\ \frac{\partial}{\partial_y} & \frac{\partial}{\partial x} \end{bmatrix}$;

$\boldsymbol{\sigma^T} = [\sigma_{xx} \quad \sigma_{yy} \quad \tau_{xy}]$**:** the stress vector;

$\boldsymbol{u^T} = [u \quad v]$: the displacement vector;

$\boldsymbol{b^T} = [b_x \quad b_y]$: the vector of body force;

$\bar{\boldsymbol{t}}$: the traction on the natural boundaries of the problem treated;

$\bar{\boldsymbol{u}}$: the displacement on the essential boundaries of the problem treated;

$\boldsymbol{n}$: the unit vector outward normal at a point on the natural boundary;

The global weak form for solids in terms of displacement is given as

3.30

$$\int_{\Omega} (\boldsymbol{L}\delta\boldsymbol{u})^T (\boldsymbol{DLu})\, d\Omega - \int_{\Omega} \delta\boldsymbol{u^T b}\, d\Omega - \int_{\Gamma_t} d\boldsymbol{u^T}\, \bar{\boldsymbol{t}}\, d\Gamma = 0$$

With $\boldsymbol{D}$ the matrix of elastic constants coming from the Hook's law for 2D elastic solids and isotropic materials expressed as

3.31

$\boldsymbol{D} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}$ for plane stress.

3.32

$$\boldsymbol{D} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)}\begin{bmatrix} 1 & \frac{1-\nu}{2} & 0 \\ \frac{1-\nu}{2} & 1 & 0 \\ 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix}$$ for plane strain.

Equation 3.30 is a weak form which is defined over Ω and to evaluate during the quadrature of its integrals, the global problem domain needs to be discretized into a set of background cells. In the same way, a set of curve background cells is necessary to evaluate the integrals of the natural boundary. It is important that these sets of background cells and curves do not overlap.

After this discretization, the problem domain needs to be represented by a set of field nodes in order to approximate the field variable, which is the displacement in our case. These last are numbered from 1 to N for the entire problem domain and the RPIM shape functions that have been explained previously will be used to compute the approximation of the field variable (displacement) at any point of interest by using the field nodes which are located in the local support domain of this point. The point of interest can be either a quadrature point or a field node.

This approximated displacement $\boldsymbol{u^h}$ at a point of interest is formulated as

3.33

$$\boldsymbol{u}^{\boldsymbol{h}}_{[2\times1]} = \sum_{I}^{n} \begin{bmatrix} \phi_I & 0 \\ 0 & \phi_I \end{bmatrix} \begin{bmatrix} u_I \\ v_I \end{bmatrix} = \sum_{I}^{n} \boldsymbol{\Phi_I u_I}$$

with

$\boldsymbol{\Phi_I}$: matrix of shape functions of node I.

$\boldsymbol{u_I}$: nodal displacements.

n: number of field nodes in the local support domain.

From this last equation, by adding the operator δ

3.34

$$\delta\boldsymbol{u}^{\boldsymbol{h}}_{[2\times1]} = \sum_{I}^{n} \boldsymbol{\Phi_I}\delta\boldsymbol{u_I} = \boldsymbol{\Phi}_{[2\times2\boldsymbol{n}]}\delta\boldsymbol{u}_{[2\boldsymbol{n}\times1]}$$

And using the strain-displacement equation and equation 3.34, the strains can be calculated using the approximated displacements.

3.35

$$\boldsymbol{\varepsilon}_{[3\times1]} = \boldsymbol{L}\boldsymbol{u^h} = \boldsymbol{L}_{[3\times2]}\boldsymbol{\Phi}_{[2\times2\boldsymbol{n}]}\boldsymbol{u}_{[2\boldsymbol{n}\times1]}$$

$$\boldsymbol{\varepsilon}_{[3\times1]} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial_y} \\ \frac{\partial}{\partial_y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} \phi_1 & 0 & \dots & \phi_n & 0 \\ 0 & \phi_1 & \dots & 0 & \phi_n \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ \dots \\ u_N \\ v_N \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial \phi_1}{\partial x} & 0 & \dots & \frac{\partial \phi_n}{\partial x} & 0 \\ 0 & \frac{\partial \phi_1}{\partial_y} & \dots & 0 & \frac{\partial \phi_n}{\partial_y} \\ \frac{\partial \phi_1}{\partial_y} & \frac{\partial \phi_1}{\partial x} & \dots & \frac{\partial \phi_n}{\partial_y} & \frac{\partial \phi_n}{\partial x} \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ \dots \\ u_N \\ v_N \end{bmatrix} = \boldsymbol{B}_{[3\times2n]} \boldsymbol{u}_{[2n\times1]}$$

And in a nodal summation form:

3.36

$$\boldsymbol{\varepsilon}_{[3\times1]} = \sum_{\boldsymbol{I}}^{\boldsymbol{n}} (\boldsymbol{B_I})_{[3\times2]} (\boldsymbol{u_I})_{[2\times1]}$$

Where $\boldsymbol{B}$ is the strain matrix and $\boldsymbol{B_I}$ the strain matrix for the node I.

And similarly, by adding δ:

3.37

$$\boldsymbol{L}\delta\boldsymbol{u}^{\boldsymbol{h}} = \sum_{\boldsymbol{I}}^{\boldsymbol{n}} (\boldsymbol{B_I})_{[3\times2]} (\delta\boldsymbol{u_I})_{[2\times1]}$$

The stress vector can now be obtained using the Hook' law at a point of the problem.

3.38

$$\boldsymbol{\sigma} = \boldsymbol{D\varepsilon} = \boldsymbol{D}_{[3\times3]} \boldsymbol{B}_{[3\times2\boldsymbol{n}]} \boldsymbol{u}_{[2\boldsymbol{n}\times1]} = \sum_{\boldsymbol{I}}^{\boldsymbol{n}} \boldsymbol{D}_{[3\times3]} (\boldsymbol{B_I})_{[3\times2]} (\boldsymbol{u_I})_{[2\times1]}$$

It should be mentioned that I and J are both numbering the nodes in the local support domain (from 1 to n respectively). The numbering system can now be changed from a local to a global that will record all the fields nodes of the problem from 1 to N (global numbering system for field nodes). Thus, I and J go now from 1 to N and if node I and node J are in a different local support domain the integrals vanish. In this manner, using equation 3.36 and 3.37, the first term of equation 3.30 can now be expressed as

3.39

$$\int_{\Omega} (\boldsymbol{L}\delta\boldsymbol{u})^T \, (\boldsymbol{DLu}) \, d\Omega = \int_{\Omega} \sum_{I}^{N} \boldsymbol{B_I^T} \sum_{J}^{N} \boldsymbol{DB_J u_J} \, d\Omega$$

and moving the integration inside the double summation:

3.40

$$\int_{\Omega} (\boldsymbol{L}\delta\boldsymbol{u})^T \, (\boldsymbol{DLu}) \, d\Omega = \sum_{I}^{N} \sum_{J}^{N} \boldsymbol{\delta u_I}^{\boldsymbol{T}} (\int_{\Omega} \boldsymbol{B_I^T D \, B_J} d\Omega) \boldsymbol{u_J} = \sum_{I}^{N} \sum_{J}^{N} \boldsymbol{\delta u_I}^{\boldsymbol{T}} \boldsymbol{K}_{IJ} \, \boldsymbol{u_J}$$

with

3.41

$$\boldsymbol{K}_{IJ_{[2\times2]}} = \int_{\Omega} (\boldsymbol{B_I^T})_{[2\times3]} \, \boldsymbol{D}_{[3\times3]} (\boldsymbol{B_J})_{[3\times2]} d\Omega$$

the nodal stiffness matrix. As above, if node I and J are in a different support domain of a same point of integration, the nodal stiffness matrix vanishes.

Finally, using the global matrix form, equation 3.40 can be expressed as

3.42

$$\int_{\Omega} (\boldsymbol{L}\delta\boldsymbol{u})^T \, (\boldsymbol{DLu}) \, d\Omega = \boldsymbol{\delta U^T K U}$$

with $\boldsymbol{K}$, the global stiffness matrix.

3.43

$$\boldsymbol{K}_{[2N\times2N]} = \begin{bmatrix} \boldsymbol{K}_{11} & \boldsymbol{K}_{12} & \dots & \boldsymbol{K}_{1N} \\ \boldsymbol{K}_{21} & \boldsymbol{K}_{22} & \dots & \boldsymbol{K}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{K}_{N1} & \boldsymbol{K}_{N2} & \dots & \boldsymbol{K}_{NN} \end{bmatrix}$$

The dimension of $\boldsymbol{K}$ is $[2N \times 2N]$ because this last is composed by assembling the nodal stiffness matrix $\boldsymbol{K}_{IJ}$ of dimension $[2 \times 2]$ with I and J varying both from 1 to N.

The vector $\boldsymbol{U}$ is called the global displacement vector, that contains all nodal displacements of all field nodes in the problem domain.

3.44

$$\boldsymbol{U}_{[2N\times1]} = \begin{bmatrix} \boldsymbol{u_1} \\ \boldsymbol{u_2} \\ \dots \\ \boldsymbol{u_N} \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \\ \dots \\ u_N \\ v_N \end{bmatrix}$$

The second term of equation 3.30 can now be developed.

3.45

$$\int_{\Omega} \delta \boldsymbol{u}^{\boldsymbol{T}} \boldsymbol{b}\, d\Omega = \int_{\Omega} (\delta \sum_{I}^{n} \boldsymbol{\Phi}_{\boldsymbol{I}} \boldsymbol{u}_{\boldsymbol{I}})^{T} \boldsymbol{b}\, d\Omega$$

And in the same manner as below, the global numbering system is now used:

3.46

$$\int_{\Omega} \delta \boldsymbol{u}^{\boldsymbol{T}} \boldsymbol{b}\, d\Omega = \int_{\Omega} (\delta \sum_{I}^{N} \boldsymbol{\Phi}_{\boldsymbol{I}} \boldsymbol{u}_{\boldsymbol{I}})^{T} \boldsymbol{b}\, d\Omega$$

Moving the integrals inside the summation:

3.47

$$\int_{\Omega} \delta \boldsymbol{u}^{\boldsymbol{T}} \boldsymbol{b}\, d\Omega = \sum_{I}^{N} \delta \boldsymbol{u}_{\boldsymbol{I}}{}^{\boldsymbol{T}} \int_{\Omega} \boldsymbol{\Phi}_{\boldsymbol{I}}{}^{\boldsymbol{T}} \boldsymbol{b}\, d\Omega = \sum_{I}^{N} \delta \boldsymbol{u}_{\boldsymbol{I}}{}^{\boldsymbol{T}} \boldsymbol{F}_{\boldsymbol{I}}^{\boldsymbol{b}}$$

where $\boldsymbol{F}_{\boldsymbol{I}}^{\boldsymbol{b}}$ is the nodal body force vector and **b** the body force vector.

3.48

$$\boldsymbol{F}_{\boldsymbol{I}}^{\boldsymbol{b}} = \int_{\Omega} \boldsymbol{\Phi}_{\boldsymbol{I}}{}^{\boldsymbol{T}} \boldsymbol{b}\, d\Omega$$

Using the global form, the second term of equation 3.30 can be rewritten as

3.49

$$\int_{\Omega} \delta \boldsymbol{u}^{\boldsymbol{T}} \boldsymbol{b}\, d\Omega = \sum_{I}^{N} \delta \boldsymbol{u}_{\boldsymbol{I}}{}^{\boldsymbol{T}} \boldsymbol{F}_{\boldsymbol{I}}^{\boldsymbol{b}} = \delta \boldsymbol{U}^{\boldsymbol{T}} \boldsymbol{F}^{\boldsymbol{b}}$$

with $\boldsymbol{F}^{\boldsymbol{b}}$ the global body force vector constructing by assembling the nodal body force vectors $\boldsymbol{F}_{\boldsymbol{I}}^{\boldsymbol{b}}$ for all the field nodes in the problem domain. i.e.

3.50

$$\boldsymbol{F}_{\boldsymbol{I}}^{\boldsymbol{b}} = \begin{bmatrix} \boldsymbol{F}_{\boldsymbol{I}}^{\boldsymbol{b}}{}_{[2\times 1]} \\ \vdots \\ \boldsymbol{F}_{\boldsymbol{I}}^{\boldsymbol{b}}{}_{[2\times 1]} \end{bmatrix}_{[2\boldsymbol{N}\times 2\boldsymbol{N}]}$$

The last term of equation 3.30 is treated in the exact same manner as that for the second term. The difference is that the body force vector $\boldsymbol{b}$ is replaced by the traction vector $\bar{\boldsymbol{t}}$ and the integrations are along the boundaries. Thus, this last term can be expressed as

3.51

$$\int_{\Gamma_t} d\boldsymbol{u}^{\boldsymbol{T}} \bar{\boldsymbol{t}}\, d\Gamma = \sum_{I}^{n} \delta \boldsymbol{u}_{\boldsymbol{I}}{}^{\boldsymbol{T}} \int_{\boldsymbol{\Gamma_t}} \boldsymbol{\Phi}_{\boldsymbol{I}}^{\boldsymbol{T}} \bar{\boldsymbol{t}} d\Gamma = \delta \boldsymbol{U}^{\boldsymbol{T}} \sum_{I}^{N} \int_{\boldsymbol{\Gamma_t}} \boldsymbol{\Phi}_{\boldsymbol{I}}^{\boldsymbol{T}}\, d\Gamma = \delta \boldsymbol{U}^{\boldsymbol{T}} \boldsymbol{F}^{\boldsymbol{t}}$$

where $\boldsymbol{F^t}$ is the global traction force vector composed by assembling all the nodal traction force vectors $\boldsymbol{F_I^t}$. Like for the global body force vector, the length of $\boldsymbol{F^t}$ is [2N×1].

3.52

$$\boldsymbol{F_I^t} = \int_{\Gamma_t} \boldsymbol{\Phi_I^T}\, d\Gamma$$

Now, using equation 3.42, 3.49 and 3.51 into equation 3.30, the following equations are obtained successively:

3.53

$$\delta \boldsymbol{U^T K U} - \delta \boldsymbol{U^T F^b} - \delta \boldsymbol{U^T F^t} = 0$$

3.54

$$\delta \boldsymbol{U^T}(\boldsymbol{KU} - \boldsymbol{F^b} - \boldsymbol{F^t}) = 0$$

And this last equation is satisfied only if the term in parenthesis is equal to the zero vector because $\delta \boldsymbol{U^T}$ is arbitrary. Thus, the previous equation is rewritten as

3.55

$$\boldsymbol{KU} = \boldsymbol{F^b} + \boldsymbol{F^t}$$

Or

3.56

$$\boldsymbol{KU} = \boldsymbol{F}$$

with $\boldsymbol{F}$ global force vector.

3.57

$$\boldsymbol{F} = \boldsymbol{F^b} + \boldsymbol{F^t}$$

Finally, the equation 3.57 is the final discretized equation of the radial point interpolation meshfree method. Thanks to it, the nodal displacement can be obtained by solving it after the enforcement of the displacement boundary condition. The strain and stress components can easily be founded using respectively equations 3.35 and 3.37.

3.3.3 Computer code

To perform global RPIM on solid mechanics problems a computer code will be used. This last has been taken and modified from (G.R. Liu, 2005). The algorithm is developed in FORTRAN 90 language and allows to approximate displacements for mechanical solid problems in two dimensions. The following flowchart illustrate the procedure on this code.

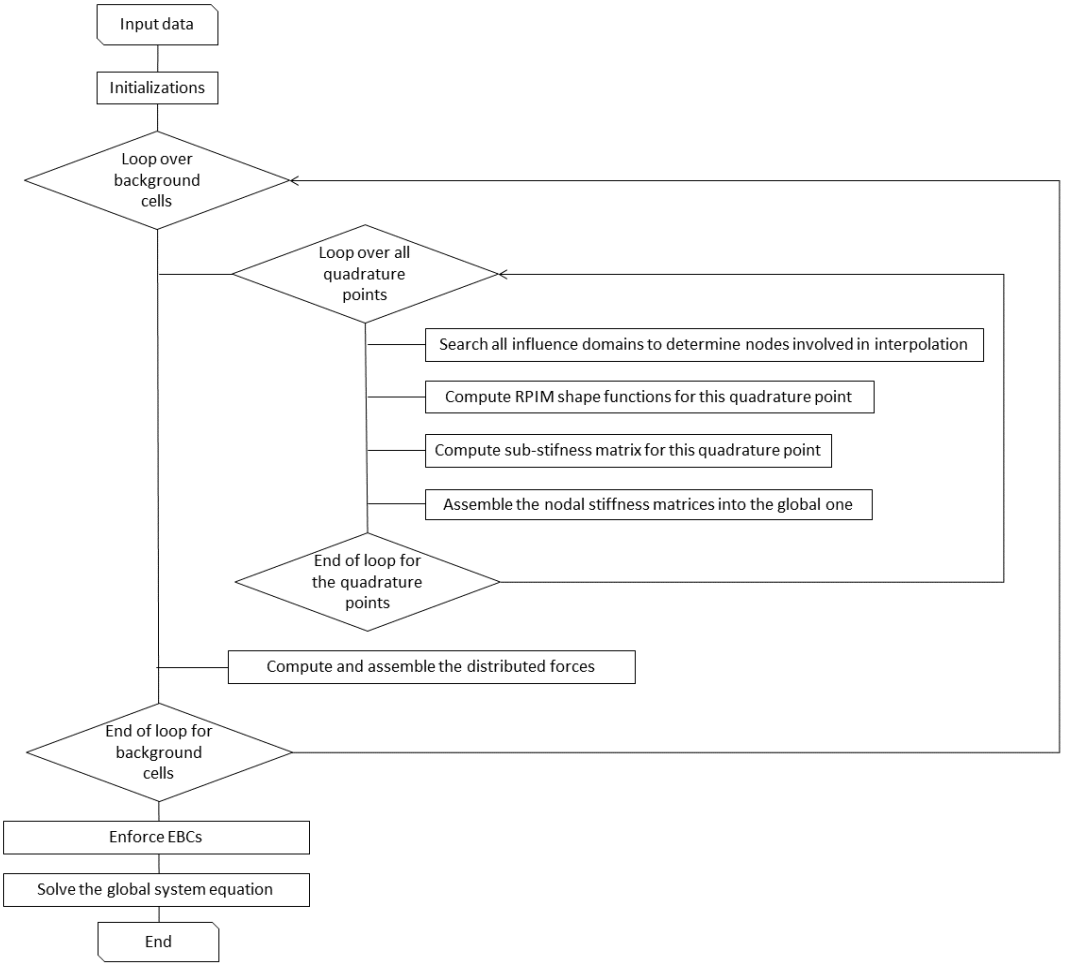

Figure 3.3 RPIM algorithm flowchart.

This report will not go through all the line of the code, but rather describe the main steps. As reminder, this report has for aim to run simple simulation in order to understand better the meshfree methods as a whole. Rather than explain all the computer code, this report will deal with meshfree challenges and parameters that the user may control. In the next sub-sections, the main implementations issue of the radial point interpolation meshfree method will be detailed. This clarification for the reader will also be valid for the computer code of the local radial point interpolation meshfree method.

3.3.3.1 Background mesh

A background mesh in needed to evaluate the integrals of the meshfree global weak form method and in particular for the RPIM. The background mesh is composed by cells that can be of various shapes (such as rectangles or triangles for 2D problem domain). In the used computer code for the RPIM, rectangular and regular background cells are used. These background cells will be the support for the gauss quadrature to perform the numerical integrations of the RPIM procedure.

3.3.3.2 Numerical integration

In the global RPIM, all integrations are over the global problem domain Ω and the global (traction) boundary Γ_t (see equation 3.30). These global integrals are evaluated by discretizing the problem domain with a background mesh mentioned above. Thanks to his, the global integrations can be now expressed as integrals summation over the cells of the background mesh. As an example, considering n_c background cells with Ω_k the domain for the cell number k, an integrand $\boldsymbol{G}$ can be expressed as

3.58

$$\int_{\Omega} \boldsymbol{G}\, d\Omega = \sum_{k}^{n_c} \int_{\Omega_k} \boldsymbol{G}\, d\Omega$$

Then, the standard Gauss quadrature scheme (common in the FEM) is used in order to numerically evaluate the integrations overt the background mesh's cells. Thus, with the example above, the integration can be expressed as

3.59

$$\int_{\Omega} \boldsymbol{G}\, d\Omega = \sum_{k}^{n_c} \int_{\Omega_k} \boldsymbol{G}\, d\Omega$$

$$\int_{\Omega} \boldsymbol{G}\, d\Omega = \sum_{k}^{n_c} \sum_{i=1}^{n_g} \widehat{w}_i\, \boldsymbol{G}(\boldsymbol{x_{Qi}}) \left|\boldsymbol{J_{ik}^D}\right|$$

with

$\widehat{w}_i$, the Gauss weight factor for the Gauss (quadrature) point number i at $\boldsymbol{x_{Qi}}$;

$\boldsymbol{x_{Qi}}$, the coordinate vector of the Gauss (quadrature) point number i;

$\boldsymbol{J_{ik}^D}$, the Jacobian matrix of the area integration of the background cell number k;

n_g, number of Gauss points used in each cell of the background mesh;

The expression for the Γ_t curve gauss quadrature is obtained similarly as

3.61

$$\int_{\Gamma_t} \boldsymbol{G}\, d\Gamma = \sum_{l}^{n_{ct}} \int_{\Gamma_{tl}} \boldsymbol{G}\, d\Gamma$$

$$\int_{\Gamma_t} \boldsymbol{G}\, d\Gamma = \sum_{l}^{n_{ct}} \sum_{i=1}^{n_{gt}} \widehat{w}_i\, \boldsymbol{G}(\boldsymbol{x_{Qi}}) \left|\boldsymbol{J_{ik}^B}\right|$$

with

$\boldsymbol{J_{ik}^B}$, the Jacobian matrix for the curve integration of the sub-boundary number l, a 1D curve integration since the problem domain is in 2D;

n_{ct}, the number of curves from the background cells that are used to discretize the global boundary Γ_t;

n_{gt}, the number of Gauss point used for each sub-curve;

Now, the nodal matrices (and so the global matrices) of the RPIM procedure can be computed numerically using the following expressions from the Gauss quadrature scheme.

Equation 3.41 of the nodal stiffness matrix $\boldsymbol{K}_{IJ}$ can be re-written as

3.61

$$\boldsymbol{K}_{IJ} = \sum_{k}^{n_c} \sum_{i=1}^{n_g} (\boldsymbol{K_{IJ}^{ik}})_{[2\times 2]}$$

with

3.62

$$(\boldsymbol{K_{IJ}^{ik}})_{[2\times 2]} = \widehat{w}_i \boldsymbol{B_I^T}(\boldsymbol{x_{Qi}}) \boldsymbol{D} \boldsymbol{B_J}(\boldsymbol{x_{Qi}}) \left|\boldsymbol{J_{ik}^D}\right|$$

Thus, the nodal matrix $\boldsymbol{K}_{IJ}$ is computed by summing all the contributions from all the Gauss (quadrature) points for which its local support domain includes both field nodes I and J. Hence, if both field node are not in the local support domain of the quadrature point number i, $\boldsymbol{K_{IJ}^{ik}}$ vanishes.

Equation of the nodal body force vector $\boldsymbol{F_I^b}$ (equation 3.48), can be re-written as

3.63

$$\boldsymbol{F_I^b} = \sum_{k}^{n_c} \sum_{i=1}^{n_g} (\boldsymbol{F_I^{ikb}})_{[2\times1]}$$

with

3.64

$$(\boldsymbol{F_I^{ikb}})_{[2\times1]} = \widehat{w}_i \boldsymbol{\Phi_I^T}(\boldsymbol{x_{Qi}}) \boldsymbol{b_I}(\boldsymbol{x_{Qi}}) \left|\boldsymbol{J_{ik}^D}\right|$$

Equation of the nodal traction force vector $\boldsymbol{F_I^t}$ (equation 3.52) can be re-written as

3.65

$$\boldsymbol{F_I^t} = \sum_{l}^{n_{ct}} \sum_{i=1}^{n_{gt}} (\boldsymbol{F_I^{ilt}})_{[2\times1]}$$

with

3.66

$$(\boldsymbol{F_I^{ilt}})_{[2\times1]} = \widehat{w}_i \boldsymbol{\Phi_I^T}(\boldsymbol{x_{Qi}}) \bar{\boldsymbol{t}}_{\boldsymbol{I}}(\boldsymbol{x_{Qi}}) \left|\boldsymbol{J_{il}^B}\right|$$

It should be mentioned that in the RPIM, all the global matrices are assembled based on Gauss (quadrature) points whose can use different support domains. In other words, the shape function matrix and the strain matrix can vary for different quadrature points.

The numerical integrations for meshfree global weak form methods such as the RPIM have been studied by many research groups and some requirements has been found empirically.

- The total number of quadrature points n_{Qtotal} should be chosen by respecting the following proportionality with the total number of field nodes N in a problem domain (for a 2D problem).

3.67

$$n_{Qtotal} > \frac{2}{3} N$$

- Considering the accuracy and convergence, the following sufficient requirement has been found on the total number of quadrature points n_{Qtotal} (for a 2D problem).

3.68

$$n_{Qtotal} = (3\sim9)\ N$$

3.3.3.3 EBCs enforcement

In the global RPIM formulation, the traction BCs are naturally formulated in the global discretized system equation, whereas the EBCs need a special treatment. The EBCs are enforced directly and

accurately in the RPIM method because the shape functions of this method possess the Kronecker delta function property. The penalty method is used to enforce the displacement essential condition and consist in replacing terms in the global stiffness matrix diagonal entry and in the global force vector as

3.69

$$K_{ii} \xrightarrow{replacing} \alpha K_{ii}$$

$$F_j \xrightarrow{replacing} \begin{cases} \alpha K_{ii} \bar{u}_i & if\ i = j \\ F_j & if\ i \neq j \end{cases}$$

The coefficient α is the penalty coefficient and it should be chosen much larger than the components of the global stiffness matrix $\boldsymbol{K}$. This EBCs enforcement technique has for advantages to limit the modification of the matrices and so implies a simple algorithm modification. Nevertheless, the penalty method only satisfies approximately the EBC and the solution accuracy of the meshfree method will be affected by the penalty coefficient selection. Empirically and thanks to FEM data, it has been shown that the penalty coefficient can be chosen as

3.70

$$\alpha = 10^4 \sim 10^6 \times (K_{ii})_{max}$$

with $(K_{ii})_{max}$, the maximum element of the diagonal of the global stiffness matrix. Thus, α is simply determined in the RPIM computer code with checking algorithm for matrix $\boldsymbol{K}$ maximum diagonal element value.

3.3.3.4 Stiffness matrix properties

Firstly, the symmetry of global matrix $\boldsymbol{K}$ and can be demonstrated thanks to the nodal stiffness matrix expression. Indeed, in its expression (equation 3.31) the matrix $\boldsymbol{D}$ is symmetric by definition and so is the product $\boldsymbol{B_I^T D B_J}$.

Thus,

3.71

$$(\boldsymbol{K_{IJ}})^{\boldsymbol{T}} = \boldsymbol{K_{IJ}}$$

And so, the global matrix $\boldsymbol{K}$ is symmetric.

Secondly, as stated previously, $\boldsymbol{K}$ computed by assembling the nodal matrices $\boldsymbol{K_{IJ}}$ and these last are non-zero only if field node I and J are within at least one support domain of a Gauss point. Therefore, if both nodes I and J are far apart and so do not have the same support domain of any Gauss point, $\boldsymbol{K_{IJ}}$ vanishes. Thus, as long as the support domain is compact and small enough, most of the $\boldsymbol{K_{IJ}}$ will be zero and so $\boldsymbol{K}$ will be sparse.

3.3.3.5 Radial basis functions shape parameters

In the section 3.2, the RPIM shape function has been detailed and especially the use of radial basis functions in the construction procedure. Depending on the RBF used, several shape parameters are implied and need to be chosen. These different shape parameters affect the accuracy of the solution, and its appropriate values can only be determined empirically by numerical examinations.

3.4 Local RPIM

3.4.1 Forewords

The global RPIM has been presented and as a reminder: in this method based on global Galerkin weak form, global background cells are necessary to evaluate the integrations and computing the system matrices. Indeed, in a global weak formulation, the system equations need to be satisfied over the entire problem domain and so a background mesh is required. Thus, such a method is, by definition, not truly a meshfree method.

The local radial point interpolation method (LRPIM) was developed by GR Liu and his co-workers from the concept of the meshless local Petrov-Galerkin method (MLPG), a numerical method proposed by Alturi and Zhu. Historically, the MLPG came in order to avoid the use of background meshes. The main difference between these two methods is that the LRPIM use RPIM shape function instead of MLS shape functions. Hence, if a local weak formulation, such as the local Galerkin weak formulation is used on field nodes, the integrations are evaluated over local quadrature domains defined for each node. This local quadrature domain may also be the local domain where the test/weight function is defined. Its shape is generally simple and regular, such as rectangles or spheres. The LRPIM has been successfully applied to problems such as solid mechanics, fluid mechanics or soil mechanics.

3.4.2 Formulation

The LRPIM formulation comes from (G.R. Liu, 2005) .A 2D linearly elastic problem defined over the domain Ω and the boundary Γ is considered. For a given node I, the governing equation 2.18 (equilibrium equation) is satisfied using a local weighted residual method that leads to a weak form equation for this node. This local weighted residual form is defined over a local quadrature domain Ω_q and its boundary Γ_q as

3.72

$$\int_{\Omega_q} \widehat{W}_I(\sigma_{ij,j} + b_i)\, d\Omega = 0$$

with $\widehat{W}_I$ the weight/test function centred at node I. This previous equation is applied to all the field nodes that describe the entire problem domain.

The Meshfree local weak form method only needs the local compatibility in the local quadrature domain Ω_q and Γ_q. That is to say that the solution exists as long as the field variable approximation is continuous at any point in the local quadrature domain and so the shape function is differentiable.

By an integration by parts, the following equation is obtained from the first term of equation 3.72:

11

$$\int_{\Omega_q} \widehat{W}_I \sigma_{ij,j}\, d\Omega = \int_{\Gamma_q} \widehat{W}_I\, n_j \sigma_{ij} d\Gamma - \int_{\Omega_q} \widehat{W}_{I,j} \sigma_{ij}\, d\Omega$$

with n_j the j-th component of the unit vector outward the normal on the boundary. Using equation 3.73 into equation 2.72, the following local-weak form expression is obtained.

3.74

$$\int_{\Gamma_q} \widehat{W}_I\, n_j \sigma_{ij} d\Gamma - \int_{\Omega_q} \left[\widehat{W}_{I,j} \sigma_{ij} - \widehat{W}_I b_i\right] d\Omega = 0$$

It should be noticed that the local quadrature domain boundary Γ_q is composed by three sub-boundaries: $\Gamma_q = \Gamma_{qi} \cup \Gamma_{qu} \cup \Gamma_{qt}$ and where

Γ_{qi}: is the boundary of the quadrature domain which does not intersect with the global boundary Γ.

Γ_{qu}: is the intersecting part of the natural boundary and the quadrature domain.

Γ_{qt}: is the intersecting part of the essential boundary and the quadrature domain.

Thanks to these sub-boundaries, equation 3.74 can be developed as

3.75

$$\int_{\Gamma_{qi}} \widehat{W}_I\, n_j \sigma_{ij} d\Gamma + \int_{\Gamma_{qu}} \widehat{W}_I\, n_j \sigma_{ij} d\Gamma + \int_{\Gamma_{qt}} \widehat{W}_I\, n_j \sigma_{ij} d\Gamma - \int_{\Omega_q} \left[\widehat{W}_{I,j} \sigma_{ij} - \widehat{W}_I b_i\right] d\Omega = 0$$

If the local quadrature domain is located entirely inside the global domain, in other words if $\Gamma_q \cap \Gamma = \emptyset$ then $\Gamma_{qi} = \Gamma_q$ and there are no integrals over the boundaries Γ_{qu} and Γ_{qt}. In this case, equation 3.75 can be rewritten as

3.76

$$\int_{\Gamma_{qi}} \widehat{W}_I\, n_j \sigma_{ij} d\Gamma - \int_{\Omega_q} \left[\widehat{W}_{I,j} \sigma_{ij} - \widehat{W}_I b_i\right] d\Omega = 0$$

These last two equations 3.75 and 3.76, which are local weak forms can be simplified by selecting the weight function $\widehat{W}_I$ so that vanishes on Γ_{qi}. That it to say that the weight function is chosen to be zero along Γ_{qi}. Such a function can be for example the cubic W1 function or the quartic spline W2. Thus, in the case of a local quadrature domain that intersects with Γ, equation 3.75 becomes

3.77

$$\int_{\Gamma_{qu}} \widehat{W}_I \, n_j \sigma_{ij} d\Gamma + \int_{\Gamma_{qt}} \widehat{W}_I \, n_j \sigma_{ij} d\Gamma - \int_{\Omega_q} \left[\widehat{W}_{I,j} \sigma_{ij} - \widehat{W}_I b_i\right] d\Omega = 0$$

and for a node with a local quadrature domain that does not intersect with Γ :

3.78

$$- \int_{\Omega_q} \left[\widehat{W}_{I,j} \sigma_{ij} - \widehat{W}_I b_i\right] d\Omega = 0$$

As a reminder the equation of the natural boundary condition (equation 3.28) is

$$\sigma n = \bar{t}$$

The equation that relies the stress ant the traction on the boundary is given by

3.79

$$\sigma_{ij} n_j = t_i$$

and using this relation with the natural boundary condition (equation 3.28) into equation 3.75:

3.80

$$\int_{\Omega_q} \widehat{W}_{I,j} \sigma_{ij} \, d\Omega - \int_{\Gamma_{qi}} \widehat{W}_I \, t_i d\Gamma - \int_{\Gamma_{qu}} \widehat{W}_I \, t_i d\Gamma = \int_{\Omega_q} \widehat{W}_I b_i \, d\Omega + \int_{\Gamma_{qt}} \widehat{W}_I \, \bar{t}_i d\Gamma$$

This last equation is composed of integrations over a local quadrature domain and so smooth out the numerical error. The local radial point interpolation meshfree method satisfies the equilibrium equation for a node and over a local quadrature domain. The size of such a domain is so a very important parameter. To get this discretized system of equations, the global domain Ω needs to be represented by a set of field nodes. The RPIM shape functions that have been described in section 3.2 are used to approximate the trial function for the displacement at a point at $\boldsymbol{x}$:

3.81

$$\boldsymbol{u}^{\boldsymbol{h}}_{[2\times1]}(\boldsymbol{x}) = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \emptyset_1 & 0 & \dots & \emptyset_n & 0 \\ 0 & \emptyset_1 & \dots & 0 & \emptyset_n \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ \dots \\ u_N \\ v_N \end{bmatrix} = \boldsymbol{\Phi}_{[2\times2n]} \boldsymbol{u}_{[2n\times1]}$$

with

n: the number of field nodes in the support domain of the point of interest at **x**. These n field nodes vary from 1 to n, it is a local numbering system used on the support domain of the point of interest;

$\boldsymbol{\Phi}_{[2\times 2n]}$: the RPIM shape functions matrix which is constructed using the n nodes;

It should be mentioned that all the field nodes are numbered from 1 to N by a global numbering system (different for the local one used for the support domains). This last will be used at the assembling step of the local nodal matrices in order to obtain the global matrix form. By an algorithmic approach, that is to say that a specific index will be necessary to collect the global number of a field node that will be used in a support domain.

In the same way as the radial point interpolation meshfree method procedure the strain and stress can be expressed as

3.82

$$\boldsymbol{\varepsilon}_{[3\times 1]} = \boldsymbol{B}_{[3\times 2\boldsymbol{n}]}\boldsymbol{u}_{[2\boldsymbol{n}\times 1]}$$

3.83

$$\boldsymbol{\sigma}_{[3\times 1]} = \boldsymbol{D}_{[3\times 3]}\boldsymbol{\varepsilon}_{[3\times 1]} = \boldsymbol{D}_{[3\times 3]}\boldsymbol{B}_{[3\times 2\boldsymbol{n}]}\boldsymbol{u}_{[2\boldsymbol{n}\times 1]}$$

with $\boldsymbol{B}_{[3\times 2\boldsymbol{n}]}$ the strain matrix and $\boldsymbol{D}_{[3\times 3]}$ the matrix of elastic constants respectively defined as

3.84

$$\boldsymbol{B}_{[3\times 2\boldsymbol{n}]} = \begin{bmatrix} \frac{\partial \phi_1}{\partial x} & 0 & \dots & \frac{\partial \phi_n}{\partial x} & 0 \\ 0 & \frac{\partial \phi_1}{\partial_y} & \dots & 0 & \frac{\partial \phi_n}{\partial_y} \\ \frac{\partial \phi_1}{\partial_y} & \frac{\partial \phi_1}{\partial x} & \dots & \frac{\partial \phi_n}{\partial_y} & \frac{\partial \phi_n}{\partial x} \end{bmatrix}$$ (Strain matrix)

3.85

$$\boldsymbol{D} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}$$ (Matrix of elastic constants for plane stress)

3.86

$$\boldsymbol{D} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)}\begin{bmatrix} 1 & \frac{1-\nu}{2} & 0 \\ \frac{1-\nu}{2} & 1 & 0 \\ 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix}$$ (Matrix of elastic constants for plane strain)

The equation 3.80 can then expressed in a matrix form:

3.87

$$\int_{\Omega_q} \widehat{\boldsymbol{V}}_{\boldsymbol{I}} \boldsymbol{\sigma}\, d\Omega - \int_{\Gamma_{qi}} \widehat{\boldsymbol{W}}_{\boldsymbol{I}}\, \boldsymbol{t} d\Gamma - \int_{\Gamma_{qu}} \widehat{\boldsymbol{W}}_{\boldsymbol{I}}\, \boldsymbol{t} d\Gamma = \int_{\Omega_q} \widehat{\boldsymbol{W}}_{\boldsymbol{I}} \boldsymbol{b}\, d\Omega + \int_{\Gamma_{qt}} \widehat{\boldsymbol{W}}_{\boldsymbol{I}}\, \bar{\boldsymbol{t}} d\Gamma$$

with $\widehat{\boldsymbol{W}}_{\boldsymbol{I}}$ and $\widehat{\boldsymbol{V}}_{\boldsymbol{I}}$ respectively the matrices of weight functions and derivatives of the weight functions.

3.88

$$\widehat{\boldsymbol{W}}_{\boldsymbol{I}} = \widehat{\boldsymbol{W}}(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{I}}) \begin{bmatrix} \widehat{W}(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{I}}) & 0 \\ 0 & \widehat{W}(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{I}}) \end{bmatrix}$$

3.89

$$\widehat{\boldsymbol{V}}_{\boldsymbol{I}} = \widehat{\boldsymbol{V}}(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{I}}) = \begin{bmatrix} \widehat{W}_{,x}(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{I}}) & 0 \\ 0 & \widehat{W}_{,y}(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{I}}) \\ \widehat{W}_{,y}(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{I}}) & \widehat{W}_{,x}(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{I}}) \end{bmatrix}$$

The traction at a point $\boldsymbol{x}$ is expressed as

3.90

$$\boldsymbol{t} = \begin{bmatrix} t_x \\ t_y \end{bmatrix} = \begin{bmatrix} n_x & 0 & n_y \\ 0 & n_y & n_x \end{bmatrix} \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{bmatrix} = \boldsymbol{n}_{[2\times 3]} \boldsymbol{D}_{[3\times 3]} \boldsymbol{B}_{[3\times 2\boldsymbol{n}]} \boldsymbol{u}_{[2\boldsymbol{n}\times 1]}$$

and

3.91

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{bmatrix} = \boldsymbol{D}_{[3\times 3]} \boldsymbol{B}_{[3\times 2\boldsymbol{n}]} \boldsymbol{u}_{[2\boldsymbol{n}\times 1]}$$

where $\boldsymbol{n}_{\Gamma}$ is the unit vector outward the normal direction on the boundary.

3.92

$$\boldsymbol{n}_{\Gamma} = \begin{bmatrix} n_x \\ n_y \end{bmatrix}$$

and

3.93

$$\boldsymbol{n}_{[2\times 3]} = \begin{bmatrix} n_x & 0 & n_y \\ 0 & n_y & n_x \end{bmatrix}$$

Using the equations 3.88 to 3.93 into equation 3.87 gives a discretised system of equation for the I-th node.

3.94

$$\int_{\Omega_q} \widehat{\boldsymbol{V}}_{\boldsymbol{I}} \boldsymbol{D}\boldsymbol{B}\boldsymbol{u}\, d\Omega - \int_{\Gamma_{qi}} \widehat{\boldsymbol{W}}_{\boldsymbol{I}}\, \boldsymbol{n}\boldsymbol{D}\boldsymbol{B}\boldsymbol{U} d\Gamma - \int_{\Gamma_{qu}} \widehat{\boldsymbol{W}}_{\boldsymbol{I}}\, \boldsymbol{n}\boldsymbol{D}\boldsymbol{B}\boldsymbol{U} d\Gamma = \int_{\Omega_q} \widehat{\boldsymbol{W}}_{\boldsymbol{I}} \boldsymbol{b}\, d\Omega + \int_{\Gamma_{qt}} \widehat{\boldsymbol{W}}_{\boldsymbol{I}}\, \bar{\boldsymbol{t}} d\Gamma$$

The matrix form of this last equation is

3.95

$$(\boldsymbol{K_I})_{[2\times 2\boldsymbol{n}]}(\boldsymbol{u})_{[2\boldsymbol{n}\times 1]} = (\boldsymbol{f_I})_{[2\times 1]}$$

with

$\boldsymbol{f_I}$, the nodal force vector composed either by the contributions of the body forces applied in the problem domain and tractions applied on the boundary.

3.96

$$\boldsymbol{f_I} = \int_{\Omega_q} \widehat{\boldsymbol{W}}_{\boldsymbol{I}} \boldsymbol{b}\, d\Omega + \int_{\Gamma_{qt}} \widehat{\boldsymbol{W}}_{\boldsymbol{I}}\, \bar{\boldsymbol{t}} d\Gamma$$

$\boldsymbol{K_I}$, the nodal stiffness matrix for the field node I.

3.97

$$\boldsymbol{K_I} = \int_{\Omega_q} \widehat{\boldsymbol{V}}_{\boldsymbol{I}} \boldsymbol{D}\boldsymbol{B}\boldsymbol{u}\, d\Omega - \int_{\Gamma_{qi}} \widehat{\boldsymbol{W}}_{\boldsymbol{I}}\, \boldsymbol{n}\boldsymbol{D}\boldsymbol{B}\boldsymbol{U} d\Gamma - \int_{\Gamma_{qu}} \widehat{\boldsymbol{W}}_{\boldsymbol{I}}\, \boldsymbol{n}\boldsymbol{D}\boldsymbol{B}\boldsymbol{U} d\Gamma$$

$\boldsymbol{u}$, the vector of the displacements for all the field nodes that are within any support domain of the quadrature/integration points in the quadrature domain of node I.

The equation 3.95 corresponds to the general form of system equations for a given field node. Furthermore, for a local quadrature domain of a given field node that is located within the global domain Ω without intersecting Γ, $\boldsymbol{K_I}$ and $\boldsymbol{f_I}$ can be simplified as

3.98

$$\boldsymbol{K_I} = \int_{\Omega_q} \widehat{\boldsymbol{V}}_{\boldsymbol{I}} \boldsymbol{D}\boldsymbol{B}\boldsymbol{u}\, d\Omega - \int_{\Gamma_{qi}} \widehat{\boldsymbol{W}}_{\boldsymbol{I}}\, \boldsymbol{n}\boldsymbol{D}\boldsymbol{B}\boldsymbol{U} d\Gamma$$

3.99

$$\boldsymbol{f_I} = \int_{\Omega_q} \widehat{\boldsymbol{W}}_{\boldsymbol{I}} \boldsymbol{b}\, d\Omega$$

3.4.3 Computer code

The numerical applications on 2D solid mechanics problem will be performed using a computer code provided in (G. R. Liu and Y. T. Gu, 2005). As for the RPIM, the algorithm is coded in FORTRAN 90 language and will be modified purposely depending on the applications and meshfree parameters investigation. The general procedure of this code is illustrated in the following Figure 3.4.

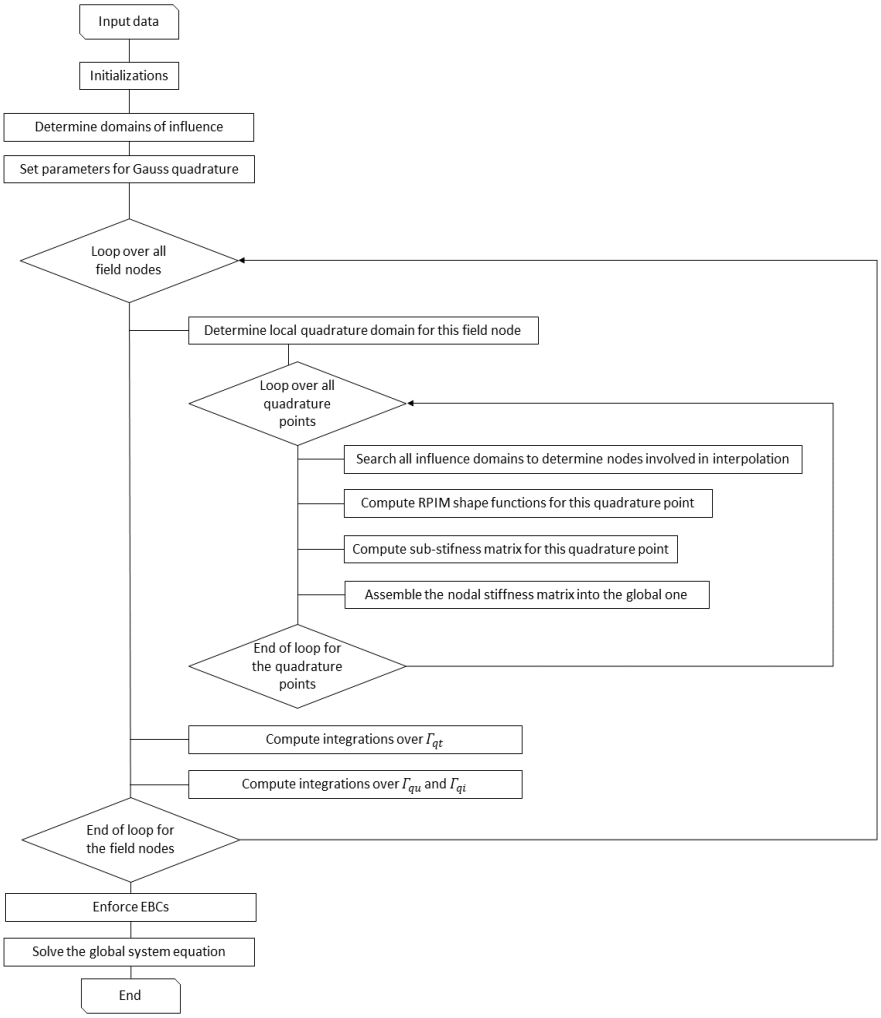

Figure 3.4 LRPIM algorithm flowchart.

3.4.3.1 Numerical integration

The gauss quadrature is used to approximate the integrals in the expressions of $\boldsymbol{K_I}$ and $\boldsymbol{f_I}$ in equation 3.95. The integrals over Ω_q are area integrals and the one along Γ_{qi}, Γ_{qu} and Γ_{qt} are curve integrals. Considering a local quadrature domain in a rectangular shape, according to the standard Gauss quadrature scheme, the equations 3.98 and 3.99 can be rewritten as

3.100

$$\boldsymbol{K_I} = \sum_{k=1}^{n_{gauss}} \widehat{w}_k \, \widehat{\boldsymbol{V}}_{\boldsymbol{I}}^{\boldsymbol{T}}(\boldsymbol{x_{Qk}}) \boldsymbol{DB}(\boldsymbol{x_{Qk}}) \left|\boldsymbol{J_{\Omega q}}\right| - \sum_{k=1}^{n_{sub-gauss}} \widehat{w}_k \, \widehat{\boldsymbol{W}}_{\boldsymbol{I}}^{\boldsymbol{T}}(\boldsymbol{x_{Qk}}) \boldsymbol{nDB}(\boldsymbol{x_{Qk}}) \left|\boldsymbol{J_{\Gamma qi}}\right| - \sum_{k=1}^{n_{sub-gauss}} \widehat{w}_k \, \widehat{\boldsymbol{W}}_{\boldsymbol{I}}^{\boldsymbol{T}}(\boldsymbol{x_{Qk}}) \boldsymbol{nDB}(\boldsymbol{x_{Qk}}) \left|\boldsymbol{J_{\Gamma qu}}\right|$$

3.101

$$\boldsymbol{f_I} = \sum_{k=1}^{n_{sub-gauss}} \widehat{w}_k \, \widehat{\boldsymbol{W}}_{\boldsymbol{I}}^{\boldsymbol{T}}(\boldsymbol{x_{Qk}})\bar{\boldsymbol{t}}\left|\boldsymbol{J_{\Gamma qt}}\right| + \sum_{k=1}^{n_{sub-gauss}} \widehat{w}_k \, \widehat{\boldsymbol{W}}_{\boldsymbol{I}}^{\boldsymbol{T}}(\boldsymbol{x_{Qk}})\boldsymbol{b}\left|\boldsymbol{J_{\Omega q}}\right|$$

With

n_{gauss}: the total number of Gauss/integrations points in the quadrature domain.

$n_{sub-gauss}$: the number of Gauss/integrations points that are used on a sub-curve.

$\boldsymbol{x_{Qk}}$: the Gauss point considered (point of interest).

$\widehat{w}_k$: the weight factor of the standard gauss quadrature for the point$\boldsymbol{x_{Qk}}$.

$\boldsymbol{J_{\Omega q}}$: the Jacobian matrix for the area integration of $\boldsymbol{\Omega_q}$ (local quadrature domain).

$\boldsymbol{J_{\Gamma qi}}$: the Jacobian matrix for the curve integration of $\boldsymbol{\Gamma_{qi}}$.

$\boldsymbol{J_{\Gamma qu}}$: the Jacobian matrix for the curve integration of $\boldsymbol{\Gamma_{qu}}$.

$\boldsymbol{J_{\Gamma qt}}$: the Jacobian matrix for the curve integration of $\boldsymbol{\Gamma_{qt}}$.

All the different integration points located in the same local quadrature domain can use different support domains. That is to say that $\boldsymbol{K_I}$ and $\boldsymbol{f_I}$ can be different for different Gauss points.

Finally, in order to assemble the global system of equations, the equation 3.95 is applied for all field node (from 1 to N) of all the problem domain using the global numbering system. That changes the system of equations from local to global by passing from 2 linear equations (for one node) to 2N linear equations (for all field nodes):

3.102

$$\boldsymbol{K}_{[2\boldsymbol{N}\times 2\boldsymbol{N}]}\boldsymbol{U}_{[2\boldsymbol{N}\times 1]} = \boldsymbol{F}_{[2\boldsymbol{N}\times 1]}$$

$$\begin{bmatrix} \boldsymbol{K}_{11} & \boldsymbol{K}_{12} & \cdots & \boldsymbol{K}_{1(2N-1)} & \boldsymbol{K}_{1(2N)} \\ \boldsymbol{K}_{12} & \boldsymbol{K}_{22} & \cdots & \boldsymbol{K}_{2(2N-1)} & \boldsymbol{K}_{2(2N)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{K}_{(2I-1)1} & \boldsymbol{K}_{(2I-1)2} & \cdots & \boldsymbol{K}_{(2I-1)(2N-1)} & \boldsymbol{K}_{(2I-1)(2N)} \\ \boldsymbol{K}_{(2I)1} & \boldsymbol{K}_{(2I)2} & \cdots & \boldsymbol{K}_{(2I)(2N-1)} & \boldsymbol{K}_{(2I)(2N)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{K}_{(2N-1)1} & \boldsymbol{K}_{(2N-1)2} & \cdots & \boldsymbol{K}_{(2N-1)(2N-1)} & \boldsymbol{K}_{(2N-1)(2N)} \\ \boldsymbol{K}_{(2N)1} & \boldsymbol{K}_{(2N)2} & \cdots & \boldsymbol{K}_{(2N)(2N-1)} & \boldsymbol{K}_{(2N)(2N)} \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ \cdots \\ u_I \\ v_I \\ \cdots \\ u_N \\ v_N \end{bmatrix} = \begin{bmatrix} f_{1x} \\ f_{1y} \\ \cdots \\ f_{Ix} \\ f_{Iy} \\ \cdots \\ f_{Nx} \\ f_{Ny} \end{bmatrix}$$

The developed form of equation 3.102 above shows that matrix $\boldsymbol{K}$ is composed by assembling all the nodal equations; both equations for node I correspond to the (2I-1) and 2I-th rows. The method to assemble the global stifness matrix $\boldsymbol{K}$ is different from that on the radial point interpolation meshfree method (based on global weak- form). For the LRPIM the nodal matrices are added row by row whereas in RPIM, the nodal matrices are assembled symmetrically into the global stiffness matrix.

The essential boundary condition has not been considered in the formulation because the RPIM shape functions possess the Kronecker delta function property. In other words, the enforcement of the EBC can be done in the same way as for the conventional radial point interpolation meshfree method. After that, the final discretized system of equations can be solved to get the displacements for all field nodes and the stress and strain can be calculated using equations 3.82 and 3.83.

3.4.3.2 Local domains used in LRPIM

As stated in the previous sub-section, the gauss quadrature is employed to evaluate the integrals in equations 3.95 The Figure 3.4 shows that for each field node, the computer code will construct a local quadrature cell Ω_q. Then, for each Gauss (quadrature) points of each quadrature cells, the RPIM shape functions will be computed. Thus, for each field node, three local domains are considered:

- The local quadrature domain (quadrature cell) Ω_q of size r_q.
- The local weight (test) function domain Ω_w of size r_w defined where w_i is non-zero.
- The local support domain Ω_s of size r_s for the Gauss point $\mathbf{x}_{\boldsymbol{Q}}$.

All these domains are considered arbitrary if the quadrature domain is smaller than the local test function domain ($\mathrm{r}_q \leq \mathrm{r}_w$). Since the weight function can be chosen purposely so that the integration along the internal boundary Γ_{qi} vanishes, to simplify both LRPIM procedure and computer code, Ω_q and Ω_w are chosen to be the same size ($\mathrm{r}_q = \mathrm{r}_w$).

Furthermore, the size of Ω_q and Ω_s for the field node I are defined as

3.103

$$\mathrm{r}_q = \alpha_{\mathrm{q}}\, d_{\mathrm{cI}}$$

and

3.104

$$r_s = \alpha_s \, d_{cI}$$

with

d_{cI}, the nodal spacing (or average nodal spacing) around the field node I;

α_q, the dimensionless size of Ω_q;

α_s, the dimensionless size of Ω_s;

3.4.3.3 Stiffness matrix properties

The final stiffness matrix $\boldsymbol{K}$ of the LRPIM is sparse like in the RPIM as long as the support domain is compact. In addition, $\boldsymbol{K}$ is generally asymmetric and banded. Its asymmetry is mainly due to causes. Indeed, in Petrov-Galerkin formulation the trial and test functions may be different as well as the sizes (or shapes) of the support domains to construct it. More, the sizes (or shapes) of the local quadrature domains may vary depending on the field node. Thus, the integration over Ω_q, Γ_{qi} and Γ_{qu} in equation 3.95 may be asymmetric.

3.4.3.4 Test/weight function

The local radial point interpolation method is derived from a local weighted method and therefore the test (weight) function is on importance. This test function can be any as long as the continuity condition is satisfied. Since a local weak formulation is employed to formulate the LRPIM, local quadrature domains are constructed for each field nodes centred on it. It seems logical to use test function that increase in magnitude when the distance with the centre decrease. More, the test function used will depend only on the distance between the point of interest and the centre, such as the cubic spline function or the 4th-order spline weight function. In addition, the test function is selected purposely to be zero over Γ_{qi}.

To sum up, if we consider the weight function $\widehat{W}$ that depend on the distance between the quadrature point considered and the centre (field node) $\boldsymbol{x} - \boldsymbol{x_i}$, the following properties need to be satisfied.

- $\widehat{W}(\boldsymbol{x} - \boldsymbol{x_i}) > 0$ within the support domain Ω_s (which is equivalent to Ω_q).
- $\widehat{W}(\boldsymbol{x} - \boldsymbol{x_i}) = 0$ outside Ω_s.
- $\widehat{W}(\boldsymbol{x} - \boldsymbol{x_i})$ monotonically decreases from the point of interest (quadrature point) at $\boldsymbol{x}$.
- $\widehat{W}(\boldsymbol{x} - \boldsymbol{x_i})$ is sufficient smooth and especially on the boundary of its support domain Ω_s.

In the computer code of LRPIM, the cubic spline function W1 is employed as test function but it could be any other which would satisfy the above properties. The W1 is expressed as

3.105

$$\widehat{W}_i(\boldsymbol{x}) = \begin{cases} \dfrac{2}{3} - 4\bar{r}_i{}^2 + 4\bar{r}_i{}^3 & (\bar{r}_i \le 0.5) \\ \dfrac{4}{3} - 4\bar{r}_i + 4\bar{r}_i{}^2 - \dfrac{4}{3}\bar{r}_i{}^3 & (0.5 < \bar{r}_i \le 1) \\ 0 & (\bar{r}_i > 1) \end{cases}$$

3.106

$$\bar{r}_i = \frac{d_i}{r_w} = \frac{|\boldsymbol{x} - \boldsymbol{x_i}|}{r_w}$$

where d_i is the distance between the point of interest at $\boldsymbol{x}$ (quadrature point) and the field node at $\boldsymbol{x_i}$ and r_w is the size of the local weight test function domain Ω_w (note that r_w has be chosen to be the same as r_q).

4. Numerical application

4.1 Cantilever beam

In this section, a rectangular cantilever beam subjected to a parabolic traction at its free end is numerically studied. This example is often used as a benchmark for numerical methods because its analytical solution is known. In consequence, this first numerical application allows to study the meshfree methods detailed in the previous sections. Thus, the different meshfree method properties and parameters could be highlighted such as the shape parameters, the convergence, the CPU time, the number of field nodes, or the effects of the different influence domains. Indeed, many factors are implied in a meshfree method, and this first application aims to quantify how these factors will affect the solution accuracy.

The studied beam is so parabolically loaded at its free end with an external force P in the y direction. Its thickness is unit ($t = 1$) and since the geometry of the problem do not depend on z its surface normal to z is free, the problem is considered as a plane stress problem (in z). Thus, the stress matrix is

4.1

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx}(x,y) & \tau_{xy}(x,y) \\ \tau_{xy}(x,y) & \sigma_{yy}(x,y) \end{bmatrix}$$

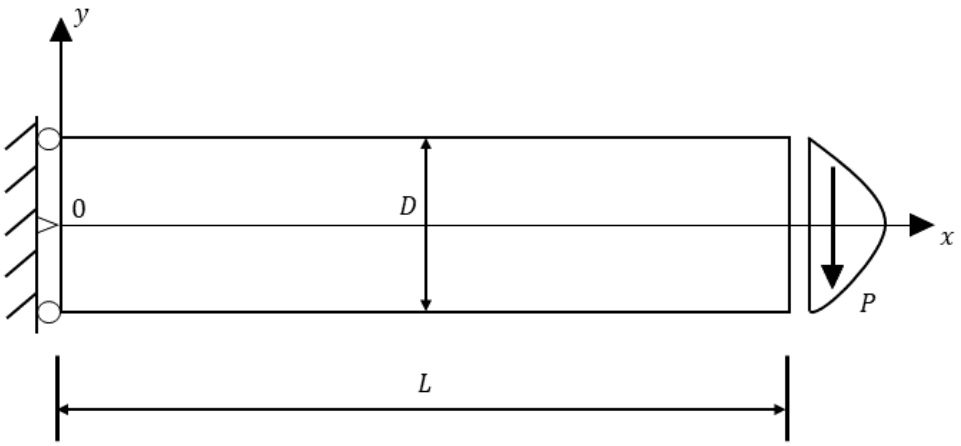

Figure 1 Cantilever beam subjected to a parabolic load at its right end.

The exact solution of this problem (illustrated on Figure 4.1) is known and listed as follow. [4]

The displacement in the x direction is :

4.2

$$u(x,y) = -\frac{Py}{6EI}\left[(6L - 3x)x + (2+\nu)\left(y^2 - \frac{D^2}{4}\right)\right]$$

where I is the moment of inertia expressed for a beam with rectangular cross-section and unit thickness as

4.3

$$I = \frac{D^3}{12}$$

The displacement in the y direction is given as

4.4

$$v(x,y) = \frac{P}{6EI}\left[3\nu y^2(L-x) + (4+5\nu)\frac{D^2}{4} + (3L-x)x^2\right]$$

The normal stress on the cross-section of the cantilever is

4.5

$$\sigma_{xx}(x,y) = -\frac{P(L-x)y}{I}$$

The normal stress in the y direction is

4.6

$$\sigma_{yy} = 0$$

The shear stress on the cross-section of the beam is

4.7

$$\tau_{xy}(x,y) = \frac{P}{2I}\left[\frac{D^2}{4} - y^2\right]$$

The parameter of the cantilever beam and its elastic material constants are listed in the following table.

Table 4.1 Table of the cantilever geometry parameters and material constants.*[1]*

Property (SI unit)	**Value**
Loading (integration of the parabolically distributed traction) (N)	$P = -1000$
Height (m)	$D = 12$
Length (m)	$L = 48$
Young's modulus ()	$E = 3 \times 10^7$
Poisson's ratio	$\nu = 0.3$

The force P is distributed in a parabolic fashion at the right free end ($x = L$) of the beam as

4.8

$$t_{xy}\big|_{x=0} = \frac{P}{2I}\left[\frac{D^2}{4} - y^2\right]$$

At the left end boundary ($x = 0$), the EBCs are expressed using the analytical equations 4.2 and 4.4.

4.9

$$u|_{x=0} = -\frac{P(2+\nu)}{6EI}\left[y^2 - \frac{D^2}{4}\right]$$

4.10

$$v|_{x=0} = -\frac{P\nu}{2EI}y^2$$

Finally, as an error indicator, the strain energy error e will be computed as an accuracy indicator for the meshfree numerical methods detailed in sections 3.3 and 3.4.

4.11

$$e = \left(\frac{1}{2}\int_{\Omega}(\varepsilon^{num} - \varepsilon^{exact})^T \boldsymbol{D}(\varepsilon^{num} - \varepsilon^{exact})\, d\Omega\right)^{\frac{1}{2}}$$

where ε^{num} and ε^{exact} are the strain vectors obtained respectively by a numerical method and the analytical method. [1]

4.1.1 Prerequisites

First the meshfree properties and the problem are established in an input data file including:

1) The definition of the problem domain geometry. The cantilever is modelled with a set of field nodes purposely chosen for the studies.
2) The background mesh. Background cells may be created to compute the error indicator or depending on the meshfree method used.
3) The EBCs possibly its coefficient (penalty coefficient) depending on the enforcement method used.
4) The shape parameters of RPIM shape functions.
5) The parameters for the Gauss quadrature, such as the number of quadrature point.
6) The sizes of influence domains.

Note that in this problem the geometry is simple, it is therefore simple to scatter field nodes (and create the background mesh) in the entire problem domain. However, for problems with complex geometries, a pre-processor step may be necessary to generate the field nodes or the background cells. Note that the concept of influence domain detailed in sub-section 3.2.1.1 is used for the applications.

4.1.1.1 Global RPIM

To run the global RPIM computer code for this cantilever problem, the beam is modelled with a uniformly distributed set of field nodes and a background mesh is generated to perform the numerical integrations. The number of field nodes used is given by

4.12

$$N = N_x \times N_y$$

with N_x and $N_y$the number of nodes in x and y directions, respectively.

The total number of cells used for the background mesh is denoted as n_{cell} and expressed as

4.13

$$n_{cell} = n_x \times n_y$$

where n_x and n_y are the number of cells in x and y directions, respectively.

n_g represents the number of Gauss (quadrature) point for a cell and the total number of quadrature points n_{Qtotal} can be calculated as

4.14

$$n_{Qtotal} = n_g \times n_{cell} = n_g \times n_x \times n_y$$

The background mesh and field nodes are independents. The nodal arrangement is illustrated in the following figure with $175\ (25 \times 7)$ regularly distributed nodes and $40\ (10 \times 4)$ regular and rectangular cells:

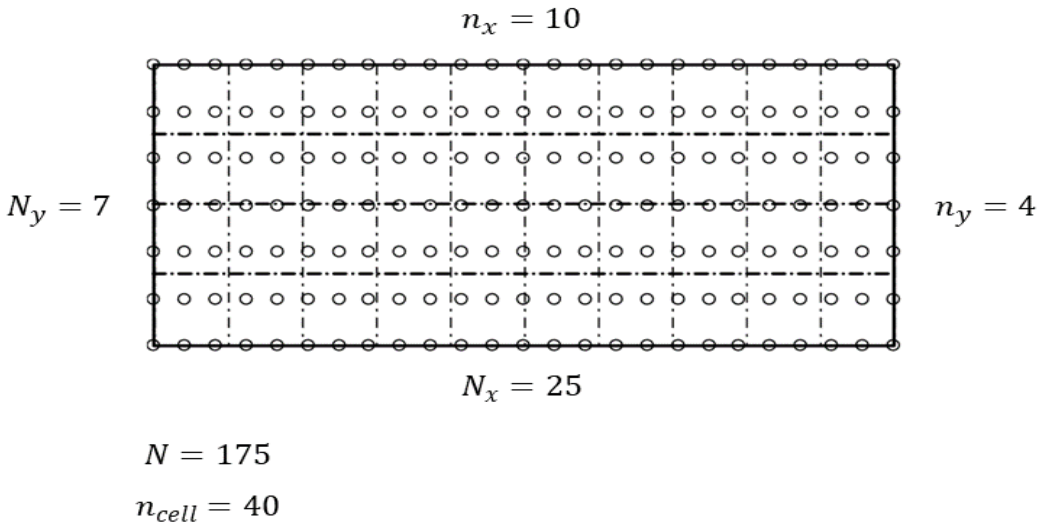

Figure 4.2 Cantilever problem representation using 175 regular nodes and 40 cells. *[3]*

4.1.1.2 Local RPIM

For the LRPIM computer code, to simulate the cantilever beam problem, the geometry is also modelled with a regular set of field nodes $N = N_x \times N_y$. Note also that no background mesh is needed to solve the problem, but one is needed in order to compute the strain energy error in equation 4.11. Furthermore, in the LRPIM algorithm, the local quadrature domains of each field nodes are divided into $n_d = n_{dx} \times n_{dy}$ sub-partitions to ensure the numerical integration accuracy. For simplification, $n_d = n_{dy} = n_{dy}$ will be used. For each sub-partition, n_g Gauss (quadrature) points are used and so its total number is $n_{Qtotal} = n_g \times N \times {n_d}^2$. The sub-local domains are illustrated in the Figure 4.3 where the local quadrature domain in divided into 4 parts. [3]

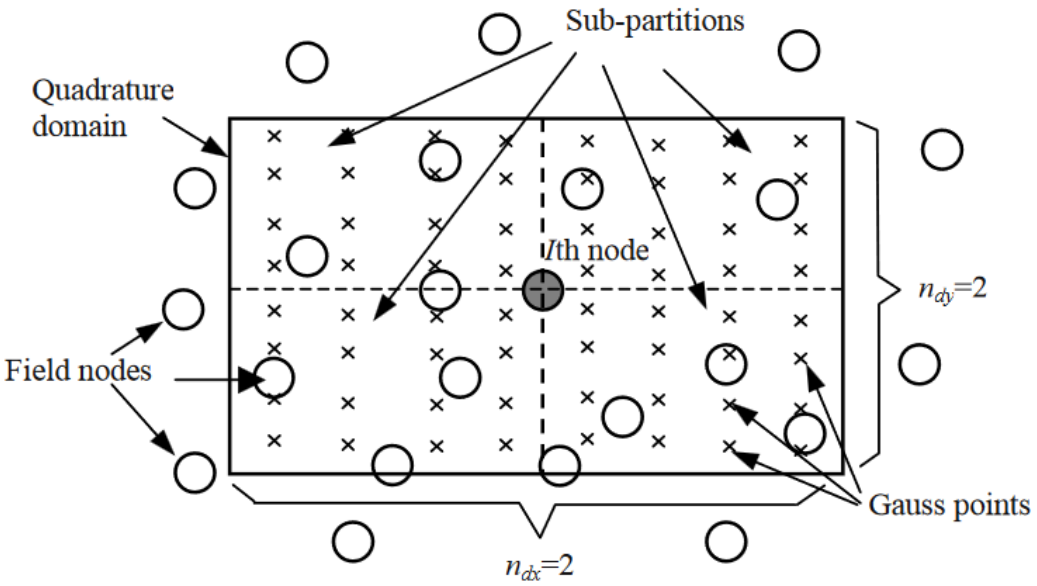

Figure 4.3 Representation of a local quadrature domain divided into 4 sub-partitions using 16 Gauss points each. *[3]*

For the LRPIM studies, the cubic spline function detailed in the sub-section 3.4.3.4 will be employed as the test function.

4.1.2 Results visualization

The rectangular cantilever problem is modelled and simulated with both global RPIM meshfree method and LRPIM meshfree method. The displacement field is obtained, and the stress, strain or energy error can be computed using the equations presented in the previous sections. All these results can easily be compared with the analytical solution since this problem and its solution equations are known. [4] All the results that will follow are plotted and studied with MATLAB and Microsoft Excel.

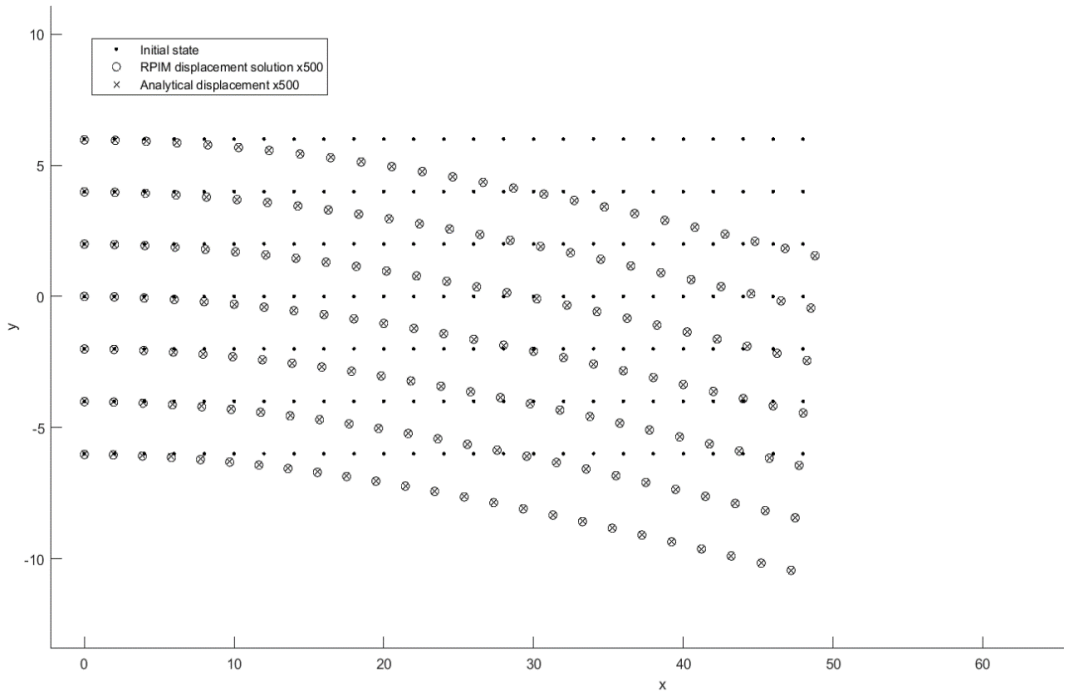

Figure 4.4 Visualization of the deflections (magnified 500 times) of the cantilever using RPIM with a set of 175 uniformly distributed nodes.

The RPIM solution is obtained for a set of 175 (25×7) regularly distributed nodes and 40 (10×4) regular and rectangular cells with 16 Gauss points for each of it. MQ-RBFs are used with $\alpha_c = 1.0$, $d_c = 2.0$ and $q = 1.03$. The dimensionless size of the support domains is chosen as $\alpha_s = 3.0$ and linear polynomials are employed. The displacement is magnified by 500 to highlight the results. [3]

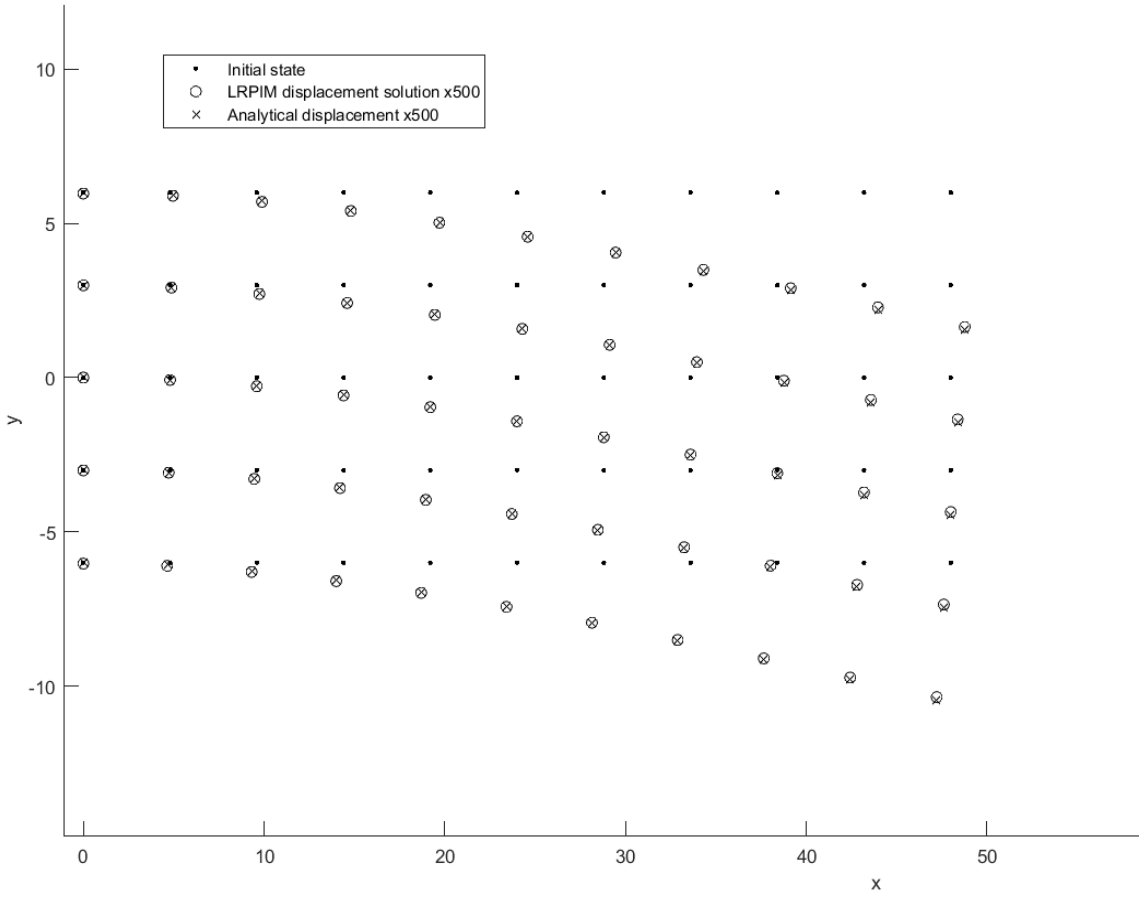

Figure 5 Visualization of the deflections (magnified 500 times) of the cantilever using LRPIM with a set of 55 uniformly distributed nodes.

The LRPIM solution is obtained for a set of $55\ (11 \times 5)$ uniformly distributed field nodes. The displacement is magnified by 500. The local quadrature domains are divided in four sub-partitions with 16 Gauss points for each of it. MQ-RBFs are used for the RPIM shape functions construction with $\alpha_c = 1.0$, $d_c = 3.0$ and $q = 1.03$. The quartic spline function W1 is employed as the test function. [3]

Another result that may be interesting to represent thanks to the data obtained with the simulations is a stress contour. It can be represented thanks to the Von Mises equivalent stress expressed for a three dimensions case as

4.14

$$\sigma_{Von\ Mises} = \frac{1}{\sqrt{2}}\sqrt{(\sigma_{xx} - \sigma_{yy})^2 + (\sigma_{yy} - \sigma_{zz})^2 + (\sigma_{zz} - \sigma_{xx})^2 + 6({\tau_{xy}}^2 + {\tau_{xz}}^2 + {\tau_{zy}}^2)}$$

with $\boldsymbol{\sigma}$ the tensor of stress.[5] [6]

4.15

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \sigma_{yy} & \tau_{zy} \\ \tau_{xz} & \tau_{zy} & \sigma_{zz} \end{bmatrix}$$

However, since the cantilever problem is considered as plane stress, $\boldsymbol{\sigma}$ is simplified as

4.16

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \sigma_{yy} & \tau_{zy} \\ \tau_{xz} & \tau_{zy} & \sigma_{zz} \end{bmatrix} \xrightarrow{\textbf{plane stress}} \boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & \tau_{xy} \\ \tau_{xy} & \sigma_{yy} \end{bmatrix}$$

and the Von mises stress equivalent may be simplified as follow.

4.17

$$\sigma_{Von\ Mises} = \frac{1}{\sqrt{2}}\sqrt{(\sigma_{xx} - \sigma_{yy})^2 + {\sigma_{yy}}^2 + {\sigma_{xx}}^2 + 6{\tau_{xy}}^2}$$

The Von Mises stress equivalent contour has been plotted from the RPIM solution obtained with a set of $175\ (25 \times 7)$ regularly distributed nodes and $40\ (10 \times 4)$ regular and rectangular cells with 16 Gauss points for each of it. And, with the following parameters $\alpha_c = 1.0$, $d_c = 2.0$, $q = 1.03$ and $\alpha_s = 3.0$.

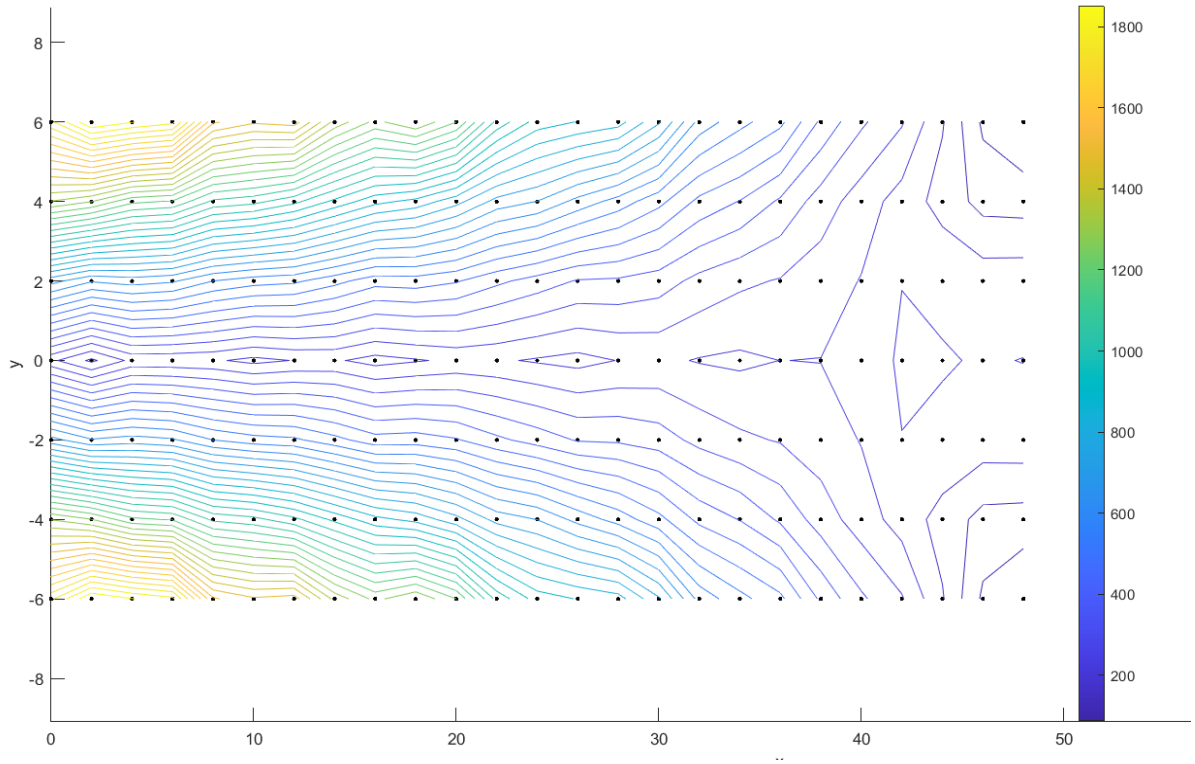

Figure 4.6 Contour of the Von Mises stress equivalent obtained with the global RPIM for a set of 175 uniformly distributed nodes.

Lastly, the different stresses (or shear stress) distributions σ_{xx}, σ_{yy} and τ_{xy} may be represented along the cross section of the beam for example. Indeed, since the analytical expressions of the different stresses are known for this problem, some comparisons between the analytical and the numerical solutions can provide information about the meshfree method accuracy (as the energy error indicator).

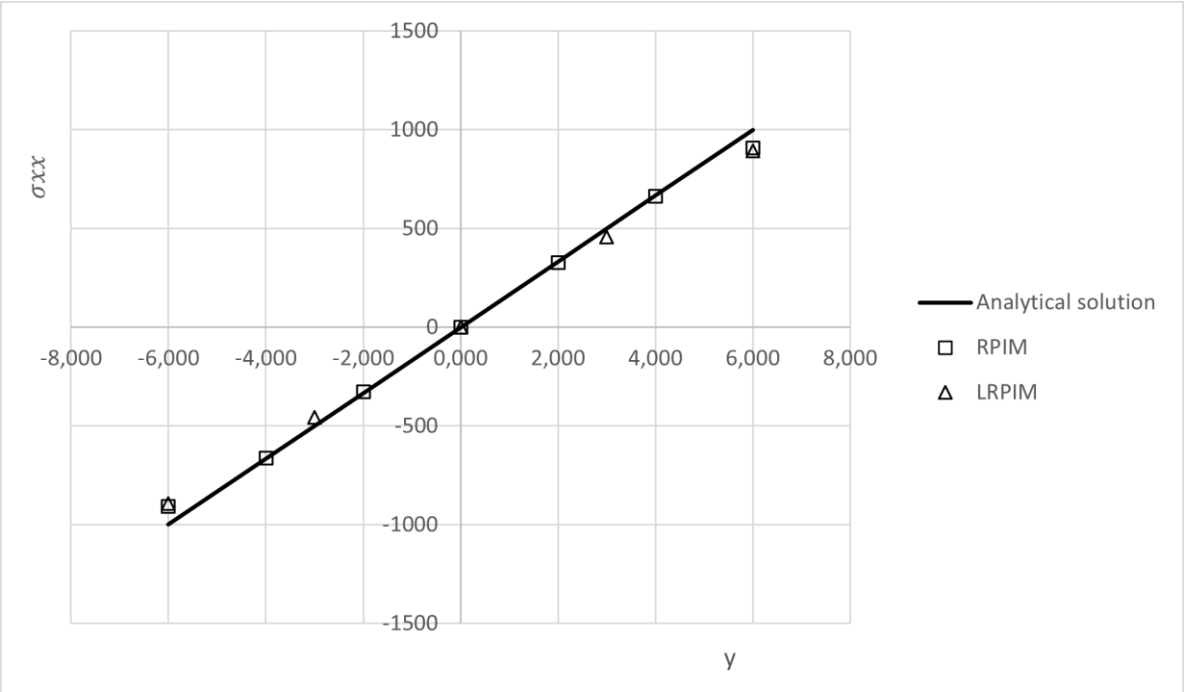

Figure 4.7 Normal stress distributions along the cross section of the cantilever obtained from the analytical solution, global RPIM and LRPIM.

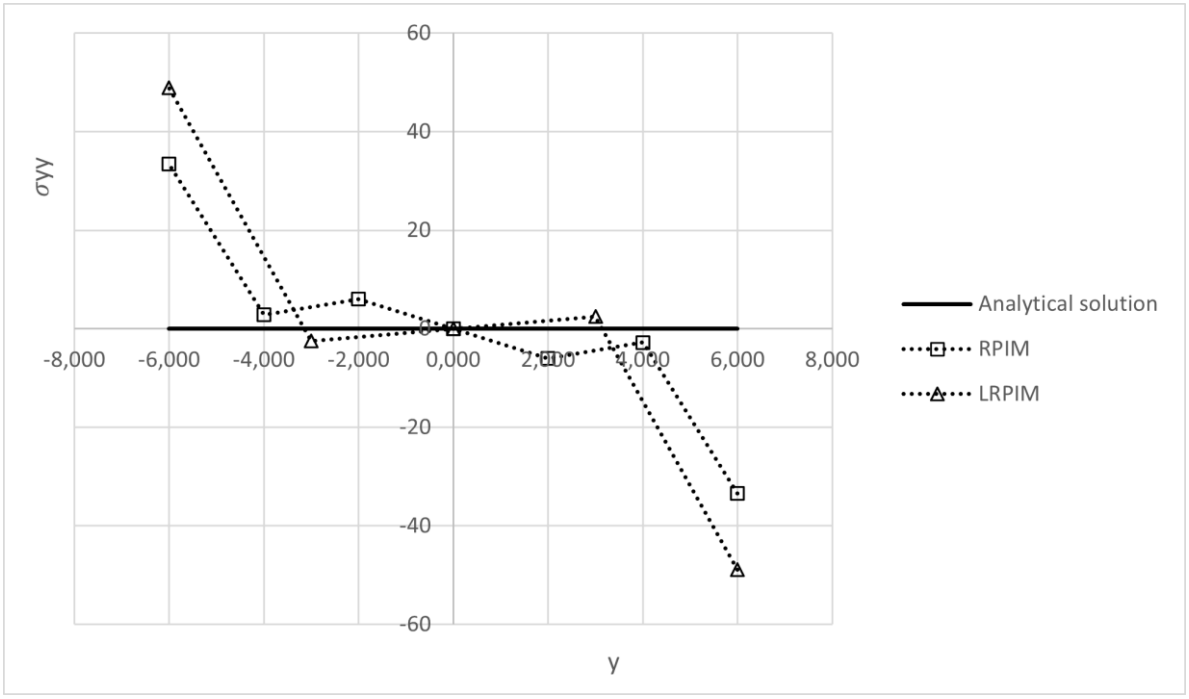

Figure 4.8 σ_{yy} stress distributions along the cross section of the cantilever obtained from the analytical solution, global RPIM and LRPIM.

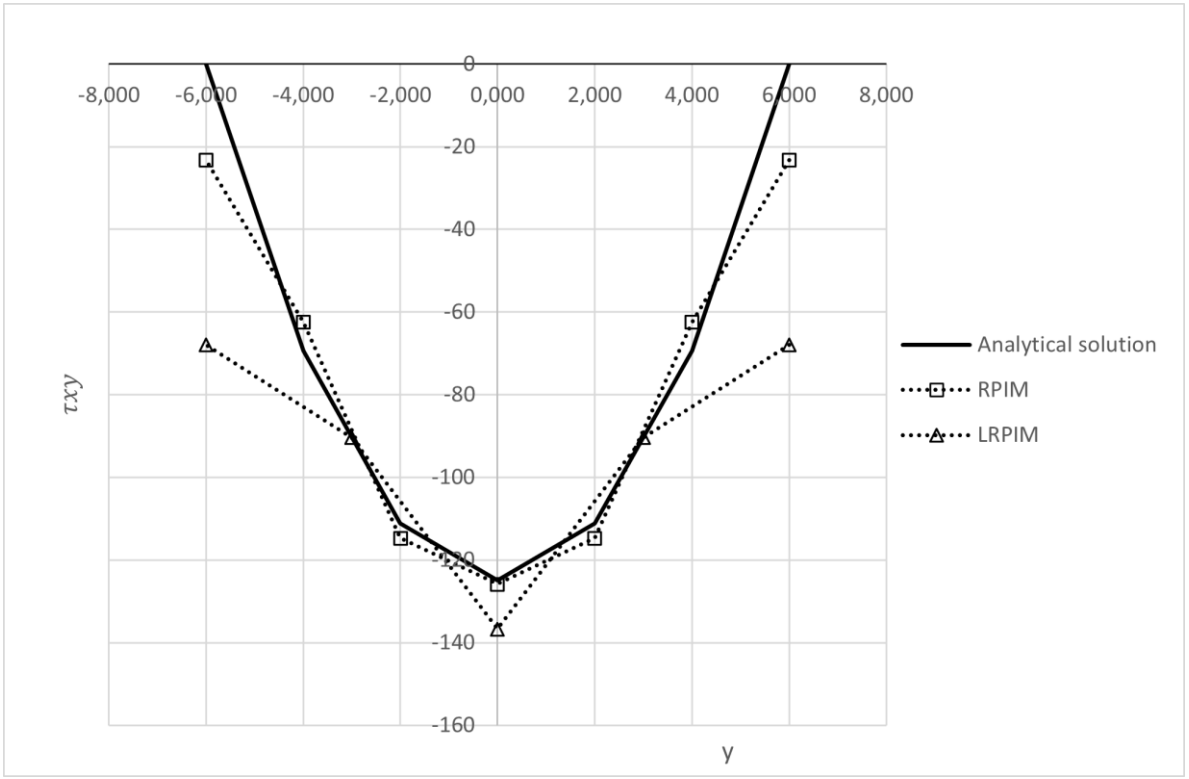

Figure 4.9 τ_{xy} shear stress distribution along the cross section of the cantilever obtained from the analytical solution, global RPIM and LRPIM

4.1.3 Studies of the parameters

4.1.3.1 Geometry and Gauss quadrature

In this first study, the analysis will focus on the modelled geometry of the cantilever problem. In other words, on the number of field nodes employed, the background mesh or the Gauss quadrature parameters.

As a reminder, the global RPIM is performed using a set of regularly distributed field nodes and a background mesh for the integrations.

Firstly, to study the effect of the background mesh and the Gauss quadrature, the energy error computed with the global RPIM algorithm have been compared for different configurations of number of cells n_{cell} and the number of Gauss (quadrature) points n_g. The total number of field nodes N stay constant with $175\ (25 \times 7)$ nodes in order to highlight sufficient values for n_{cell} and n_g.

Table 4.2 Energy errors and ratios for different background meshes and number of gauss points.

n_g \ n_{cell}	**4×1**	**8×2**	**10×4**	**12×4**
2×2	$e = \infty$ $\frac{n_{Qtotal}}{N} = 0.09$	$e = 6.308$ $\frac{n_{Qtotal}}{N} = 0.37$	$e = 22.92$ $\frac{n_{Qtotal}}{N} = 0.91$	$e = 0.058$ $\frac{n_{Qtotal}}{N} = 1.10$
3×3	$e = \infty$ $\frac{n_{Qtotal}}{N} = 0.21$	$e = 1.021$ $\frac{n_{Qtotal}}{N} = 0.82$	$e = 0.095$ $\frac{n_{Qtotal}}{N} = 2.06$	$e = 0.631$ $\frac{n_{Qtotal}}{N} = 2.47$
4×4	$e = 24.16$ $\frac{n_{Qtotal}}{N} = 0.37$	$e = \infty$ $\frac{n_{Qtotal}}{N} = 1.46$	$e = 0.091$ $\frac{n_{Qtotal}}{N} = 3.65$	$e = 0.634$ $\frac{n_{Qtotal}}{N} = 4.39$

The previous table highlights the empirical requirements found by the research groups mentioned in sub-section 3.3.3.2. The entries of the table with red background show that the total number of quadrature points n_{Qtotal} should be at least greater that $\frac{2}{3}$ in order to lead to a significant solution. Moreover, the entries with green background show sufficient values of n_g and n_{cell} to give meaningful results and an error lower than 1%.

Secondly, the field node density (uniformly distributed) has been studied. The number of background cells and quadrature points are constants are chosen purposely in order to obtain a ratio between n_{Qtotal} and N close enough compare the experiments. To study the different sets of field nodes, the error energy and the CPU time are used. The results are presented in the Table 4.3.

Table 4.3 Energy error, CPU time (s) and ratio for different set of field nodes and Gauss quadrature points.

	Field nodes $N = N_x \times N_y$			
	$49(7 \times 7)$	$91(13 \times 7)$	$175(25 \times 7)$	$325(25 \times 13)$
Energy error e	1.419	1.051	0.095	0.112
CPU time (s)	0.953	0.703	0.73	1.55
Quadrature points in one cell n_g	2×2	3×3	4×4	4×4
Background cells n_{cell}	4×10	4×10	4×10	4×10
n_{Qtotal}/N	3.90	3.96	3.65	1.96

Thus, the previous Tables (4.2 and 4.3) highlight the fact that the number of field nodes should not be chosen too small or too high but wisely by respecting some ration between the total number of quadrature points and the total number of field nodes to lead to accurate solutions without a too long computational time. For example, a problem modelled with a large set of field nodes but also a wide backgrounds mesh may lead to inaccurate results compare with a set of field nodes and background cells wisely sectioned. Finally, there are no proper law to choose either the number of field nodes or background cells, but some sufficient requirements may be founded empirically.

4.1.3.2 EXP-RBF shape parameter

This second study will investigate the effect of the shape parameters of the Gaussian exponential radial basis function that is used in the RPIM shape functions constructions. Since both meshfree methods LRPIM and RPIM use the same meshfree shape function, the study can rely on both.

LRPIM algorithm is used to investigate the influence of α_c with the energy error as an indicator. The error is plotted for different values of α_c with the following parameters for the meshfree method:

1) A set of 55 (11×5) uniformly distributed field nodes.
2) EXP-RBF for the RPIM shape function construction.
3) 40 (10×4) background cells to perform the integrations for the energy error.
4) Four sub-partitions for each local quadrature domain.
5) 16 Gauss points for each sub-partition.
6) $\alpha_i = 3.0$ is used for the construction of the influence domain.

7) A constant nodal spacing of $d_c = \frac{48}{10} = 4.8$.

The experience has shown some good accuracy solving the cantilever problem with LRPIM and the previous parameters for α_c within the range of $0.008 \sim 0.1$. For $\alpha_c = 0.009$ and error of $e = 0{,}0119$ and so this value for the shape parameter of EXP-RBF may be retained for latter studies.

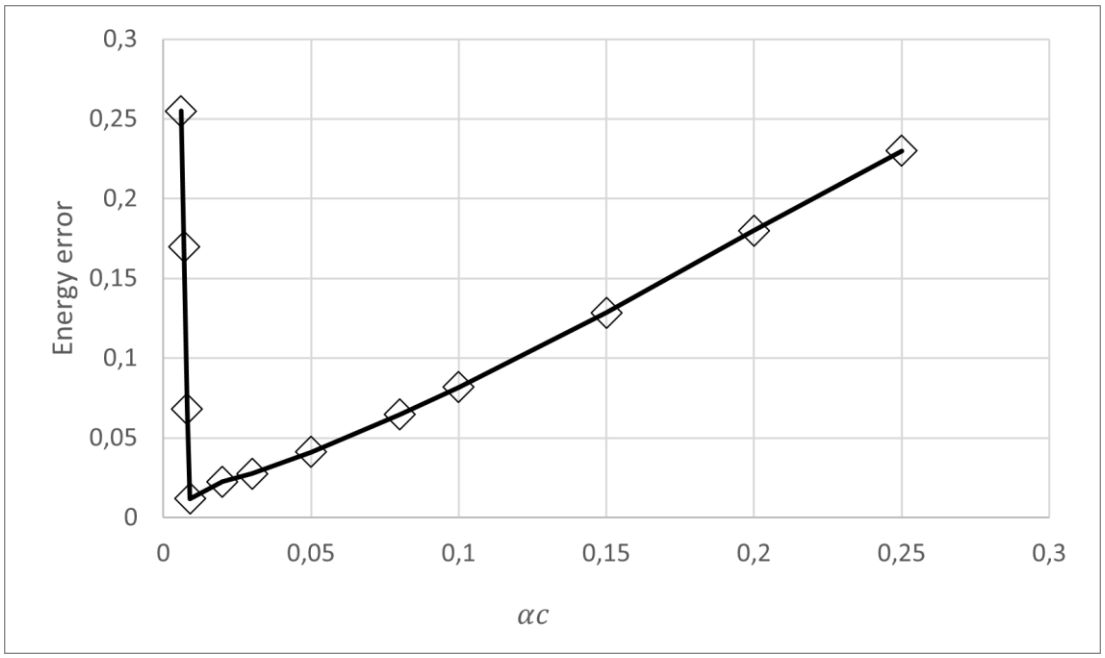

Figure 4.10 Effect of the EXP-RBF shape parameter on the energy error indicator calculated with global RPIM.

For further investigations, the RPIM meshfree method is performed on the cantilever problem with the following parameters based on the previous results and sufficient requirements:

1) A set of 175 (25×7) uniformly distributed field nodes.
2) EXP-RBF for the RPIM shape function construction.
3) 40 (10×4) background cells to perform the integrations with 16 gauss points for each of it.
4) A nodal spacing of $d_c = \frac{48}{24} = 2.0$. Note that since the field nodes are uniformly distributed, the nodal spacing d_c is a constant and can be found easily.
5) Size of influence domain $\alpha_i = 3.0$

The energy error is also used as an indicator to investigate the influence of the shape parameter α_c. The error is computed for different values of the EXP-RBF shape parameter as showed in the Figure 4.10.

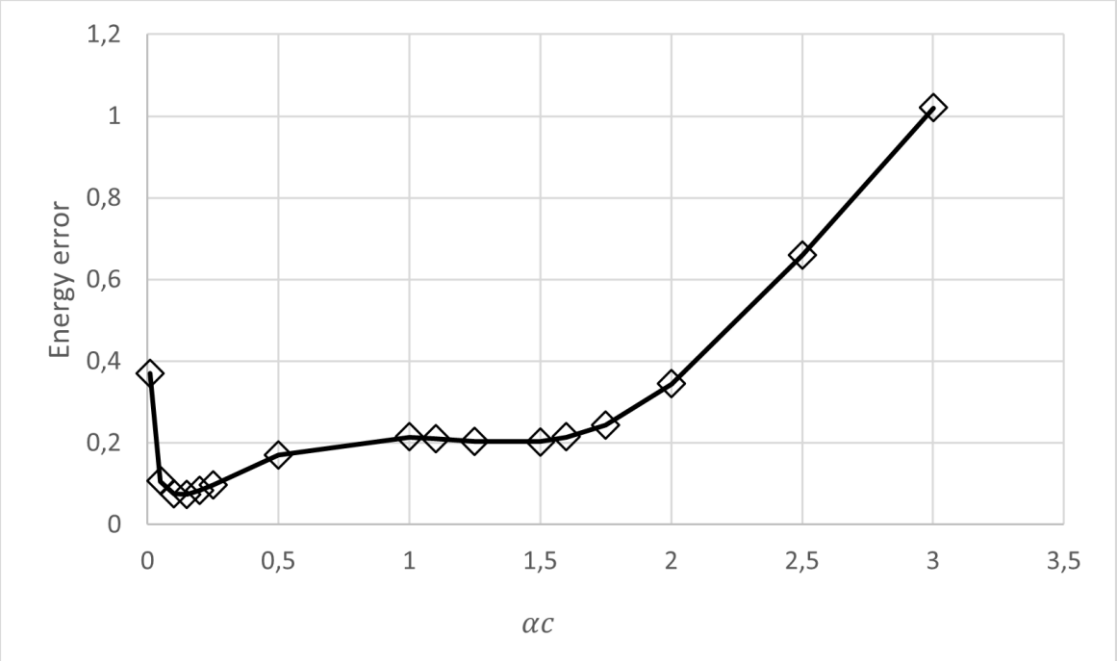

Figure 4.11 Effect of the EXP-RBF shape parameter on the energy error indicator calculated with LRPIM.

The results of this investigations show that good results are obtained for α_c within the range $0.1{\sim}0.25$. More, an $\alpha_c = 0.15$ has led to an error of 0.0743 which is very accurate.

4.1.3.3 Size of influence domains

The concept of influence for meshfree shape functions construction has been discussed in the section 3.2.1.1 and its expression is given in equation 3.3. In this work, only regularly distributed set of field nodes are uses so the nodal spacing in x and y direction are constants. Hence, the size on the influence domain is controlled by α_{ix} and α_{iy}, its dimensionless sizes. To simplify the investigations on the influence domain size, $\alpha_{ix} = \alpha_{iy} = \alpha_i$ will be used in this study.

Thus, various value of α_i are compared with the energy error in this study. The LRPIM meshfree method is used with the following parameters:

1) A set of 55 (11×5) uniformly distributed field nodes.
2) EXP-RBF for the RPIM shape function construction with $\alpha_c = 0.009$.
3) 40 (10×4) background cells to perform the integrations for the energy error.
4) Four sub-partitions for each local quadrature domain.
5) 16 Gauss points for each sub-partition.
6) A constant nodal spacing of $d_c = \frac{48}{10} = 4.8$.

In the following figures, the influence of the size of influence (support) domain on the solution accuracy and CPU time is studied.

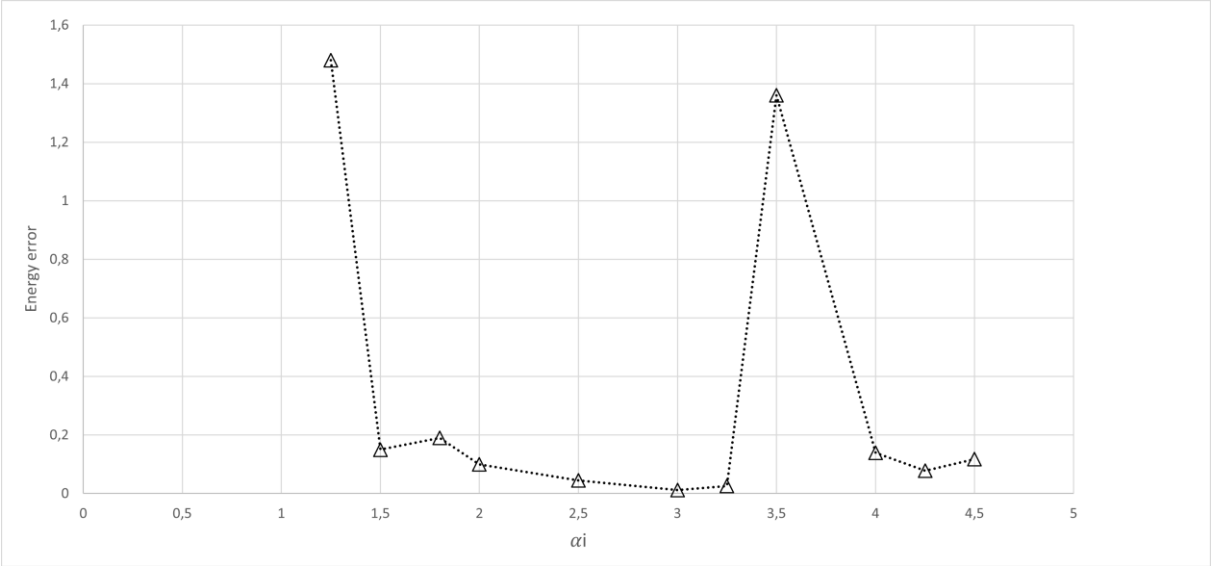

Figure 4.12 Effect of the size of influence domains on the energy error using LRPIM.

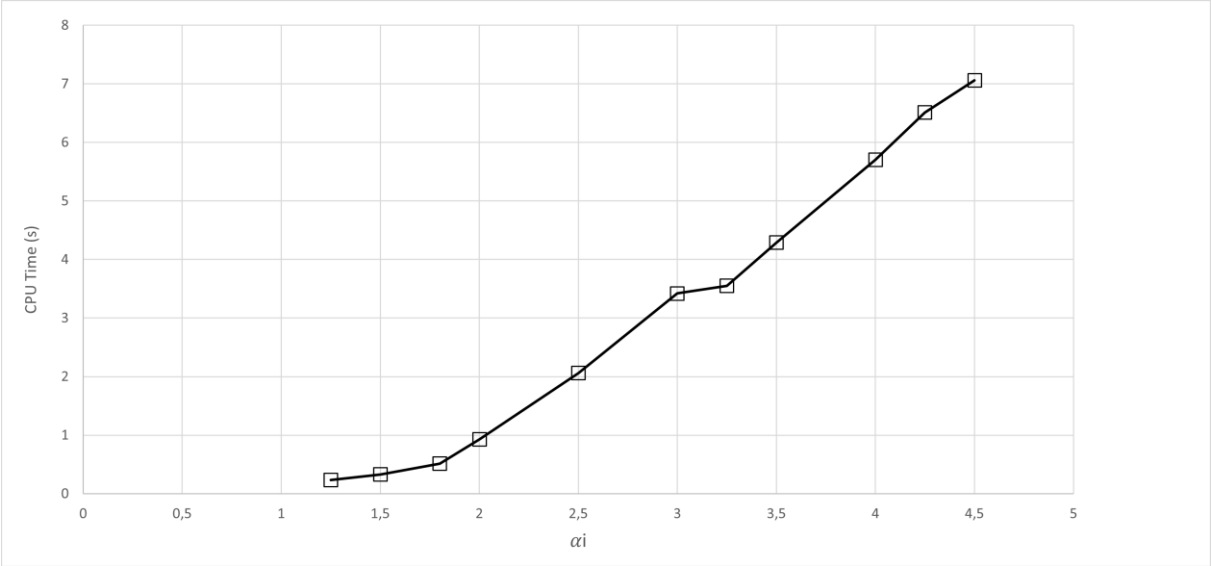

Figure 4.13 Effect of the size of influence domains on CPU time using LRPIM.

This study shows that the influence domain has an influence on the solution accuracy and should be chosen empirically, but furthermore, it also shows a great influence on the CPU time. The figures show that good results can be obtained for $1.5 \leq \alpha_i \leq 3.25$ and sufficient results for $\alpha_i \geq 4.0$. Nevertheless, increasing the size of the influence domains has a great influence on the computing time. Indeed, the larger the domain and the greater number of field nodes will be included in the interpolation and so the

computational time will increase. Also note that for $\alpha_i \leq 1.0$ the computer code fails because when the influence domain is too small, the number of field nodes used in the interpolation is not large enough.

Values within the range of $2.0 \leq \alpha_i \leq 3.25$ lead to good results and compromise for the CPU time. In particular, $\alpha_i = 3.0$ can be used.

4.1.3.4 Local quadrature domains (LRPIM)

In LRPIM procedure, the quadrature domains are constructed locally for each node thanks to a loop over all the nodes. Its sizes are defined in equation 3.103. The following study will focus on the effect of the local quadrature domain sizes with the following LRPIM parameters (according to the last studies):

1) A set of 55 (11×5) uniformly distributed field nodes.
2) EXP-RBF for the RPIM shape function construction with $\alpha_c = 0.009$.
3) 40 (10×4) background cells to perform the integrations for the energy error.
4) Four sub-partitions for each local quadrature domain.
5) 16 Gauss points for each sub-partition.
6) A constant nodal spacing of $d_c = \frac{48}{10} = 4.8$.
7) An influence domain size controlled by $\alpha_i = 3.0$

Like for the influence domain size, the local quadrature domain size is managed by α_{qx} and α_{qy} its dimensionless size in x and y directions. For simplification, $\alpha_q = \alpha_{qx} = \alpha_{qy}$ will be considered. The results are presented on figure 4.14 and highlight the influence of α_q on the accuracy. An acceptable range of its value is $1.0 \sim 3.0$. When the local quadrature domain is large enough, good results can be obtained but if this last is chosen too large, the Gauss quadrature necessary to compute the error will be less accurate.

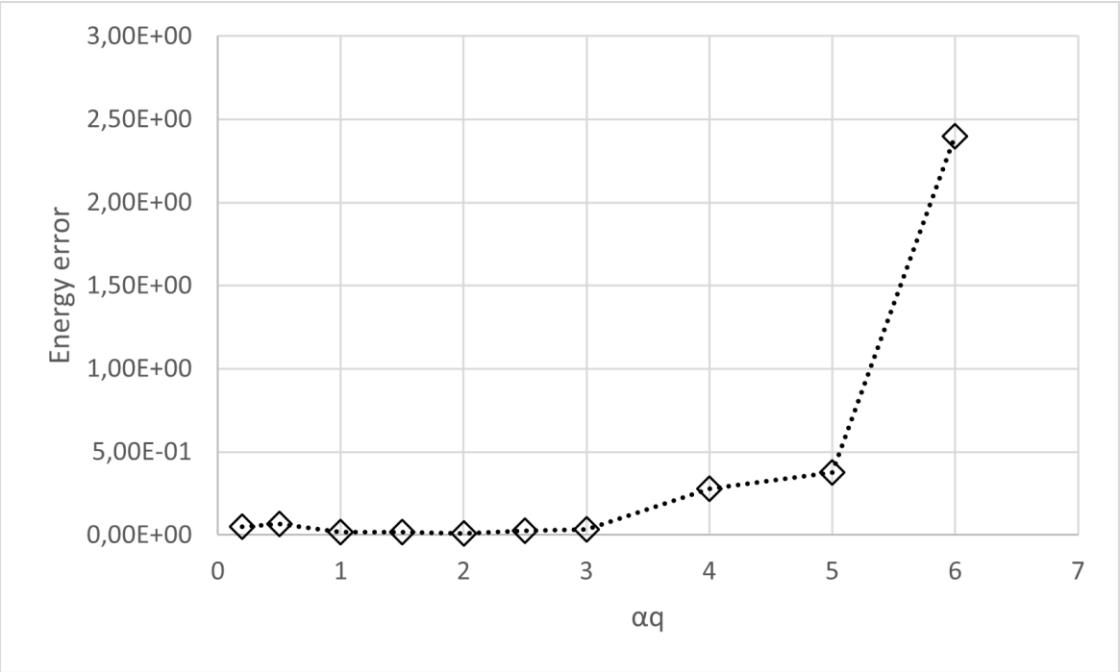

Figure 4.14 Effect of the size of local quadrature domains on energy error using LRPIM.

As mentioned in the prerequisites, the LRPIM is based on a local quadrature domain which is itself divided into sub-partitions. The number of these sub-partitions is controlled by $n_d = n_{dx} \times n_{dy}$ and could be changed to study the effect of its value. In this next study, same parameters as in the previous study are used and with $\alpha_q = 2.0$. The error is obtained for local domains divisions as follow:

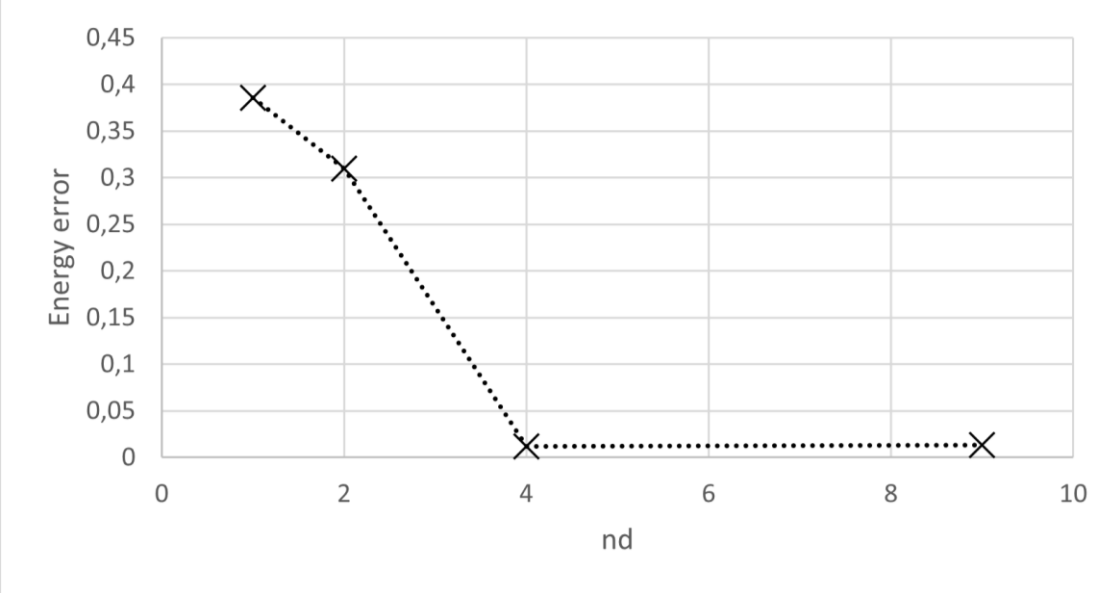

Figure 4.15 Effect of the number of sub-partitions on CPU time using LRPIM.

The previous figure shows that the energy error decrease when the number of sub-partitions increase. In fact, augmenting the number of sub-partitions improves the integral evaluation accuracy because more quadrature point will be computed. However, adding a sub-partition impact the calculation time. If one partition is added, there will be $16 \times N$ (the number of field nodes) more quadrature points and so impact the computational cost.

4.1.4 Convergence and conclusions

4.1.4.1 Global RPIM convergence

As a final study, the convergence of a meshfree method can be investigate. The global RPIM convergence is studied with four different models created with different uniform distribution of nodes. Then, the logarithmic error and nodal spacing in x direction are plotted. The different model are sets of 31×7, 25×7, 17×7 and 13×7 field nodes with the same parameters. The nodal displacement in x direction is noted h and on the plot, the error seems to decrease linearly with the nodal spacing. The rate convergence is estimated from the trend curve and is lower than the theoretical value of 1.0.

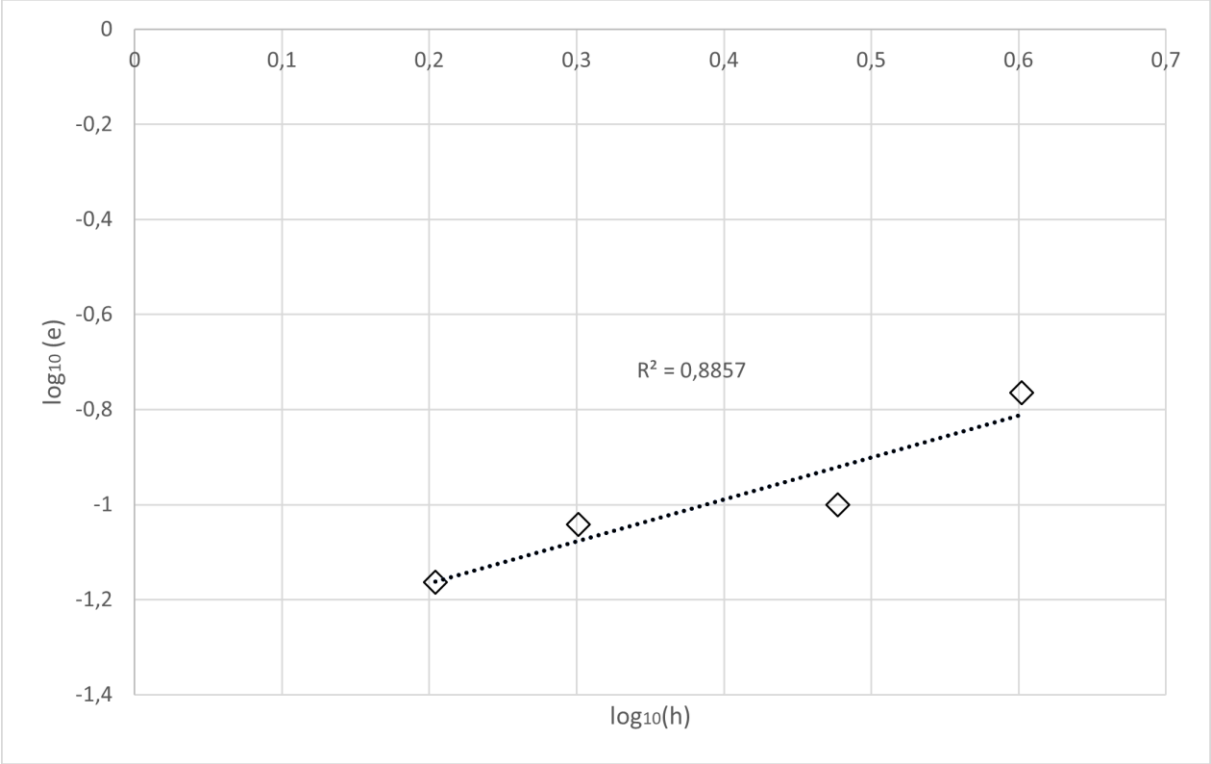

Figure 4.16 Study of the convergence for the global RPIM meshfree method.

4.1.4.1 Conclusions

This section went through the main parameter that characterize a meshfree method. It has been highlighted that many parameters can affect the meshfree solution and efficiency. Thus, the geometry of the problem needs to be correctly represented by relevant sets of field nodes and, if it is necessary, background cells. The Gauss quadrature should be also considered because it has a direct influence on the

precision for the evaluation of the integrals of the problem. The parameters used for the meshfree shape function construction can be modified and must be determined empirically by investigations. Finally, the size of the different domains (influence or quadrature) directly impacts the accuracy, but rather the computational time. Indeed, it affects the number of field nodes used for interpolation or the total number of quadrature points.

A relevant meshfree method model and its parameters is determined by considering a good accuracy for low computational costs. This last may concern the calculation time but also the pre- and post-processing. For example, in an meshfree method, a pre-processor creating non-uniformly distributed set of field nodes may be compared to the element's creation in the FEM.

It should be mentioned that in this study, only uniformly distributed sets of nodes have been studied for simplicity. Nevertheless, it is also a parameter that influences the accuracy. Irregularly distributed nodes can lead to better results but also requires modification in the algorithms, such as the average nodal spacing calculation or the use of a pre-processor to create it. Another point that could be studied is the EBCs enforcement method. Indeed, depending on the method (direct method, penalty method) the computational time and solution accuracy will vary. [1]

5. Conclusion

This master thesis went through meshfree methods, and especially the global RPIM and the LRPIM from the meshfree and shape function procedure, to the analysis of a numerical example. Furthermore, the procedures of these last two methods have been established, discussed, and related to a computer code approach. Unfortunately, the transposition of the 2D code to be efficient in 3D (mentioned in the introduction) could not be completed. Nevertheless, this failure has also been the opportunity to gain experience with algorithms coding for numerical methods and make aware of the issues of such a process.

The global RPIM and LRPIM have been applied for simple mechanics problems in order to stay focused on the meshfree study perspective. By means of the cantilever beam deflection example, the mains parameters that control the meshfree methods and affects its accuracy have been highlighted. In addition, the state of the art and the studies of global RPIM and LRPIM allowed to lead to the following conclusions:

- The global RPIM allows to overtake the singularity problem of the PIM. More, this meshfree method is robust, stable and lead to accurate results even for arbitrarily distributed field nodes. However, this is not truly a meshfree method because this last is based on a global weak form and so it requires a background mesh to evaluate the integrations.

- The LRPIM allows to get rid of the background mesh and its implementations are as simple as that in the FEM. However, its computational cost increase because of the of the asymmetry of the stiffness matrix. Another drawback is the use of additional parameters to control the local quadrature domain and test functions, and so more parameters need to be taken into account to obtain sufficient solutions

- Many parameters must be considered in meshfree methods. Here are the mains influencing parameters: the problem representation with proper set of field nodes, the quadrature parameters, the shape parameters, the sizes of influence, the local quadrature domain (if it is needed) and the background mesh (if it is needed).

Thus, a numerical method combined with proper parameters can lead to relevant and accurate solutions for good computational times.

6. Bibliography

[1] G. R. Liu, *Meshfree Methods: Moving Beyond the Finite Element Method, Second Edition*. CRC Press, 2009.

[2] G. R. Liu et S. S. Quek, *Finite Element Method: A Practical Course*, 1er édition. Butterworth-Heinemann, 2003.

[3] G. R. Liu et Y. T. Gu, *An Introduction to Meshfree Methods and Their Programming*. Springer Science & Business Media, 2005.

[4] S. Timoshenko, *Theory of Elasticity [by] S.P. Timoshenko [and] J.N. Goodier*. McGraw-Hill [1969, ©1970], 1977.

[5] S. M. A. Kazimi, *Solid Mechanics*. Tata McGraw-Hill Education, 2001.

[6] H. Ford et S. H. Ford, *Advanced Mechanics of Materials*. Wiley, 1963.