

**DESCRIPTION OF THE
MESHLESS LOCAL PETROV-GALERKIN
APPROACH**

Annamaria Mazzia

Padova, Marzo 2005

Rapporto Tecnico N° 97

Abstract

In this paper we describe the Meshless Local Petrov-Galerkin (MLPG) method and its numerical implementation for the solution of elliptic problems.

1 Introduction

Recently, meshless methods have attracted some attention due to their flexibility in solving several engineering problems, especially with reference to discontinuities or moving boundaries. Among these, the Meshless Local Petrov-Galerkin (MLPG) method [2, 3] has the feature of being "truly" meshless, as it does not require "finite element" or "boundary element mesh" and, for this reason, it appears to be more flexible and easier to use when dealing with non-linear problems than the conventional FE or other meshless methods.

The weight function in the Moving Least Square (MLS) approximation is also employed as a test and trial function in the local symmetric weak form (LSWF) that subdivides the original domain into local regularly shaped subdomains over which particular integrals are calculated thus leading to a discretized system of equations. The final step is the application of the MLS approximation to the solution vector of the algebraic system in order to calculate the value of the unknown variable and its derivatives.

A crucial point for a good MLPG performance is the selection of the size of the local domains with the aim at preserving the local feature of the MLS approximation. Several examples from the Laplace and Poisson equations show that the choice of the parameter set in the MLPG implementation is not so straightforward.

The present paper is organized as follows. In the next Section the MLPG formulation is described in its fundamental steps. Some preliminar definitions of spatial domains useful in the construction of the method are then given. Next, the MLS approximation is presented with special attention on the description of the weight functions. Then, the local symmetric weak form that characterizes the MLPG is analyzed and the final system of equation to be solved is given. At this point the steps to numerical implement the MLPG are presented. Numerical examples illustrate the implementation and convergence of the MLPG described in the paper. Finally, the appendix suggests some notes about the numerical quadrature and application of the MLPG in elasto-statics problems.

2 Meshless Local Petrov-Galerkin formulation

The Meshless Local Petrov-Galerkin method can be described by the analysis of the following steps:

- the moving least squares approximation (MLS) scheme;
- the choice of the weight function that is to be used as test and trial function (observe that we can use two different functions as test and trial function [1]: in the present work we present the first and most simple approach of the MLPG);
- the local symmetric weak form (LSWF) of the original problem on local subdomains;

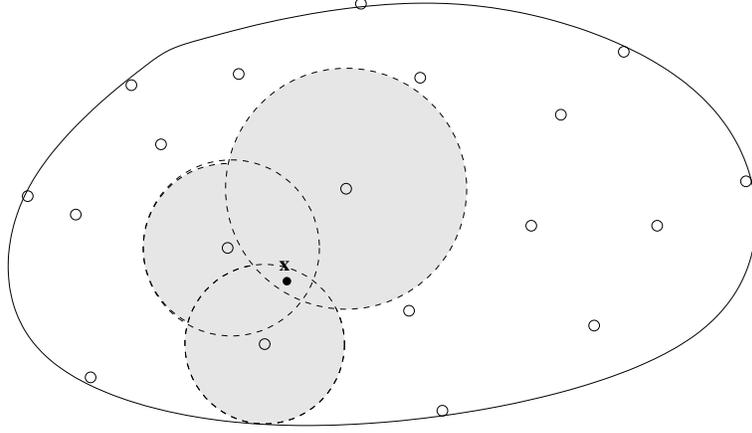


FIGURE 1: *Domain of definition of \mathbf{x} as union of supports of nodes \mathbf{x}_i , $i = 1, \dots, n^x$.*

- the discretization of the weak form that leads to the solution of a discretized system of equations.

This method does not require elements or meshes as standard Finite Elements methods but only the choice of the nodal values on the domain in which to approximate the solution of the problem. All integrals are carried out on circles in 2D or spheres in 3D at each point in question. A crucial point is the definition of the radius of these circles or spheres as we will see later.

3 Preliminar definitions

Let n be the number of nodes \mathbf{x}_i , $i = 1, \dots, n$, in which to approximate the solution of the problem, and w_i the corresponding weight functions in the MLS approximation.

- The **domain of definition of point \mathbf{x}** (or domain of definition of an MLS approximation for the trial function at any point \mathbf{x}) is a domain Ω_x which covers all the nodes whose weight functions do not vanish at \mathbf{x} . In other words Ω_x is a domain containing the nodes \mathbf{x}_i $i = 1, \dots, n^x$, $n^x \leq n$, such that $w_i(\mathbf{x}) \neq 0$.
- The **support of node \mathbf{x}_i** is the support of the weight function w_i , that is a circle of radius r_i centered at \mathbf{x}_i [7].

The domain of definition of an evaluation point \mathbf{x} can be described as the union of n^x overlapping circles, each centered at \mathbf{x}_i and radius r_i , for which $w_i(\mathbf{x}_i) \neq 0$ (see Figure 1).

Special case is if all radii r_i , $i = 1, \dots, n^x$ are equal to r : this is equivalent to a domain of definition of \mathbf{x} as a circle of radius r centered at \mathbf{x} .

- The **domain of influence for a node \mathbf{x}_i** is useful when calculating the local integral that derives by the local symmetric weak form for each node, and determines the nodal contacts of the stiffness matrix - that is the non zero entries in the system stiffness matrix depend upon all the nodes located within the domain of influence of the node in question.

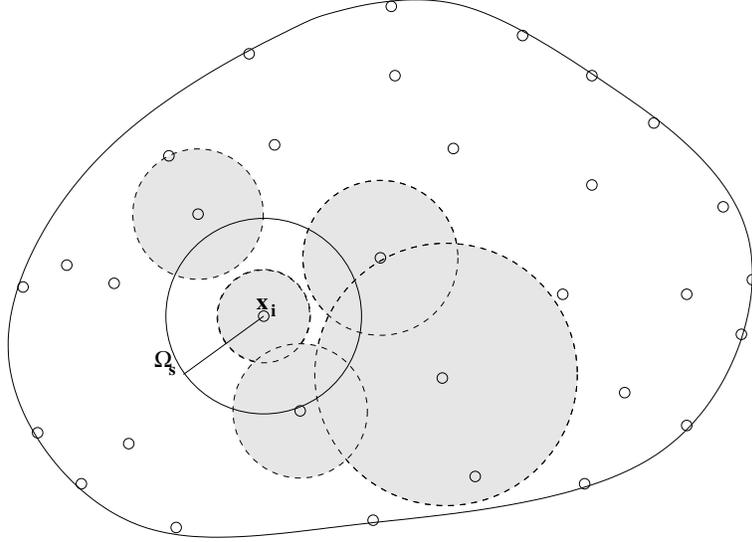


FIGURE 2: *Domain of influence for \mathbf{x}_i as union of supports of nodes \mathbf{x}_j , $j = 1, \dots, n_i$.*

The domain of influence is related to the domain of integration for the LSWF, generally called Ω_s , a circular domain of radius r_0 and center at the node \mathbf{x}_i considered. Generally, the radius r_0 is chosen as the distance between the node \mathbf{x}_i and the nearest neighboring node.

A definition for the domain of influence for \mathbf{x}_i says that this domain is given by the union of all domains of definition of points \mathbf{x} belonging to Ω_s , that is $\bigcup_{\mathbf{x} \in \Omega_s} \Omega_{\mathbf{x}}$.

Taking into account the definition of the domain of definition of \mathbf{x} it is simple to see that the domain of influence for \mathbf{x}_i is given by the union of the supports of the nodes \mathbf{x}_j , $j = 1, \dots, n_i$, $n_i \leq n$, such that the intersection of support of \mathbf{x}_j with Ω_s is not empty (see Figure 2).

4 Moving Least Squares Approximation

In the MLPG method the variable $u(\mathbf{x})$ of the problem is approximated by the Moving Least Squares approximation scheme [4].

This scheme is based on:

- the choice of a weight function with compact support for each node;
- the choice of a polynomial basis functions;
- a set of coefficients that depend on the position \mathbf{x} (with coordinates (x, y)) of the point in which to approximate u .

Let $u^h(\mathbf{x})$ the unknown trial approximant of the function $u(\mathbf{x})$. It is defined by:

$$u^h(\mathbf{x}) = \sum_{i=1}^m p_i(\mathbf{x}) a_i(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \mathbf{a}(\mathbf{x}) \quad (1)$$

where $\mathbf{p}^T(\mathbf{x}) = [p_1(\mathbf{x}), p_2(\mathbf{x}), \dots, p_m(\mathbf{x})]^T$ is a vector of the complete monomial basis of size m , and $\mathbf{a}(\mathbf{x}) = [a_1(\mathbf{x}), a_2(\mathbf{x}), \dots, a_m(\mathbf{x})]^T$ is the vector of the unknown coefficients $a_j(\mathbf{x})$, $j = 1, \dots, m$.

In 2D problems, a linear basis is given by $m = 3$ while a quadratic basis is given by $m = 6$: that is:

$$\begin{aligned} m = 3 \quad \mathbf{p}^T(\mathbf{x}) &= [1, x, y] \\ m = 6 \quad \mathbf{p}^T(\mathbf{x}) &= [1, x, y, x^2, xy, y^2]. \end{aligned}$$

The coefficient vector $\mathbf{a}(\mathbf{x})$ is determined by minimizing a weighted discrete L_2 norm, that is by minimizing the functional

$$J(\mathbf{x}) = \sum_{i=1}^{n^x} w_i(\mathbf{x}) [\mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) - \hat{u}_i]^2 \quad (2)$$

where w_i is the weight function associated with the node \mathbf{x}_i , n^x is the number of nodes in Ω_x , the domain of definition of \mathbf{x} , for which the weight functions $w_i(\mathbf{x}) > 0$, and \hat{u}_i is the fictitious nodal value and need not equal to $u^h(\mathbf{x}_i)$.

In matrix form, the functional (2) can be expressed as

$$J(\mathbf{x}) = [P \cdot \mathbf{a}(\mathbf{x}) - \hat{\mathbf{u}}]^T \cdot W \cdot [P \cdot \mathbf{a}(\mathbf{x}) - \hat{\mathbf{u}}] \quad (3)$$

where P and W are defined as matrices of sizes $n^x \times m$ and $n^x \times n^x$ respectively:

$$P = \begin{bmatrix} \mathbf{p}^T(\mathbf{x}_1) \\ \mathbf{p}^T(\mathbf{x}_2) \\ \dots \\ \mathbf{p}^T(\mathbf{x}_{n^x}) \end{bmatrix}$$

$$W = \begin{bmatrix} w_1(\mathbf{x}) & 0 & \dots & 0 \\ 0 & w_2(\mathbf{x}) & 0 & \dots \\ \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & w_{n^x}(\mathbf{x}) \end{bmatrix}$$

and the vector $\hat{\mathbf{u}}$ is the vector of the fictitious nodal values

$$\hat{\mathbf{u}}^T = [\hat{u}_1, \hat{u}_2, \dots, \hat{u}_{n^x}]^T.$$

Finding the extremum of $J(\mathbf{x})$ with respect to $\mathbf{a}(\mathbf{x})$ leads to the following linear relation between $\mathbf{a}(\mathbf{x})$ and $\hat{\mathbf{u}}$:

$$A(\mathbf{x})\mathbf{a}(\mathbf{x}) = B(\mathbf{x})\hat{\mathbf{u}} \quad (4)$$

where

$$A(\mathbf{x}) = \sum_{i=1}^{n^x} w_i(\mathbf{x}) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}) = P^T W P = B(\mathbf{x}) P \quad \text{is a matrix } m \times m \quad (5)$$

and

$$B(\mathbf{x}) = [w_1(\mathbf{x})\mathbf{p}(\mathbf{x}_1), w_2(\mathbf{x})\mathbf{p}(\mathbf{x}_2), \dots, w_{n^x}(\mathbf{x})\mathbf{p}(\mathbf{x}_{n^x})] = P^T W \quad \text{is a matrix } m \times n^x \quad (6)$$

Solving for $\mathbf{a}(\mathbf{x})$ from equation (4) yields

$$\mathbf{a}(\mathbf{x}) = A^{-1}(\mathbf{x})B(\mathbf{x})\hat{\mathbf{u}}. \quad (7)$$

Therefore, the unknown coefficients of $\mathbf{a}(\mathbf{x})$ can be obtained only if $A(\mathbf{x})$ is non-singular. A necessary condition for a well-defined MLS approximation is that at least m weight functions are non-zero for each \mathbf{x} . In this way the rank of P equals m and A is non-singular.

Substituting for $\mathbf{a}(\mathbf{x})$ into equation (1) gives the following relation:

$$u^h(\mathbf{x}) = \Phi^T(\mathbf{x}) \cdot \hat{\mathbf{u}} = \sum_{i=1}^{n^x} \Phi_i(\mathbf{x})\hat{u}_i \quad u^h(\mathbf{x}_i) \equiv u_i \neq \hat{u}_i, \quad \mathbf{x} \in \Omega_x \quad (8)$$

where Φ is a vector of n^x components:

$$\Phi^T(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})A^{-1}(\mathbf{x})B(\mathbf{x}) \quad (9)$$

and each component $\Phi_i(\mathbf{x})$, $i = 1, \dots, n^x$ is given by

$$\Phi_i(\mathbf{x}) = \sum_{j=1}^m p_j(\mathbf{x})[A^{-1}(\mathbf{x})B(\mathbf{x})]_{ji}. \quad (10)$$

The function $\Phi_i(\mathbf{x})$ is usually called the shape function of the MLS approximation corresponding to node \mathbf{x}_i .

The partial derivatives of $\Phi_i(\mathbf{x})$ are:

$$\frac{\partial \Phi_i(\mathbf{x})}{\partial x^k} = \sum_{j=1}^m \left[\frac{\partial p_j(\mathbf{x})}{\partial x^k} (A^{-1}B)_{ji} + p_j \frac{\partial (A^{-1}B)_{ji}}{\partial x^k} \right] \quad k = 1, 2 \quad (11)$$

where we have used the convention that $x^1 = x$ and $x^2 = y$. The derivative $\frac{\partial (A^{-1}B)_{ji}}{\partial x^k}$ is equal to

$$\frac{\partial (A^{-1}B)_{ji}}{\partial x^k} = (A^{-1} \frac{\partial B}{\partial x^k})_{ji} + (\frac{\partial A^{-1}}{\partial x^k} B)_{ji}$$

Taking into account the relation very simple to demonstrate

$$\frac{\partial A^{-1}}{\partial x^k} = -A^{-1} \frac{\partial A}{\partial x^k} A^{-1}$$

we obtain

$$\frac{\partial \Phi_i(\mathbf{x})}{\partial x^k} = \sum_{j=1}^m \left\{ \frac{\partial p_j(\mathbf{x})}{\partial x^k} (A^{-1}B)_{ji} + p_j \left[(A^{-1} \frac{\partial B}{\partial x^k})_{ji} - (A^{-1} \frac{\partial A}{\partial x^k} A^{-1} B)_{ji} \right] \right\} \quad (12)$$

Partial derivatives of A are obtained by partial derivatives of the weight functions:

$$\begin{aligned} \frac{\partial A(\mathbf{x})}{\partial x^k} &= \frac{\partial \sum_{i=1}^{n^x} w_i(\mathbf{x}) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i)}{\partial x^k} \\ &= \sum_{i=1}^{n^x} \frac{\partial w_i(\mathbf{x})}{\partial x^k} \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i) \end{aligned} \quad (13)$$

5 The weight function

In implementing the MLS approximation the basis functions ($m = 3$ or $m = 6$ in 2D) and the weight functions w_i are to be chosen.

The choice of the weight function is important for the subsequent behavior of the MLS approximation [4, 5]. For example, if we consider a weight function associated to each node constant over the entire domain, the minimization process correspond to the standard least squares approximation.

Two weight functions are commonly used in meshless methods, the Gaussian weight function and the quartic spline weight function.

The Gaussian weight function is described as:

$$w_i(\mathbf{x}) = \begin{cases} \frac{\exp[-(d_i/c_i)^{2k}] - \exp[-(r_i/c_i)^{2k}]}{1 - \exp[-(r_i/c_i)^{2k}]} & 0 \leq d_i \leq r_i \\ 0 & d_i \geq r_i \end{cases} \quad (14)$$

where $d_i = \|\mathbf{x} - \mathbf{x}_i\|$ is the distance between \mathbf{x} and \mathbf{x}_i , c_i is a constant controlling the shape of weight function w_i , and r_i is the size of the support for the weight function and determines the support of node \mathbf{x}_i . The parameter k is usually taken equal to 1.

So far, there is no theory in the definition of the parameter c_i and it is chosen empirically in such a way the weight function covers sufficient number of nodes to ensure the non-singularity of matrix A . At the same time, the size of the support, r_i , should be chosen large enough to have a sufficient number of nodes covered in the domain of definition of every sample point, to ensure the regularity of A . A very small r_i may results in a relatively large numerical errors when using a Gaussian numerical quadrature to calculate the entries in the system matrix. On the other hand, r_i should also be small enough to maintain the local character of the MLS approximation. A robust theory concerning this subject is an open research topic, therefore the choice on the numerical experiments is done by empirical considerations. In this work, following [6] c_i is chosen as the distance from node \mathbf{x}_i to the third nearest neighboring node and the radius of the domain of influence r_i is chosen in such a way $\frac{r_i}{c_i} \geq 3.5$ so that the weight function w_i covers sufficient number of nodes to ensure the non-singularity of A . To this aim a proper scaling parameter σ is set so that $r_i = \sigma c_i$.

A picture of the Gauss weight function for $c_i = 1$ and $r_i = 4$ is depicted in Figure 3.

Cross sections along the positive axis of the Gaussian weight function for different values of c_i , leaving unchanged the radius $r_i = 1$, are shown in Figure 4. As we can see, when c_i becomes larger and larger the shape of the function is almost the same (the ratio $\frac{r_i}{c_i} < 1$) and diverges from the known shape of a gaussian function. When leaving unchanged the parameter $c_i = 1$ and varying the radius r_i we can see (Figure 5) again that the shape does not more change for larger values of r_i but this time the ratio $\frac{r_i}{c_i}$ is greater than 4 and the profile is that typical of a gaussian function. Finally, Figure 6 shows the cross sections of the weight function with different values of c_i and r_i and with σ constant and equal to 4.

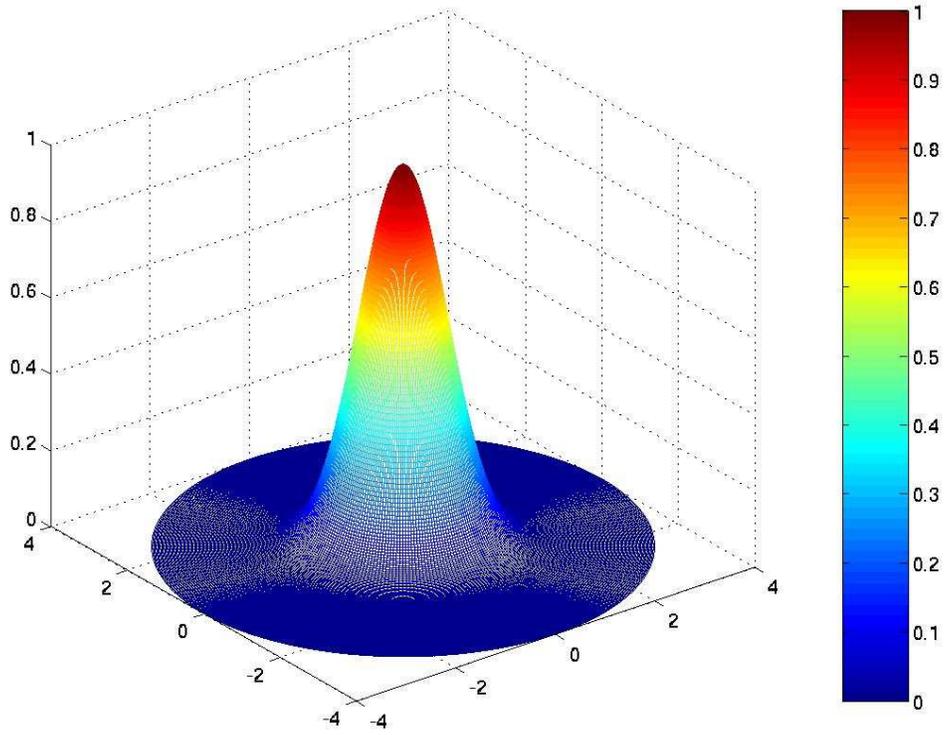


FIGURE 3: *The Gaussian weight function for $c_i = 1$ and $r_i = 4$.*

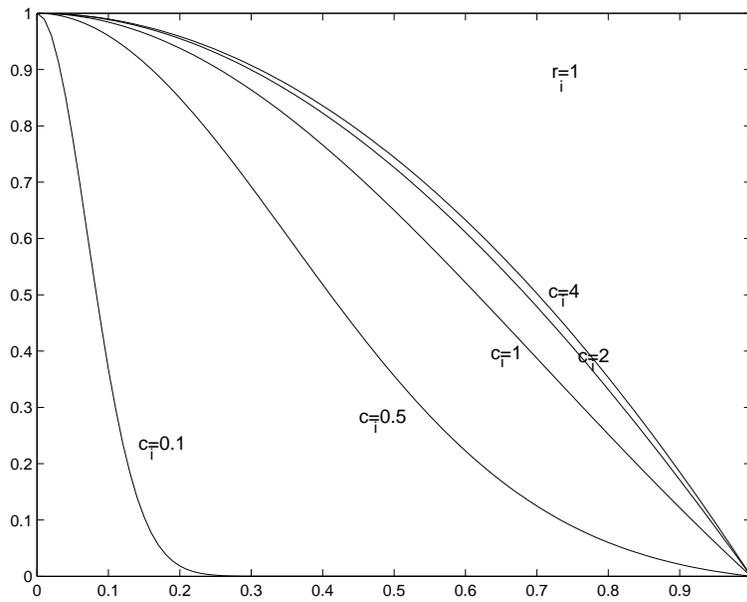


FIGURE 4: *Cross section of the Gaussian weight function for different choices of parameter c_i and r_i unchanged $r_i = 1$.*

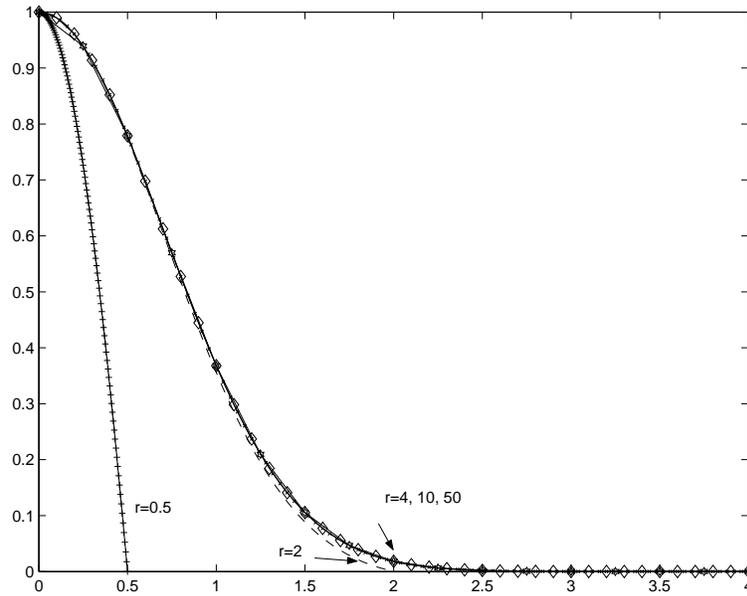


FIGURE 5: Cross section of the Gaussian weight function when c_i is unchanged $c_i = 1$ and r_i changes $r_i = 0.5, 2, 4, 10, 20$. Zoom of the interval $[0, 4]$.

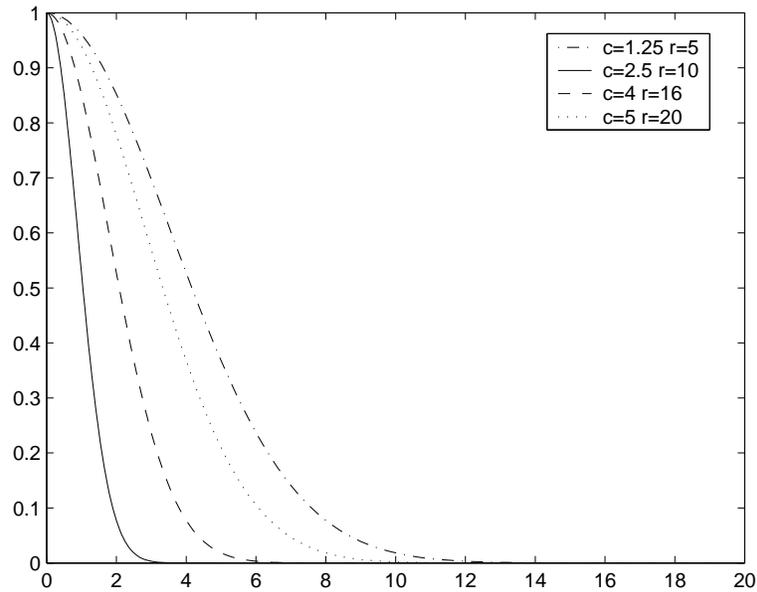


FIGURE 6: Cross section of the Gaussian weight function for different choices of c_i and r_i with $\sigma = 4$.

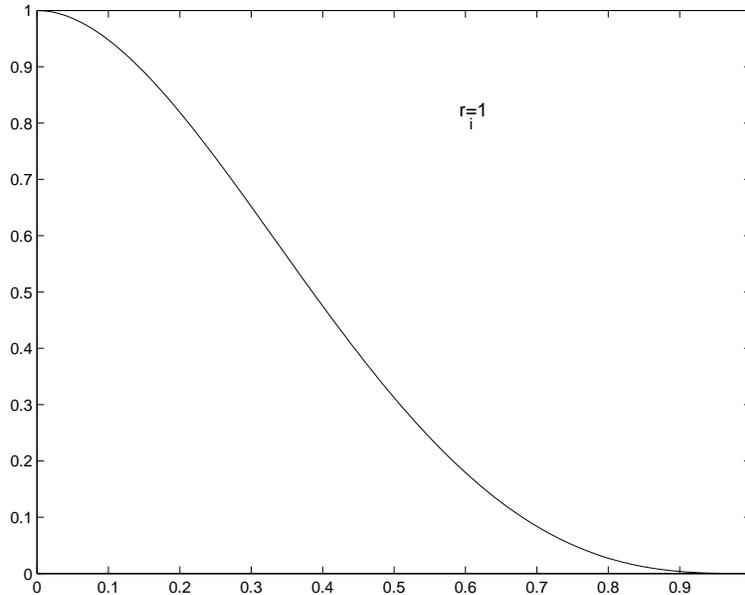


FIGURE 7: Cross section along the positive axis of the quartic spline weight function.

A different weight function is represented by the quartic spline:

$$w_i(\mathbf{x}) = \begin{cases} 1 - 6\left(\frac{d_i}{r_i}\right)^2 + 8\left(\frac{d_i}{r_i}\right)^3 - 3\left(\frac{d_i}{r_i}\right)^4 & 0 \leq d_i \leq r_i \\ 0 & d_i \geq r_i \end{cases} \quad (15)$$

whose profile is plotted in Figure 7.

6 The local symmetric weak form

Let us consider the linear Poisson's equation

$$\nabla^2 u(\mathbf{x}) = p(\mathbf{x}) \quad \text{on } \Omega \quad (16)$$

that is

$$\frac{\partial^2 u(\mathbf{x})}{\partial x^2} + \frac{\partial^2 u(\mathbf{x})}{\partial y^2} = p(\mathbf{x}) \quad (17)$$

with boundary conditions

$$\begin{aligned} u &= \bar{u} \quad \text{on } \Gamma_u \\ \frac{\partial u}{\partial \vec{n}} &\equiv q = \bar{q} \quad \text{on } \Gamma_q \end{aligned} \quad (18)$$

where \bar{u} and \bar{q} are the prescribed potential and normal flux, respectively, on the essential (or Dirichlet) boundary Γ_u and on the flux (or Neumann) boundary Γ_q , and \vec{n} is the outward normal direction to the boundary $\Gamma = \Gamma_u \cup \Gamma_q$.

We consider a weak form over a local sub-domain Ω_s , located entirely inside the global domain Ω . Usually, Ω_s is taken to be a circle in 2D or a sphere in 3D, centered at the point \mathbf{x} in question. Next we use the MLS approximation to develop a meshless method.

A generalized local weak form of the differential equation (16) and boundary conditions (18), over Ω_s , can be written as:

$$\int_{\Omega_s} (\nabla^2 u(\mathbf{x}) - p(\mathbf{x}))v \, d\Omega - \alpha \int_{\Gamma_{su}} (u - \bar{u})v \, d\Gamma = 0 \quad (19)$$

where u is the trial function, v is the test function, Γ_{su} is a part of the boundary $\partial\Omega_s$ of Ω_s , over which the essential boundary conditions are specified. The boundary $\partial\Omega_s$ can be generally written as $\partial\Omega_s = \Gamma_s \cup L_s$ where $\Gamma_s = \partial\Omega_s \cap \Gamma$ is the intersection of the local boundary on the global boundary where boundary conditions are specified, while $L_s = \partial\Omega_s - \Gamma_s$ is the remaining part of the boundary where no boundary conditions are specified.

The parameter α in equation (19) is a penalty parameter, $\gg 1$, used to impose the essential boundary conditions.

Using the relation

$$\frac{\partial \left(\frac{\partial u}{\partial x^k} v \right)}{\partial x^k} = \frac{\partial^2 u}{\partial^2 x^k} v + \frac{\partial u}{\partial x^k} \frac{\partial v}{\partial x^k}, \quad k = 1, 2$$

the integral $\int_{\Omega_s} \nabla^2 uv \, d\Omega$ becomes

$$\int_{\Omega_s} \nabla^2 uv \, d\Omega = \int_{\Omega_s} \left(\frac{\partial \left(\frac{\partial u}{\partial x} v \right)}{\partial x} + \frac{\partial \left(\frac{\partial u}{\partial y} v \right)}{\partial y} - \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} \right) d\Omega$$

Application of the divergence theorem yields the following expression:

$$\int_{\Omega_s} \nabla^2 uv \, d\Omega = \int_{\partial\Omega_s} \left(\frac{\partial u}{\partial x} v \right) n_x + \left(\frac{\partial u}{\partial y} v \right) n_y \, d\Gamma - \int_{\Omega_s} \left(\frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} \right) d\Omega$$

where $(n_x, n_y) = \vec{n}$ is the outward normal to the boundary. Being $\left(\frac{\partial u}{\partial x} \right) n_x + \left(\frac{\partial u}{\partial y} \right) n_y = \frac{\partial u}{\partial \vec{n}} \equiv q$ we can write:

$$\int_{\Omega_s} \nabla^2 uv \, d\Omega = \int_{\partial\Omega_s} \left(\frac{\partial u}{\partial \vec{n}} v \right) d\Gamma - \int_{\Omega_s} \left(\frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} \right) d\Omega$$

Taking into account that the boundary $\partial\Omega_s$ is equal to $L_s \cup \Gamma_{su} \cup \Gamma_{sq}$, where Γ_{sq} is the portion of $\partial\Omega_s$ over which the Neumann condition $q = \bar{q}$ is specified, equation (19) can be written as:

$$\int_{L_s} qv \, d\Gamma + \int_{\Gamma_{su}} qv \, d\Gamma + \int_{\Gamma_{sq}} \bar{q}v \, d\Gamma - \int_{\Omega_s} \left(\frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} + pv \right) d\Omega - \alpha \int_{\Gamma_{su}} (u - \bar{u})v \, d\Gamma = 0 \quad (20)$$

To simplify the above equation, we select a test function v such that it vanishes over L_s . To this aim we use the same functions as weight functions in the MLS approximation and test function, with the only difference that the support of the test function is r_0 , the radius of the local domain Ω_s such that the test function vanishes on the circle of radius r_0 . In this way, we obtain the following local symmetric weak form:

$$\int_{\Omega_s} \left(\frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} \right) d\Omega + \alpha \int_{\Gamma_{su}} uv \, d\Gamma - \int_{\Gamma_{su}} qv \, d\Gamma = \int_{\Gamma_{sq}} \bar{q}v \, d\Gamma + \alpha \int_{\Gamma_{su}} \bar{u}v \, d\Gamma - \int_{\Omega_s} pv \, d\Omega. \quad (21)$$

This equation is for any point \mathbf{x} , and the problem becomes one as if we are dealing with a localized boundary value problem over a 2D or 3D sphere Ω_s . Note that the radius of the circle (or sphere) will affect the solution.

7 Final system of equations

Equation (21) involves one point \mathbf{x} and therefore yields only one linear equation involving $\hat{\mathbf{u}}$. We want to solve problem (16) on the nodes of the domain. Therefore we need as many local domains Ω_s as the number of nodes in the global domain, centered at the prescribed nodes.

The trial function u is chosen by the MLS approximation as

$$u(\mathbf{x}) = \sum_{j=1}^n \Phi_{ij}(\mathbf{x}) \hat{u}_j$$

and the test function is chosen as a weight function of the MLS approximation with radius r_0 , $v = v(\mathbf{x}, \mathbf{x}_i)$ for each node \mathbf{x}_i .

Substitution of expression of u into equation (21) for each node \mathbf{x}_i , $i = 1, \dots, n$ leads to a system of linear equations

$$K \hat{\mathbf{u}} = \mathbf{f} \quad (22)$$

where the i -th row of the system is:

$$\begin{aligned} & \int_{\Omega_s} \left(\sum_{j=1}^n \frac{\partial \Phi_j}{\partial x} \frac{\partial v(\mathbf{x}, \mathbf{x}_i)}{\partial x} + \frac{\partial \Phi_j}{\partial y} \frac{\partial v(\mathbf{x}, \mathbf{x}_i)}{\partial y} \right) \hat{u}_j \, d\Omega + \alpha \int_{\Gamma_{su}} \sum_{j=1}^n \Phi_j v(\mathbf{x}, \mathbf{x}_i) \hat{u}_j \, d\Gamma \\ & - \int_{\Gamma_{su}} \sum_{j=1}^n \left(\frac{\partial \Phi_j}{\partial x} n_x + \frac{\partial \Phi_j}{\partial y} n_y \right) v(\mathbf{x}, \mathbf{x}_i) \hat{u}_j \, d\Gamma = \\ & \int_{\Gamma_{sq}} \bar{q}v(\mathbf{x}, \mathbf{x}_i) \, d\Gamma + \alpha \int_{\Gamma_{su}} \bar{u}v(\mathbf{x}, \mathbf{x}_i) \, d\Gamma - \int_{\Omega_s} pv(\mathbf{x}, \mathbf{x}_i) \, d\Omega. \end{aligned} \quad (23)$$

Therefore

$$\begin{aligned} K_{ij} &= \int_{\Omega_s} \left(\frac{\partial \Phi_j}{\partial x} \frac{\partial v(\mathbf{x}, \mathbf{x}_i)}{\partial x} + \frac{\partial \Phi_j}{\partial y} \frac{\partial v(\mathbf{x}, \mathbf{x}_i)}{\partial y} \right) d\Omega + \alpha \int_{\Gamma_{su}} \sum_{j=1}^n \Phi_j v(\mathbf{x}, \mathbf{x}_i) \, d\Gamma \\ & - \int_{\Gamma_{su}} \sum_{j=1}^n \left(\frac{\partial \Phi_j}{\partial x} n_x + \frac{\partial \Phi_j}{\partial y} n_y \right) v(\mathbf{x}, \mathbf{x}_i) \, d\Gamma \\ f_i &= \int_{\Gamma_{sq}} \bar{q}v(\mathbf{x}, \mathbf{x}_i) \, d\Gamma + \alpha \int_{\Gamma_{su}} \bar{u}v(\mathbf{x}, \mathbf{x}_i) \, d\Gamma - \int_{\Omega_s} pv(\mathbf{x}, \mathbf{x}_i) \, d\Omega \end{aligned}$$

The stiffness matrix K is banded but unsymmetric: the locations of the non-zero entries in K depends on the domain of influence of each node.

In the present work the test function v is chosen to be the same as the weight function w in the MLS approximation for the trial function u (the difference is in the radius of the support).

8 Numerical implementation

The implementation of the MLPG method is carried out in according with the following steps:

1. Choose the n nodes in the domain Ω and on the boundary Γ ;
2. Decide the basis function (linear or quadratic) and weighth functions (Gaussian or quartic spline) such that the MLS approximation can be defined;
3. For each node $\mathbf{x}_i, i = 1, \dots, n$:
 - determine the local sub-domain Ω_s and the corresponding local boundary $\partial\Omega_s$
 - determine the support for the weight functions (that is the radius r_i)
4. For each node $\mathbf{x}_i, i = 1, \dots, n$:
 - determine the Gaussian quadrature points \mathbf{x}_Q in the sub-domain Ω_s and on its boundary $\partial\Omega_s$;
 - for each \mathbf{x}_Q determine the nodes \mathbf{x}_j located in the domain of definition of the MLS approximation for the trial function at point \mathbf{x}_Q - that is the nodes for which $w_j(\mathbf{x}_Q) > 0$;
 - for those nodes in the domain of definition of trial function at \mathbf{x}_Q calculate the value of the shape function $\Phi_j(\mathbf{x}_Q)$ and its derivatives;
 - evaluate the numerical integrals inside the sub-domain Ω_s along its boundary in order to give the contribution for the entries of the stiffness matrix and the vector \mathbf{f} ;
 - assemble the contributions to the linear system.
5. Solve the linear system for the fictitious nodal values $\hat{\mathbf{u}}$;
6. Apply the MLS approximation (8) to calculate the value of the unknown variable \mathbf{u} and its derivates at those sample points under consideration (e.g. the nodes).

9 Numerical examples

We present some numerical examples to illustrate the implementation and convergence of the present MLPG approach.

The error estimation is calculated as the L_2 norm of the difference between the vector of the numerical solution \mathbf{u} and the vector of the analytical solution \mathbf{u}^{exact} :

$$|e| = \frac{\sqrt{\sum_{i=1}^n (u_i - u_i^{exact})^2}}{\sqrt{\sum_{i=1}^n (u_i^{exact})^2}}. \quad (24)$$

9.1 Patch test

Consider a Dirichlet problem with $p = 0$ and exact solution $u^{exact} = x + y$ in a domain of dimension 2×2 . This is a standard patch test.

To perform a patch test, each exact solution is prescribed as the essential boundary conditions in the problem, and the problem is analyzed with the MLPG algorithm. To pass the patch test, the MLPG algorithm must reproduce the exact solution at all interior nodes of the model to machine accuracy.

Let us consider 9 nodes on the domain, 8 belonging to the boundary and 1 internal point, as shown in Figure 8.

It is interesting to note that the behavior of the solution is dependent on the choice of the Gaussian quadrature points. If we choose, for example, 9 Gauss points on each section of Γ_s and 5×8 points in the local domain Ω_s for numerical quadratures, we observe that the MLPG analysis reproduce the exact solution at the interior node. Changing the coordinates of node 5, we continue to observe the exact solution at node 5.

But if we change the number of Gaussian points for the numerical quadrature, the numerical solution may not reproduce the exact value! Table 1 displays this behavior for node 5 with coordinates (1.0, 1.0), by changing the number of Gaussian quadrature points in the local domain Ω_s . The number 8 for the quadrature along the y -axis seems to be essential to reproduce the exact solution. The other parameters used for this simulation are: linear basis functions, Gaussian weight functions, scaling parameter for the weight function $\sigma = 4$, and penalty parameter for the essential boundary conditions $\alpha = 10^8$. It can be observed that may be sufficient a smaller penalty parameter to obtain a sufficient accuracy degree in the numerical results, but we prefer to assign a larger value (as 10^8) to be sure of a correct imposition of the essential boundary conditions. For this reason we do not change this value in the numerical simulations presented in this work.

The linear basis function for the MLS approximation is chosen because the exact solution is a linear function of x and y (therefore a linear basis function is able to represent this solution). Similar results are obtained by applying both Gaussian and quartic weight function, due to the simplicity of the test case. But more investigation is required in order to the selection of the proper weight function (and of the proper scaling parameter for the Gaussian function).

This study is done numerically in the following test case.

TABLE 1: *Patch test*

u^{exact}	u	Gaussian points in Ω_s
2.0	2.0	$i \times 8 \ i = 1, \dots, 9$
2.0	2.0000076	1×9
2.0	2.0000070	2×9
2.0	2.0000072	$i \times 9 \ i = 3, \dots, 9$
2.0	2.0000045	1×7
2.0	2.0000041	2×7
2.0	2.0000042	$i \times 7 \ i = 3, \dots, 9$
2.0	2.0000005	3×6
2.0	1.9999757	6×5
2.0	2.0009592	6×4
2.0	1.9804683	9×3

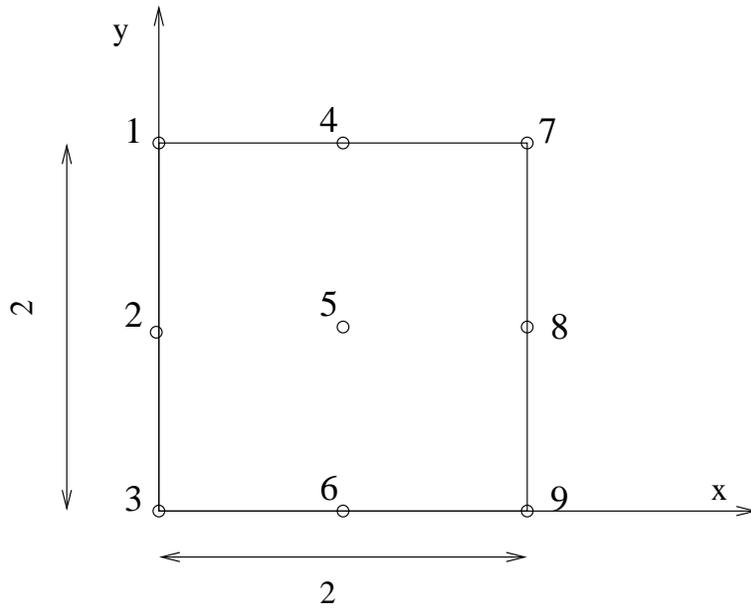


FIGURE 8: *Nodes for the patch test*

TABLE 2: Comparison of L_2 errors on different grids, between $\sigma = 4$ and $\sigma = 10$

$m = 3$		$m = 6$	
$\sigma = 4,$	$\sigma = 10$	$\sigma = 4$	$\sigma = 10$
2.0407e-2	2.0407e-2	1.0381e-2	1.0381e-2
1.3918e-3	1.3918e-3	9.1498e-4	9.1394e-4
2.1687e-4	2.1686e-4	1.0463e-4	1.0459e-4
7.5900e-5	7.5904e-5	2.3185e-5	2.3155e-5

9.2 Laplace equation

In the same 2×2 domain we solve the Laplace equation with exact solution

$$u^{exact} = -x^3 - y^3 + 3x^2y + 3xy^2. \quad (25)$$

A Dirichlet problem is solved, for which the essential boundary condition is imposed on all sides, and a mixed problem, for which the essential boundary condition is imposed on top and bottom sides and the flux boundary condition is prescribed on left and right sides of the domain.

We apply the MLPG scheme by considering both linear and quadratic basis functions, both Gaussian (with different scaling parameters) and quartic spline weight functions, different number of quadrature points, in order to study the performance and the convergence of the method. To this aim we consider 4 regular meshes of 9, 25, 81, 289 nodes obtained by starting from a subdivision of 3×3 of the grid (as depicted in Figure 8). Each mesh is further refined by adding a node in the midpoint of the edge connecting two nodes of the coarser mesh along the x - and y - axis: therefore we have 5×5 subdivisions, next 9×9 and finally 17×17 subdivisions.

9.2.1 Dirichlet problem

From the results obtained, we can observe that the parameter σ is not so relevant to respect to the choice of the number of quadrature points. Indeed, if the parameter is less than 3.5 (as explained in Section 5), errors decrease when the linear basis is selected but blow up when selecting a quadratic basis.

On the contrary, by setting the value of σ greater than 3.5, the differences are not so relevant, as Table 2 shows for the Gaussian weight function in combination with 9 points for numerical quadrature on the sections of local domains and 9×9 points for the numerical quadrature on the local domains, and a penalty parameter $\alpha = 10^8$. This fact suggests that the proper selection of the parameter c_i is more crucial to respect the scaling parameter σ (as we will see later) since there is no theory regarding an optimal choice of c_i (and, consequently, of r_i). At the same time, the choice of the weight function and of the number of quadrature points gives different accuracy results. For the same test case, we consider the Gaussian and the quartic spline weight functions, and different pairs of numbers of quadrature points (6×6 , 5×8 and 9×9 , respectively) for the local domains while it is not changed the number of quadrature points for the integration on the section

TABLE 3: *Dirichlet problem: different pairs of points for numerical integration.*

Gaussian		quartic spline	
$m = 3$	$m = 6$	$m = 3$	$m = 6$
6 × 6 quadrature points			
2.0664e-2	1.0755e-2	2.9543e-2	8.8281e-3
2.6776e-3	2.4244e-3	2.9583e-3	1.9990e-3
1.3942e-3	1.5477e-3	1.1863e-3	1.2303e-3
7.7546e-4	8.9083e-4	6.1133e-4	6.5449e-4
5 × 8 quadrature points			
2.0411e-2	1.03807e-2	2.9497e-2	8.6620e-3
1.4061e-3	9.1927e-4	2.2860e-3	1.1276e-3
2.3653e-4	1.4677e-4	5.8555e-4	6.1587e-4
8.6890e-5	7.3725e-5	2.0076e-4	9.0665e-5
9 × 9 quadrature points			
2.0407e-2	1.0381e-2	2.9496e-2	8.6616e-3
1.3918e-3	9.1498e-4	2.2674e-3	1.1083e-3
2.1687e-4	1.0463e-4	5.7414e-4	6.0583e-4
7.5900e-5	2.3185e-5	1.8238e-4	9.7371e-5

of the local domains (set equal to 9). Comparison of the results shows that the Gaussian weight function together with 9×9 quadrature points gives the better results, as displayed in Table 3.

It is known from the literature that the Gaussian weight function usually performs better than the spline weight functions. Therefore, we do not surprise about the better accuracy obtained in our test case. In the following we continue to compare both weight functions just for completeness.

A question remains open: why in this test case 9×9 points for numerical quadrature give better results to respect 5×8 points as in the patch test?

And: is it appropriate the selection of parameter c_i , by the distance from the node \mathbf{x}_i to the third nearest neighboring node as suggested by [6]?

To answer the last question, we change (in empiric way) the definition of c_i (and consequently of r_i) for the solution of this test case by using the Gaussian weight function with 9×9 points of quadrature, $\alpha = 10^8$ and $\sigma = 4$.

The results are reported in Table 4. We can observe that selecting $c_i = 1.25h$, with h being the minimum distance between \mathbf{x}_i and its neighboring nodes (or, equivalently, the mesh size), then the numerical errors are less than that obtained with c_i chosen as proposed by [6]. When doubling c_i , the error seems to reduce but at the final mesh it grows: this behavior may be due to the fact that the value c_i is very large to respect the mesh size and, consequently, the radius r_i is too large and the local character of the MLS approximation is lost.

If we observe the rate of convergence by doubling the number of nodes that discretize the domain, we observe that the results obtained by choosing c_i as proposed by [6] are reasonable, since we reach 1.5 with the weight function associated to the linear basis and

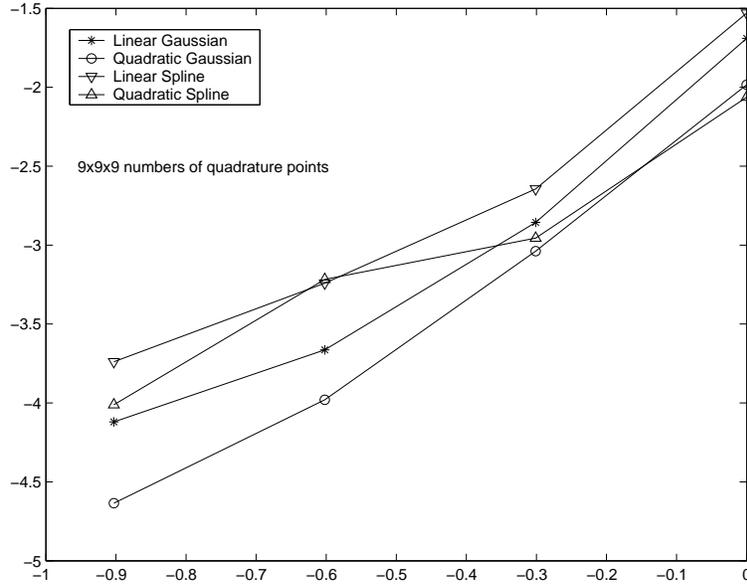


FIGURE 9: Convergence of the MLPG scheme by applying Gaussian or quartic spline, with $9 \times 9 \times 9$ points of quadrature.

TABLE 4: Results for the Laplace test case with the Gaussian weight function selected in such a way the parameter c_i varies with h , being h the mesh size.

$c_i = 1.25h$		$c_i = 2.5h$		$c_i = 5h$	
$m = 3$	$m = 6$	$m = 3$	$m = 6$	$m = 3$	$m = 6$
1.8908e-2	2.0799e-2	2.1977e-2	2.2540e-2	2.2940e-2	2.2986e-2
1.3974e-3	7.6863e-4	2.4516e-4	5.0256e-5	7.2959e-6	6.1052e-6
1.8893e-4	4.5024e-5	1.2124e-5	1.1150e-5	1.1168e-5	1.1143e-5
3.5724e-5	2.2724e-5	2.3105e-5	2.3272e-5	2.30730e-5	2.2941e-5

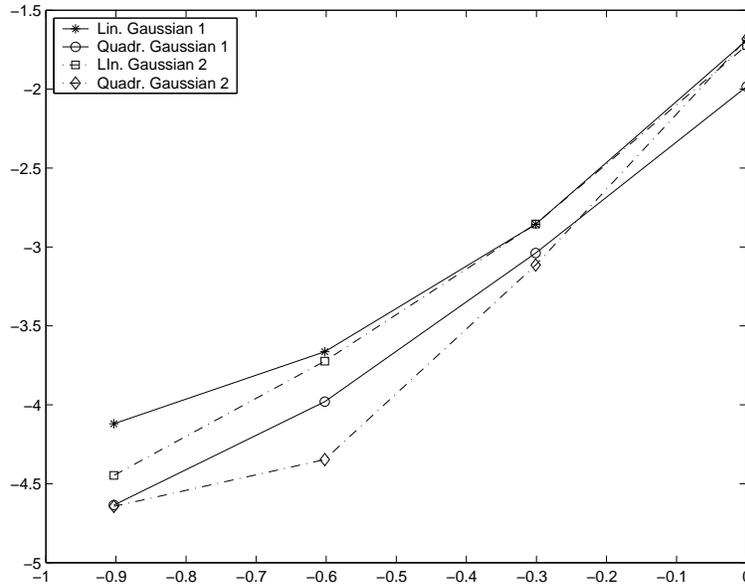


FIGURE 10: Convergence differences between the choice of c_i as in [6] - in legend as Gaussian 1 - and $c_i = 1.25h$ - in legend as Gaussian 2.

TABLE 5: Mixed problem.

Gaussian		quartic spline	
$m = 3$	$m = 6$	$m = 3$	$m = 6$
3.1561e-2	1.2599e-2	4.1412e-2	1.0765e-2
5.0901e-3	1.8861e-3	2.9115e-3	2.0574e-3
1.3929e-3	3.9915e-4	1.0219e-3	7.8156e-4
4.8607e-4	8.8918e-5	5.2296e-4	5.7335e-4

2 with the weight function associated to the quadratic basis function, while for $c_i = 1.25h$, we reach more than 2 for the linear and just 1 for the quadratic basis function (see also Figure 10 where there is a log-log plot of the errors norms of the above simulations).

For all these reasons we prefer to continue to use the c_i chosen as proposed by [6].

9.2.2 Mixed problem

We now consider the mixed problem for which Dirichlet boundary conditions are imposed on top and bottom sides of the domain and Neumann conditions are prescribed on left and right sides of the domain.

We use 9 points for the numerical quadrature along each section of the boundary of local domains Ω_s and 9×9 points for the numerical integration in the local domain Ω_s . The other parameters are: $\alpha = 10^8$, $\sigma = 4$, c_i as the third smallest distance between \mathbf{x}_i and its surrounding nodes.

Table 5 reports the errors for the linear and quadratic Gaussian weight function and for the linear and quadratic quartic spline.

TABLE 6: *Poisson's equation: results for the Dirichlet and mixed problem.*

Gaussian		quartic spline	
$m = 3$	$m = 6$	$m = 3$	$m = 6$
Dirichlet problem			
1.61948e-2	6.4761e-3	2.5200e-2	5.7524e-3
1.30904e-3	6.9793e-4	2.0951e-3	9.3172e-4
1.87122e-4	7.8010e-5	5.0216e-4	5.7797e-4
6.08480e-5	1.3535e-5	2.0066e-4	8.4223e-5
mixed problem			
2.5815e-2	7.8145e-3	3.5522e-2	7.2430e-3
4.2892e-3	1.4303e-3	2.6806e-3	1.7054e-3
1.1202e-3	3.1092e-4	7.9725e-4	7.1306e-4
3.7758e-4	6.6765e-5	4.1744e-4	5.5808e-4

The results obtained are in agreement with that obtained with the Dirichlet problem.

9.3 Poisson's equation

We consider the Poisson equation(16) with a source function $p = x + y$ in the same domain 2×2 considered in the previous numerical examples, for which the exact solution is:

$$u^{exact} = -\frac{5}{6} (x^3 + y^3) + 3x^2y + 3xy^2. \quad (26)$$

As the same manner of the Laplace equation studied in the previous section, we consider a Dirichlet and a mixed problem and apply the MLPG method on the same meshes of 9, 25, 81 and 289 nodes used in the previous example.

The parameters c_i , σ , α , as well as the number of quadrature points are the same as those used in the mixed problem of the Laplace equation. Also, we compare the results obtained with linear and quadratic basis as well as Gaussian and quartic spline weight functions.

The results are shown in Table 6.

A About the numerical quadrature

The integrals on the local domain Ω_s are calculated by applying the Gauss-Legendre formulae by proper change of variables.

Given the circle of center $\mathbf{x} = (x, y)$ and radius r_0 , the generic point of coordinates (x_q, y_q) can be written as:

$$x_q = x + \rho \cos \theta \quad (27)$$

$$y_q = y + \rho \sin \theta \quad (28)$$

where $\rho \in [0, r_0]$ and $\theta \in [\theta_1, \theta_2]$, being θ_1 and θ_2 the angle that characterize Ω_s ($\theta_1 = 0$ and $\theta_2 = 2\pi$ for internal points while for points whose circle intersects the boundary, the angles θ_1 and θ_2 are selected by considering the intersection of the circle with the boundary itself).

The Jacobian of the transformation $(x_q, y_q) \rightarrow (\rho, \theta)$ is

$$J = \begin{pmatrix} \frac{\partial x_q}{\partial \rho} & \frac{\partial y_q}{\partial \rho} \\ \frac{\partial x_q}{\partial \theta} & \frac{\partial y_q}{\partial \theta} \end{pmatrix}$$

whose determinant is equal to ρ .

A first change of variables leads to:

$$\int_{\Omega_s} f(x_q, y_q) d\Omega = \int_0^{\rho} \int_{\theta_1}^{\theta_2} F(\rho, \theta) \rho d\rho d\theta$$

where $F(\rho, \theta) = f(x + \rho \cos \theta, y + \rho \sin \theta)$.

To apply the Gauss-Legendre quadrature rule, we need again a change of variables to operate in the square $[-1, 1] \times [-1, 1]$.

To this aim we introduce the variables ξ and η by the affine transformations:

$$\rho = \frac{r_0}{2}\xi + \frac{r_0}{2} \quad (29)$$

$$\theta = \frac{1}{2}\theta_1(1 - \eta) + \frac{1}{2}\theta_2(1 + \eta) \quad (30)$$

whose Jacobian is:

$$J = \begin{pmatrix} \frac{r_0}{2} & 0 \\ 0 & \frac{-\theta_1}{2} + \frac{\theta_2}{2} \end{pmatrix}$$

Its determinant is equal to $\det J = \frac{r_0(\theta_2 - \theta_1)}{4}$

Therefore, we obtain

$$\int_0^{\rho} \int_{\theta_1}^{\theta_2} F(\rho, \theta) \rho d\rho d\theta = \int_{-1}^1 \int_{-1}^1 G(\xi, \eta) d\xi d\eta$$

where $G(\xi, \eta) = F(\rho(\xi, \eta), \theta(\xi, \eta)) \rho(\xi, \eta) \frac{r_0(\theta_2 - \theta_1)}{4}$.

Now we can apply the Gauss-Legendre formula by calculating first the integral along the ξ -axis and by considering η as a constant, next by integrating to respect the η variable. We obtain the formula:

$$\int_{-1}^1 \int_{-1}^1 G(\xi, \eta) d\xi d\eta \approx \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} w_i w_j G(\xi_j, \eta_i)$$

where n_1 and n_2 are the number of quadrature points that are to be chosen for the application of the Gauss-Legendre formula to approximate the integral to respect the η and ξ axis, respectively.

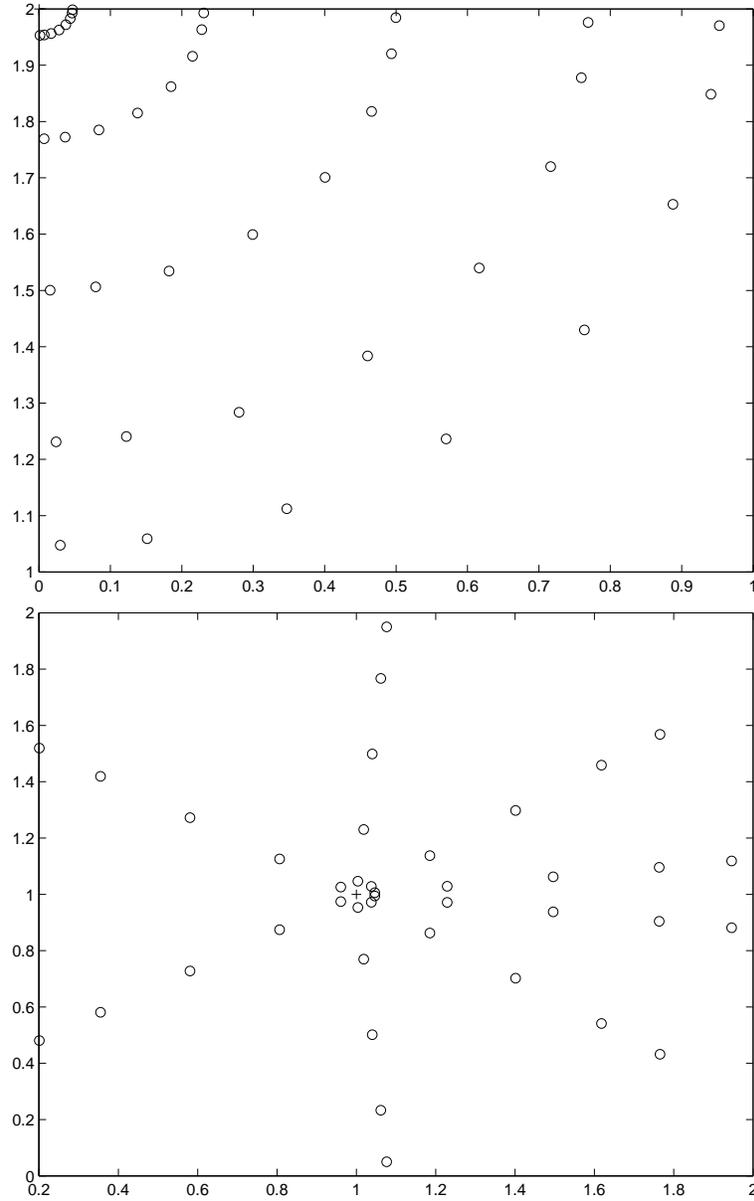


FIGURE 11: *Distribution of 5×8 Gaussian points x_q, y_q in the domains Ω_s relatively to node 1 of coordinates $(0, 2)$ (top) and node 5 of coordinates $(1, 1)$ (bottom) on the mesh with 9 nodes.*

Therefore the integral

$$\int_{\Omega_s} f(x_q, y_q) \, d\Omega \approx \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} w_i w_j G(\xi_j, \eta_i).$$

REMARK A.1. *The fact that we firstly integrate along the ξ -axis by considering the η variable constant and next integrate to respect η , may be the answer to the question posed when applying the patch test and regarding the different results obtained with different choices of quadrature points.*

B Application of the MLPG approach in elasto-statics problems

The MLPG approach is extensively applied for the solution of elasto-statics problems

$$\sigma_{ij,j} + b_j = 0 \quad \text{in } \Omega \quad (31)$$

where σ_{ij} is the stress tensor, which corresponds to the displacement field u_i , b_j is the body force, a repeated index implies summation over the range of index and $(\cdot)_{,j}$ denotes the partial derivative $\frac{\partial(\cdot)}{\partial x^j}$.

The corresponding boundary conditions are given as follows:

$$u_i = \bar{u}_i \quad \text{on } \Gamma_u \quad (32)$$

$$t_i \equiv \sigma_{ij} \bar{n}_j = \bar{t}_i \quad \text{on } \Gamma_t \quad (33)$$

where \bar{u}_i and \bar{t}_i are the prescribed displacements and tractions, respectively, on the boundary Γ_u and the boundary Γ_t , and \bar{n}_i is the unit outward normal to the boundary Γ . Γ_u and Γ_t are complementary subset of Γ and represent the displacement and the traction boundary, respectively.

B.1 Review of stress, strain and displacement

The elasticity of a body can be described by three variables: the stress, the strain and the displacement.

The stress The concept of stress originated from the study of strenght and failure of solids. The stress field is the distribution of internal "tractions" that balance a given set of external tractions and body forces.

Surface tractions, or stresses acting on an internal datum plane, are typically decomposed into three mutually orthogonal components. The component normal to the surface represents *direct or normal stress*. The other two components are tangential to the surface and represent *shear stress*. They vary smoothly with respect to the rotation angle and there exist a pair of particular angles where the stresses take on special values. By appropriate transformation equations there is an angle that defines the principal directions where the

only stresses are normal stresses. These stresses are thus called *principal stresses* and are found from the original stresses. The principal stresses are useful to define the plane stress.

Direct stresses tend to change the volume of the material while the shear stresses tend to deform the material without changing its volume.

The stress state can be organized into the matrix known as the stress tensor (or stress matrix):

$$\sigma = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{pmatrix}$$

where σ_{xy} have the meaning of the stress on the x plane along the y direction. Note that the shear stresses across the diagonal are identical (i.e. $\sigma_{xy} = \sigma_{yx}$, $\sigma_{yz} = \sigma_{zy}$) as a result of static equilibrium.

The strain In one dimension, the strain can be defined as the ratio of elongation with respect to the original length. Generalizing to three dimensions, the strain is related to the displacement \vec{u} by the symmetric gradient as:

$$\begin{aligned} \epsilon &= \nabla_s \vec{u} = \frac{1}{2} (\nabla \vec{u} + (\nabla \vec{u})^T) \\ &= \frac{1}{2} \left[\begin{pmatrix} \frac{\partial}{\partial x^1} \\ \frac{\partial}{\partial x^2} \\ \frac{\partial}{\partial x^3} \end{pmatrix} (u \ v \ w) + \begin{pmatrix} \left(\frac{\partial}{\partial x^1} \right)^T (u \ v \ w) \\ \left(\frac{\partial}{\partial x^2} \right)^T (u \ v \ w) \\ \left(\frac{\partial}{\partial x^3} \right)^T (u \ v \ w) \end{pmatrix} \right] \end{aligned}$$

In terms of components we obtain:

$$\epsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x^j} + \frac{\partial u_j}{\partial x^i} \right),$$

where $\vec{u} = (u_1, u_2, u_3) = (u, v, w)$ is the displacement vector, x is coordinate, and the two indices i and j can range over the three coordinates $\{1, 2, 3\}$ (in three dimensional space $(x^1, x^2, x^3) = (x, y, z)$).

We focus now our attention in 2 dimensions.

The strain tensor can be written as

$$\epsilon = \begin{pmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{pmatrix} = \begin{pmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}$$

where $\gamma_{xy} = 2\epsilon_{xy}$.

Calling \bar{D} the matrix of the partial derivatives:

$$\bar{D} = \begin{pmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{pmatrix}$$

we obtain $\epsilon = \bar{D}\vec{u}$.

The stress vector σ is related to the strain vector by a matrix \mathbf{E} that depends on whether the problem is one of plane strain or plane stress.

The plane stress The plane stress is said when one principal stress is much smaller than the other two. By assuming that this small principal stress is zero, the three-dimensional stress state can be reduced in two dimensions and the remaining two principal stresses lie in a plane, the so called plane stress.

An alternative definition is that a plane stress problem is one in which the thickness is normally small compared to the profile.

The plane strain The plane strain is said when the strain in one direction is much less than the strain in the two orthogonal directions: the smallest strain is ignored.

An other definition is that a plane strain problem is one in which the thickness is normally very large compared to the cross section.

Stress-strain relationship The relationship between stress and strain is the following:

$$\sigma = \mathbf{E}\epsilon + \sigma_0$$

where $\sigma = (\sigma_x \ \sigma_y \ \sigma_{xy})^T$ with the meaning of $\sigma_x = \sigma_{xx}$ and $\sigma_y = \sigma_{yy}$ as previously defined; $\epsilon = (\epsilon_x \ \epsilon_y \ \gamma_{xy})^T$, $\sigma_0 = \mathbf{E}\epsilon_0$ (that we consider negligible in the following).

For a plane stress problem \mathbf{E} is equal to:

$$\mathbf{E} = \frac{E}{1 - \nu^2} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1 - \nu)/2 \end{pmatrix}$$

For a plane strain problem:

$$\mathbf{E} = \frac{E}{(1 + \nu)(1 - 2\nu)} \begin{pmatrix} (1 - \nu) & \nu & 0 \\ \nu & (1 - \nu) & 0 \\ 0 & 0 & (1 - 2\nu)/2 \end{pmatrix}$$

The parameters E and ν are the Young modulus and the Poisson ratio respectively.

B.2 The MLPG and the linear elasticity problem

Equation (31) can be written as

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + b_1 = 0 \quad (34)$$

$$\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + b_2 = 0 \quad (35)$$

In matrix form we get:

$$\overline{D}^T \sigma + \vec{b} = 0$$

where $\vec{b} = (b_1, b_2)^T$.

By taking into account the stress-strain relation and the strain-displacement relation we have:

$$\overline{D}^T \sigma + \vec{b} = \overline{D}^T \mathbf{E} \epsilon + \vec{b} = \overline{D}^T \mathbf{E} \overline{D} \vec{u} + \vec{b} = 0$$

Let us consider a generalized local weak form of the system (34) and boundary conditions (18), with $\vec{u} = (u_1, u_2)^T$ and $\vec{v} = (v_1, v_2)^T$ the trial and test function, respectively:

$$\int_{\Omega_s} \left(\frac{\partial \sigma_x}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + b_1 \right) v_1 \, d\Omega - \alpha \int_{\Gamma_{su}} (u_1 - \bar{u}_1) v_1 \, d\Gamma = 0 \quad (36)$$

$$\int_{\Omega_s} \left(\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + b_2 \right) v_2 \, d\Omega - \alpha \int_{\Gamma_{su}} (u_2 - \bar{u}_2) v_2 \, d\Gamma = 0 \quad (37)$$

Taking into account the relationship

$$\frac{\partial fg}{\partial x} = \frac{\partial f}{\partial x} g + f \frac{\partial g}{\partial x}$$

where f and g are arbitrary differentiable functions, applying the divergence theorem, and following the same lines of Section 6, we obtain the system:

$$\int_{\partial\Omega_s} (\sigma_x v_1 n_x + \sigma_{xy} v_1 n_y) \, d\Gamma - \int_{\Omega_s} \left(\sigma_x \frac{\partial v_1}{\partial x} + \sigma_{xy} \frac{\partial v_1}{\partial y} - b_1 v_1 \right) \, d\Omega - \alpha \int_{\Gamma_{su}} (u_1 - \bar{u}_1) v_1 \, d\Gamma = 0 \quad (38)$$

$$\int_{\partial\Omega_s} (\sigma_{xy} v_2 n_x + \sigma_y v_2 n_y) \, d\Gamma - \int_{\Omega_s} \left(\sigma_{xy} \frac{\partial v_2}{\partial x} + \sigma_y \frac{\partial v_2}{\partial y} - b_2 v_2 \right) \, d\Omega - \alpha \int_{\Gamma_{su}} (u_2 - \bar{u}_2) v_2 \, d\Gamma = 0 \quad (39)$$

Let us now consider the boundary $\partial\Omega_s = L_s \cup \Gamma_{st} \cup \Gamma_{su}$ with the same meaning seen in Section 6. The previous system becomes:

$$\int_{L_s} t_1 v_1 \, d\Gamma + \int_{\Gamma_{su}} t_1 v_1 \, d\Gamma + \int_{\Gamma_{st}} \bar{t}_1 v_1 \, d\Gamma - \int_{\Omega_s} \left(\sigma_x \frac{\partial v_1}{\partial x} + \sigma_{xy} \frac{\partial v_1}{\partial y} - b_1 v_1 \right) \, d\Omega - \alpha \int_{\Gamma_{su}} (u_1 - \bar{u}_1) v_1 \, d\Gamma = 0 \quad (40)$$

$$\int_{L_s} t_2 v_2 \, d\Gamma + \int_{\Gamma_{su}} t_2 v_2 \, d\Gamma + \int_{\Gamma_{st}} \bar{t}_2 v_2 \, d\Gamma - \int_{\Omega_s} \left(\sigma_{xy} \frac{\partial v_2}{\partial x} + \sigma_y \frac{\partial v_2}{\partial y} - b_2 v_2 \right) \, d\Omega - \alpha \int_{\Gamma_{su}} (u_2 - \bar{u}_2) v_2 \, d\Gamma = 0 \quad (41)$$

By choosing the test functions v_i $i = 1, 2$ that vanish on L_s we get:

$$\int_{\Omega_s} (\sigma_x \frac{\partial v_1}{\partial x} + \sigma_{xy} \frac{\partial v_1}{\partial y}) d\Omega + \alpha \int_{\Gamma_{su}} u_1 v_1 d\Gamma - \int_{\Gamma_{su}} t_1 v_1 d\Gamma = \int_{\Gamma_{st}} \bar{t}_1 v_1 d\Gamma + \alpha \int_{\Gamma_{su}} \bar{u}_1 v_1 d\Gamma + \int_{\Omega_s} b_1 v_1 d\Omega \quad (42)$$

$$\int_{\Omega_s} (\sigma_{xy} \frac{\partial v_2}{\partial x} + \sigma_y \frac{\partial v_2}{\partial y}) d\Omega + \alpha \int_{\Gamma_{su}} u_2 v_2 d\Gamma - \int_{\Gamma_{su}} t_2 v_2 d\Gamma = \int_{\Gamma_{st}} \bar{t}_2 v_2 d\Gamma + \alpha \int_{\Gamma_{su}} \bar{u}_2 v_2 d\Gamma + \int_{\Omega_s} b_2 v_2 d\Omega \quad (43)$$

In matrix form we can write:

$$\begin{aligned} & \int_{\Omega_s} \begin{pmatrix} \frac{\partial v_1}{\partial x} & 0 & \frac{\partial v_1}{\partial y} \\ 0 & \frac{\partial v_2}{\partial y} & \frac{\partial v_2}{\partial x} \end{pmatrix} \begin{pmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{pmatrix} d\Omega + \alpha \int_{\Gamma_{su}} \begin{pmatrix} v_1 & 0 \\ 0 & v_2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} d\Gamma - \int_{\Gamma_{su}} \begin{pmatrix} v_1 & 0 \\ 0 & v_2 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} d\Gamma = \\ & = \int_{\Gamma_{st}} \begin{pmatrix} v_1 & 0 \\ 0 & v_2 \end{pmatrix} \begin{pmatrix} \bar{t}_1 \\ \bar{t}_2 \end{pmatrix} d\Gamma + \alpha \int_{\Gamma_{su}} \begin{pmatrix} v_1 & 0 \\ 0 & v_2 \end{pmatrix} \begin{pmatrix} \bar{u}_1 \\ \bar{u}_2 \end{pmatrix} d\Gamma + \int_{\Omega_s} \begin{pmatrix} v_1 & 0 \\ 0 & v_2 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} d\Omega \quad (44) \end{aligned}$$

By setting

$$\epsilon_v = \begin{pmatrix} \frac{\partial v_1}{\partial x} & 0 & \frac{\partial v_1}{\partial y} \\ 0 & \frac{\partial v_2}{\partial y} & \frac{\partial v_2}{\partial x} \end{pmatrix}$$

the strain matrix from the test functions;

$$\mathbf{v} = \begin{pmatrix} v_1 & 0 \\ 0 & v_2 \end{pmatrix}$$

and the vectors

$$\vec{u} = \begin{pmatrix} \bar{u}_1 \\ \bar{u}_2 \end{pmatrix} \quad \vec{t} = \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} \quad \vec{\bar{t}} = \begin{pmatrix} \bar{t}_1 \\ \bar{t}_2 \end{pmatrix}$$

system (44) can be written as:

$$\int_{\Omega_s} \epsilon_v \sigma d\Omega + \alpha \int_{\Gamma_{su}} \mathbf{v} \vec{u} d\Gamma - \int_{\Gamma_{su}} \mathbf{v} \vec{t} d\Gamma = \int_{\Gamma_{st}} \mathbf{v} \vec{\bar{t}} d\Gamma + \alpha \int_{\Gamma_{su}} \mathbf{v} \vec{\bar{u}} d\Gamma + \int_{\Omega_s} \mathbf{v} \vec{b} d\Omega \quad (45)$$

The functions v_i and u_i , $i = 1, 2$ are taken by the MLS approximation. The trial function \vec{u} is of the form:

$$\vec{u} = \sum_{j=1}^n \Phi_j(\mathbf{x}) \tilde{\mathbf{u}}_j$$

where $\tilde{\mathbf{u}}_j = (\tilde{u}_j^1, \tilde{u}_j^2)^T$ represents the displacement along the x and y axis of the j -th node.

Since $\sigma = \mathbf{E}\epsilon = \mathbf{E}\bar{D}\vec{u}$, we get:

$$\sigma = \mathbf{E} \begin{pmatrix} \sum_{j=1}^n \frac{\partial \Phi_j}{\partial x} \tilde{u}_j^1 & 0 \\ 0 & \sum_{j=1}^n \frac{\partial \Phi_j}{\partial y} \tilde{u}_j^2 \\ \sum_{j=1}^n \frac{\partial \Phi_j}{\partial y} \tilde{u}_j^1 & \sum_{j=1}^n \frac{\partial \Phi_j}{\partial x} \tilde{u}_j^2 \end{pmatrix}$$

By denoting with B_j the matrix

$$B_j = \begin{pmatrix} \frac{\partial \Phi_j}{\partial x} & 0 \\ 0 & \frac{\partial \Phi_j}{\partial y} \\ \frac{\partial \Phi_j}{\partial y} & \frac{\partial \Phi_j}{\partial x} \end{pmatrix}$$

the integral $\int_{\Omega_s} \epsilon_v \sigma \, d\Omega$ applied on the local domain relatively to the node \mathbf{x}_i can be equivalently written as $\sum_{j=1}^n \int_{\Omega_s} \epsilon_v(\mathbf{x}, \mathbf{x}_i) \mathbf{E} B_j \tilde{\mathbf{u}}_j \, d\Omega$

About the integrals along Γ_{su} we have to consider that u_1 and u_2 can be assigned or not as Dirichlet conditions over Γ_{su} . To this aim we introduce the matrix

$$S = \begin{pmatrix} S_{u_1} & 0 \\ 0 & S_{u_2} \end{pmatrix}$$

where

$$S_{u_i} = \begin{cases} 1 & \text{if } u_i \text{ is prescribed along } \Gamma_{su} \\ 0 & \text{if } u_i \text{ is not prescribed along } \Gamma_{su} \end{cases} \quad i = 1, 2$$

Therefore the integral $\alpha \int_{\Gamma_{su}} \mathbf{v} \vec{t} \, d\Gamma$ becomes $\sum_{j=1}^n \alpha \int_{\Gamma_{su}} \mathbf{v}(\mathbf{x}, \mathbf{x}_i) S \Phi_j \tilde{\mathbf{u}}_j \, d\Gamma$.

By considering that $\vec{t} = \begin{cases} \sigma_x n_x + \sigma_{xy} n_y \\ \sigma_{xy} n_x + \sigma_y n_y \end{cases}$ it is clear that $\vec{t} = N \sigma = \sum_{j=1}^n N \mathbf{E} B_j \tilde{\mathbf{u}}_j$

where

$$N = \begin{pmatrix} n_x & 0 & n_y \\ 0 & n_y & n_x \end{pmatrix}$$

The integral $\int_{\Gamma_{su}} \mathbf{v} \vec{t} \, d\Gamma$ becomes equal to $\sum_{j=1}^n \int_{\Gamma_{su}} \mathbf{v}(\mathbf{x}, \mathbf{x}_i) N \mathbf{E} B_j S \tilde{\mathbf{u}}_j \, d\Gamma$.

Taking into account all the contributions given by each node, we obtain a system of linear equations :

$$K \mathbf{u} = \mathbf{f}$$

where, for $i, j = 1, \dots, n$

$$K_{ij} = \int_{\Omega_s} \epsilon_v(\mathbf{x}, \mathbf{x}_i) \mathbf{E} B_j \, d\Omega + \alpha \int_{\Gamma_{su}} \mathbf{v}(\mathbf{x}, \mathbf{x}_i) S \Phi_j \, d\Gamma - \int_{\Gamma_{su}} \mathbf{v}(\mathbf{x}, \mathbf{x}_i) N \mathbf{E} B_j S \, d\Gamma$$

$$\mathbf{f}_i = \int_{\Gamma_{st}} \mathbf{v}(\mathbf{x}, \mathbf{x}_i) \vec{t} \, d\Gamma + \alpha \int_{\Gamma_{su}} \mathbf{v}(\mathbf{x}, \mathbf{x}_i) S \vec{u} \, d\Gamma + \int_{\Omega_s} \mathbf{v}(\mathbf{x}, \mathbf{x}_i) \vec{b} \, d\Omega$$

and \mathbf{u} is the vector of $2n$ components $\mathbf{u} = (\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2, \dots, \tilde{\mathbf{u}}_n)^T$.

References

- [1] S. N. ATLURI AND S. SHEN, *The Meshless Local Petrov Galerkin (MLPG) Method*, Tech Science Press, Forsyth, GA, USA, 2002.
- [2] S. N. ATLURI AND T. ZHU, *A new Meshless Local Petrov-Galerkin (MLPG) approach in computational mechanics*, Computational Mechanics, 22 (1998), pp. 117–127.
- [3] —, *New concepts in meshless methods*, Int. J. Numer. Methods Eng., 47 (2000), pp. 537–556.
- [4] T. BELYTSCHKO, Y. KRONGAUZ, D. ORGAN, M. FLEMING, AND P. KRYSL, *Meshless methods: an overview and recent developments*, Comp. Methods App. Mech. Eng., 139 (1996), pp. 3–47.
- [5] H.-K. CHING, *Solution of Linear Elastostatic and Elastodynamic Plane Problems by the Meshless Local Petrov-Galerkin Method*, PhD thesis, Virginia Polytechnic Institute and State University, July 2002.
- [6] Y. Y. LU, T. BELYTSCHKO, AND L. GU, *A new implementation of the element free Galerkin method*, Comp. Methods App. Mech. Eng., 113 (1994), pp. 397–414.
- [7] Y. X. MUKHERJEE AND S. MUKHERJEE, *On boundary conditions in the element-free Galerkin method*, Computational Mechanics, 19 (1997), pp. 264–270.