Politecnico di Milano

School of Industrial and Information Engineering

Master of Science in Space Engineering

Dipartimento di Scienze e Tecnologie Aerospaziali

Implementation of Mesh Free Methods for Linear Elasticity Problems

Advisor: Prof. Riccardo Vescovini

Master Thesis by:

Serena Saitta
ID: 883498

Academic Year 2020-2021

Abstract

For decades, modeling and simulation of engineering systems have relied on the use of the Finite Element Method which is a robust, well-established technique used to solve numerical problems. Being the currently most used method, the drawbacks of the FEM are well known, especially in terms of mesh generation. The process is, in fact, extremely time consuming, expensive and cumbersome when the geometry considered is of complex shape. The source of the problem is the use of the *elements* and the concept of *mesh* in general. That is the reason why, in recent years, more and more attention has been given to the so-called Meshless or Mesh Free methods. These new techniques have been proposed to perform the same task as the Finite Element Method but without using any element or mesh. This work aims to investigate these methods as a potential alternative to FEM analysis in the field of structural mechanics. A computer program has been developed based on these methods and applied to some test cases such as bending, free vibration and buckling analysis of plates. The accuracy of the results is then compared to the analytical solution, where available, and to the FEM solution to assess the quality of the results.

Sommario

Per decenni, la modellazione e simulazione di sistemi ingegneristici ha fatto affidamento sull'utilizzo del Metodo ad Elementi Finiti (FEM), uno strumento robusto e largamente diffuso usato per risolvere problemi numerici. Essendo il metodo attualmente più utilizzato, gli svantaggi del FEM sono ben conosciuti, specialmente in termini di generazione della mesh. Infatti, questo processo richiede una grande quantità di tempo, è costoso e non intuitivo quando le geometrie trattate hanno forme complesse. La base del problema risiede nell'utilizzo degli *elementi* e nel concetto di *mesh* in generale. Per questo motivo, di recente, sempre più attenzione è stata dedicata ai così detti Metodi Mesh Free. Queste nuove tecniche si propongono di svolgere le stesse operazioni del FEM ma senza l'uso di elementi o mesh. L'obbiettivo di questo lavoro è di appurare se questi metodi abbiano il potenziale di essere una valida alternativa all'analisi FEM nel campo della meccanica strutturale. A tal fine, un nuovo algoritmo è stato scritto sulla base di queste tecniche e applicato ad alcuni casi test tra cui la flessione, le vibrazioni libere e l'analisi ad instabilità di vari tipi di piastre. La precisione dei risultati è stata confrontata con la soluzione analitica, quando disponibile, e alle soluzioni ottenute con l'analisi FEM così da appurarne la qualità dei risultati ottenuti.

Acknowledgements

I would like to thank my advisor, Professor Riccardo Vescovini, for his constant support and for always being available to help me, even in the middle of a pandemic. Without him, I would have never developed the passion and interest I feel now for the field of structural mechanics.

Thank you to all the people I shared this journey with, Alessandro, Pietro, Federica, Andrea who stayed with me from my very first year. Thanks to Emanuele for his constant support throughout not just University but our 22 years of friendship. Thanks to Federica for all the shared weekend breakfasts. Thanks to Peppe for the long walks and beautiful pictures. Thanks to Francesco, who was always right in the end. Thanks to my dear friends Sofia, Egle and Andrea for showing me that true friendship knows no distance. To Marco for his support, his jokes and all the pizzas. To Oscar for his patience and the long chats.

Finally, I would like to say thank you to my family. To my mom, Luana, who never lets me put myself down and my dad, Giuseppe, who always knows how to cheer me up. Thank you for all your constant love, support, patience and for believing in me even when I didn't. Thank you for putting up with me throughout all these years. Thanks to my grandparents for always cheering for me and encouraging me. I would have not made it this far without all of you.

Contents

List of Figures

List of Tables

1 Introduction

Building complex, advanced engineering systems requires numerical modeling and simulation procedures to solve the partial differential equations that govern the system. The ideal numerical method to be used should allow to reduce the time for the generation of the models as well as the time for the numerical solution.
For decades, the most widely used technique has been the Finite Element Method (FEM) which was applied to a number of different fields, from structural mechanics to fluid dynamics and more. The superiority of this method against competing techniques lies in three main features [7]. A geometrically complex domain is discretised as a series of small domains of simple geometrical shapes, the Finite Elements. Over each of these elements, an approximation function, or shape function, is obtained considering that a continuous function can be represented by a linear combination of polynomials. Finally, the algebraic relations among the unknown coefficient, or nodal values, are obtained by solving the governing equations.
The pivoting aspect of FEM, the mesh, is also its major drawback. The generation of the mesh is a costly procedure in terms of time and often requires human intervention for especially complex geometries. As a result, new techniques to solve the partial differential equations governing a certain physical phenomenon started to be researched. Mesh Free Methods are a relatively new family of numerical methods to be collocated in this framework as an answer to the limitations of FEM. The different types of methods can be classified, according to Liu [4], in three main categories.
Finite integral representation methods include the smoothed particle hydrodynamics (SPH) method which is one of the earliest Mesh Free Methods implemented. It was developed by Lucy (1977) and Gingold and Monaghan (1977) and has been later applied in problems, such as astrophysics problems, by Monaghan in 1992. In the same category, the Reproducing Kernel Particle Method (RKPM) can be found, which was developed more recently by W. K. Liu et. al in 1995.
To the second category, finite series representation methods, belong some of the currently most used Mesh Free Methods. The Moving Least Squares (MLS) method was developed by Lancaster and Salkauskas in 1981 and was first used to construct Mesh Free shape functions by Nayroles et al. in 1992. The polynomial Point Interpolation Method (PIM), see Section 3.1, was designed shortly after by Liu and Gu in 1999 and was followed by the radial Point Interpolation Method (RPIM) developed by Wang and Liu in 2000 to solve some of the issues encountered in the polynomial PIM, as described in Section 3.3.
The last category, finite differential representation methods, includes the Finite Difference Method, developed by Onate et al. in 1996 [4], which makes use of regular grids, and the finite difference method with irregular grids developed by Liszka and Orkisz in 1980 and by Jensen, also in 1980.
Mesh Free Methods were initially developed to solve issues that arise with the Finite Element Methods while meshing problems in which the material can move around, like in fluid dynamics, or when large deformations are expected. In these cases, in fact, the relationship between the nodes of the mesh may not be maintained therefore introducing errors. The re-mesh approaches that have been proposed require additional computational cost and time needed to generate the new mesh.
Over the past three decades, Mesh Free methods have successfully been applied to different scientific areas from classical astronomical problems to fluid flow and heat transfer problems. In structural mechanics they are particularly appreciated in the area of fracture mechanics (Element Free Galerkin Method), but are also applied in vibration and buckling analysis. In a more general way, Mesh Free Methods are potential candidates to provide an optimization of the numerical solution of all kind of partial differential equations while being free from the problems, limitations and costs, in terms of time and man power, that the use of a mesh imposes.
The investigation of Mesh Free Methods is still an open field of research. There are, of course,

some limitations typical of the methods that have to be overcome. For instance, differently from what happens in FEM, Mesh Free shape functions, in general, do not posses the Kronecker delta function property. Therefore, the imposition of the essential boundary conditions of the problem can be not as straightforward as it is in FEM. The Point Interpolation Method and Radial Point Interpolation Method overcome this problem and possess the Kronecker delta function property. Another obstacle lies in the need for these methods of a background mesh needed to evaluate the integrals that appear from the weak formulation of the problem. Although the mesh used in the integration is different from the concept of mesh as understood in FEM, the research is still ongoing to have a truly Mesh Less method. It has been shown that it is possible to not use weak forms and have Mesh Free Methods operating on strong forms instead. An example is the finite point method (Liszka and Orkisz, 1980; Jensen, 1980; Onate et al., 1996; Cheng and Liu, G. R., 1999; Xu and Liu, G. R., 1999; Song et al., 1999), which is based on the the Taylor series representation of a function. The problem with these methods is the fact that they are not particularly stable and the more the nodal distribution is irregular the less accurate are the results. Research on the stabilization of these methods is still ongoing and moving in the direction of using radial functions.
It is clear how the development of Mesh Free Methods is far from complete and how active and dynamic the research on these methods is.
The aim of this work is to develop a computer program implementing a Mesh Free Method to perform preliminary assessments regarding the potential of the method for problems in the field of linear elasticity. Here, Mesh Free methods are applied to the field of structural mechanics. They are used to study the static deflection of beams and membranes as well as the static deflection, free vibrations and buckling analysis of both isotropic and laminated plates. The problems are studied using finite series representation methods and specifically, the Point Interpolation Method and Radial Point Interpolation Method. The results obtained from the analysis are compared with analytical solution, when available, and the FEM solution to make a comparison between the two methods.
The work is divided into three main parts. An initial introduction of Mesh Free methods is provided, which contains the explanation on their main characteristics and the difference with the FEM. It is followed by a detailed description of the PIM and RPIM, shape functions constructions, their advantages, peculiarities and limitations. In the second part, a quick review of the main structural mechanics theories, implemented in the program, is given. The equations of motion in their weak form, are discretized and the nodal and global system matrices are obtained and assembled. The third and final parts show the results obtained with the computer program and aim at verifying the accuracy of the Mesh Free Methods implemented.

2 Mesh Free Methods

The design of complex, advanced engineering systems involves governing partial differential equations (PDE) which are significantly difficult to solve. Hence, the needs of numerical methods. The Finite Element Method (FEM) is the most used technique for solving systems' governing differential equations. The idea of the method is to discretize the continuum by means of a mesh. Different nodes are introduced in the domain, each one connected to other defining a region of space called element. In other words, a mesh provides a relationship between nodes. At this point, the continuum has been modelled as an assembly of Finite Elements interconnected at nodal points.
Numerical methods allow to work with algebraic equations rather than complex PDE. This is achieved by introducing shape functions which are substituted to unknown field variables, such as displacements and strains. From the global domain we move to a fixed local domain which is that of the elements. Using a parametric formulation, a mapping from the physical domain to a reference or computational domain is operated and a new set of coordinates is introduced. At this point, employing a variational principle, as well as the discretization, it is possible to simplify the governing equations of the system assembling sets of algebraic equations for the elements.
Although FEM is a robust, thoroughly developed method, some limitations are quite evident. The creation of the mesh is an extremely time consuming operation. Refinement of a mesh for successive analysis of the same body, can take away a significant amount of time. Different kind of mesh exist, which are based on elements of different shapes which means different nodal distributions. Although not necessarily required from the Finite Element Method, it is preferred to employ a mesh built on regularly distributed nodes which makes it easier to build the elements.
Mesh Free methods are yet another tool to perform the same operation as FEM. They have the potential to overcome the aforementioned limitations of FE methods. To achieve this objective, FEM and Mesh Free methods diverge at the step of constructing the mesh and the subsequent derivation of the shape functions.
Being the diffusion of Mesh Free Methods quite recent, they are supported by a relatively limited amount of literature. Hence, it is interesting to asses whether they can effectively accomplish the task of overcoming the limitations imposed by the Finite Element Method.

2.1 Geometry discretization

As mentioned, conventional FEM implies the construction of a mesh made of nodes and elements. In most cases, linear elements and flat surfaces are used to approximate even curved areas and boundaries. The accuracy in this cases depends on the umber of elements used or, in other words, on how fine the mesh is. But finer mesh means more elements which in turns means more time spent on mesh generation. The constraint on time also poses a constraint on the number of elements used.
Mesh Free methods allow to move from PDE to algebraic governing equations but without the generation of any mesh. They only require the generation of a distribution of nodes scattered on the domain and on the boundaries. The problem is *represented*, not discretized, by the nodes. The absence of a proper mesh means that there is no a priori information on the nodal relationship required for the field variable interpolation. Moreover, these methods introduce a huge degree of flexibility in terms of adding or deleting nodes. For instance, if after generating the initial nodal distribution a higher degree of accuracy is required in a certain area, it is possible to increase the nodal density in that particular area without altering the pre-existent nodal distribution.
Having no mesh means that the mesh generating time is spared in modeling and simulation.
Although being appealing for this characteristic, it must be stated that a mesh has to be

generated in any case. The shape functions used to represent the field variables are introduced in the weak form of the governing equations expressed through an energy principle. In other words, there are integrals to be evaluated. In this work, Gauss integration is used for this purpose. This requires the generation of a so-called background mesh, not different from a regular FEM mesh. In this case though, the mesh plays no part in building the shape functions, it is only used for the evaluation of the integrals. There still is an advantage with respect to FEM. While in the latter mesh and nodes are part of the same mathematical structure in which the relationship between the nodes are predefined, in Mesh Free methods the nodes used to discretize the continuum and the background mesh used to evaluate the integrals are two independent objects. The nodal distribution of the problem doesn't have to be the same as the one used to build the background mesh. The only constraint on how fine this background mesh has to be is posed by the construction of the shape functions. It will be shown that to have an accurate representation of the problem, a certain minimum number of nodes have to be involved in the shape functions construction. The Gauss points relative to the background mesh are actively involved in this process, they have to be sufficient in number and distribution to fulfill the task.
A visualization of the concept of background Mesh is presented in Figure 1 where the two different meshes are put to comparison. In the Mesh Free case one can note how the nodes have no relation with the background grid while the opposite happens in the FEM case.

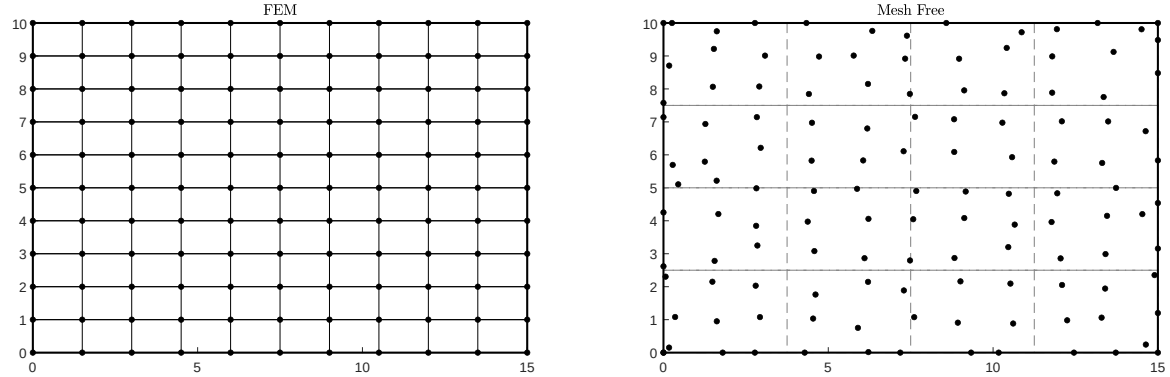


Figure 1: Comparison between FEM mesh (left) and Mesh Free nodal distribution and background mesh (right)

2.2 Shape functions overview

The other aspect in which FEM and Mesh Free Methods diverge is at the stage of shape functions construction.
FEM shape functions are predetermined before the analysis even begins. Each different kind of element is associated with a certain shape function and elements of the same type have the same shape function. It is clear that the domain in which they are developed is the fixed domain of the element they refer to and to which a natural coordinate system is associated to move from the global domain of the problem to the local one of the element.
In Mesh Free Methods the shape functions are not preexisting functions to choose from. They are constructed during the analysis. Moreover, due to the absence of elements and mesh, they are referred to a certain point of interest and change as the location of the point of interest changes. To better explain this mechanism, it is necessary to introduce the concept of support domain.

2.2.1 Support domain

The absence of a mesh requires that the shape functions used to represent a field variable, which can be, for instance, a component of the displacement field, have to be constructed in a different way. The displacement at a certain point $\mathbf{x} = (x, y, z)$ is interpolated using the displacements of the nodes that fall within a certain small local domain called *support domain* of point $\mathbf{x}$.

$$u(\mathbf{x}) = \sum_{i=1}^{n} \Phi_i(\mathbf{x}) u_i = \mathbf{\Phi}(\mathbf{x}) \mathbf{U}_s \tag{1}$$

where n is the number of nodes included in the support domain of $\mathbf{x}$, u_i is the field variable of the i-th node in the support domain, while $\mathbf{U}_s$ is the vector collecting all the field variables at the nodes. Finally, $\Phi_i(\mathbf{x})$ is the the shape function of the i-th node. The shape function is therefore built using all the nodes falling within the support domain of $\mathbf{x}$.
A support domain is an area of a certain shape and extension, within the global domain of the problem, which is centered in a point of interest and that encloses a certain number of nodes. The shape functions are built using information only from the nodes falling inside the support domain. It can have different shapes and dimensions and different points of interest can have support domains having different characteristics. To clarify this concept, Figure 2 shows two examples of different support domains. In this case, the global domain has elliptical shape and is represented by a set of arbitrarily scattered nodes. The two support domains represented have circular and rectangular shapes and are centered in a randomly chosen node of interest. The nodes falling within the border of the support domain are the ones used to construct the shape function for the specific node in which the domain is centered.

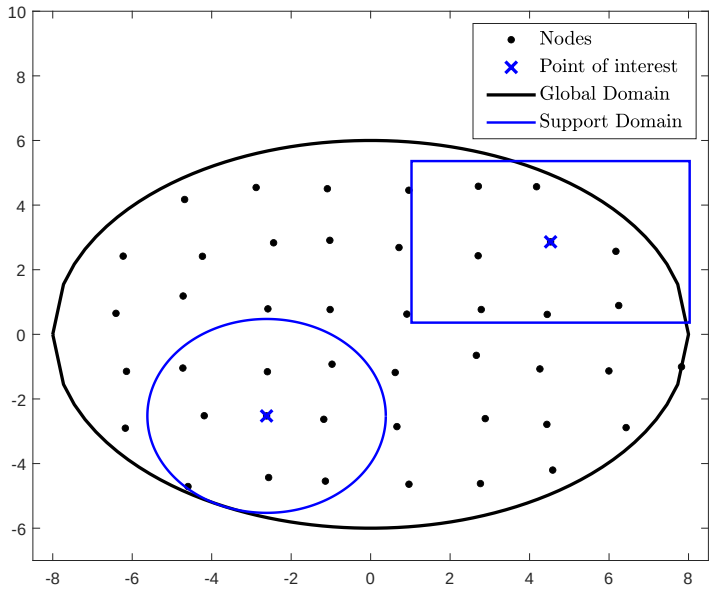


Figure 2: Examples of support domains of different shapes and dimension

In FEM, the elements define a domain which is fixed and remains equal to itself. In Mesh Free Methods the support domain can be imagined as a certain geometry which moves from point of interest to point of interest. The most used shapes are circular and rectangular. For instance, considering a circular support domain, the procedure to build the shape functions will be the following. The circular domain will be first centered in the i-th point of interest. The nodes falling inside it will be identified and the shape function will be created using only those nodes. When all the operations to be performed are completed, the support domain will be moved so that its center will coincide with the i-th+1 point. The operations described will be repeated

and the procedure will be applied to all the nodes in the global domain. An example is shown in Figure 3. As the support domain is moved from one node to the next, it can be clearly seen how the nodes falling inside it are different in the two cases.

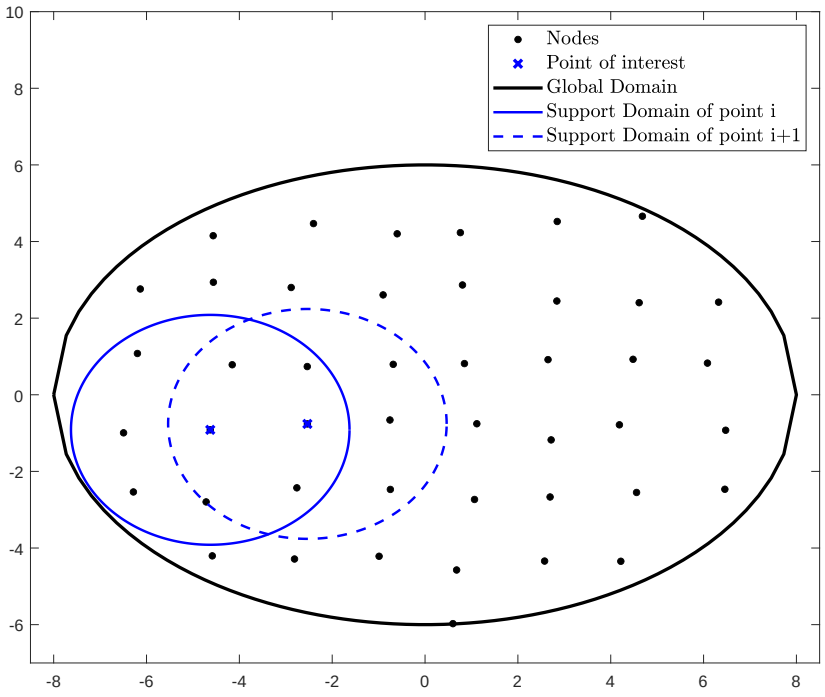


Figure 3: Example of two successive support domains of circular shape. The nodes falling inside the local domain of point i are not the same as the ones falling inside the domain of node i+1. As a consequence, also the two shape functions will be different.

Since the shape function depends directly on the number of nodes involved in its construction, the concept of support domain makes sense if the nodal distribution doesn't vary too much in the global domain. Whenever this is not the case and the nodal density is significantly different in the domain, it may occur that unbalanced selections of nodes are used for the shape function construction. In this work, the nodal representation of the problem involves nodes either equally spaced or randomly distributed, but still homogeneously spread.
From the considerations above, it follows that the number of nodes involved in the construction of the shape function influence the accuracy of the interpolation. Hence, the dimension of the support domain must be such that an adequate number of nodes is included. The dimension d_s of the support domain of a point can be determined as:

$$d_s = \alpha_s d_c \tag{2}$$

where, d_c is a characteristic distance related to the nodal spacing. If the nodes are all evenly spaced, then d_s is just the space between two consecutive nodes. However, this situation may not always be feasible. In this cases d_c is an average nodal spacing which can be estimated with formulas available in the literature [4]. For 1D cases

$$d_c = \frac{D_s}{(n_{D_s} - 1)} \tag{3}$$

where D_s is an estimation of the support domain dimension which doesn't have to be very accurate but at least reasonable, and n_{D_s} is the number of nodes falling inside a known support domain D_s.

For a 2D case:

$$d_c = \frac{\sqrt{A_s}}{\sqrt{n_{A_s}} - 1} \tag{4}$$

where A_s is an estimation of the area covered by the support domain d_s and n_{A_s} is the number of nodes falling inside the estimated domain of area A_s.
Finally, for the 3D case:

$$d_c = \frac{\sqrt[3]{V_s}}{\sqrt[3]{n_{V_s}} - 1} \tag{5}$$

where V_s is an estimation of the volume covered by the support domain d_s and n_{V_s} is the number of nodes falling inside the estimated domain of volume V_s.
In this work, the estimation of d_c in the case of randomly distributed nodes, is approached in a slightly different way. The random nodal distribution is built starting from an even distribution in which the nodes are then perturbed of a very small quantity both in the x and y direction. The order of magnitude of the perturbation ranges from 10^{-3} to 10^{-2} times a characteristic dimension of the problem. Bearing in mind that mainly squared geometries are analysed in this work, and being the value of the perturbation this small, the characteristic length of the support domain is computed in the same way for both regular and irregular nodal distributions:

$$d_{c_x} = \frac{L_x}{n_x} \tag{6}$$

$$d_{c_y} = \frac{L_y}{n_y} \tag{7}$$

Here, L_x and L_y are the lengths of the two sides of the rectangle in x and y respectively and n_x and n_y are the number of nodes along the two directions. Note that d_{c_x} and d_{c_y} are the exact nodal spacing of a regular nodal distribution but are different from each other as it is not necessarily true that the same number of nodes is found in the two directions. Hence, in this work, the characteristic length is always computed as if referred to a set of evenly distributed nodes, no matter the actual nodal distribution.
The other parameter, α_s, is the dimensionless size of the support domain. Tuning this parameter allows to control the number of nodes falling inside each support domain, hence, the accuracy of the interpolation. In this work, depending on the size of the problem, $\alpha_s = 1.3 \div 3$ have been proven to give good results.

2.2.2 Shape functions properties

From the previous discussion emerges that the challenge in constructing Mesh Free shape functions is to use nodes that are not linked by any predefined relationship but are just arbitrarily scattered in the domain. There can be some requirements that are identified to classify a shape function construction method as adequate [4].

I The shape functions must be constructed starting from a reasonably *arbitrary* nodal distribution.

II The casual distribution of nodes in the domain introduces uncertainties in the process. The algorithm to build the shape functions must be *stable* and robust to these uncertainties.

III As in FEM, as the number of nodes increases, the numerical solution has to converge to the exact one. To make sure this happens, shape functions must be of a certain order of *consistency*. A common way to assess the degree of consistency of a Mesh Free Method is by checking the order of the polynomial field that can be represented exactly by means of the approximation. If the approximation obtained using a certain shape function is able to exactly reproduce a polynomial of up to the k-th order, then the approximation has

k-th order consistency (C^k). This definition is the same as FEM. In the latter, the concept of *completeness* is discussed too. In Mesh Free Methods, this translates in guaranteeing that an approximation of C^k consistency has to be consistent from 0 to k, meaning it has to include all the lower orders.

IV It has been explained that shape functions in Mesh Free methods are built using the nodes falling inside the support domain of a point of interest. The dimensions of the domain have also been discussed. In practice, the local domain should be *small* compared to the global domain of the problem. If this happens, the system matrices are banded and this leads to a good computational efficiency.

V These methods are developed to be a viable alternative to FEM codes. Therefore, it makes sense for them to be as *efficient* and of the same degree of complexity as FEM codes. It would be pointless to develop a method whose shape functions computation involves a too expensive procedure. No matter how good it would prove to be, such a method would be impractical and certainly poorly competitive.

There are two more properties about Mesh Free shape functions that have to be pointed out. On the one hand, a compulsory property for the shape functions to possess is the *partition of unity* which is necessary to ensure the ability to reproduce any rigid body motion. This property is expressed as:

$$\sum_{i=1}^{n} \Phi_i(\mathbf{x}) = 1 \tag{8}$$

On the other hand, a preferable property is the *Kronecker delta function property*, which states:

$$\phi_i(\mathbf{x} = \mathbf{x}_j) = \begin{cases} 1 & i = j, \quad j = 1, 2, \ldots, n \\ 0 & i \neq j, \quad j = 1, 2, \ldots, n \end{cases} \tag{9}$$

This property ensures a simple imposition of the essential boundary conditions, namely, it allows to impose them in the same way as in FEM. It is considered "preferable" instead of "compulsory" because even if not satisfied there still are other ways to impose the essential boundary conditions. The price to pay is additional computational expenses.

3 Shape Functions construction

In this chapter the step-by-step procedure to construct the two different kind of shape functions chosen for this work, polynomial and radial Point Interpolation Method, is analysed.
There are different methods that can be used to build different kinds of shape functions, but in this work only the finite series representation methods are employed. They are well know and also widely developed in FEM. They make use of a basis function that ensure consistency and, if augmented with special terms, can improve the accuracy of the results.

3.1 Point Interpolation Method

The Point Interpolation Method can be considered a finite series representation method, which are also well developed in FEM. For this method the approximation of the displacement field is obtained by letting the interpolation function pass through the function values at each node falling inside the support domain of the point of interest. The general procedure for the PIM is the following.
Let's consider a function *u(**x**)* defined in the problem domain Ω represented by a set of nodes distributed on the domain itself and on its boundaries. The PIM interpolates the function *u(**x**)* using the nodal values at the nodes within the support domain of a certain point $\mathbf{x}_Q$:

$$u^h(\mathbf{x}, \mathbf{x}_Q) = \sum_{i=1}^{n} B_i(\mathbf{x}) a_i(\mathbf{x}_Q) \tag{10}$$

where $B_i(x)$ are basis functions defined in Cartesian coordinates, n is the number of nodes inside the support domain of $\mathbf{x}_Q$ and $a_i(\mathbf{x}_Q)$ is the coefficient for $B_i(x)$. The PIM used in this work, makes use of two kinds of basis functions: polynomial and radial.
The formulation for the polynomial PIM starts with the following definition:

$$u^h(\mathbf{x}, \mathbf{x}_Q) = \sum_{i=1}^{n} p_i(\mathbf{x}) a_i(\mathbf{x}_Q) = \mathbf{p}^T(\mathbf{x}) \mathbf{a}(\mathbf{x}_Q) \tag{11}$$

where $a_i(\mathbf{x}_Q)$ is the coefficient for the monomial defined as:

$$\mathbf{a}^T(x) = \{a_1, a_2, a_3, \cdots, a_n\} \tag{12}$$

and $p_i(\mathbf{x})$ is the monomial basis function built using the Pascal triangle shown in Figure 4.

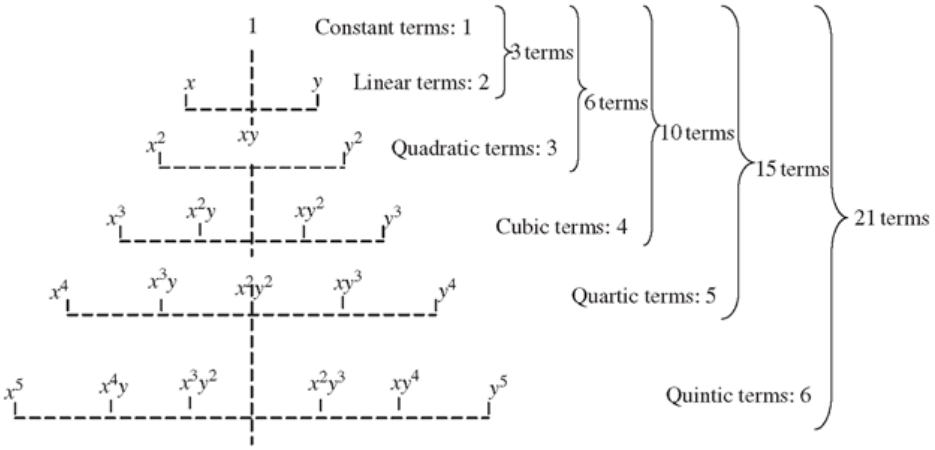


Figure 4: 2D Pascal triangle

Note that the order of the terms inside the polynomial basis may influence the accuracy of the

results.
The coefficients a_i can be determined by imposing that Eq.(11) is satisfied at all of the n nodes inside the support domain. For the *i-th* node it becomes:

$$u_i = \mathbf{p}^T(\mathbf{x}_i)\mathbf{a} \qquad i = 1 \sim n \tag{13}$$

where u_i is the nodal value of u at $\mathbf{x} = \mathbf{x}_i$. Eq.(13) is written in matrix form as:

$$\mathbf{U}_s = \mathbf{P}_Q \mathbf{a} \tag{14}$$

where $\mathbf{U}_s$ is the vector containing the values of the field variables at all the n nodes in the support domain and $\mathbf{P}_Q$ is called *moment matrix* which, for a 2D case, is given by:

$$\mathbf{P}_Q = \begin{bmatrix} p^T(x_1) \\ p^T(x_2) \\ \vdots \\ p^T(x_n) \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 & x_1^2 & y_1^2 & x_1^2y_1 & x_1y_1^2 & x_1^3 & \cdots \\ 1 & x_2 & y_2 & x_2y_2 & x_2^2 & y_2^2 & x_2^2y_2 & x_2y_2^2 & x_2^3 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & y_n & x_ny_n & x_n^2 & y_n^2 & x_n^2y_n & x_ny_n^2 & x_n^3 & \cdots \end{bmatrix} \tag{15}$$

The coefficients a_i in Eq.(11) can be computed as:

$$\mathbf{a} = \mathbf{P}_Q^{-1}\mathbf{U}_s \tag{16}$$

where the assumption of invertible moment matrix is made. A more detailed discussion on the matter will be given in the following paragraphs.
Substituting Eq.(16) into Eq.(11):

$$u^h(\mathbf{x}) = \sum_{i=1}^{n} \Phi_i(\mathbf{x})u_i = \mathbf{\Phi}(\mathbf{x})\mathbf{U}_s \tag{17}$$

where $\Phi(x)$ is a matrix of PIM shape function defined as:

$$\mathbf{\Phi} = \mathbf{p}^T(\mathbf{x})\mathbf{P}_Q^{-1} = [\Phi_1(x), \Phi_2(x), \Phi_3(x), \cdots, \Phi_n(x)] \tag{18}$$

The l-th derivatives of the PIM shape functions, which still has a simple polynomial form, can be immediately obtained as:

$$\mathbf{\Phi}^{(l)}(\mathbf{x}) = [\mathbf{p}^{(l)}(\mathbf{x})]^T\mathbf{P}_Q^{-1} \tag{19}$$

An interesting and useful characteristic of the PIM shape functions is the fact that they possess the Kronecker delta property. Such property allows to easily impose the essential boundary conditions following the same procedure used in conventional FEM.

The consistency of the PIM shape functions depends on the orders of monomial $p_i(x)$ used in Eq.(11) which implies that it depends on the number of nodes in the support domain. If the monomial is of order n the shape functions possess C^n consistency.
If the moment matrix $\mathbf{P}_Q$ is invertible, the shape functions depend only on the distribution of the nodes. They possess the following properties [4]:

1. PIM shape functions are linearly independent in the support domain. In fact, the basis functions are linearly independent themselves and the assumption of invertible moment matrix is made.

2. The PIM shape functions possess the Kronecker delta function property:

$$\phi_i(x = x_j) = \begin{cases} 1 & i = j, \quad j = 1, 2, \ldots, n \\ 0 & i \neq j, \quad j = 1, 2, \ldots, n \end{cases} \tag{20}$$

   This property allows for an easy imposition of the essential boundary conditions which follows the same procedure of conventional FEM.

3. The shape function for node i, ϕ_i, is the partition of unity:

$$\sum_{i=1}^{n} \phi_i(x) = 1 \tag{21}$$

if the constant term is included in the monomial basis. This property is also a good check, during numerical implementation, to ensure that the computation of the shape functions is done correctly.

4. Shape functions have reproducing properties

$$\sum_{i=1}^{n} \phi_i(x) = x \tag{22}$$

if the first order monomial is included in the basis. Considering $u(\mathbf{x}) = \mathbf{x}$, then:

$$\mathbf{U}_s = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}^T \tag{23}$$

Using Eq.(17), it becomes:

$$u^h(\mathbf{x}) = \sum_{i=1}^{n} \Phi_i(\mathbf{x}) u_i = \mathbf{x} \tag{24}$$

5. The derivatives of the shape functions have polynomial form.

6. If the chosen support domain is compact, the shape function is of compact support as well.

7. PIM shape function is not compatible, hence the approximation using the PIM can be discontinuous when x_Q changes and the nodes in the support domain are updated as a consequence of the moving local domain.

3.2 Example: PIM shape functions in 1D

Let's consider a 1D space in which the support domain $[-1, 1]$ is considered. Let's say that five evenly distributed nodes fall inside such support domain which will be centered in one of the nodes. The nodal distribution is shown in Figure 5.

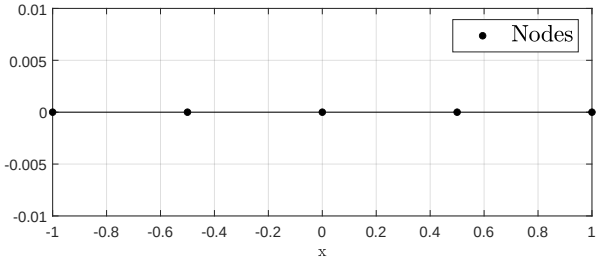


Figure 5: 1D nodal distribution for shape functions construction.

In this 1D case, the polynomial basis to consider is:

$$\mathbf{p}^T(x) = \{1, x, x^2, x^3, x^4\} \tag{25}$$

Note that in 1D cases the inverse of the moment matrix $\mathbf{P}_Q$ always exists.
According to where the support domain is centered and assuming that all the nodes always fall inside it, the five resulting shape functions (one for each node) are shown in Figure 6.
The first derivative of the shape functions is computed by following the same procedure but

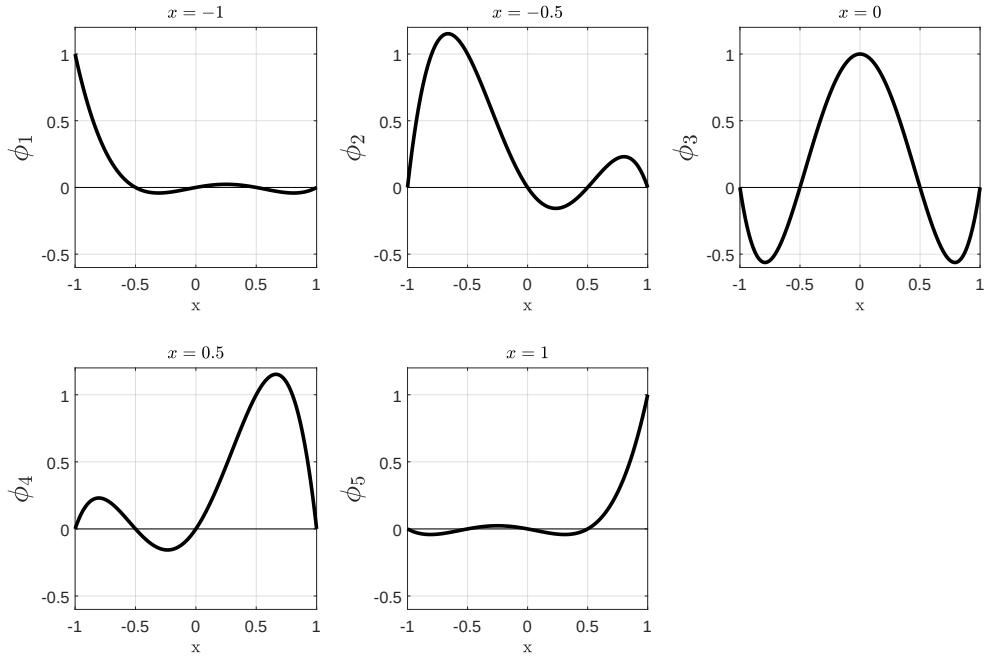


Figure 6: 1D Polynomial PIM Shape functions for each node in the domain

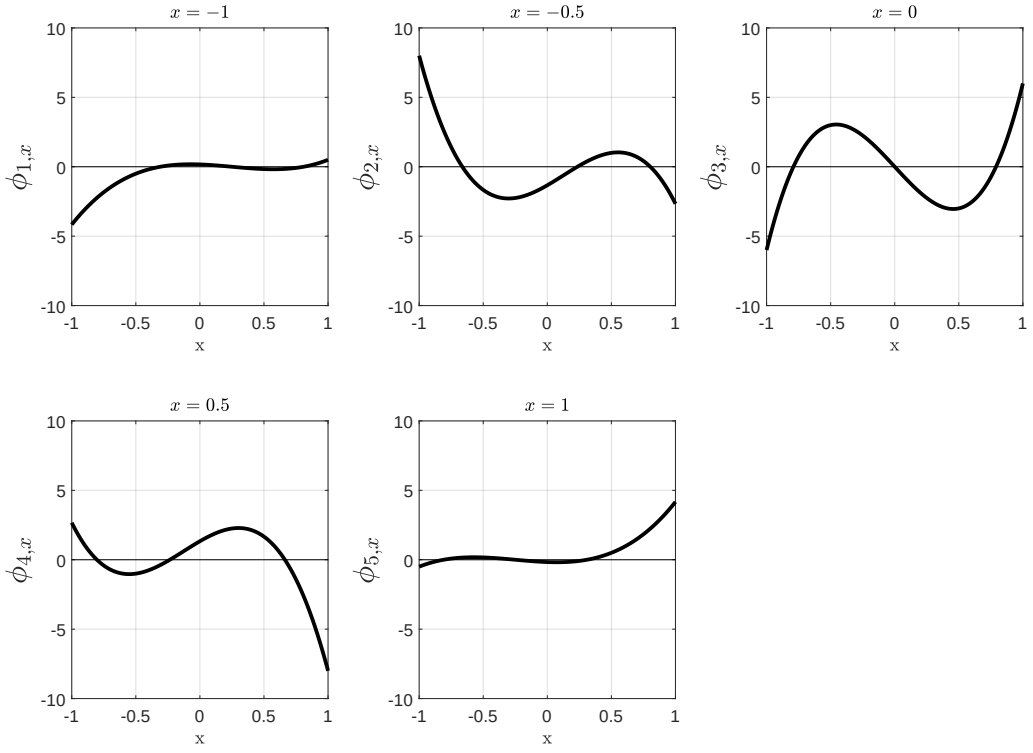


Figure 7: 1D Polynomial PIM shape functions first derivative

considering the derivative of the polynomial basis:

$$\mathbf{p}^{\mathsf{T}}(\mathrm{x}) = \{0, 1, 2\mathrm{x}, 3\mathrm{x}^2, 4\mathrm{x}^3\} \tag{26}$$

Figure 7 shows the derivatives of the shape functions for a support domain centered in each node.

3.2.1 Example: PIM shape functions in 2D and 3D

The shape functions for the 2D and 3D cases can be easily obtained by modifying the polynomial basis used. The arrangement of the polynomial terms inside the basis may vary. Considering the Pascal triangle, from the quadratic terms on, the arrangement may consider the terms of higher order first and the other following with decreasing exponents value. In this work however, it will be proven that a symmetric arrangement of the polynomial terms yields better results. Accordingly, the polynomial basis chosen is in the form:

$$\mathbf{p}^{\mathsf{T}}(x) = \{1, x, y, x^2, xy, y^2, x^3, x^2y, xy^2, y^3, \cdots\} \tag{27}$$

for the 2D case, and

$$\mathbf{p}^{\mathsf{T}}(x) = \{1, x, y, z, x^2, xy, y^2, yz, xz, z^2, \cdots\} \tag{28}$$

for the 3D case.
The shape functions for the 2D case are computed for a $[-1, 1] \times [-1, 1]$ squared domain containing 9 nodes as shown in Figure 8. Note how the coordinates of the middle node have been slightly perturbed to avoid singularity issues in the Moment Matrix $\mathbf{P}_Q$. An example of

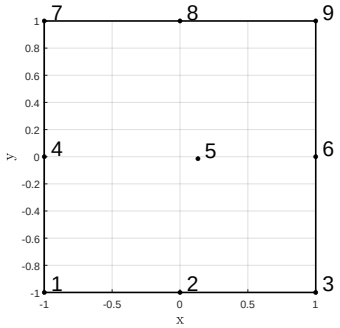


Figure 8: 2D domain for 2D polynomial PIM shape functions construction.

2D PIM shape functions and their derivatives with respect to x and y is shown in Figures 9 to 11.

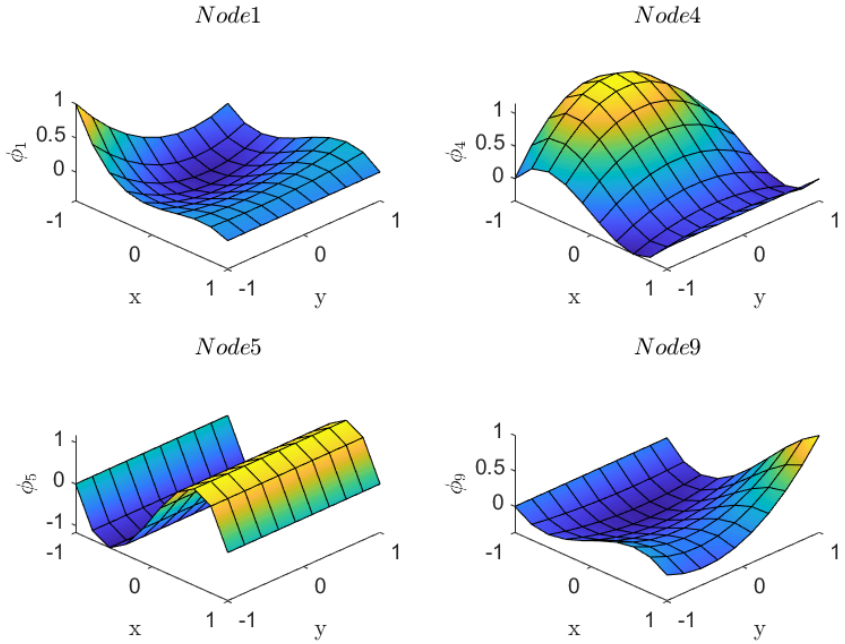


Figure 9: 2D Polynomial PIM shape functions for nodes 1, 4, 5 and 9.

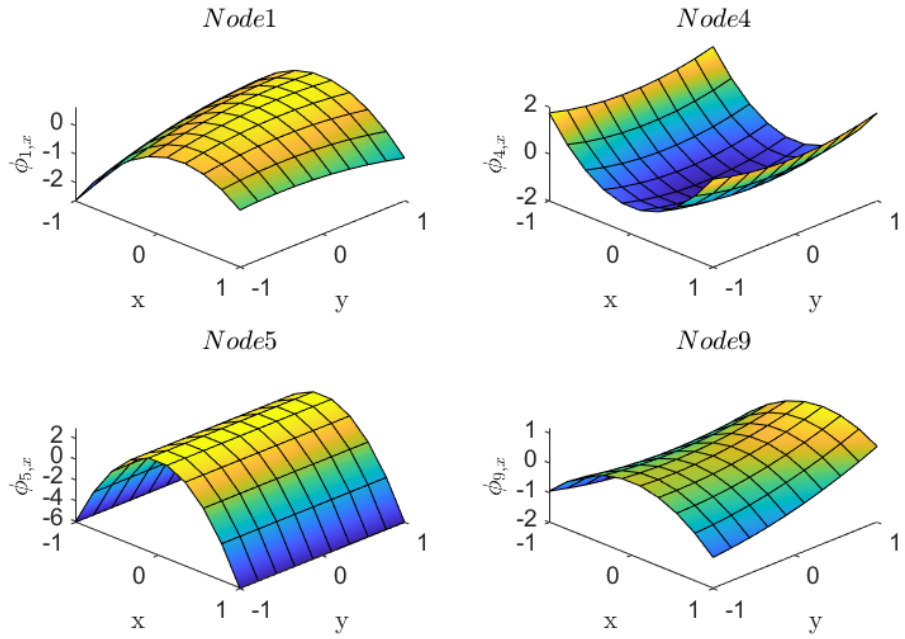


Figure 10: 2D Polynomial PIM shape functions x derivatives for nodes 1, 4, 5 and 9.

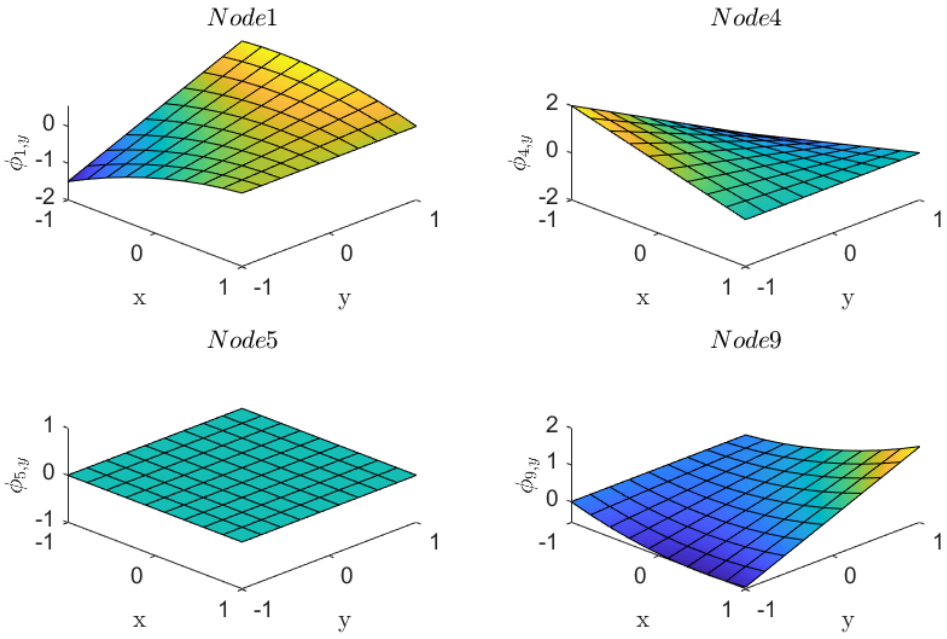


Figure 11: 2D Polynomial PIM shape functions y derivatives for nodes 1, 4, 5 and 9.

3.3 Critical aspects of PIM

3.3.1 Support domain

Since elements are not present in Mesh Free methods, as mentioned, the field variable is interpolated using the displacements at the nodes falling inside a small local domain called *support domain*. In other words, the support domain of a point $\mathbf{x}$ determines the number of nodes to be used to approximate the variable of interest at $\mathbf{x}$.
The support domain is centered in a point of interest, which can be either a node or another point. Gauss integration is used in this work and the support domain, throughout the whole analysis, is centered in the Gauss integration points. In theory, centering the local domain in either nodes or Gauss integration points makes no difference. However, it is common practice to use the Gauss points as points of interest and in this work, the choice is also supported by numerical reasons. In fact, as seen in Section 3.1, the construction of the moment matrix makes use of the polynomial basis and of the coordinates of the nodes inside the support domain. In some cases, for instance when both the coordinate and the related exponent are 0, for numerical reasons, the program will deliver an incorrect result. For the example given, x^{n}, if both x and n are 0, the result will be 1. Moreover, when solving the system the leads to the computation of the shape functions, these zeros may cause additional errors. Choosing the Gauss integration points as center of the support domain, reduces the possibility of having 0 among the coordinates, avoiding this kind of numerical errors and also sparing the computational cost needed to prevent them from occurring.
The support domain can have different shapes and also its dimensions can vary even from point of interest to point of interest. Two examples of support domain shapes have been shown earlier (see Figure 2).
Usually, for the PIM, the dimensions of the support domain are chosen such that the number of nodes it contains ranges from 7 to 15 for 2D problems. If too few points are used in the construction of the shape functions, the final results may be affected by significant errors. Hence, the dimension of the support domain plays a critical role in the accuracy of the method.
A simple way to choose its size has been shown in Eq.(2). The key for the definition of the support size is the non-dimensional parameter α_s which can be tuned to achieve a good amount of nodes to be used in constructing the shape functions. Due to the singularity problem of the moment matrix, this tuning is of the uttermost importance: usually, an $\alpha_s = 2.0$ or 3.0 leads to good results but it is not always the case. Hence, a trial and error procedure is required to make sure that no singularity occurs and that the results are reliable and consistent with the problem.

3.3.2 Nodal distribution and singular moment matrix

The main issue of the point interpolation method is the possibility of having a singular moment matrix $\mathbf{P}_Q$. This occurrence makes it impossible to compute the shape functions and it occurs more often with regular nodal distributions. Hence, to avoid the problem without using any additional algorithm, the regularly distributed nodes can be perturbed by a small quantity. Perturbation values as low as 10^{-2} times a characteristic dimension of the problem have proven to be effective in avoiding singularity in the moment matrix.
Still, the problem persists when the support domain includes nodes located, for instance, on a regular boundary. In fact, it has been assessed that the moment matrix $\mathbf{P}_Q$ becomes singular when the support domain contains three or more nodes aligned either in the x or y direction. When dealing with rectangular plates or 2D solids in general, it is inevitable for the nodes on the boundary to be aligned in one of the two directions. There are two possible solutions to this issue. The first requires to increase the perturbation of the nodes as to compensate for the regularity of the boundary nodes falling inside the local domain. Since this strategy may prove ineffective, another possibility is the tuning of the α_s parameter. As the support domain size

is increased or reduced, the regularity of the nodes on the boundary get compensated by the randomly distributed nodes in the rest of the domain.
None of the two proposed solutions is always effective and both require a trial and error procedure to either find an optimal perturbation value or an ideal dimension for the support domain.

3.3.3 Support domain overlap

Let's consider a problem in which the nodal distribution is such that no singularity in the moment matrix occurs. There is another condition to be fulfilled in order to guarantee the convergence to the exact solution of the problem. In fact, the continuity and consistence of the PIM depend on the number of nodes used in the interpolation and the number of nodes that fall in the overlapping area of two neighbouring support domain of two sampling points. If two support domains overlap, at least two nodes must be included in the overlapping area to guarantee C^0 continuity [5]. Figure 12 shows an example in which this condition is satisfied. In the figure, two neighbouring rectangular support domains overlap. The nodal distribution is such that three nodes fall inside the overlapping area, thus satisfying the requirement. Note also that neighbouring support domain does not mean necessarily that the support domains are centered in two neighbouring nodes. Figure 12 shows how the overlapping domains are relative to two nodes far from one another.

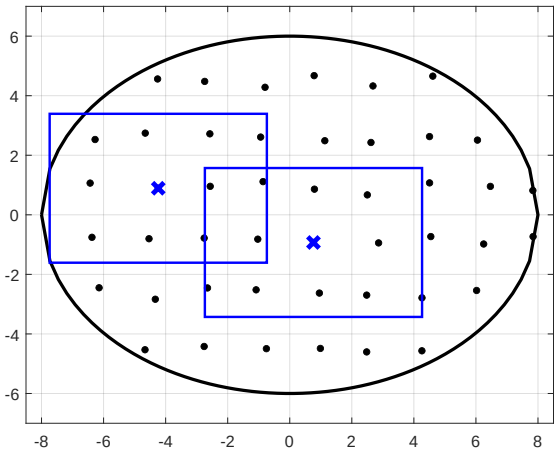


Figure 12: Support domains with overlapping area containing three nodes

3.4 Radial Point Interpolation Method

The worst issue of polynomial PIM is the possible occurrence of a singular moment matrix $\mathbf{P}_Q$. To make sure this doesn't happen, radial basis functions are introduced in PIM theory to construct the shape functions.
Referring to Eq.(10), the basis $B_i(x)$ is chosen as a set of radial functions:

$$u^h(\mathbf{x}, \mathbf{x}_Q) = \sum_{i=1}^{n} R_i(\mathbf{x}) a_i(\mathbf{x}_Q) = \mathbf{R}^T(\mathbf{x})\mathbf{a}(\mathbf{x}_Q) \tag{29}$$

where the definition of vector $\mathbf{a}$ is given in Eq.(12). R_i is a radial basis function. Denoting with r the distance between two points $\mathbf{x}$ and $\mathbf{x}_i$ the radial basis function in 2D is defined as:

$$r = [(x - x_i)^2 - (y - y_i)^2]^{1/2} \tag{30}$$

The vector of radial basis functions is written as:

$$\mathbf{R}^T(\mathbf{x}) = [R_1(\mathbf{x}), R_2(\mathbf{x}), R_3(\mathbf{x}), \cdots, R_n(\mathbf{x})] \tag{31}$$

Different kinds of radial basis functions have been developed over the years; the most common are summarized in Table 1:

Name	Expression	Shape Parameters
Multiquadrics (MQ)	$R_i(x, y) = (r_i^2 + C^2)^q = [(x - x_i)^2 + (y - y_i)^2 + C^2]^q$	C,q
Gaussian (EXP)	$R_i(x, y) = exp(-cr_i^2) = exp\{-c[(x - x_i)^2 + (y - y_i)^2]\}$	c
Thin plate spline (TPS)	$R_i(x, y) = r_i^\eta = [(x - x_i)^2 + (y - y_i)^2]^\eta$	η
Logarithmic RBF	$R_i(r_i) = r_i^\eta \log r_i$	η

Table 1: Radial basis functions

The two most used radial basis functions are the multiquadric (MQ) basis and the Gaussian radial basis, or EXP. Each radial basis function depends on some parameters which can be tuned differently for different problems to achieve better performance. In this work, only MQ basis functions will be used.
Recalling Eq.(11), $u(\mathbf{x})$ has to pass through each node falling inside the support domain for the point of interest $\mathbf{x}$. Hence, we have a moment matrix which, for RPIM, is written as:

$$\mathbf{R}_Q = \begin{bmatrix} R_1(r_1) & R_2(r_1) & \cdots & R_n(r_1) \\ R_1(r_2) & R_2(r_2) & \cdots & R_n(r_2) \\ \vdots & \vdots & \vdots & \vdots \\ R_1(r_n) & R_2(r_n) & \cdots & R_n(r_n) \end{bmatrix} \tag{32}$$

The radial moment matrix thus obtained is always invertible for an arbitrary distribution of nodes. The only singularity issue that may occur is due to the values of the shape parameters. Specific values of these shape parameters that lead to singularity issues are known and hence avoidable. In any case, if the singularity problem occurs, even slight changes of the shape parameters can solve it. Therefore, $\mathbf{R}_Q$ can always be symmetric and invertible and this is the main advantage in moving from polynomial PIM to radial PIM.
In Eq.(30) the distance r_k is expressed as:

$$r_k = [(x_k - x_i)^2 + (y_k - y_i)^2]^{1/2} \tag{33}$$

Since the distance is directionless

$$R_i(r_j) = R_j(r_i) \tag{34}$$

and the matrix $\mathbf{R}_Q$ is symmetric.
The vector of coefficients $\mathbf{a}(\mathbf{x}_Q)$ in Eq.(29) is obtained by enforcing the interpolating function passes through all the nodes in the support domain. At the k-th point, it is given by:

$$u_k = u(x_k, y_k) = \sum_{i=1}^{n} a_i R_i(x_k, y_k) \qquad k = 1, 2, \cdots, n \tag{35}$$

In matrix form it can be written as:

$$\mathbf{U}_s = \mathbf{R}_Q \mathbf{a} \tag{36}$$

where $\mathbf{U}_s$ is a vector collecting the field nodal variables at the n nodes in the support domain. If the inverse of $\mathbf{R}_Q$ exists, a unique solution for the vector of coefficients $\mathbf{a}$ can be obtained as:

$$\mathbf{a} = \mathbf{R}_Q^{-1} \mathbf{U}_s \tag{37}$$

Substituting Eq.(37) into Eq.(29):

$$u^h(\mathbf{x}) = \mathbf{R}^T(\mathbf{x}) \mathbf{R}_Q^{-1} \mathbf{U}_s = \mathbf{\Phi}(x) \mathbf{U}_s \tag{38}$$

where $\mathbf{\Phi}$ is the matrix containing the n shape functions:

$$\mathbf{\Phi}(\mathbf{x}) = [R_1(x), R_2(x), \cdots, R_k(x), \cdots, R_n(x)] \mathbf{R}_Q^{-1} \tag{39}$$

$$= [\Phi_1(x), \Phi_2(x), \cdots, \Phi_k(x), \cdots, \Phi_n(x)] \tag{40}$$

$$\tag{41}$$

Here, $\Phi_k(x)$ represents the shape function for the k-th node:

$$\Phi_k(x) = \sum_{i=1}^{n} R_i(x) R^{-1}_{Q_{ik}} \tag{42}$$

In the foregoing equation, $R^{-1}_{Q_{ik}}$ is the (i,k) element of matrix $\mathbf{R}_Q^{-1}$ which is a constant matrix for given locations of the n nodes in the support domain.
The derivatives of the shape functions

$$\frac{\partial \Phi_k}{\partial x} = \sum_{i=1}^{n} \frac{\partial R_i}{\partial x} R^{-1}_{Q_{ik}} \tag{43}$$

$$\frac{\partial \Phi_k}{\partial y} = \sum_{i=1}^{n} \frac{\partial R_i}{\partial y} R^{-1}_{Q_{ik}} \tag{44}$$

$$\tag{45}$$

are no longer simple polynomials. Still, the partial derivatives for MQ radial functions have the following simple expression:

$$\frac{\partial R_i}{\partial x} = 2q(r_i^2 + C^2)^{q-1}(x - x_i) \tag{46}$$

$$\frac{\partial R_i}{\partial y} = 2q(r_i^2 + C^2)^{q-1}(y - y_i) \tag{47}$$

$$\tag{48}$$

3.4.1 RPIM Moment Matrix

The only difference between the PIM and RPIM is the basis function used. While PIM uses polynomial basis functions, RPIM uses radial basis functions which have been proven making the radial moment matrix $\mathbf{R}_Q$ always symmetric and invertible. This is true for any arbitrarily scattered nodal distributions. Numerical issues may arise if some specific shape parameters are used but, since these values are known, they are easy to avoid. For instance, it is preferred to not set the parameter C equal to 2. In structural problems, in fact, this value leads to significant errors in the computations of the deflections. On the other hand, when the non-dimensional parameter q is set to 1, the original radial MQ function degenerate to an unstable polynomial form which may lead to very high errors. Hence, for stability reasons, q should never be set equal to 1.
Hence, this is the major advantage of this method with respect to the PIM: the possibility of inverting $\mathbf{R}_Q$.

3.5 Dimensionless shape parameters for radial functions

As shown in Table 1, the radial basis functions exhibit a dependence on some shape parameters that can be tuned. Liu et al. [4] have recently proven that it is difficult to standardize these parameters and decided to propose a set of radial basis functions that have dimensionless parameters. Some of the new radial basis functions, which are obtained just by making some minor modifications to the original ones, are shown in Table 2, where the last column represents the relationship between the original and dimensionless parameters . Here, d_c is a characteristic

Name	Expression	Parameters	Relations
Multiquadrics (MQ)	$R_i(x,y) = (r_i^2 + (\alpha_C d_c)^2)^q$	$\alpha_C \geqslant 0, q$	$\alpha_C = C/d_c,\ q{=}q$
Gaussian (EXP)	$R_i(x,y) = exp(-\alpha_c (r_i/d_c)^2)$	α_c	$\alpha_c = cd_c$
Thin plate spline (TPS)	$R_i(x,y) = r_i^\eta$	η	$\eta = \eta$
Logarithmic RBF	$R_i(r_i) = r_i^\eta \log r_i$	η	$\eta = \eta$

Table 2: Radial basis functions with dimensionless parameters

length related to the nodal spacing of the nodes within the support domain of a point of interest $\mathbf{x}_Q$. Usually it is the average nodal spacing inside the domain.
Note that the shape parameters in TPS and logarithm RBFs are already dimensionless and no change is needed.
The first and second-order partial derivatives will change accordingly and, for MQ function, they are obtained as:

$$\frac{\partial R_i}{\partial x} = 2q(r_i^2 + (\alpha_C d_c)^2)^{q-1}(x - x_i) \tag{49}$$

$$\frac{\partial R_i}{\partial y} = 2q(r_i^2 + (\alpha_C d_c)^2)^{q-1}(y - y_i) \tag{50}$$

$$R_{i,xx} = 2q[r_i^2 + (\alpha_C d_c)^2]^{q-1} + 4q(q-1)[r_i^2 + (\alpha_C d_c)^2]^{q-2}(x - x_i)^2 \tag{51}$$

$$R_{i,xy} = 4q(q-1)[r_i^2 + (\alpha_C d_c)^2]^{q-2}(x - x_i)(y - y_i) \tag{52}$$

$$R_{i,yy} = 2q[r_i^2 + (\alpha_C d_c)^2]^{q-1} + 4q(q-1)[r_i^2 + (\alpha_C d_c)^2]^{q-2}(y - y_i)^2 \tag{53}$$

$$\tag{54}$$

3.6 Example: RPIM shape functions in 1D

Let's consider the same 1D domain as per the 1D PIM shape functions example: five nodes evenly distributed in a support domain of $[-1, 1]$. The MQ-PIM shape functions for each node are computed by centering the support domain in the various nodes and performing

computations. The 1D Multiquadrics radial basis function and its first derivative are written as:

$$R_{MQ_i}(x) = [(x - x_i)^2 + C^2]^q \tag{55}$$

$$\frac{\partial R_{MQ_i}(x)}{\partial x} = 2q[(x - x_i)^2 + C^2]^{q-1}(x - x_i) \tag{56}$$

$$\tag{57}$$

where the shape parameters C and q, used for the computation, have values equal to 1.0 and 0.5, respectively. The resulting shape functions and their derivatives are shown in Figures 13 and 14.

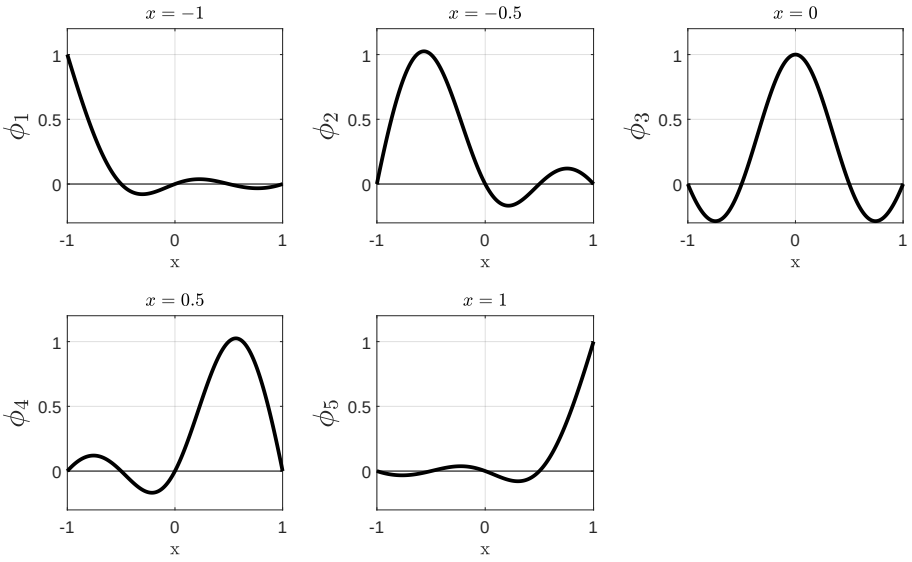


Figure 13: 1D RPIM shape functions.

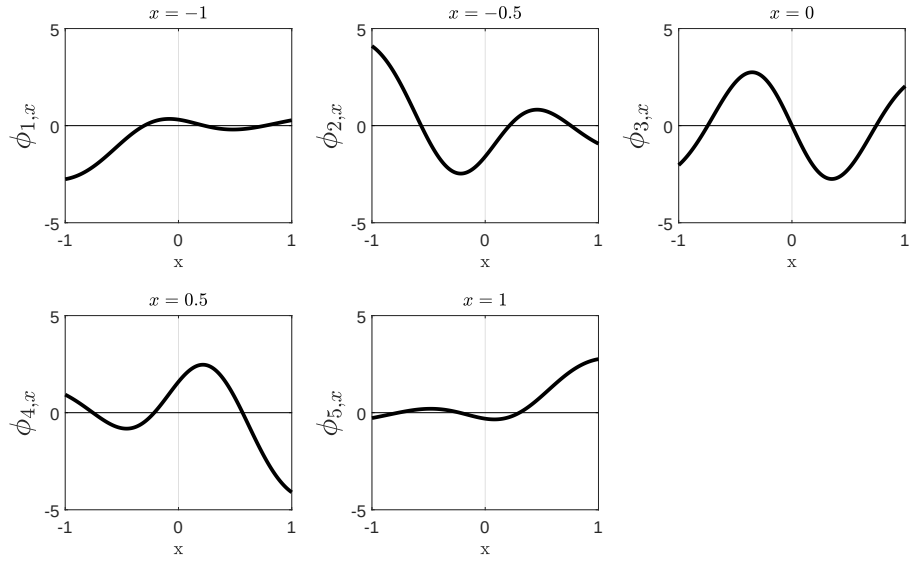


Figure 14: 1D RPIM shape function first derivative.

It is clear from Figure 13 that the RPIM shape functions poss the Kronecker delta property.

3.6.1 Example: RPIM shape functions in 2D and 3D

Similarly to the PIM case, let's consider the squared domain of Figure 8. Having solved the singularity issue with the moment matrix, there is no need, with the RPIM method, to perturb the mid node.
The MQ radial basis are in the form:

$$R_{MQ_i}(x, y) = [(x - x_i)^2 + (y - y_i)^2 + C^2]^q \tag{58}$$

for the 2D case, and

$$R_{MQ_i}(x, y) = [(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 + C^2]^q \tag{59}$$

for the 3D case.
Three sample graphs of the 2D shape functions and derivatives are shown in Figure 15 to 17. The plots refer to nodes 1, 5 and 9.

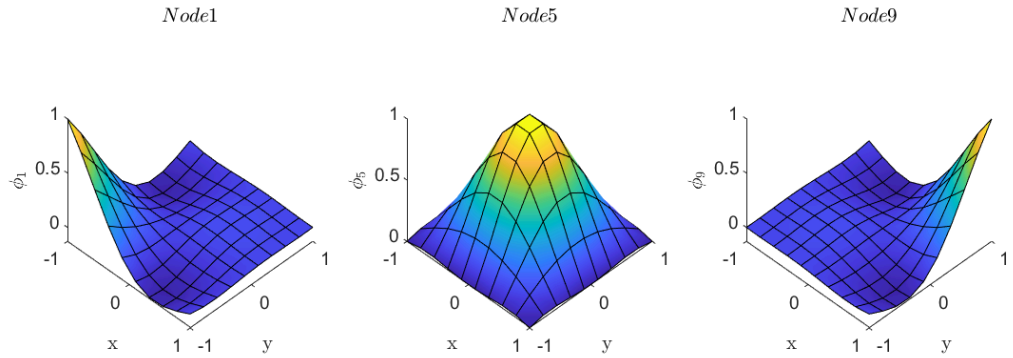


Figure 15: 2D MQ-RPIM shape functions for nodes 1, 5 and 9.

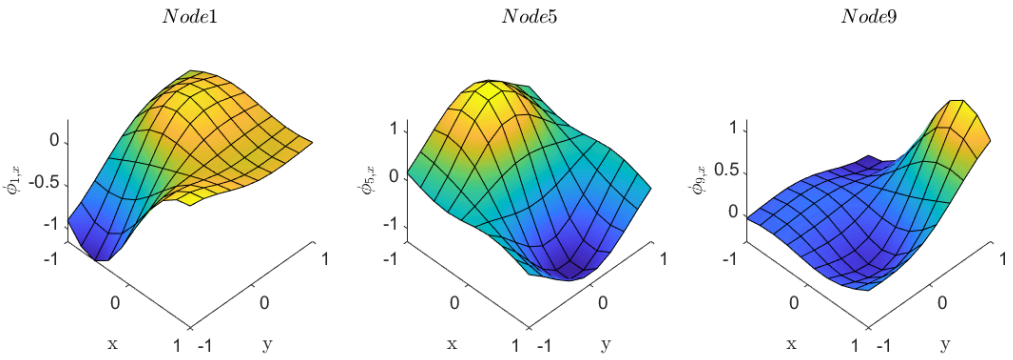


Figure 16: 2D MQ-RPIM shape functions x derivatives for nodes 1, 5 and 9.

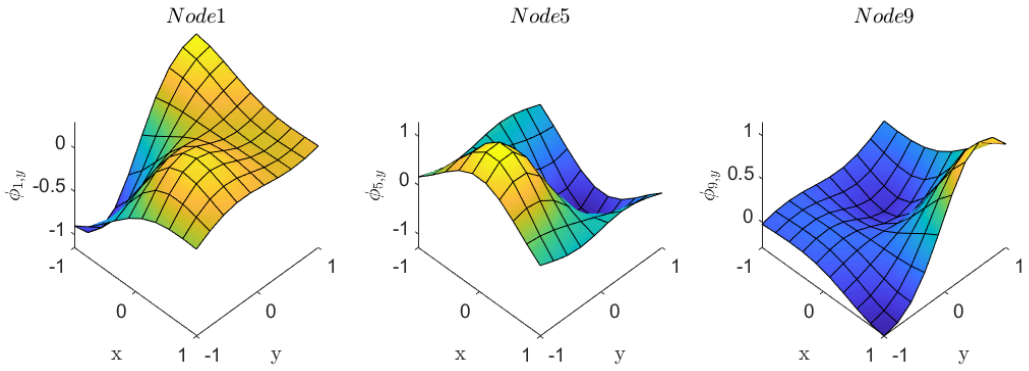


Figure 17: 2D MQ-RPIM shape functions y derivatives for nodes 1, 5 and 9.

3.7 Importance of parameters and support domain dimension

Although the singularity problem of the moment matrix is solved, the RPIM still has some criticalities to deal with.
Just as in the case of the polynomial PIM, the dimension of the support domain influences the accuracy of the solution. Referring to Eq.(2), this means that the tuning of the parameter α_s is still an important step when tuning the method and several trials may be needed to achieve good results.
An issue that is, instead, typical of RPIM only, concerns the shape parameters on which the radial basis functions depend on. In this work, only MQ-PIM shape functions are used, hence two parameters have to be tuned: C and q. These values can strongly influence the results, especially the computation of the stresses. It has been observed that different values of the dimensionless parameter C yield different results in the computation of the stresses. The sensitivity of the solution to these parameters can be relatively large. For instance, choosing C equal to 1 gives extremely different results than C equal to 1.42.
During the analysis performed in this work, it has been noticed that variations of the parameter q have almost no effect on the solution. On the other hand, the major influence is given by parameter C which can be varied in combination with α_s. Although these parameters are dimensionless, it has been observed that for problems of significant dimensions, larger values of C and α_s are preferable. The best choices range around $1 \div 1.42$ for C and $2 \div 3$ for α_s. For small domains however, these values may need to be re-calibrated. The parameter α_s undergoes the minor of changes since its value may still range in the interval $1.3 \div 2.5$. C on the other hand, has to be almost halved. The best results are achieved by considering values around $0.5 \div 0.75$.
Even narrowing so much the values of these parameters, depending on the geometry of the problem and the number of nodes, significant variations of the results may occur with even slight variation of each parameter.
For instance, let's consider to perform an analysis on a body with a certain nodal density and to tune the parameter to achieve the best possible results. Let's say that, after the analysis, it is necessary to increase the number of nodes in the global domain. Keeping constant the global dimensions and the parameter α_s, the number of nodes falling inside the support domain increases. For the construction of RPIM shape functions, it is preferable to use 9 to 15 nodes. If the dimension of the support domain remains the same for an increasing nodal density, the nodes used to build the shape functions may become too many. Moreover, if in one case a parameter $C = 0.75$ is adequate, in the other a value of 0.5 may be required. Parameters which differ so little in values may yield very different results.

4 Application of Mesh Free Methods to 1D and 2D Structures

In this chapter, the basic equations of solid mechanics are presented which will later be applied to obtain those of the discretized problem.
The structural analysis is performed under the assumption of small displacements and elastic materials.

4.1 1D structures

In the 1D case, the only unknown of the problem is given by the axial displacement component u. The linear strain-displacement relation is given by the relation

$$\varepsilon_x = \frac{\partial u}{\partial x} \tag{60}$$

The constitutive equation, or Hooke's law, for uniaxial state of stress reads:

$$\sigma = E\varepsilon \tag{61}$$

Similarly, the equation for the dynamic equilibrium is:

$$\frac{\partial \sigma_x}{\partial x} + b_x = \rho \ddot{u} \tag{62}$$

4.2 2D solids

For 2D solids the assumption is that the dependent variables are a function of the in-plane coordinates x, y only. All the loading conditions are applied in the x-y plane and the component of strains and stresses related to the z direction are zero.
The displacement vector, in this case, has the form

$$\mathbf{u} = \begin{Bmatrix} u \\ v \end{Bmatrix} \tag{63}$$

where u and v are the displacement components in the x and y direction respectively. The strain-displacement relation is

$$\varepsilon_{xx} = \frac{\partial u}{\partial x} \tag{64}$$

$$\varepsilon_{yy} = \frac{\partial v}{\partial y} \tag{65}$$

$$\varepsilon_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \tag{66}$$

$$\tag{67}$$

Considering a plane stress condition, Hooke's law is written as:

$$\boldsymbol{\sigma} = C\boldsymbol{\varepsilon} \tag{68}$$

where matrix C for isotropic materials is

$$C = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{(1-\nu)}{2} \end{bmatrix} \tag{69}$$

The dynamic equilibrium is written as

$$\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{yx}}{\partial y} + b_x = \rho \ddot{u} \tag{70}$$

$$\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + b_y = \rho \ddot{v} \tag{71}$$

$$\tag{72}$$

where

$$\mathbf{b} = \begin{Bmatrix} b_x \\ b_y \end{Bmatrix} \tag{73}$$

is the vector collecting the external body forces.

4.3 3D solids

In 3D solids all the three displacement components are considered:

$$\mathbf{u} = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \tag{74}$$

In general, there are six independent components of stress on each point considering that the stress tensor is symmetric. Accordingly, there are six components of strain that have to be considered: $\boldsymbol{\varepsilon}^T = \{\varepsilon_{xx} \quad \varepsilon_{yy} \quad \varepsilon_{zz} \quad \gamma_{yz} \quad \gamma_{xz} \quad \gamma_{xy}\}$, where

$$\varepsilon_{xx} = \frac{\partial u}{\partial x} \tag{75}$$

$$\varepsilon_{yy} = \frac{\partial v}{\partial y} \tag{76}$$

$$\varepsilon_{zz} = \frac{\partial w}{\partial z} \tag{77}$$

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \tag{78}$$

$$\gamma_{xz} = \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \tag{79}$$

$$\gamma_{yz} = \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \tag{80}$$

The constitutive equation is the same as in Eq.(68) but in this case, C is a symmetric 6×6 matrix whose components have to be determined experimentally.
The dynamic equilibrium equations must also be modified to account for the dependence on the z direction:

$$\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{yx}}{\partial y} + \frac{\partial \sigma_{zx}}{\partial z} + b_x = \rho \ddot{u} \tag{81}$$

$$\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{zy}}{\partial z} + b_y = \rho \ddot{v} \tag{82}$$

$$\frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} + b_z = \rho \ddot{w} \tag{83}$$

$$\tag{84}$$

4.4 Principle of Virtual Work

The equations presented in the previous sections represents a set of partial differential equations (PDE), which form the strong form of the system of governing equations. To obtain an algebraic set of equations, in other words to apply the Mesh Free Methods explained so far, it is more convenient to deal with the weak formulation of the problem.
The variational principle used in this work is the Principle of Virtual Work (PVW) which, for a generic 3D continuum, is written as

$$\int_V \delta\boldsymbol{\varepsilon}^T \boldsymbol{\sigma} dV = \int_V \delta\mathbf{u}^T \mathbf{f}^b dV + \int_S \delta\mathbf{u}^T \mathbf{f}^S dV + \sum_i \delta\mathbf{u}_i^T \mathbf{F}_i^C \tag{85}$$

where $\mathbf{f}^b$ represents the field of body forces on the volume of the solid, $\mathbf{f}^S$ the surface traction applied on the surface S and $\mathbf{F}_i^C$ the concentrated forces applied at node i.
It is convenient to express Eq.(85) in terms of displacements only. Keeping into account the displacement-strain relationships given in Equations 60, 67 and 80, the generic displacement strain relationship can be written in vector form as:

$$\boldsymbol{\varepsilon} = \mathbf{L}\mathbf{u} \tag{86}$$

where $\mathbf{L}$ is a matrix containing the differential operators. For a 3D case it is given by

$$\mathbf{L} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{bmatrix} \tag{87}$$

The 1D and 2D cases can be easily obtained neglecting the appropriate terms.
Remembering the constitutive equation, Eq.(69) and substituting Eq.(86) in Eq.(85), the Principle of Virtual Work can written as function of displacements only:

$$\int_V (\mathbf{L}\delta\mathbf{u})^T \mathbf{C} (\mathbf{L}\delta\mathbf{u}) dV = \int_V \delta\mathbf{u}^T \mathbf{f}^b dV + \int_S \delta\mathbf{u}^T \mathbf{f}^S dS + \sum_i \delta\mathbf{u}_i^T \mathbf{F}_i^C \tag{88}$$

The approximation of the displacement field, as written in Eq.(1), can now be directly introduced in Eq.(88) and the discretized set of equations can be obtained and specialized according to the degree of complexity of the model. To this aim, a series of paragraphs are now presented that illustrates how the matrices of the system varies for the different cases analysed in this work and whose results are shown later in the final chapters. This is expressed as a set of integrals which have to be numerically evaluated, hence the need for a background mesh.

4.5 Equations for membrane problems

In this section, the formulation for 2D solids is developed as an explicative case in which the main aspects of the formulation are presented. The 1D and 3D cases can be obtained as a special case by extending the present formulation.
The starting point for the derivation of the equations is always the PVW in the form shown in Eq.(88). The displacement field for 2D solids, as shown in Eq.(63), can be interpolated as:

$$\mathbf{u}^h = \begin{Bmatrix} u \\ v \end{Bmatrix}^h = \sum_{i=1}^{n} \begin{bmatrix} \Phi_i & 0 \\ 0 & \Phi_i \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \end{Bmatrix} = \sum_{i=1}^{n} \boldsymbol{\Phi}_i \mathbf{u}_i \tag{89}$$

where $\mathbf{u}^h$ is the displacement field of the node of interest, n is the number of nodes falling inside the support domain and used for the interpolation and $\mathbf{\Phi}_i$ is the matrix of shape functions for node i. In this work, the shape functions matrix $\mathbf{\Phi}$ contains either PIM or RPIM shape functions.
The interpolation can now be plugged inside Eq.(88):

$$\int_\Omega \delta\left(\mathbf{L}\sum_{i=1}^{n}\mathbf{\Phi}_i\mathbf{u}_i\right)^T\mathbf{C}\left(\mathbf{L}\sum_{j=1}^{n}\mathbf{\Phi}_j\mathbf{u}_j\right)d\Omega = \int_\Omega \delta\left(\sum_{i=1}^{n}\mathbf{\Phi}_i\mathbf{u}_i\right)^T\mathbf{f}^b\,d\Omega + \int_{\Gamma_t}\delta\left(\sum_{i=1}^{n}\mathbf{\Phi}_i\mathbf{u}_i\right)^T\mathbf{f}^S\,d\Gamma \tag{90}$$

Remembering that $\mathbf{L}$ is a matrix containing the derivation operators, the strain matrix $\mathbf{B}_i$ can be obtained as:

$$\mathbf{L}\sum_{i=1}^{n}\mathbf{\Phi}_i\mathbf{u}_i = \sum_{i=1}^{n}\begin{bmatrix}\frac{\partial}{\partial x} & 0\\ 0 & \frac{\partial}{\partial y}\\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x}\end{bmatrix}\begin{bmatrix}\Phi_i & 0\\ 0 & \Phi_i\end{bmatrix}\mathbf{u}_i = \sum_{i=1}^{n}\begin{bmatrix}\Phi_{i,x} & 0\\ 0 & \Phi_{i,y}\\ \Phi_{i,y} & \Phi_{i,x}\end{bmatrix}\mathbf{u}_i = \sum_{i=1}^{n}\mathbf{B}_i\mathbf{u}_i \tag{91}$$

Eq.(90) then becomes:

$$\int_\Omega \delta\left(\sum_{i=1}^{n}\mathbf{B}_i\mathbf{u}_i\right)^T\mathbf{C}\left(\sum_{j=1}^{n}\mathbf{B}_j\mathbf{u}_j\right)d\Omega = \int_\Omega \delta\left(\sum_{i=1}^{n}\mathbf{\Phi}_i\mathbf{u}_i\right)^T\mathbf{f}^b\,d\Omega + \int_{\Gamma_t}\delta\left(\sum_{i=1}^{n}\mathbf{\Phi}_i\mathbf{u}_i\right)^T\mathbf{f}^S\,d\Gamma \tag{92}$$

It is now possible to identify the matrices of the system as:

$$\mathbf{K}_{ij} = \int_\Omega \mathbf{B}_i^T\mathbf{C}\mathbf{B}_j\,d\Omega \tag{93}$$

which is called nodal stiffness matrix. If n nodes fall inside the nodal domain, $\mathbf{K}_{ij}$ have dimensions $2n \times 2n$.
The nodal force vector is given by:

$$\mathbf{F}_i = \int_\Omega \mathbf{\Phi}_i^T\mathbf{f}^b\,d\Omega + \int_S \mathbf{\Phi}_i^T\mathbf{f}^S\,dS_\Omega \tag{94}$$

which will be a column vector of $2n$ rows.
The nodal parameter vectors, $\mathbf{u}_i = \{u_i \quad v_i\}^T$, can be collected in the global displacement parameter vector

$$\mathbf{U} = \begin{Bmatrix}\mathbf{u}_1\\ \mathbf{u}_2\\ \mathbf{u}_3\\ \vdots\\ \mathbf{u}_n\end{Bmatrix} \tag{95}$$

which if of length $2n$.
These matrices and vectors have to be defined for each node in the global domain and assembled to obtain the final set of equations governing the problem:

$$\mathbf{K}\mathbf{U} = \mathbf{F} \tag{96}$$

4.6 Equations for plates

The interpolation of the displacement field is now specialized to obtain the equations for plates modelled with two different theories: Kirchhoff and Mindlin. Laminated plates are also treated in this section.

4.6.1 Kirchhoff Plates

In this work, Kirchhoff plates are only analysed in bending. Hence, due to uncoupling phenomena between bending and axial behaviour for the material, the only displacement parameter is the transverse displacement w. The displacement field is then function of w only and can be written as:

$$\mathbf{u} = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{Bmatrix} -z\dfrac{\partial}{\partial x} & -z\dfrac{\partial}{\partial y} & 1 \end{Bmatrix}^{T} w \tag{97}$$

Variable w is then interpolated

$$w(x, y) = \sum_{i=1}^{n} \Phi_i u_i = \mathbf{\Phi}_i \mathbf{u}_i \tag{98}$$

and introduced in Eq.(88) thus obtaining a strain matrix in the form:

$$\mathbf{B}_i = \begin{bmatrix} -\Phi_{i,xx} \\ -\Phi_{i,yy} \\ -2\Phi_{i,xy} \end{bmatrix} \tag{99}$$

Note that for a plate loaded in bending, the constitutive law for an isotropic plate reads:

$$\mathbf{D} = \frac{Eh^3}{12(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{(1-\nu)}{2} \end{bmatrix} \tag{100}$$

where E is the Young's modulus, h is the thickness and ν is the Poisson's ratio.

4.6.2 Mindlin Plates

The Mindlin plate theory includes the effect of transverse shear deformations. The displacement field for a thick plate can be defined as a function of the transverse displacement w and the two rotations around axis x and y:

$$\mathbf{u} = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{Bmatrix} z\theta_x \\ z\theta_y \\ w \end{Bmatrix} \tag{101}$$

The interpolation of the generalized displacement components is written as:

$$\mathbf{u} = \begin{bmatrix} w \\ \theta_x \\ \theta_y \end{bmatrix} = \sum_{i=1}^{n} \begin{bmatrix} \Phi_i & 0 & 0 \\ 0 & \Phi_i & 0 \\ 0 & 0 & \Phi_i \end{bmatrix} \begin{bmatrix} w_i \\ \theta_{x_i} \\ \theta_{y_i} \end{bmatrix} = \sum_{i=1}^{n} \mathbf{\Phi}_i \mathbf{u}_i \tag{102}$$

where the three unknowns w, θ_x and θ_y are interpolated using the same shape functions.
The interpolation is plugged into Eq.(88), as usual. For Mindin plates though, the stiffness matrix is slightly more complex due to the contribution introduced by the shear term:

$$\mathbf{K}_{ij} = \int_{\Omega} \mathbf{B}_{b_i}^{T} \mathbf{D} \mathbf{B}_{b_j} \, d\Omega + \int_{\Omega} \mathbf{B}_{s_i}^{T} \mathbf{A}_s \mathbf{B}_{s_j} \, d\Omega \tag{103}$$

where $\mathbf{B}_b$ and $\mathbf{B}_s$ are the strain matrices for bending and shear, defined respectively as:

$$\mathbf{B}_{b_i} = \begin{bmatrix} 0 & \Phi_{i,x} & 0 \\ 0 & 0 & \Phi_{i,x} \\ 0 & \Phi_{i,y} & \Phi_{i,x} \end{bmatrix} \tag{104}$$

$$\mathbf{B}_{s_i} = \begin{bmatrix} \Phi_{i,x} & \Phi_i & 0 \\ \Phi_{i,y} & 0 & \Phi_i \end{bmatrix} \tag{105}$$

If n nodes are used for the interpolation, the two matrices above have dimensions $3 \times 3n$ and $2 \times 3n$ respectively. The nodal stiffness matrix has dimension $3n \times 3n$.
Moreover, the two constitutive matrices in Eq.(103) are the bending matrix $\mathbf{D}$ shown in Eq.(100) and the shear matrix $\mathbf{A}_s$ in the form:

$$\mathbf{A}_s = kGh \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{106}$$

where k is the shear correction factor and G is the shear modulus.

4.6.3 Laminated Plates

To describe laminated plates, the first order shear deformation theory is used. Thus, the displacement field is the same as in Mindlin plates. To address the any coupling effects between in-plane and out-of-plane behaviour, such as in the case of unsymmetric laminates, in-plane displacements are also considered

$$\mathbf{u} = \begin{Bmatrix} u \\ v \\ w \\ \theta_x \\ \theta_y \end{Bmatrix} \tag{107}$$

Even in this case, the same shape function is used to interpolate all the components in the field variable vector.
The nodal stiffness matrix is made of 5 contributions:

$$\begin{aligned} &\mathbf{K}_{ij} = \mathbf{K}_{ij}^{mm} + \mathbf{K}_{ij}^{bm} + \mathbf{K}_{ij}^{mb}\mathbf{K}_{ij}^{bb} + \mathbf{K}_{ij}^{ss} = \\ &= \int_\Omega \mathbf{B}_{m_i}^T \mathbf{A} \mathbf{B}_{m_j} \, d\Omega + \int_\Omega \mathbf{B}_{b_i}^T \mathbf{B} \mathbf{B}_{m_j} \, d\Omega + \int_\Omega \mathbf{B}_{m_i}^T \mathbf{B} \mathbf{B}_{b_j} \, d\Omega + \int_\Omega \mathbf{B}_{b_i}^T \mathbf{D} \mathbf{B}_{b_j} \, d\Omega + \int_\Omega \mathbf{B}_{s_i}^T \mathbf{A}_s \mathbf{B}_{s_j} \, d\Omega \end{aligned} \tag{108}$$

In Eq.(108), it can be defined:

$$\mathbf{B}_{m_i} = \begin{bmatrix} \Phi_{i,x} & 0 & 0 & 0 & 0 \\ 0 & \Phi_{i,y} & 0 & 0 & 0 \\ \Phi_{i,y} & \Phi_{i,x} & 0 & 0 & 0 \end{bmatrix} \tag{109}$$

as the membrane strain matrix (of size $3 \times 5n$),

$$\mathbf{B}_{b_i} = \begin{bmatrix} 0 & 0 & 0 & \Phi_{i,x} & 0 \\ 0 & 0 & 0 & 0 & \Phi_{i,y} \\ 0 & 0 & 0 & \Phi_{i,y} & \Phi_{i,x} \end{bmatrix} \tag{110}$$

as bending strain matrix (of size $3 \times 5n$), and

$$\mathbf{B}_{b_i} = \begin{bmatrix} 0 & 0 & \Phi_{i,x} & \Phi_i & 0 \\ 0 & 0 & \Phi_{i,y} & 0 & \Phi_i \end{bmatrix} \tag{111}$$

as shear strain matrix (of size $2 \times 5n$).
The material matrices of Eq.(108) came from the laminate constitutive equations [8]:

$$\begin{bmatrix} [\mathbf{A}] & [\mathbf{B}] \\ [\mathbf{B}] & [\mathbf{D}] \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} & B_{11} & B_{12} & B_{16} \\ A_{12} & A_{22} & A_{26} & B_{12} & B_{22} & B_{26} \\ A_{16} & A_{26} & A_{66} & B_{16} & B_{26} & B_{66} \\ B_{11} & B_{12} & B_{16} & D_{11} & D_{12} & D_{16} \\ B_{12} & B_{22} & B_{26} & D_{12} & D_{22} & D_{26} \\ B_{16} & B_{26} & B_{66} & D_{16} & D_{26} & D_{66} \end{bmatrix} \tag{112}$$

where

$$A_{ij} = \sum_{k=1}^{N} \bar{Q}_{ij}^{k} (z_{k+1} - z_k) \tag{113}$$

$$B_{ij} = \frac{1}{2} \sum_{k=1}^{N} \bar{Q}_{ij}^{k} (z_{k+1}^2 - z_k^2) \tag{114}$$

$$D_{ij} = \frac{1}{3} \sum_{k=1}^{N} \bar{Q}_{ij}^{k} (z_{k+1}^3 - z_k^3) \tag{115}$$

$$\tag{116}$$

4.6.4 Global stiffness matrix assembly

The matrix $\mathbf{K}_{ij}$ presented in these sections has been called nodal stiffness matrix since it is referred to one single node. The global stiffness matrix is obtained by assembling all the nodal contributions. To each node in the global domain is associated an index vector containing the field variables which are selected according to the model used to describe the structure. The assembly of the global stiffness matrix is not a mere sum of all the nodal contributions, it has to respect the index order.
Let's consider a plane problem and let's say that three nodes, 3 5 and 7, fall inside the local domain at the generic i-th step of the analysis. The vector of generalized displacements (of size $2n \times 1$) is given by:

$$\mathbf{u}_i = \{u_3 \quad v_3 \quad u_5 \quad v_5 \quad u_7 \quad v_7\}^T \tag{117}$$

The corresponding nodal stiffness matrix will be a $2n \times 2n$ matrix. To correctly assemble the global stiffness matrix it must always be considered how its dimension depends on the generalized displacement field. In this case, two generalised coordinates are used and the size of the problem is doubled. A vector of indexes can be computed to correctly perform the assembly. For the generic j node, these indexes can be computed as:

$$u_{index} = 2j - 1 \tag{118}$$

$$v_{index} = 2j \tag{119}$$

For the three nodes of this example, the contributions of the nodal stiffness matrix, once assembled in the global one, will occupy the rows and column indicated in the following vector:

$$\text{indexes} = \{5 \quad 6 \quad 9 \quad 10 \quad 13 \quad 14\} \tag{120}$$

The extension to cases in which different generalized nodal coordinates are used, follows the same principle.

5 Numerical implementation

Before digging into the analysis and show the results, it may be useful to have a look at the overview of the computer program built to solve a problem using a Mesh Free Method. All the codes presented in this work are developed in Matlab®.
The structure of the program can be divided into three sections, pre-process, matrix assembly and analysis and post-process, each one made of several routines some of which dependent on additional subroutines.

I. Pre-process

   a. Geometry generation

   b. Nodes generation

   The node generation step is where the two methods, PIM and RPIM, slightly diverge. It has been already explained how the moment matrix of the RPIM cannot have any singularity issue. This means that regularly distributed nodes is in enough for the RPIM. Hence, the generation of the nodal distribution is straightforward and just makes use of the `combvec` Matlab command.
   On the other hand, polynomial PIM suffers of singularity of the moment matrix for regular nodal distributions. As already explained, the nodes have to be perturbed and this is the stage at which this method differs from the RPIM in the pre-process. Once the regular nodal distribution has been generated, the line of command: `-p + (p+p)*rand(N)` is used to generate the vector of perturbations. Here, N is the number of nodes in the global domain and p is the magnitude of the perturbation. Note that the nodes on the boundaries are left untouched both to preserve a correct representation of the edges of the body and because it is not necessary to move them if the rest of the nodal distribution is already random.

   c. Support domain definition

   d. Background mesh generation

   e. Computation of Gauss points, weights and Jacobian

II. Matrix assembly and analysis

   a. Stiffness matrix construction

      - Construct shape functions and their derivatives

      Differently from FEM, in which the shape functions are known a priori, Mesh Free Methods require to construct them at each iteration. A loop is initiated on the Gauss points in which the support domain is centered. The nodes falling inside the domain are identified and the shape functions are constructed accordingly, as shown in Sections 3.1 and 3.4.

      - $\mathbf{B}$ matrices computation
      - Global stiffness matrix assembly

   b. Load vector construction

      - Construct shape functions and their derivatives
      - Global load vector assembly

   c. Boundary conditions imposition

Both PIM and RPIM possess the Kronecker delta function property. Hence, the essential boundary conditions can be imposed in the same way as in FEM codes, by deleting the rows and columns corresponding to the constrained degree of freedom.

d. System solution

III. Post process

a. Stress recovery

- Construct shape functions and their derivatives
- $\mathbf{B}$ matrices computation

b. Plot results

6 Results and discussion

In this chapter, the two Mesh Free Methods discussed earlier are used to perform the analysis of one and two dimensional structures. The results obtained are compared to the ones obtained with FEM or with the analytical solution, when available.

6.1 Cantilever beam

The first case deals with the analysis of a cantilever beam, modeled via membrane elements to address the in-plane bending response. Having known analytical solution, this case is often used as a benchmark for numerical methods. In this case, the study serves a double purpose. Not only it is used to validate Mesh Free Methods against the analytical solution, it has to prove that the implementation carried out in this work reflects the results available in the literature. In fact, Mesh Free solutions are available from [4] and [6] and are here used to validate the computer program results.
The cantilever beam here considered is subjected to a parabolic traction at the free end as shown in Figure 18. A plane stress problem is considered and all the data are available from Table 3.

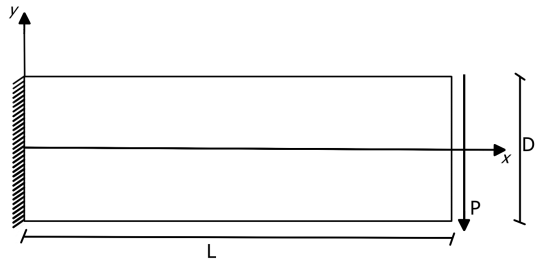


Figure 18: Cantilever beam with parabolic traction load

Name	Value
Length	$L = 48m$
Height	$D = 12m$
Thickness	$t =$unit
Load	$P = -1000N/m$
Young's modulus	$E = 3 \times 10^7 Pa$
Poisson's ratio	$\nu = 0.3$

Table 3: Cantilever beam data

The exact solution of the problem is [11]:

$$u(x,y) = -\frac{Py}{6EI}\left[(6L-3x)x + (2+\nu)\left(y^2 - \frac{D^2}{4}\right)\right] \tag{121}$$

$$v(x,y) = \frac{P}{6EI}\left[3\nu y^2(L-x) + (4+5\nu)\frac{D^2 x}{4} + (3L-x)x^2\right] \tag{122}$$

where I is the moment of inertia which, for a beam of unit thickness and rectangular cross-section, is given by:

$$I = \frac{D^3}{12} \tag{123}$$

The normal stress on the cross-section in the y direction and the shear stress on the cross section are, respectively:

$$\sigma_{xx}(x,y) = -\frac{P(L-x)y}{I} \tag{124}$$

$$\sigma_{yy}(x,y) = 0 \tag{125}$$

$$\tau_{xy}(x,y) = \frac{P}{2I}\left(\frac{D^2}{4} - y^2\right) \tag{126}$$

The analysis is performed both with polynomial PIM and radial PIM. In Ref.[6], the analysis is performed using RPIM only. The RPIM results are compred directly with these results, while serving the double purpose of assessing if the program written for the PIM is correct as well. The parameters to perform the analysis are shown in Table 4.

Nodes	175 (25×7)
RPIM	$\alpha_s = 3$ $C = 1.42$ $q = 1.03$
PIM	$\alpha_s = 2$ $p = 0.06\mathfrak{m}$

Table 4: Data for Mesh Free analysis

Note that among the PIM parameters and additional parameter p is considered. The need to alter the nodal distribution of the PIM has been assessed several times in the previous discussion. Referring to the MATLAB implementation of the code, see Section 5, it is of interest to consider the magnitude p of the perturbation as an additional parameter, to show how much it is necessary to alter an initially regular nodal distribution to avoid the singularity of the moment matrix $\mathbf{P}_Q$. On the contrary, RPIM directly uses evenly spaced nodes.
To performed the analysis, a 24×6 background integration mesh is used and, in each cell, 2×2 Gauss points are used for the evaluation of the integrals.
The analysis is performed using the polynomial PIM method first. The results reported in Ref.[6] and here used for comparison, develop the results for the RPIM only. Therefore, the PIM results, which are shown in Figure 19, are compared directly with the analytical solution. For this method the analysis is complicated by the usual singularity problem of the moment matrix. As shown in Figure 19, the deflection is in good agreement with the analytical results but the shear stress is completely out of trend. The reasons for this behaviour may be researched in the irregular nodal distribution or the continuity issues that characterises the method due to the "jump" of the moving domain from one Gauss point to the next.
To explain why the deflection in Figure 19 is compared with the analytical solution at the top

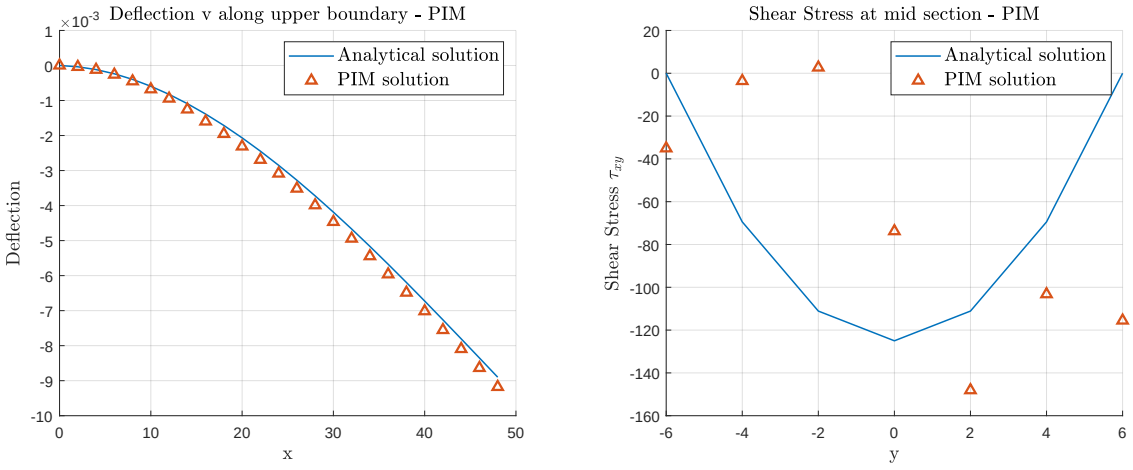


Figure 19: Deflection and Shear stress using PIM

boundary of the problem rather than the mid section, one should consider the altered nodal distribution characteristic of the PIM: due to the perturbation introduced, there is no longer a group of nodes clearly defining the mid section of the domain.
The results obtained with the RPIM are presented in two separate graphs. Figure 20 shows

the RPIM results of the problem against the analytical solution from Ref.[6] compared with the RPIM results obtained with the program developed for this work. On the other hand, Figure 21 shows the same comparison but in terms of shear stresses.

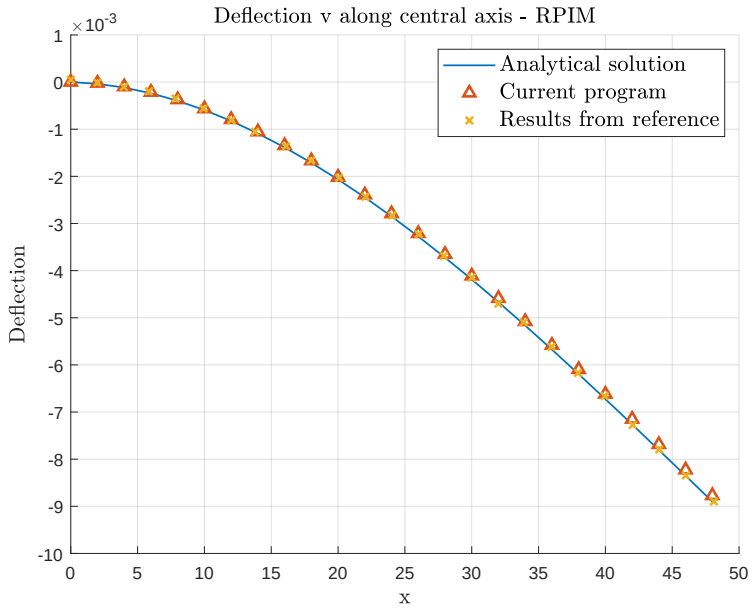


Figure 20: Deflection comparison between analytical solution, current program and results from Ref. [6]

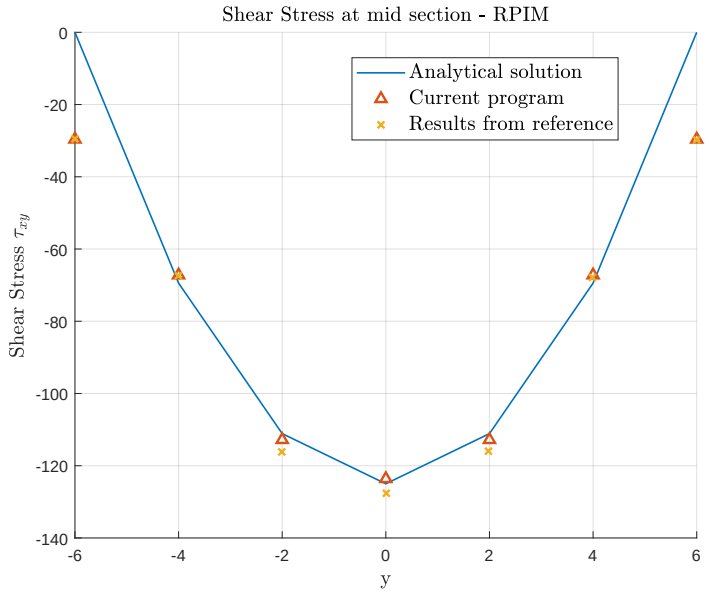


Figure 21: Shear stress comparison between analytical solution, current program and results from Ref. [6]

The results obtained are in good agreement with the analytical solution and mirror the ones obtained in Ref.[6], meaning that the development of the code is overall correct.

6.1.1 Effects of parameters

In this section, the practical effects of the variation of the parameters on which the methods depend are shown through a series of figures.
The polynomial PIM is mainly affected by the dimension of the support domain. Hence, different analysis are performed with different values of α_s. Figure 22 shows the results of the analysis for four different values of the parameter α_s, both in terms of deflection and shear stress. From the left plot of Figure 22, it can be seen how the error on the computation of the deflection increases as α_s changes. In particular, for the low value of 1.5 the error is the greatest and by imposing $\alpha_s = 1$ the program fails to complete the analysis as the moment matrix becomes singular.
Although the recovery of the shear stress with the PIM doesn't give good results, it was also shown on the right plot of Figure 22. Altering the parameter doesn't bring any improvements, it rather makes the results even worst.

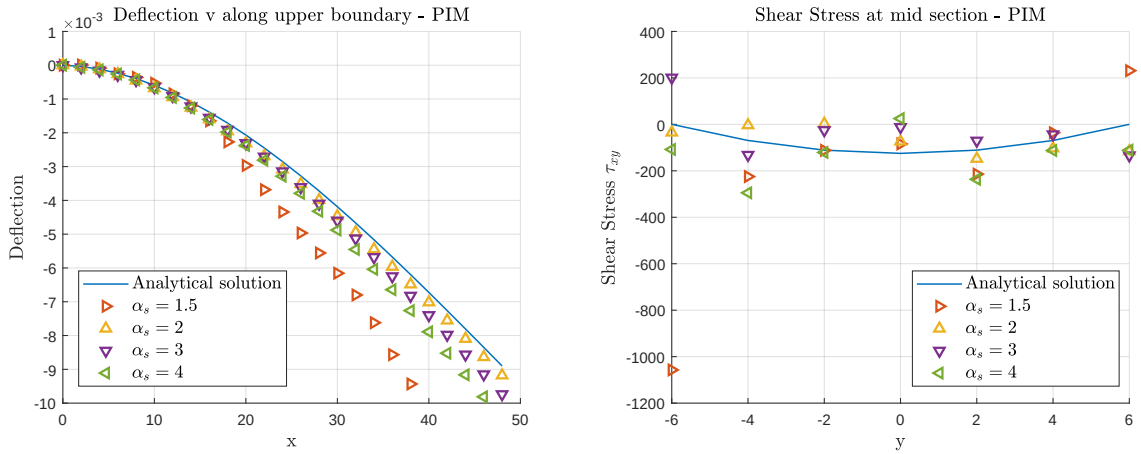


Figure 22: Deflection and shear stress of cantilever beam for different values of α_s.

Drastic alterations can also be observed in the case of the RPIM. Although depending on three non-dimensional parameters, variations in the values of C and q do not bring any significant change in the results. Changing parameter α_s, however, brings non negligible alterations as shown in Figure 23. The RPIM sensitivity for low values of the parameter α_s is greater than the PIM one. It can be observed in Figure 23 how for $\alpha_s = 1.5$ the solution both in terms of displacement and stresses, diverges significantly from the analytical solution. In any case, altering this parameter introduces non negligible errors in the results and this behaviour makes it clear how crucial the tuning of the parameters is in Mesh Free Methods and the issue that derives from such tuning.

In fact, the cases analysed in this work always had an analytical solution available in literature or, at least, a FEM simulation to make a comparison with. Once the code is validated, if the results are not correct all it takes is a trial and error procedure on the tuning of the parameters. If the problem is specifically linked to α_s, a visual checking is also an option. It has been explained, in fact, that continuity of these methods depends on the number of nodes falling in the overlapping areas of two neighbouring support domains. Namely, at least two nodes have to be found in this area. This is also a visual check that requires a trial and error procedure.
The real issue arises for problems in which there is no known solution, analytical or computed with FEM codes. In these cases, unless the results are obviously diverging from any reasonable prediction, it is not possible to asses whether the solutions obtained are correct or not. This is a matter of Mesh Free Methods that may be worth investigating in future works.

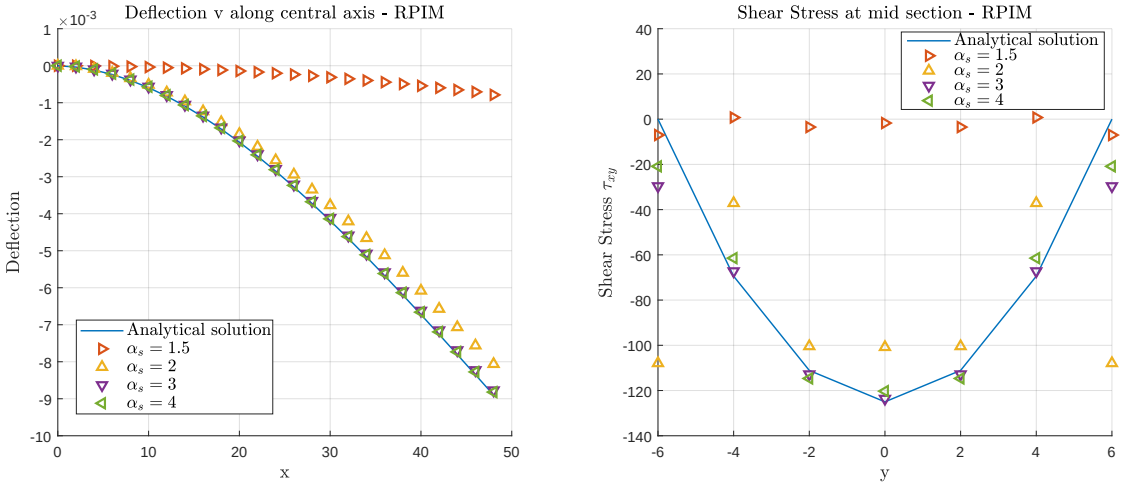


Figure 23: Deflection and shear stress of cantilever beam for different values of α_s.

6.2 Membrane with center hole

A membrane with a center hole, loaded in traction, is now analysed. Again, this is a plane stress problem and due to its double symmetry only a quarter of the structure is considered for analysis. The geometry of the problem and the data for the membrane are shown in Figure 24 and Table 5.

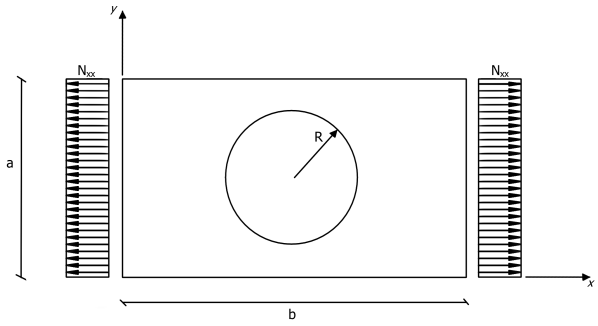


Figure 24: Membrane with center hole and traction load

Name	Value
Height	$a = 100mm$
Length	$b = 150mm$
Thickness	$t = 1mm$
Radius	$R = 25mm$
Load	$N_{xx} = 20N/mm$
Young's modulus	$E = 72 \times 10^3 Pa$
Poisson's ratio	$\nu = 0.3$

Table 5: Membrane data

Since only a quarter of the problem is considered for symmetry reasons, the nodes on the bottom boundary are considered constrained in the y direction while the ones on the side on the left are constrained in the x direction.
As done for the cantilever beam, both PIM and RPIM are used for the analysis and the relative data are available from Table 6. The results obtained with Mesh Free Methods are compared

Nodes	60
RPIM	$\alpha_s = 2$ $C = 1.42$ $q = 1.03$
PIM	$\alpha_s = 2.5$ $p = 0.06mm$

Table 6: Data for Mesh Free analysis

with a reference FEM code. To ensure the comparison to be consistent, in all three cases 60

nodes are used for the discretization of the structure. For the Mesh Free codes, the background mesh used for the integration is built using all of the 60 nodes. Figure 25 shows the nodal distribution and background mesh used to perform the analysis. Figure 25 is directly referred to the RPIM while for the PIM remember that the nodes are slightly perturbed.
In both cases, the integration is performed using the 44 cells shown in the Figure and 2×2 Gauss points for each cell.

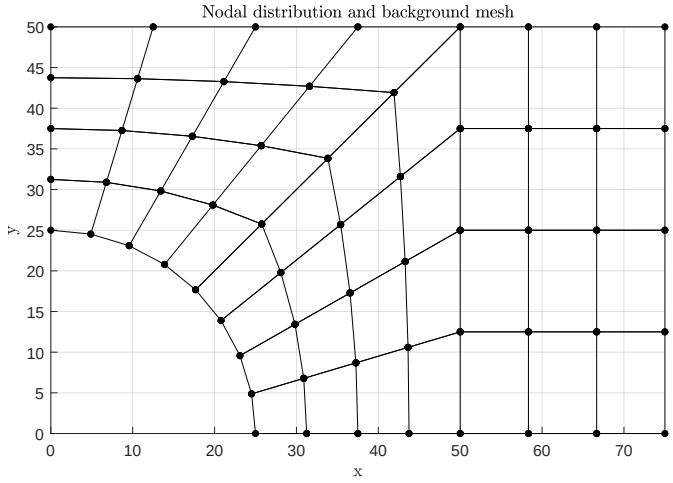
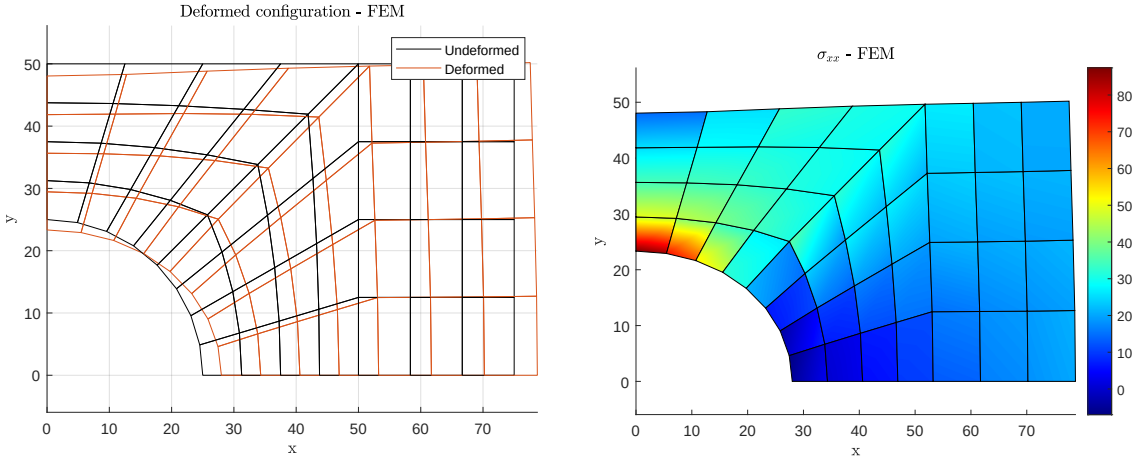


Figure 25: Nodal distribution and background mesh

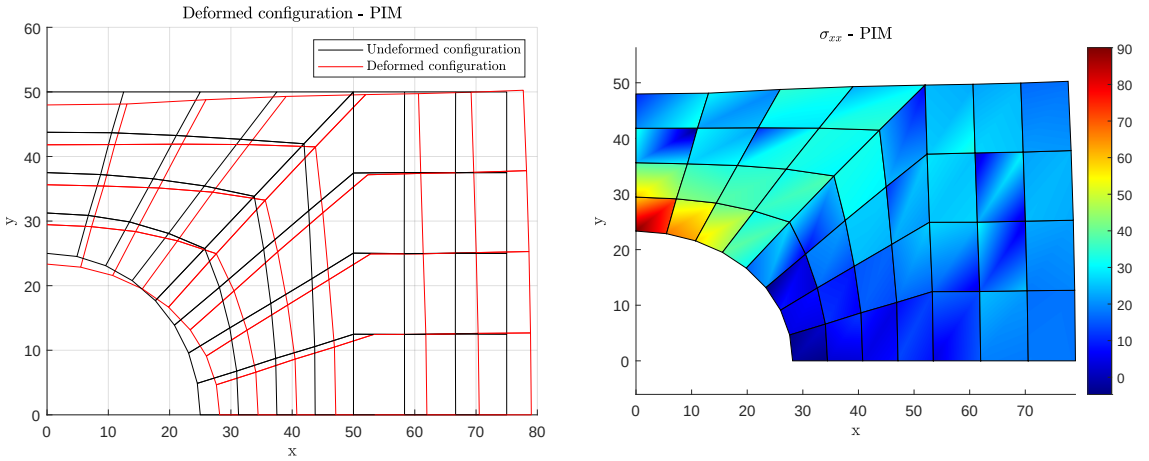
The results of the analysis are shown in Figures 27 and 28 in terms of deformed configuration and stress component σ_{xx}. The reference FEM results are shown in Figure 26.
The results show a similar behaviour to that observed in the cantilever beam case. The deformed configuration of both PIM and RPIM is in good agreement with the results obtained with the FEM code. On the other hand, the stress component σ_{xx} computed with PIM is not as accurate as it is in RPIM.



(a) Membrane deformed configuration with FEM program

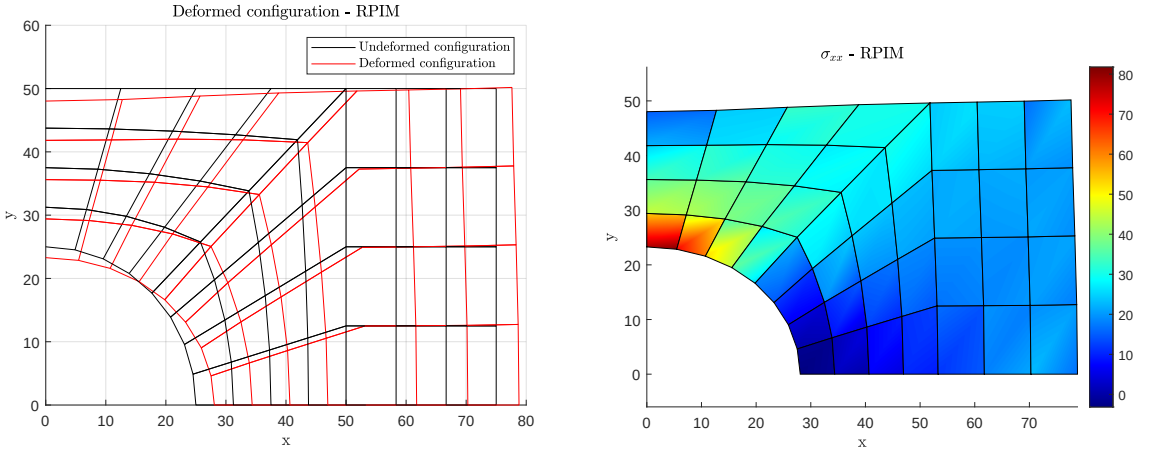
(b) Membrane axial stress with FEM program

Figure 26: FEM results



(a) Membrane deformed configuration with Mesh Free PIM program **(b)** Membrane axial stress with Mesh Free PIM program

Figure 27: Mesh Free PIM results

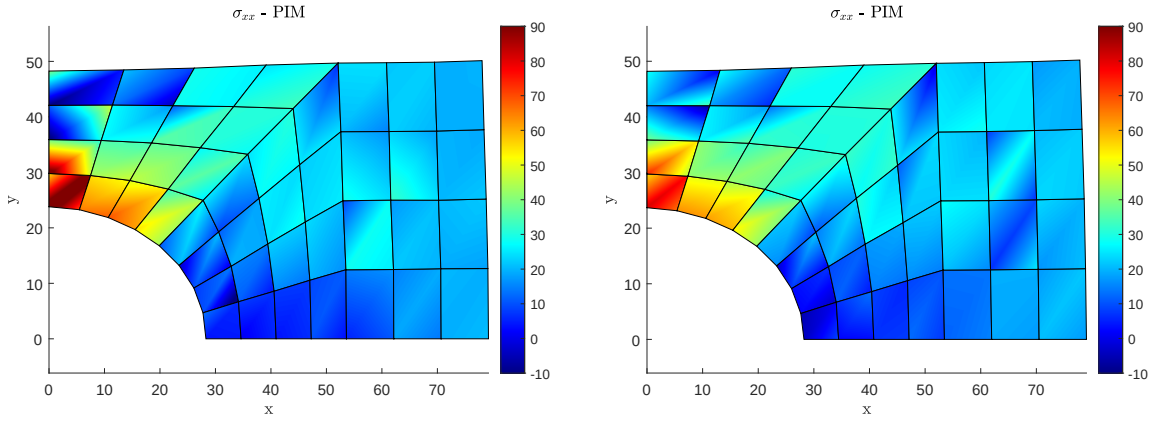


(a) Membrane deformed configuration with Mesh Free RPIM program

(b) Membrane axial stress with Mesh Free RPIM program

Figure 28: Mesh Free RPIM results

Moreover, there is always another aspect to consider when dealing with polynomial PIM. As discussed in Section 5, the command `rand` is used every time the program is launched to generate the irregular nodal distribution. This means that every time the program is launched, it works with a different nodal distribution than the previous launches. As the nodal distribution varies, even considering a parameter p as small as indicated in Table 6, the results obtained are different. Not so much concerning the deformed configuration but certainly in the computation of the stress. Two examples of two successive launches of the program are shown in Figure 29. Note that no modifications to the parameter p nor any other variable in the program has been performed to obtain the results in Figures 29. The differences in the results are only due to the different nodal distribution, as generated by the command `rand`, for that particular launch.
Although the trend of the stress is not disrupted, it still doesn't resemble enough the results obtained with the FEM program. To improve the results and try to make the program more stable, different solutions have been applied. Radical changes in the nodal distribution and variations of the parameter α_s seemed the most promising but both gave the same outcome. In almost all cases the moment matrix $\mathbf{P}_Q$ becomes singular. Even if it doesn't, the program



(a) Case 1 **(b)** Case 2

Figure 29: Mesh Free PIM stress results from two subsequent executions of the program

shows several warning on the ill conditioning and very small determinant of $\mathbf{P}_Q$ and exhibits results, both in terms of deformations and stresses, which are completely wrong.
An additional attempt to make the PIM program more robust to changes in the nodal distribution was to alter the background mesh used for the integration. As explained, the support domain in this work is centered in the Gauss points, not the nodes. Varying the background mesh, the number and position of the Gauss points changes accordingly. The α_s parameter is then tuned to accommodate the changes and to make sure that a number of nodes between 9 and 15 are used for the interpolation. Attempt with both finer and coarser background meshes have been made, but still singularity issues, warnings and wrong results occurred.
When properly tuned, the PIM gives results that are even more accurate than the ones obtained with the RPIM and all without depending on so many parameters. It may be worth investigating further the reasons why it fails and, if possible, to make it more robust to changes of the domain nodal representation. In this work, however, the problems encountered with the PIM suggested to proceed in the next analysis using the RPIM only. Hence, although the value and power of the PIM is fully recognised, further work is needed on it. Without need to specify it every time, all the results shown from the next chapter on are obtained using only the Radial Point Interpolation Method (RPIM).

6.3 Rectangular plates

In this section, a simple plate will be analysed in bending, buckling or free vibrations and modelled with the different theories shown in the previous sections. The properties of the plate are summarized in Table 7 while a plate representation is shown in Figure 30.

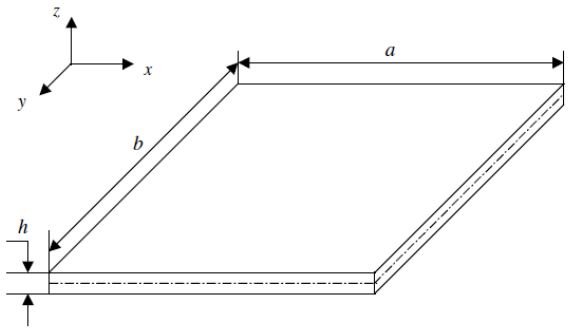


Figure 30: Rectangular plate

Name	Value
Height	$\mathtt{a} = 1\mathtt{m}$
Length	$\mathtt{b} = 1\mathtt{m}$
Young's modulus	$\mathtt{E} = 10920\mathtt{Pa}$
Poisson's ratio	$\nu = 0.3$

Table 7: Plate data

6.3.1 Kirchhoff isotropic plate

The first example considered is a simply-supported (SSSS) square plate loaded with a uniform transverse pressure. In this case, the plate has a thickness $\mathtt{t} = 0.01\mathtt{m}$. The results are compared in terms of non-dimensional transverse displacement:

$$\bar{w} = w\frac{D}{pa^4}10^2 \tag{127}$$

where the bending stiffness D is written as:

$$D = \frac{Et^3}{12(1-\nu^2)} \tag{128}$$

For this analysis, the parameters characterising the RPIM method are set as $C = 1.42$, $q = 1.03$, $\alpha_s = 1.8$.
The results are compared with the exact solution from Timoshenko and Woinowsky-Krieger, see Ref.[10], and the FEM solution reported in Ref.[2]. Different number of nodes are considered and the FEM analysis makes use of Q4 elements. The results are shown in Table 8.

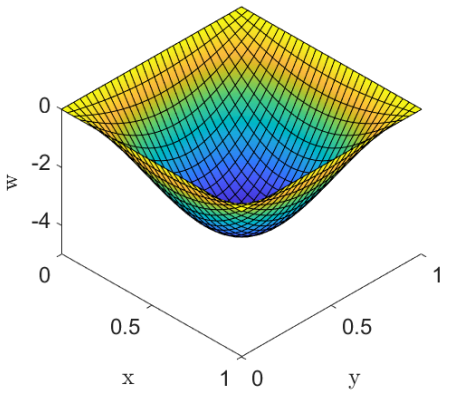


Figure 31: RPIM result for deflection of SSSS Kirchhoff plate (30×30 nodes)

Nodes	FEM	RPIM
5×5	0.3976	0.4287
10×10	0.4136	0.3986
20×20	0.4081	0.4044
30×30	0.4071	0.4054
Exact	0.4062	

Table 8: Mesh Free vs FEM results comparison

It is interesting to notice how Mesh Free methods grants more accuracy of the solution already with a 20×20 nodes mesh, while FEM codes needs at least 30×30 nodes mesh to achieve good results. Table 9 shows the percentage error on the results in Table 8:

Nodes	RPIM error	FEM error
5×5	5.5391%	2.1172%
10×10	1.8710%	1.8218%
20×20	0.4431%	0.4677%
30×30	0.1969%	0.2216%

Table 9: Percentage error on analysis results

Using Q4 elements the solution as computed using FEM does not converge with the nodal distribution used in Ref.[2]. Nor does the RPIM using the same nodal distribution. Still, as seen in Table 9, the RPIM becomes more accurate then FEM for a 20×20 nodal distribution and the gap between the error computed for the two methods increases as the number of nodes increases: both values of the error decreases as the numerical solution approaches the exact one, but the numerical results of RPIM remain the most accurate.
A slight variation of this example consists in considering a point load ($\mathsf{P} = 100\mathsf{N}$) applied at the mid node. This case is interesting to observe once again the influence of the parameters on the Mesh Free RPIM. The values of the distributed pressure case are maintained except for the value for parameter C. The comparison is performed on the non-dimensional deflection of the mid point, $\beta = w_{max}\mathsf{D}/\mathsf{P}a^2$, and the exact solution is given in Ref.[10].
As shown in Table 10, keeping the same value of C as in the previous case yield good results but

C	Nodes	β
1.42	15×15	0.0118
1.42	19×19	0.0118
0.5	15×15	0.0117
0.5	19×19	0.0116
Exact		0.0116

Table 10: Effects of parameters on non-dimensional deflection

the solution does not convergence to the exact one. Changing the number of nodes, although of just 4, doesn't bring any significant improvement. On the other hand, using C= 0.5 gives better results with less nodes and converges when 19×19 nodes are used. This is a case in which using a non optimal tuning of the parameters doesn't alter the results but still doesn't allow to exploit the full potential of the method. In a case in which the exact solution would have not been available, with this numbers, it would have been quite hard to realise that a better solution was achievable. This is another example of the major drawback of RPIM Mesh Free Method.

6.3.2 Orthotropic plate

An additional variation is to consider an orthotropic plate made of a single ply oriented at 0°. The geometrical data of the plate is given in 7. While the material properties are shown in Table 11. Once again, the results are compared in terms of non-dimensional central deflection:

$$\bar{w} = w\frac{\mathsf{D}_{12} + 2\mathsf{D}_{66}}{\mathsf{p}a^4}10^3 \tag{129}$$

For this example, the tuning of the RPIM parameters proved to be quite cumbersome up to the point that no optimal combination of parameters was found and several warnings about

Property	Value
E_1	31.8MPa
E_2	1.02MPa
$\boldsymbol{\nu}_{12}$	0.31
G_{12}	0.96MPa

Table 11: Material properties

the small determinant of the moment matrix $\mathbf{R}_{\mathrm{Q}}$ were given. In fact, remember that RPIM solves the problem of the singularity of the moment matrix concerning the nodal distribution, but cases in which the determinant is extremely small are still possible and entirely due to the parameters selection.
The FEM analysis makes use of non conforming elements and the results of the analysis are given in Table 12.

Method	$\bar{w}$
FEM [2]	0.9213
RPIM	0.9201
Exact [7]	0.9225

Table 12: Orthotropic plate bending results

Mindlin plate: isotropic case

In this section, an isotropic square plate whose dimensions are given in Table 7, is analysed while subjected to different boundary conditions. The plate is analysed in bending, free vibrations and buckling and results are compared with the available exact solution and the FEM results found in Ref.[2]. The results of the FEM analysis are obtained using Q4 elements and both the FEM and Mesh Free background integration use 2×2 Gauss points. Moreover, a shear correction factor k= 5/6 is used for the analysis.
The bending case is still compared in terms of non-dimensional mid deflection as expressed in Eq.(127) but this time the factor 10^2 is not considered.

Thickness	Nodes	FEM	RPIM
0.1	20×20	0.001503	0.001490
	30×30	0.001503	0.001501

Table 13: Non-dimensional transverse displacements for clamped (CCCC) boundary conditions and RPIM parameters C= 0.5, q= 1.03. $\alpha_{\mathrm{s}} = 2$.

Thickness	Nodes	FEM	RPIM	Exact [2]
0.1	20×20	0.004270	0.00458	
	30×30	0.004271	0.00458	0.004270
0.0001	20×20	0.004059	0.002549	
	30×30	0.004060	0.003985	0.004060

Table 14: Non-dimensional transverse displacements for clamped (SSSS) boundary conditions and RPIM parameters C= 0.5, q= 1.03. $\alpha_{\mathrm{s}} = 2$.

For the CCCC case, shown in Table 13, the exact solution is not available. Hence, the RPIM results are compared with the FEM results from Ref. [2]. In this case, the two methods are in

good agreement while the SSSS case, shown in Table 14, is not that satisfactory. While FEM seems to converge for both values of thickness, RPIM shows poor results for a larger thickness. For a small value of thickness, the results are still in good agreement with the analytical solution but do not reach convergence. In both cases, variations of the parameters do not bring any significant improvement to the results. The deflection computed with the RPIM program with both simply-supported and clamped boundary conditions are shown in Figure 32.

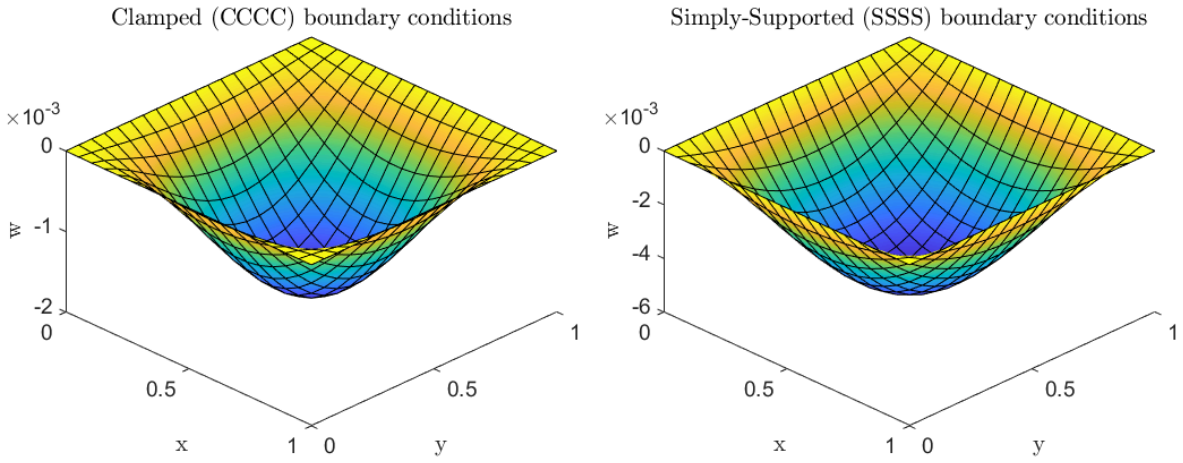


Figure 32: Mindlin plate deflections for different boundary conditions (20×20 mesh)

The next analysis performed is the study of the free vibration of the same Mindlin plate discussed so far. The results are compared in terms of non-dimensional natural frequency given by:

$$\bar{\omega} = \omega_{mn} a \sqrt{\frac{\rho}{G}} \tag{130}$$

where ρ is the material density set to $1 kg/m^3$ and G is the shear modulus. The indices m and n represent the vibration half waves along the two axes x and y.
As done for the previous comparisons, Q4 elements are used for the FEM mesh and both methods use 2×2 Gauss integration points. The results of the analysis are shown in Tables 15 and 16.

Thickness	Nodes	FEM [2]	RPIM	Ref [1]
0.01	20×20	0.1765	0.1759	
	25×25	0.1761	0.1755	0.1754

Table 15: Non-dimensional natural frequency for clamped (CCCC) boundary conditions with k= 0.8601 and RPIM parameters C= 0.5, q= 1.03. $\alpha_s = 2.5$.

Thickness	Nodes	FEM	RPIM	Ref [1]
0.01	20×20	0.0965	0.0962	
	25×25	0.0965	0.0960	0.0963

Table 16: Non-dimensional natural frequency for simply supported (SSSS) boundary conditions with k= 0.833 and RPIM parameters C= 0.5, q= 1.03. $\alpha_s = 2.5$.

An additional comparison is made with the analytical solution given by Mindlin, see Ref. [3], for the first four frequencies of the system. The results are obtained for a SSSS plate with

$t = 0.01$, $k = 0.833$ and a 20×20 nodal distribution and are shown in Table 17.

Mode	FEM (Q4 [2])	RPIM	Mindlin [3]
1	0.0965	0.0962	0.0963
2	0.2430	0.2411	0.2406
3	0.2430	0.2411	0.2406
4	0.3890	0.3843	0.3847

Table 17: Non-dimensional natural frequency for simply supported (SSSS) boundary conditions with k= 0.833 and RPIM parameters C= 0.75, q= 1.03. $\alpha_s = 2.3$.

Differently from the bending case, the results of the free vibration analysis with Mesh Free methods are more precise than FEM ones with both sets of boundary conditions. The percentage error on the two analysis are reported in Table 18 and 19.

Nodes	Boundary Conditions	FEM error	RPIM error
20×20	CCCC	0.6271%	0.2851%
25×25	CCCC	0.3991%	0.0570%
20×20	SSSS	0.2077%	0.1038%
25×25	SSSS	0.2077%	0.1038%

Table 18: Percentage error on plate free vibrations with exact results from Ref.[1]

Mode	FEM error	RPIM error
1	0.2077%	0.1038%
2	0.9975%	0.2078%
3	0.9975%	0.2078%
4	1.1178%	0.1040%

Table 19: Percentage error on plate free vibrations with exact results from Ref.[3]

Similar results are also obtained with a buckling analysis under uni-axial initial stress and 10×10 nodes. The buckling factor used for comparison is written as:

$$\bar{\lambda} = \frac{\lambda a^2}{\pi^2 D} \tag{131}$$

Thickness	FEM [2]	RPIM	Exact [3]
0.001	4.0897	4.0902	4.0001
$0.001_{(21\times21)}$		4.0001	4.0001
0.05	4.0153	3.8471	3.944
0.1	3.8097	3.5218	3.786

Table 20: Buckling factor for simply supported (SSSS) boundary conditions with k= 0.8601 and RPIM parameters C= 0.75, q= 1.03. $\alpha_s = 2.5$.

In Table 20, the computations for a $t = 0.001$ have been also performed by using 21×21 nodes. The results are shown to converge to the exact solution. The same has been done for the other values of thickness but no significant improvements were observed.
Once again, RPIM shows good agreement with the exact solution although the accuracy of the results seems to degrade as the thickness increases.

6.4 Mindlin plate: laminated case

Mindlin theory is now applied to different kinds of laminates, starting with the so-called Srinivas (Ref. [9]) problem in which the analysis is performed on a simply supported laminated sandwich 3-layer square plate under uniform pressure load. The material has the properties shown in the matrix below and the relationship between properties of the core and skins is given as $\bar{Q}_{skin} = R\bar{Q}_{core}$.

$$\bar{Q}_{core} = \begin{bmatrix} 0.999781 & 0.231192 & 0 & 0 & 0 \\ 0.231192 & 0.524886 & 0 & 0 & 0 \\ 0 & 0 & 0.262931 & 0 & 0 \\ 0 & 0 & 0 & 0.266810 & 0 \\ 0 & 0 & 0 & 0 & 0.159914 \end{bmatrix} \tag{132}$$

The thickness is taken as $t/10$ for the skins and $4t/5$ for the core, with $t = 0.1m$.
The results are compared in terms of non-dimensional displacement of the mid point of the whole plate and axial and shear stresses of the mid point of each layer.
The FEM results, see Ref. [2], are obtained with a mesh made of 10×10 nodes and Q4 elements are used. The results of the two methods are shown in Table 21.

Method	$\bar{w}$	$\bar{\sigma}_x^{(1)}$	$\bar{\sigma}_x^{(2)}$	$\bar{\sigma}_x^{(3)}$	$\bar{\tau}_{xz}^{(1)}$	$\bar{\tau}_{xz}^{(2)}$
Analytical [9]	258.97	60.35	46.62	9.34	4.36	3.27
FEM	259.30	58.44	46.75	9.35	2.98	14.92
RPIM	258.53	57.65	46.12	9.22	3.38	16.91

Table 21: Sandwich plate results for R= 5 and using 11×11 nodes. The superscripts in the stresses refer to the layer of the plate.

The parameters used for the RPIM are C= 0.75, q= 1.03, α_s = 1.3. The non-dimensional displacement and the stresses are in good agreement with the analytical and FEM results, some of them being even more precise than the latter. The only result that largely differs from the analytical solution is the shear stress in the second layer. But the results are still consistent with the FEM solution which leads to believe that it may be an issue linked to the modelling of the problem rather than an error in the program.

The last case shown in this sections is the buckling analysis of a symmetric $(0/90)_s$ laminate under uniform compression. For this example, there is no FEM solution available but the analytical solution by Reddy, see Ref [8], is used as a term for comparison. The information on the material, as provided in Ref. [8], are shown in Table 22:

Property	Value
E_1/E_2	Variable
G_{12}	$0.5E_2$
G_{12}	$0.5E_2$
G_{23}	$0.2E_2$
ν_{12}	0.25

Table 22: Reddy Material properties

The results are expressed in terms of non-dimensional critical load, written as:

$$\bar{N} = \frac{N_0 b^2}{\pi^2 D_{22}} \tag{133}$$

For this example, the thickness is fixed as $t = 0.01$ and different ratios a/b and E_1/E_2 are considered.

As shown in Table 23, once again the program manages to achieve results in good agreement

a/b	$E_1/E_2 = 5$		$E_1/E_2 = 10$	
	RPIM	Ref.[8]	RPIM	Ref.[8]
0.5	13.789	13.900	17.879	18.126
1.0	5.632	5.650	6.324	6.347
1.5	5.191	5.233	5.240	5.277

Table 23: Laminate results with a 21×21 nodal distribution. The parameters used for the RPIM are C= 0.75, q= 1.03, $\alpha_s = 1.8$.

with the known available ones with percentage errors given in Table 24.

a/b	$E_1/E_2 = 5$	$E_1/E_2 = 10$
0.5	0.7986 %	1.3627 %
1.0	0.3186%	0.3624 %
1.5	0.8026 %	0.7012%

Table 24: Percentage error on buckling analysis

6.5 Mindlin plate: Variable stiffness case

For this final section, the case of variable stiffness laminates is presented. Differently from the classical straight fiber laminates, variations of the stiffness properties increases the number of variables in the design, thus allowing for wider tailoring capabilities, increasing the performance of these materials. It could be interesting to test the performance of the new numerical methods investigated in this work on a topic of increasing interest as variable stiffness panels.
The geometrical data for the plate are given in Table 7 and the properties of the material, as given in Ref.[12], are reported in Table 25

Property	Value
E_{11}	173000 MPa
$E_{22} = E_{33}$	7200 MPa
$G_{12} = G_{13} = G_{23}$	3760 Mpa
$\nu_{12} = \nu_{13} = \nu_{23}$	0.29
ρ	1540 kg/m^3

Table 25: Material properties for variable stiffness panel

The peculiarity of these laminates lays in the variation of their stiffness not only along the thickness but also in one of the two directions x and y. The case analysed in this section, studies the free vibrations of a laminate with fiber variation along the x axis only. The orientation angle, function of the x coordinate, is expressed as (Ref. [12]):

$$\theta(x) = \frac{(T_1 - T_0)}{a}|x| + T_0 \tag{134}$$

where T_0 and T_1 are the orientation angles of the fibers at the center of the panel and at the panel edge respectively and $|x|$ is the absolute value of the x coordinate of the current node (or

Gauss point).
The results are compared in terms of non-dimensional frequencies, defined as:

$$\bar{\omega} = \omega \frac{a^2}{t} \sqrt{\frac{\rho}{E_{22}}} \tag{135}$$

and the laminate considered is simply-supported and characterised by a lay-up of $[\langle 0|45\rangle_x, \langle -45|-60\rangle_x, \langle 0|45\rangle_x]$.
The Mesh Free analysis is implemented using a 25×25 regular nodal distribution and 2×2 Gauss points for the integration. The RPIM parameters are tuned as q= 1.03 and $\alpha_s = 1.8$ while C is set as 0.75 when $t = 0.01$ and 1.42 when $t = 0.1$.
The results, shown in Table 26 are compared with analysis performed with different methods, as indicated in Ref.[12] and two values of thickness are considered.

Thickness	Method	Modes			
		1	2	3	4
t=0.001	p-version Ref. [12]	16.5794	27.2818	44.4149	49.7264
	LGDQ Ref. [12]	16.5198	27.2333	44.4554	49.7264
	ED2 Ref. [12]	16.5234	27.2558	44.4729	49.6648
	RPIM	17.0706	28.0863	45.3679	50.8938
t=0.1	p-version Ref. [12]	13.5724	21.6825	32.3781	33.8731
	LGDQ Ref. [12]	13.1984	21.3644	32.2639	33.9147
	ED2 Ref. [12]	13.8819	22.3681	33.7378	35.2302
	RPIM	13.3548	21.3825	32.9206	34.0433

Table 26: Results for variable stiffness panel

The evaluation of the error of the RPIM is performed first against the two methods p-version and LGDQ from Ref. [12], as seen in Table 27. In Table 28 the variable kinematic approach, as developed in [12], and the RPIM are compared in accuracy with respect to the LGDQ method.
From the computation of the error it is clear how the accuracy of the RPIM is not as high as

Thickness	Method	Modes error			
		1	2	3	4
t=0.001	p-version	2.9627%	2.9489%	2.1457%	2.3476%
	LGDQ	3.3342%	3.1322%	2.0526%	2.3476%
t=0.1	p-version	1.6033%	1.3836%	1.6755%	0.5025%
	LGDQ	1.1850%	0.0847%	2.0354%	0.3792

Table 27: Percentage error comparison

in previous cases even if, in the worst case scenario the error value lays around 3%. Table 28 in particular, shows how the accuracy of the results drops with respect to the ED2 method as the thickness decreases while achieving even better results than the latter for thicker panels.

Thickness	Method	Modes error			
		1	2	3	4
t=0.001	ED2	0.34 %	0.10 %	0.09 %	0.12%
	RPIM	3.3342%	3.1322%	2.0526%	2.3476%
t=0.1	ED2	2.28%	3.16%	4.20%	4.01%
	RPIM	1.1850%	0.0847%	2.0354%	0.3792

Table 28: Error comparison with results from Ref. [12]

7 Conclusions and Remarks for future improvements

A selection of linear elasticity problems have been analysed using Mesh Free Methods. A computer program implementing polynomial Point Interpolation Method and Radial Point Interpolation Method has been developed and the results of the analysis have been compared with analytical solutions when available and conventional FEM.
The absence of a mesh made the representation of the problems, by means of the nodal distribution, extremely easy and flexible to changes and alterations. Moreover, the characteristic typical of PIM and RPIM of possessing the Kronecker delta function property allowed the imposition of the essential boundary conditions in the same way as in traditional FEM programs.
A preliminary validation of the Methods showed the efficiency of the developed computer program and the limitation of the individual methods.
The Polynomial Point Interpolation Method is an extremely powerful tool to use in structural analysis. Its shape functions and their derivatives are simple polynomials and possess the Kronecker delta function property. Having no dependence on any parameter, the PIM possess a huge potential to be exploited, for the solution accuracy will only depend on the nodal distribution and the number of nodes in each support domain. All the power of the method is limited by the fact that a solution to the issue of the singular moment matrix must be developed first. This problem led to abandon the Polynomial PIM after the verification study. Moreover, stress recovery using PIM proved to undergo numerical issues demanding for further research studies.
The Radial Point Interpolation Method is an equally powerful tool that also solves the limitations of the polynomial PIM in terms of Moment Matrix singularity and stress recovery. The issue, in this case, is that the method is strongly limited by its dependence on the non-dimensional parameters, whose tuning has been proved to be a critical aspect of the method. Three main values of the C parameter, 0.5, 0.75 and 1.42, have been identified in this work that are able to give very good results. Similarly, values of the parameter α_s between 1.8 and 3 have been proven to be extremely effective. Having an analytical solution available, accurate solution can be obtained by trial and error of different combinations of these values. Real problems may arise when the solution is not available a priori and there is no way to know if the results of the methods are correct. Moreover, while a certain combination of parameters may give good results, changing one of the tuning may grant better convergence. But this is hard to asses if no reference solution is available.
Overall, Mesh Free methods have proven to be a valid alternative to FEM codes, providing results in good agreement with both the latter and the analytical solution and sometimes granting even greater accuracy than FEM itself. It is clear that further investigation is required on how the methods work and to understand how to improve their performance. Concerning the Polynomial PIM, developing a technique to prevent the moment matrix to become singular would virtually solve all the issue of the method. For the RPIM case, an optimization of the parameters is certainly needed as well as a better understanding of their effect on the solution of the analysis.

Bibliography

[1] D.J. Dawe and O.L. Roufaeil. "Rayleigh-Ritz Vibration Analysis of Mindlin Plates". In: *J. Sound and Vibration* 69.3 (1980), pp. 345–359.

[2] A. J. M. Ferreira and N. Fantuzzi. *MATLAB Codes for Finite Element Analysis - Solids and Structures*. Springer, 2020.

[3] E. Hinton. *Numerical Methods and Software for Dynamic Analysis of Plates and Shells*. Pineridge Press, 1988.

[4] G. R. Liu. *Mesh Free Methods - Moving Beyond the Finite Element Method*. CRC Press, 2002. ISBN: 0849312388.

[5] G. R. Liu and Y. T. Gu. "A Point Interpolation Method for Two-Dimensional Solids". In: *Int. J. Numerical Methods in Engineering* (2001).

[6] G. R. Liu and Y. T. Gu. *An Introduction to Meshfree Methods and Their Programming*. Springer, 2005.

[7] J.N. Reddy. *An Introduction to the Finite Element Method*. 3rd ed. McGraw-Hill International Editions, 2005.

[8] J.N. Reddy. *Mechanics of Laminated Composite Plates and Shells*. CRC Press, Boca Raton, 2004.

[9] S. Srinivas. "A Refined Analysis of Composite Laminates". In: *J. Sound and Vibration* 30 (1973), pp. 495–507.

[10] S. Timoshenko and S. Woinowsky-Krieger. *Theory of Plates and Shells*. McGraw-Hill, 1989.

[11] S. P. Timoshenko and J. N. Goodier. *Theory of Elasticity*. 3rd ed. McGraw-Hill, 1970.

[12] R. Vescovini and L. Dozio. "A Variable-Kinematic Model for Variable Stiffness Plates: Vibration and Buckling Analysis". In: *J. Composite Structure* 142 (2016), pp. 15–26.