COMPARISON BETWEEN NUMERICAL INTEGRATION TECHNIQUES IN MLPG-1 APPLIED TO 2D ELASTICITY PROBLEMS

Franco De Mello, J. P.

Santiago, J. A. F.

jpfdemello@coc.ufrj.br

santiago@coc.ufrj.br

COPPE / Universidade Federal do Rio de Janeiro

Centro de Tecnologia - Av. Horácio Macedo, 2030, 21941-450, Rio de Janeiro, RJ, Brazil

Abstract. *Numerical integration in the local subdomains is the most complex process in the solution of problems by a truly meshless method, especially if the boundary of the local subdomain surpasses the global boundary of the problem. This paper proposes an alternative integration method and a comparison with the commonly used integration methods. The development of a new integration technique aims an increase in effectiveness, precision, and a reduction in the computational cost of the integration process. Numerical integration in MLPG is performed by quadrature techniques applied over local subdomains. One of the mostly used techniques is the Gauss quadrature. Traditionally, in order to apply the gaussian quadrature, Gauss points are created over the domain and the boundary of the local subdomains. Subsequently, new subdomains are created using the Gauss points as their central point, which are used as the quadrature's domain. Nonetheless, the traditional technique enforces some restrictions on the positioning of subdomains, avoiding the intersection of the subdomain boundary and the global problem boundary. These restrictions are necessary because the traditional method do not embrace subdomains with a circular segment geometry. This paper proposes a technique which adopts the subdomains created by the Moving Least Squares (MLS) method, used to find the shape functions, as the integration domain of the quadrature. Moreover, this technique can be applied to subdomains with a circular segment geometry.*

Keywords: *Meshless, MLPG, Numerical Integration*

Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering P.O. Faria, R.H. Lopez, L.F.F. Miguel, W.J.S. Gomes, M. Noronha (Editores), ABMEC, Florianópolis, SC, Brazil, November 5-8, 2017.

INTRODUCTION

Meshless methods are increasingly gaining credibility in the numerical methods field, as they have been proven to be a reliable alternative to FEM. However, the absence of a mesh discretization incurs in a problem of numerical integration, especially in local methods, such as MLPG. Since MLPG functions have a rational characteristic, it is difficult to perform an efficient integration procedure. This leads to a more demanding routine in terms of computational processing. Furthermore, in what concerns to local-weak formulation methods, such as MLPG, the mismatch between local subdomains and global domain's boundaries consists in another troublesome spot in MLPG analysis. Commonly, the solution to this problem is to create a new subdomain for each integration point and to limit the positioning of the field nodes, so that the local subdomains do not intersect the global boundary.

This work proposes an alternative integration method for the MLPG-1, by applying the numerical integration in the same subdomain used in the Moving Least Squares (MLS) approximation, thus dismissing the need to create a new subdomain for each integration point. In order to prove the consistency of this integration technique, a simple elasticity problem will be solved by MLPG-1 method and compared to the analytical solution.

1. MLS Approximation

The MLPG method uses an approximated function as the solution of the differential equation problem, named trial function. Differently from the Finite Element Methods, the trial function is calculated throughout the solving process by interpolations between the points distributed over the domain.

The interpolation technique used during the MLPG analysis must produce a trial function which satisfies some requirements:

- Locality, having zero value in the regions outside of the local subdomain;
- Continuity, the field variable must be continuous over all analyzed points;
- Consistency, an increase in the number of points must decrease the error between the approximated answer and the analytical answer.

The locality is necessary in order to generate sparse matrices, guaranteeing the computational efficiency of the method. The continuity of the interpolation is essential to assure that the numerical integration can be performed over the weak formulation of the problem. And, finally, the consistency secures the convergence of the weak formulation based in the interpolation method.

In this paper, the interpolation method used is the Moving Least Squares (MLS) method, which is an already established method in MLPG analysis. According to Atluri and Shen (2005), the MLS method has an exquisite precision and computational efficiency, well suitable for what is proposed by Meshfree methods.

Following, a broader explanation over the MLS method.

Supposing a circular neighborhood around a point of interest x_i is delimited. This neighborhood is the local support or the subdomain Ω_i in which the Moving Least Squares (MLS) approximation of the trial function is defined. The approximated form u^h of the function u can be defined, over points arbitrarily distributed inside the subdomain Ω_i, as:

Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering P.O. Faria, R.H. Lopez, L.F.F. Miguel, W.J.S. Gomes, M. Noronha (Editores), ABMEC, Florianópolis, SC, Brazil, November 5-8, 2017.

$$u^{\boldsymbol{h}}(\boldsymbol{x}) = \boldsymbol{p}^T(\boldsymbol{x})\boldsymbol{a}(\boldsymbol{x}) \;\; \forall\, \boldsymbol{x} \in \; \Omega_i \tag{1}$$

where $a(\boldsymbol{x}) = [a_1(\boldsymbol{x}), a_2(\boldsymbol{x}), \dots, a_m(\boldsymbol{x})]^T$ is the vector containing the coefficients that represent function of space coordinates $\boldsymbol{x}$; and $\boldsymbol{p}^T(\boldsymbol{x}) = [p_1(\boldsymbol{x}), p_2(\boldsymbol{x}), \dots, p_m(\boldsymbol{x})]$ is a monomial basis composed by a set of n-continuous polynomials. A complete second order monomial basis will be used in this paper:

$$\boldsymbol{p}^T(\boldsymbol{x}) = \; [1, x, y, x^2, xy, y^2] \tag{2}$$

An approximation which uses the linear basis (2) is suitable to represent any smooth function and its first derivative, and therefore, perfect to be used on elasticity problems.

The radius of the circular subdomain is regulated by the type of monomial basis. The subdomain must include a number of points at least equal to the number of components of the vector $\mathbf{p}(\mathbf{x})$, ensuring the good conditioning of the approximation.

The best approximation for the trial function u^h is obtained by determining $a(\boldsymbol{x})$ through minimizing the weighted discrete L_2 norm:

$$J(\boldsymbol{x}) = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} w_i(\boldsymbol{x})[u^h - \hat{u}_{\boldsymbol{i}}]^2 \tag{3}$$

Substituting (1) in (3):

$$J(\boldsymbol{x}) = \sum_{i=1}^{n} w_i(\overline{\boldsymbol{x}})[\boldsymbol{p}^T(x_i)\boldsymbol{a}(\boldsymbol{x}) - \hat{u}_i]^2 \tag{4}$$

where $w_i(\boldsymbol{x})$ is the weight function associated with the position $\boldsymbol{x}$ of the points inside the subdomain Ω_i; n is the number of points inside the subdomain; $\hat{u}_i$ represents fictional nodal values of the function u.

The vector $\boldsymbol{p}^T(x_i)$ is related to the coordinates of the field points inside the subdomain analyzed.

From the minimization of (3) and substituting it into (1):

$$\boldsymbol{u}^h = \sum_{i=1}^{n} \phi_i(\boldsymbol{x})\hat{u}_i \,, \quad \boldsymbol{x} \in \boldsymbol{\Omega_x} \tag{5}$$

Which represents the approximation of the trial function by in the form of an interpolation function. $\phi_i(\boldsymbol{x})$ is the vector of Shape functions of the MLS approximation associated with the nodal point x_i.

$$\phi_i(\boldsymbol{x}) = \sum_{\boldsymbol{j=1}}^{\boldsymbol{m}} \boldsymbol{p_j}\,(\boldsymbol{x})[\boldsymbol{A}^{-1}(\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x})]_{ji} \tag{6}$$

in which the matrices $\boldsymbol{A}(\boldsymbol{x})$ and $\boldsymbol{B}(\boldsymbol{x})$ are defined by:

$$\boldsymbol{A}(\boldsymbol{x}) = \sum_{i=1}^{\boldsymbol{n}} \boldsymbol{P}^T\boldsymbol{W}(\boldsymbol{x})\boldsymbol{P} \tag{7}$$

$$\boldsymbol{B}(\boldsymbol{x}) = \boldsymbol{P}^T\boldsymbol{W}(\boldsymbol{x}) \tag{8}$$

where, $\boldsymbol{P}$ is the matrix which groups the vectors $\boldsymbol{p}(\boldsymbol{x})$; and $\boldsymbol{W}$ is the matrix that groups the values of $w_i(\boldsymbol{x})$. Both expressed by the following forms:

$$\boldsymbol{P} = \begin{bmatrix} p_1(x_1) & p_1(x_1) & \cdots & p_1(x_1) \\ p_1(x_1) & p_1(x_1) & \cdots & p_1(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(x_1) & p_1(x_1) & \cdots & p_1(x_1) \end{bmatrix} \tag{9}$$

$$\boldsymbol{W}(\boldsymbol{x}) = \begin{bmatrix} w(x-x_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w(x-x_n) \end{bmatrix} \tag{10}$$

The weight function must follow some specifications: $w_i(\boldsymbol{x}) > 0$ for every $\boldsymbol{x}$ inside the local support and $w_i(\boldsymbol{x}) = 0$ for every $\boldsymbol{x}$ outside.

The weight function used in this paper is a 4th order spline function, defined as:

$$w_i(x) = \begin{cases} 1 - 6\left(\frac{r_j}{R_i}\right)^2 + 8\left(\frac{r_j}{R_i}\right)^3 - 3\left(\frac{r_j}{R_i}\right)^4, & 0 \le r_j \le R_i \\ 0 \qquad\qquad\qquad\qquad\qquad\quad , & r_j \ge R_i \end{cases} \tag{11}$$

where r_j is the distance between the point base x_i and the field points x_j inside the subdomain Ω_i; and R_i is the radius of the subdomain c.

Therefore, the radius R_i delimitates de neighborhood of the point x_i, regulating the area of influence of the weight function $w_i(x)$. Since the shape function vector ϕ_i is related to the weight function $w_i(x)$, $\phi_i = 0$ for every point outside the subdomain Ω_i. Thus, guarantying the local sense of the Moving Least Squares approximation.

In this paper, the radius R_i is calculated by an iterative method, starting with an initial value equal to the smallest distance between two points increasing until the subdomain is large enough to contain at least the same number of terms of the linear basis (2), a necessary condition for a well-conditioned matrix $\boldsymbol{A}$.

The size of the support must be small enough to maintain the regularity of the matrix $\boldsymbol{A}$**,** however a subdomain too small may cause numerical errors during the numerical integration (Gaussian Quadrature). However, subdomains very large results in the loss of the local character of the approximation.

2. MLPG-1 in Elasticity

Differently from predecessor meshless methods, such as Element Free Galerkin, the Meshless Local Petrov-Galerkin method a is truly meshless method, since the numerical integrations are performed without auxiliary background meshes. Moreover, MLPG method is based on a local weak formulation over small subdomains, thus the method has a sense of locality inside the domain.

The MLPG can be presented in six different version, depending on the choice of the trial and weight functions used in during the analysis. This paper will use the MPLG-1, which obtains the govern equations of the discrete problem by a symmetric local-weak formulation, utilizes as test function the weight function from the MLS approximation.

Since in the MLPG method does not exist mesh nor element, nodes arbitrarily distributed are used to represent the domain and the boundary of the analysed problem. The global domain is, then, modified in small overlapping subdomains, defined by an influence neighborhood around a point, namely base point. Nevertheless, it is important to guarantee that the union of all subdomains covers the whole domain in order to avoid integration errors.

The local-weak formulation used in this meshless method is obtained through the Weighted Residual Method, however, this local-weak formulation is restricted, and only valid, to the subdomains. Therefore, the formulation must be only satisfied locally for each subdomain, and it must be integrated over each subdomain independently.

This necessity to confine the integration inside the local supports, with arbitrary geometries, caused a distress working with the Galerkin formulation, which picks the weight and the test functions from the same space of functions. So, to increase the variability of possible function, the MLPG uses the Petrov-Galerkin formulation, which picks weight and test functions from different spaces. (Liu G.R., 2010).

Next, it will be presented the MLPG-1 formulation for 2D elasticity problems.

Considering, a two-dimensional problem in linear elasticity on the domain Ω limited by the boundary Γ:

$$\sigma_{ij,j} + b_i = 0 \qquad em\ \Omega_s \tag{12}$$

With the corresponding boundary conditions:

$$u_i = \bar{u}_i \qquad em\ \Gamma_u \tag{13}$$

$$t_i \equiv \sigma_{ij}\boldsymbol{n} = \bar{t}_i \qquad em\ \Gamma_t \tag{14}$$

where σ_{ij} is the stress tensor corresponding to the displacement field u_i; b_j is the body forces vector; $\bar{u}_i$ and $\bar{t}_i$ correspond, respectively, to the prescribed displacements and the prescribed tractions over the displacement boundary Γ_u and the traction boundary Γ_t.

Considering the weak formulation over a local subdomain Ω_i:

$$\int_{\Omega_s} v_i \left(\frac{\partial \sigma_{ij}}{\partial x_j} + b_j\right) d\Omega - \alpha \int_{\Gamma_{su}} v_i (u_i - \bar{u}_i) d\Gamma = 0 \tag{15}$$

in which, v_i is the test or ponderation function; α is the penalty coefficient.

Since the MLPG-1 uses the weight function w_i as the ponderation function of the weak formulation, the function to be integrated have a non-polynomial aspect. The rational behavior of the weight function produces a difficulty into applying the displacement boundary conditions over the problem. The penalty method is, then, used to impose the boundary conditions.

Applying the divergence theorem on (15):

$$\int_{\Omega_s} \sigma_{ij} \frac{\partial v_i}{\partial x_j} d\Omega - \int_{\Omega_s} v_i b_j d\Omega - \int_{\partial\Omega_s} v_i \sigma_{ij} n\, d\Gamma + \alpha \int_{\Gamma_{su}} v_i (u_i - \bar{u}_i) d\Gamma = 0 \tag{16}$$

in which $\partial\Omega_s$ is the boundary of the subdomain Ω_s, being expressed by $\partial\Omega_s = L_s \cup \Gamma_s$; and Γ_s is the part of the subdomain boundary that intersects with the global domain, $\Gamma_s = \partial\Omega_s \cap \Gamma$, and consequentially L_s is the subdomain boundary which does not intersects with the global boundary, as shown in the Figure 1:

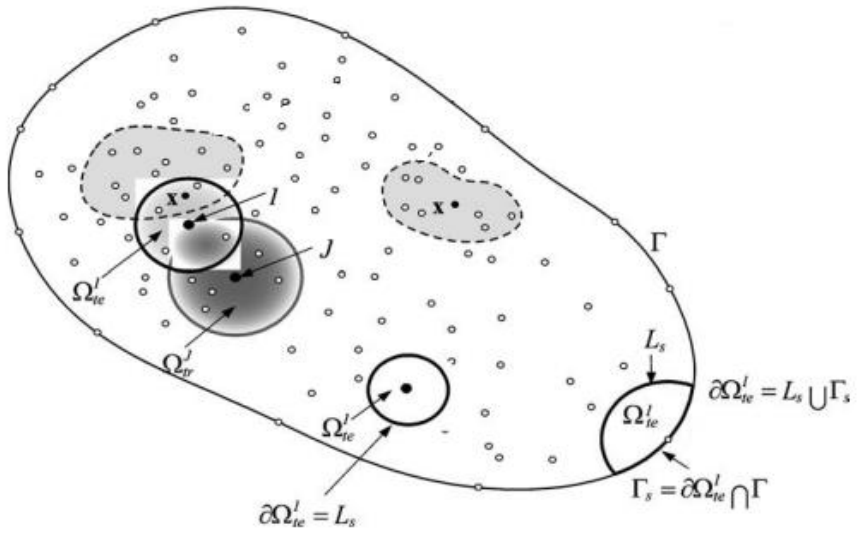

Figure 1: Representation of Global Domain and Subdomains. (Atluri e Shen, 2005)

Substituting the boundary conditions (13) and (15) in (16):

$$\int_{\Omega_s} \frac{\partial v_i}{\partial x_j}\sigma_{ij}d\Omega - \int_{\Omega_s} v_i b_j d\Omega + \alpha \int_{\Gamma_{su}} v_i(u_i - \bar{u}_i)d\Gamma - \int_{\Gamma_{St}} v_i\, \bar{t}_i d\Gamma - \int_{\Gamma_{Su}} v_i\, t_i d\Gamma - \int_{L_s} v_i\, t_i d\Gamma = 0 \quad (17)$$

As expressed before, trial function u_i and test function v_i can be picked from different spaces, therefore, it is possible to choose a v_i such as it has value equal to zero over L_s, nullifying the last term of (15):

$$\int_{\Omega_s} \frac{\partial v_i}{\partial x_j}\sigma_{ij}d\Omega + \alpha \int_{\Gamma_{su}} v_i u_i d\Gamma - \int_{\Gamma_{Su}} v_i\, t_i d\Gamma = \int_{\Gamma_{St}} v_i\, \bar{t}_i d\Gamma + \alpha \int_{\Gamma_{su}} v_i \bar{u}_i d\Gamma + \int_{\Omega_s} v_i b_j d\Omega \quad (18)$$

Evaluating (16) for each field point in the global domain and writing it in matrix form:

$$\int_{\Omega_s} \boldsymbol{\varepsilon}_v \boldsymbol{\sigma} d\Omega + \alpha \int_{\Gamma_{su}} \boldsymbol{v}\boldsymbol{u} d\Gamma - \int_{\Gamma_{Su}} \boldsymbol{v}\boldsymbol{t}\, d\Gamma = \int_{\Gamma_{St}} \boldsymbol{v}\bar{\boldsymbol{t}}\, d\Gamma + \alpha \int_{\Gamma_{su}} \boldsymbol{v}\bar{\boldsymbol{u}} d\Gamma + \int_{\Omega_s} \boldsymbol{v}\boldsymbol{b} d\Omega \quad (19)$$

in which $\boldsymbol{\varepsilon}_v$ is the strain matrix from the test functions v_i; and $\boldsymbol{\sigma}$ is the stress vector from the trial functions u_i. Since this paper solves a 2D elasticity problem, we have:

$$\boldsymbol{\sigma} = \begin{Bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\end{Bmatrix};\ \boldsymbol{\varepsilon}_v = \begin{bmatrix}\frac{\partial v_1}{dx_1} & 0 & \frac{\partial v_1}{dx_2}\\ 0 & \frac{\partial v_2}{dx_2} & \frac{\partial v_2}{dx_1}\end{bmatrix} \quad (20)$$

$$\boldsymbol{v} = \begin{bmatrix}v_{11} & v_{12}\\ v_{21} & v_{22}\end{bmatrix};\ \boldsymbol{u} = \begin{Bmatrix}u_1\\ u_2\end{Bmatrix};\ \bar{\boldsymbol{u}} = \begin{Bmatrix}\bar{u}_1\\ \bar{u}_2\end{Bmatrix};\ \boldsymbol{t} = \begin{Bmatrix}t_1\\ t_2\end{Bmatrix};\ \bar{\boldsymbol{t}} = \begin{Bmatrix}\bar{t}_1\\ \bar{t}_2\end{Bmatrix};\ \boldsymbol{b} = \begin{Bmatrix}b_1\\ b_2\end{Bmatrix} \quad (21)$$

Theoretically, the equilibrium equations and the boundary conditions from (17) will be fulfilled as long as the whole global domain is covered by local subdomains.

3. Discretization

Equation (17) can be represented in a way which the only unknown is the displacement u, by introducing the stress-strain and strain-displacement correlations, respectively:

$$\boldsymbol{\sigma} = \boldsymbol{D}\boldsymbol{\varepsilon} \tag{22}$$

$$\boldsymbol{\varepsilon} = \boldsymbol{\mathcal{B}}\boldsymbol{u} \tag{23}$$

in which $\boldsymbol{D}$ is the material properties matrix; and $\boldsymbol{\mathcal{B}}$ is a differential operator, defined by:

$$\boldsymbol{\mathcal{B}} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \tag{24}$$

In linear elasticity, the matrix $\boldsymbol{D}$ can be defined by:

$$\boldsymbol{D} = \frac{E}{1-\upsilon^2}\begin{bmatrix} 1 & \upsilon & 0 \\ \upsilon & 1 & 0 \\ 0 & 0 & \frac{1-\upsilon}{2} \end{bmatrix} \tag{25}$$

for plane stress state problems, or:

$$\boldsymbol{D} = \frac{E}{(1+\upsilon)(1-2\upsilon)}\begin{bmatrix} 1-\upsilon & \upsilon & 0 \\ \upsilon & 1-\upsilon & 0 \\ 0 & 0 & \frac{1-2\upsilon}{2} \end{bmatrix} \tag{26}$$

for plane strain state problems. In both cases, E is the elasticity modulus of the material and υ is the Poisson's ratio.

Moreover, the unknown $\boldsymbol{u}$ is replaced by its approximated form (5), obtained from the MLS approximation. And the test function v is replaced by the weight function w_i also from the MLS approximation.

After these considerations, the Eq. (17) has the following form:

$$\sum_{j=1}^{NP}\int_{\Omega_s}\boldsymbol{\varepsilon_w}\boldsymbol{D}\boldsymbol{\mathcal{B}}\,\boldsymbol{\phi_j}\hat{\boldsymbol{u}}_{\boldsymbol{j}}\,d\Omega_s + \alpha\sum_{j=1}^{NP}\int_{\boldsymbol{\Gamma_{su}}}\boldsymbol{w_i}(\boldsymbol{x})\boldsymbol{S}\,\boldsymbol{\phi_j}\hat{\boldsymbol{u}}_{\boldsymbol{j}}d\Gamma - \sum_{j=1}^{NP}\int_{\boldsymbol{\Gamma_{su}}}\boldsymbol{w_i}(\boldsymbol{x})\boldsymbol{NDS}\,\boldsymbol{\mathcal{B}}\boldsymbol{\phi_j}\hat{\boldsymbol{u}}_{\boldsymbol{j}}d\Gamma = \int_{\boldsymbol{\Gamma_{st}}}\boldsymbol{w_i}(\boldsymbol{x})\bar{\boldsymbol{t}}d\Gamma + \alpha\int_{\boldsymbol{\Gamma_{su}}}\boldsymbol{w_i}(\boldsymbol{x})\boldsymbol{S}\bar{\boldsymbol{u}}\,d\Gamma + \int_{\Omega_s}\boldsymbol{w_i}(\boldsymbol{x})\boldsymbol{b}\,d\Omega_s \tag{27}$$

where **N** is the matrix that contains the normal vectors from every point on the boundary; and **S** is the matrix used to verified if the field points on the boundary have a Dirichlet boundary condition attached. Thus, **N** and **S** can be described as:

$$\boldsymbol{N} = \begin{bmatrix} n_x & 0 & n_y \\ 0 & n_y & n_x \end{bmatrix} \tag{28}$$

$$\boldsymbol{S} = \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix} \tag{29}$$

The terms of the matrix **S** can be equal to 1 if the displacement in the direction x or y is prescribed.

Equation (25) can be expressed as a linear algebraic system, very similar to the one generated by FEM analysis:

$$\sum_{j=1}^{NP}\boldsymbol{K}_{ij}\hat{\boldsymbol{u}}_j = \boldsymbol{f}_i\,,\quad i = 1,2,\dots,N \tag{30}$$

in which, N is the number of nodes,

$$\boldsymbol{K}_{ij} = \int_{\Omega_s} \boldsymbol{\varepsilon_w D B}\, \boldsymbol{\phi_j}\, d\Omega_s + \alpha \int_{\boldsymbol{\Gamma_{su}}} \boldsymbol{w_i}(\boldsymbol{x}) \boldsymbol{S}\, \boldsymbol{\phi_j} d\Gamma - \int_{\boldsymbol{\Gamma_{su}}} \boldsymbol{w_i}(\boldsymbol{x}) \boldsymbol{NDS}\, \boldsymbol{\mathcal{B}} \boldsymbol{\phi_j} d\Gamma \tag{31}$$

$$\boldsymbol{f}_i = \int_{\boldsymbol{\Gamma_{st}}} \boldsymbol{w_i}(\boldsymbol{x}) \bar{\boldsymbol{t}} d\Gamma + \alpha \int_{\boldsymbol{\Gamma_{su}}} \boldsymbol{w_i}(\boldsymbol{x}) \boldsymbol{S} \bar{\boldsymbol{u}}\, d\Gamma + \int_{\Omega_s} \boldsymbol{w_i}(\boldsymbol{x}) \boldsymbol{b}\, d\Omega_s \tag{32}$$

From Eq. (29) is possible to affirm that the locations of the null values will depend on the nodes subdomains. In 2D elasticity, every field point x_i will be represented by two lines from the matrix **K**, one for each direction, and the columns with non-zero values will be the ones that represents the nodes x_j encircled by the subdomain Ω_i.

4. Numerical Integration

Numerical integration is the costly process in a MLPG analysis, due to the non-polynomial character and the lack of the Kronecker's delta properties on the functions (integrant).

This process is responsible to guarantee the convergence of the numerical solution. Therefore, optimize the integration techniques is fundamental to endorse the solution of more robust problems and development of more efficient algorithms.

The numerical integration performed by the MLPG uses the local subdomains generated in the MLS approximation as its domain. The literature presents diverse integration techniques; however, this paper will use specifically the Gauss-Legendre Quadrature technique.

Generally, numerical integration rules approximate continuous integrals on the domain Ω through a discrete sum:

$$\int_{\Omega} f(\xi) \approx \sum_{i=1}^{n_q} f(\hat{\xi}_i) W_i \tag{33}$$

where, n_q is the number of Gauss point; $\hat{\xi}_i$ is the coordinate of the Gauss point; and W_i is the weight attributed to Gauss point.

The Gaussian quadrature is a well-known technique, commonly used in several meshless methods (Belytschko T. et al., 1996). Its popularity is due to its capacity of integrate exactly unidimensional polynomials, allowing the exact integration of polynomial with order $2n_q - 1$ through a Gaussian Quadrature rule with order n_q (Stroud e Secrest ,1966).

However, as said before, MLPG functions have a rational, and not a polynomial, characteristic. And, furthermore, local subdomains with circular or circular section geometry generate a disequilibrium between the subdomain and the global boundaries, as show in Figure 2. This paper will use only rectangular domains.

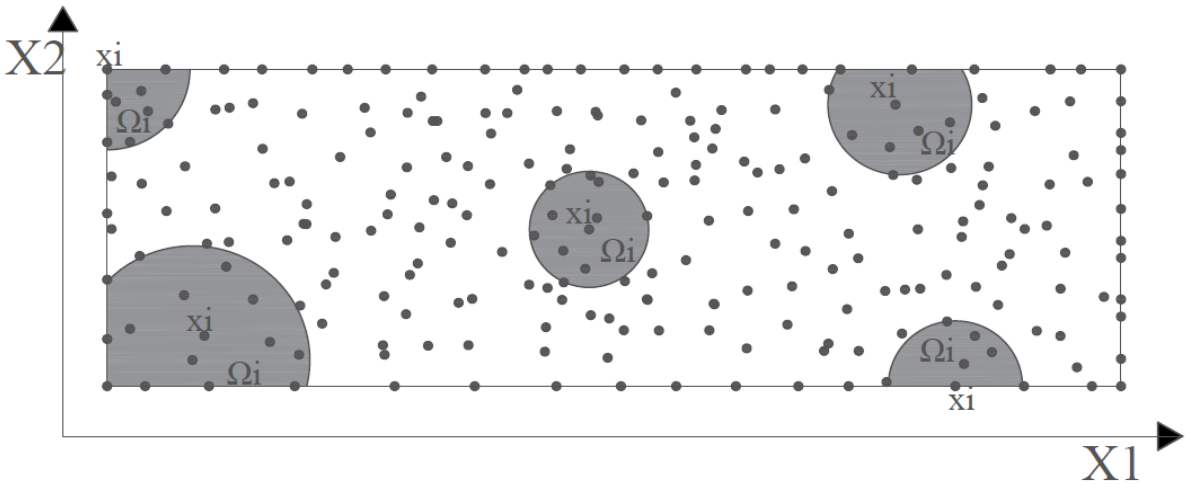

Figure 2

This mismatch between local and global boundaries is the main issue discussed in this paper. Commonly, to avoid this problem, the numerical integration at MLPG-1 limits the positioning of nodes, in a way which the local subdomains never intersect the global boundary, enabling only complete circular subdomains, as described by Mazzia et al. (2007). Furthermore, it generates other local subdomains for each Gauss point, and afterwards, performs a MLS approximation using these subdomains as its support, increasing the number of operations necessary to solve a problem and diminishing the computational efficiency.

This paper proposes an alternative approach to the numerical integration. The position of the nodes is not limited, local supports may surpass the global boundaries and the integral limits for the numerical integration are the supports used during the MLS approximation.

Despite leading to more delicate geometries to be integrated, the employment of several similar parameters already calculated for the MLS approximation may introduce a more efficient solution method. The only change between the parameters already calculated at MLS approximation is the substitution of the vector $\boldsymbol{p}^T(x_i)$ related to field points for the vector $\boldsymbol{p}^T(ng_i)$ related to each Gauss point inside the subdomain. Therefore, it is not necessary to calculate a matrix A and B for each integration point, utilizing, instead, the matrix A and B as a constant inside the local subdomains. Nevertheless, the integration procedure must recalculate $\boldsymbol{\phi_j}$ and $\boldsymbol{w_i}$ for every integration point inside every local support.

Due to the local sense of the MLPG-1 method, each field point x_i is analyzed separately; consequently, every subdomain is integrated separately. Therefore, it is necessary to perform a coordinate translation from a global system to a local system centered in each x_i, Figure 3.a and 3.b. Since, the local subdomains have a circular geometry, a transformation from cartesian coordinates to polar coordinates is performed to simplify the calculations, Figure 3.c.

Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering P.O. Faria, R.H. Lopez, L.F.F. Miguel, W.J.S. Gomes, M. Noronha (Editores), ABMEC, Florianópolis, SC, Brazil, November 5-8, 2017.

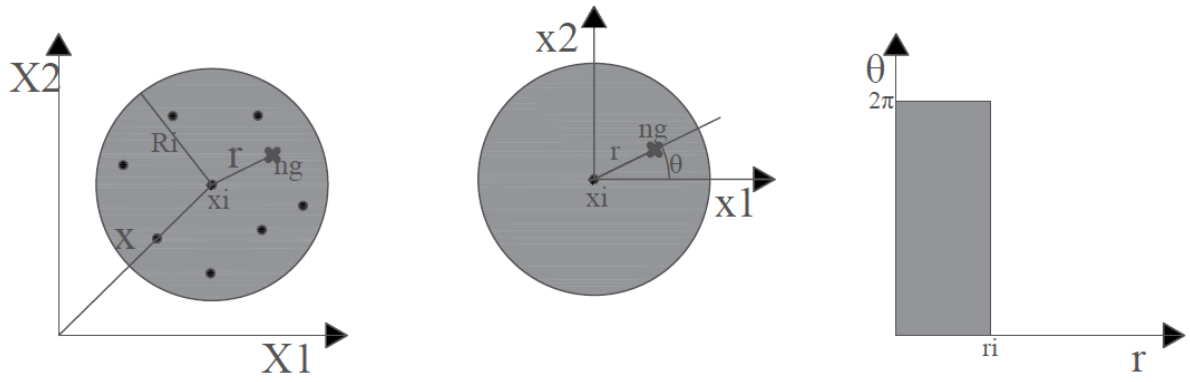

Figure 3

Initially, we can represent the local subdomain in cartesian coordinates:

$$x_1^2 + x_2^2 = R_i^2 \tag{34}$$

where R_i is the radius of the subdomain Ω_i.

Performing a transformation to a polar coordinates system:

$$0 \leq r \leq R_i \tag{35.a}$$

$$\theta_1 \leq \theta \leq \theta_2 \tag{35.b}$$

with the following relation between polar and cartesian systems:

$$x_1 = rcos(\theta) \tag{36.a}$$

$$x_2 = rsin(\theta) \tag{36.b}$$

where r is the radial distance between the point of Gauss and x_i; and θ the angle of the Gauss point. θ_1 and θ_2 are the limits of the subdomain, depicted in Figure 4.

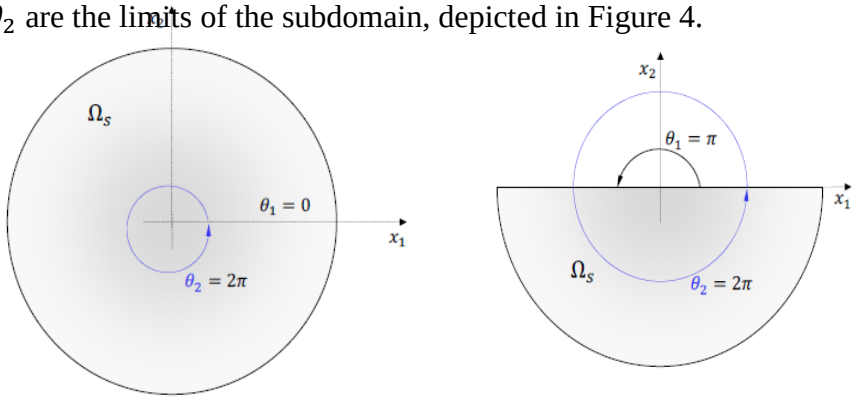

Figure 4: Definition of θ_1 and θ_2. (Edivaldo,2014)

The determinant of the Jacobian of this transformation:

$$|\boldsymbol{J_1}| = \begin{vmatrix} \frac{\partial x_1}{\partial r} & \frac{\partial x_2}{\partial r} \\ \frac{\partial x_1}{\partial \theta} & \frac{\partial x_2}{\partial \theta} \end{vmatrix} = \begin{vmatrix} \cos(\theta) & \sin(\theta) \\ -rsin(\theta) & rcos(\theta) \end{vmatrix} = r \tag{37}$$

However, a more generic implementation is necessary to simplify the Gaussian Quadrature. Therefore, the subdomain is represented by a normalized quadrilateral [-1,1] × [-1,1]. The polar coordinates are related with the normalized variables ξ and η by the following expression:

$$r = \frac{R_i}{2}(1+\xi), \;\; \xi \in [-1,1] \tag{38.a}$$

$$\theta = \left(\frac{\theta_2-\theta_1}{2}\right)\eta + \frac{\theta_1-\theta_2}{2}, \;\; \eta \in [-1,1] \tag{38.b}$$

The determinant of the Jacobian of this transformation:

$$|\boldsymbol{J_2}| = \begin{vmatrix} \frac{\partial r}{\partial \xi} & \frac{\partial r}{\partial \eta} \\ \frac{\partial \theta}{\partial \xi} & \frac{\partial \theta}{\partial \eta} \end{vmatrix} = \begin{vmatrix} \frac{R_i}{2} & 0 \\ 0 & \frac{\theta_2-\theta_1}{2} \end{vmatrix} = \frac{R_i(\theta_2-\theta_1)}{4} \tag{39}$$

After these transformations and using a usual Gauss-Legendre Quadrature, Eq. 29 and Eq. 30 can be expressed by:

$$\boldsymbol{K}_{ij} = |\boldsymbol{J_2}||\boldsymbol{J_1}| \sum_{j=1}^{ng}\sum_{i=1}^{ng} \alpha_i\alpha_j \, \boldsymbol{\varepsilon_w}(\xi,\eta)\boldsymbol{D}\boldsymbol{\mathcal{B}}\boldsymbol{\phi_j}(\xi,\eta) + \alpha|\boldsymbol{J_3}| \sum_{i=1}^{ng} \alpha_i \, \boldsymbol{w_i}(\xi)\boldsymbol{S}\boldsymbol{\phi_j}(\xi,\eta) + |\boldsymbol{J_3}| \sum_{i=1}^{ng} \alpha_i \, \boldsymbol{w_i}(\xi)\boldsymbol{N}\boldsymbol{S}\boldsymbol{D}\boldsymbol{\mathcal{B}}\boldsymbol{\phi_j}(\xi,\eta) \tag{40}$$

$$\boldsymbol{f}_i = |\boldsymbol{J_3}| \sum_{i=1}^{ng} \alpha_i \boldsymbol{w_i}(\xi)\bar{\boldsymbol{t}} + \; \alpha|\boldsymbol{J_3}| \sum_{i=1}^{ng} \alpha_i \boldsymbol{w_i}(\xi)\boldsymbol{S}\bar{\boldsymbol{u}} + |\boldsymbol{J_3}| \sum_{i=1}^{ng}\sum_{j=1}^{ng} \alpha_i\alpha_j \boldsymbol{w_i}(\xi,\eta)\boldsymbol{b} \tag{41}$$

The boundary terms in Eq.29 and Eq.30 suffers a linear transformation to the normalized domain ξ [-1,1], and the determinant of the Jacobian of this transformation is:

$$|\boldsymbol{J_3}| = \frac{d_i}{2} \tag{42}$$

in which, d_i is the extent of the boundary of the subdomain Ω_i.

A formulation was used to determine the number of Gauss points should be created for each subdomain. According to Pecher (2006), an optimal number of Gauss points in a 2D circle is equal to:

$$ng = m\big(m - mod_2(m)\big) + mod_2(m) \tag{43}$$

where, $mod_2(m)$ is the remainder on the division of m by 2.

5. Numerical Application

The problem analyzed by this paper is a cantilever beam, supported at one end and a concentrated load applied at the free end, portrayed by Figure 5.

Figure 5: Cantilever beam with concentrated load at the free end.

The beam's geometrical properties are 1000 mm x 100 mm. The Young's modulus for the material is equal to 205000 MPa, while the Poisson's ratio equals to 0.25. There is an applied load of 1 N acting in the opposite extreme of the support.

A set of 427 was used to discretized the problem above, equally distributed by 7 nodes height and 61 nodes length.

Another parameter considered was the penalty coefficient α is 10E+14, in order to force the boundary conditions on the supported end.

Gauss-Legendre quadrature was applied over each local subdomain using 49 (forty-four) Gaussian points, seven in each axis. As a result, the distribution of integration points inside each local subdomain is represented by Figure 6.

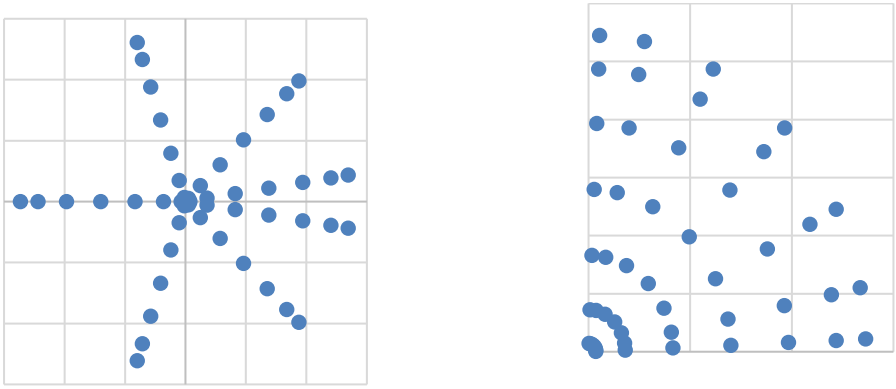

Figure 6: Integration points for a subdomain located totally inside the global domain and for a subdomain located on a corner.

The comparison between the displacement determined by an analytical solution and by the MLPG-1 method with the alternative method described above is portrayed in Figure 7.

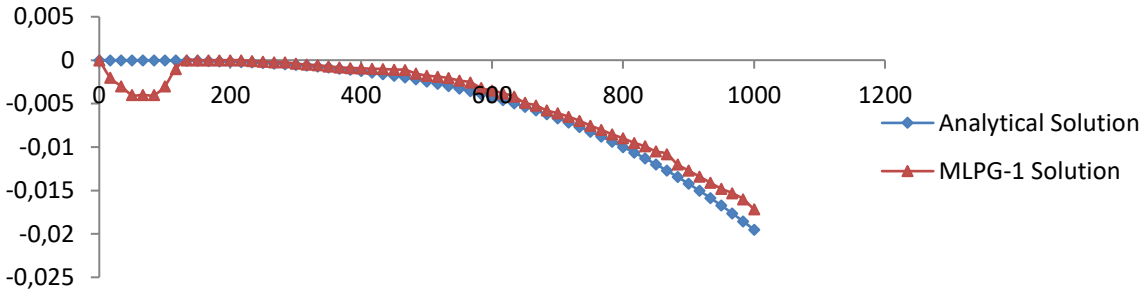

Figure 7: Comparison of the displacements.

Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering P.O. Faria, R.H. Lopez, L.F.F. Miguel, W.J.S. Gomes, M. Noronha (Editores), ABMEC, Florianópolis, SC, Brazil, November 5-8, 2017.

A difference of 12% between the analytical and the numerical solution is noticed. This error is probably caused by the penalty method used to impose the boundary conditions at the supported end. The application of a non-optimal α may induce to error nearby the area with imposed Dirichlet conditions.

6. Conclusion

Although there is still a perturbation in the aforementioned area, this error is expected to be eliminated until the presentation date.

Besides the integration technique described in this work, it is necessary to emphasize two different methods which can be used as another integration procedure.

The first one, already mentioned in this work, uses different subdomains for the MLS approximation and for the numerical integration. The field node's subdomain is delimited by the distance to the closest field node, spreading Gauss points over it and applying the whole procedure presented in this work for each Gauss point. This integration process is very time consuming and can become impracticable for large problems.

Another integration scheme, not as time consuming, uses the same techniques previously described in this work; however, it uses a reduced version of the MLS' subdomain, therefore using the same field node as base point and all the matrices computed by the MLS approximation. The goal of using a reduced subdomain is to reduce the number of Gauss points necessary to represent efficiently the problem. The reduced integration subdomain must be delimited by the distance to the closest field node, in order to guarantee full coverage of the global domain. Nevertheless, further studies are required to fully evaluate the efficiency and accuracy of this technique.

REFERENCES

Atluri SN, Zhu T (1998b): A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Comput. Mech.* 22: 117-127

Atluri SN, Zhu T (2000): The meshless local Petrov- Galerkin (MLPG) approach for solving problems in elasto-statics, *Comput. Mech.* 25: 169-179.

Atluri SN, Zhu T (2000): New concepts in meshless methods. *Int. J. Numer. Mech. Engrg.* 47: 537-556.

Atluri SN (2002): The Meshless Local Petrov–Galerkin (MLPG) Method, Tech Science Press.

Atluri SN, Shen S (2002): The meshless local Petrov–Galerkin (MLPG) method: a simple and less-costly alternative to the finite element and boundary element methods, Comput. Model Eng. Sci. 3 11–51.

Belytschko T, Krongauz Y, Organ D, Fleming M, Krysl P (1996): Meshless methods: An overview and recent developments. *Comput. Methods Appl. Mech. Engrg.* 139:3-47

Gu YT, Liu GR (2001): A meshless local Petrov-Galerkin (MLPG) formulation for static and free vibration analysis of thin plates. *CMES: Computer Modeling in Engineering & Sciences* 2 (4): 463-476.

Lancaster P, Salkauskas K. (1981): Surfaces generated by moving least squares methods. Math Comput;37:141–58.

Liu GR (2003): Mesh free methods: moving beyond the finite element method. Boca Raton: CRC Press.

Liu GR, Gu YT (2005): An introduction to meshfree methods and their programming. Berlin Heidelberg New York: Springer.

Long S, Atluri SN (2002): A meshless local Petrov-Galerkin (MLPG) method for solving the bending problem of a thin plate. *CMES: Computer Modeling in Engineering & Sciences* 3.

Mazzia,A., Ferronato, M., Pini, G. "A comparison of numerical integration rules for the meshless local Petrov-Galerkin method", *Numerical Algorithms,* v.45, n. 1-4, pp. 61-74, 2007.