DISCUSSION OF THE TEST AND TRIAL FUNCTIONS' RADIUS SIZE ON THE MESHLESS LOCAL PETROV-GALERKIN METHOD APPLIED TO ELASTICITY PROBLEMS

Suzana Matos França de Oliveira

Creto Augusto Vidal

Joaquim Bento Cavalcante-Neto

Laise Lima de Carvalho Sousa

suzanam@lia.ufc.br

cvidal@dc.ufc.br

joaquimb@dc.ufc.br

laise@lia.ufc.br

CRAb – Computer Graphics, Virtual Reality and Animation Group, Department of Computing, Federal University of Ceará (UFC). Humberto Monte ave, Block 910 - Pici campus, 60455-760, Fortaleza, Ceará, Brazil

Tereza Denyse Pereira de Araújo

denyse@ufc.br

Department of Structural Engineering and Construction, Federal University of Ceará (UFC). Humberto Monte ave, Block 710 - Pici campus, 60455-760, Fortaleza, Ceará, Brazil

Abstract. *The meshless local Petrov-Galerkin (MLPG) method has been widely used as an alternative method to the Finite Element Method (FEM) and to the Element Free Galerkin (EFG) method. In MLPG, a set of scattered nodes is used to perform the simulation. Each of those nodes has a trial and a test function, which can belong to different function spaces. We use the method called MLPG5, in which the test function is a Heaviside step function, whose support is used for the integration of the Local Weak Form; and the trial function is approximated using the Moving Least Square (MLS) method. The essential boundary conditions are imposed using the MLS collocation method. We discuss that the choice of radius sizes greatly influences the simulation results.*

Keywords: *meshless method, meshless local Petrov-Galerkin (MLPG) method, numerical simulation*

1 INTRODUCTION

Simulation methods have been studied for a long time. A physically-based simulation helps determine whether a building or any physical object can withstand the predicted loads before a prototype is actually built, since a compatible model can be analyzed. This can help to prevent disasters and reduce the costs of constructing prototypes. Simulations can also be used to produce animations in movies and commercials, when physical realism is taken into consideration.

The governing equations of complex structures usually do not have analytical solutions, so plenty of methods were proposed in order to obtain approximate solutions. The Finite Element Method (FEM) is probably the most popular and widely used method. However, it still has some disadvantages in certain types of simulations (Sladek et al., 2013; Liu, 2009). The FEM requires that the entire domain be discretized with a high-quality mesh composed of elements (e.g.: triangle, quadrilateral, tetrahedron), whose shapes must be as regular as possible in order to guarantee accurate results.

Because of the difficulty of generating good-quality meshes in all situations, a number of Meshless Methods have been developed. In those methods, the domain is represented through a set of scattered points (nodes), with no connection between any pair of nodes, as illustrated in Fig. 2. The nodes can be easily removed or added in the model. Some proposed meshless methods are: smooth particle hydrodynamics (SPH) (Gingold & Monaghan, 1977); element free Galerkin (EFG) (Belytschko et al., 1994); reproducing kernel particle method (RKPM) (Liu et al., 1995); partition of unity finite element method (PUFEM) (Babuska & Melenk, 1997); natural element method (NEM) (Sukumar et al., 1998). The greatest difference between all those Galerkin methods resides in how they construct their shape functions (Nguyen et al., 2008). They do not require a mesh to interpolate the test and trial functions, they need a background mesh in order to integrate the global weak form, which makes them not a true meshless method.

A truly meshless method, named Meshless Local Petrov-Galerkin (MLPG), was proposed by Atluri & Zhu (1998). That method does not use any mesh either to interpolate the trial and test functions, or to integrate the local weak formulation. The integrals are computed over simple subdomains (e.g.: circles, quadrilaterals, spheres), associated with each node. Since it is based on the Petrov-Galerkin approach, the trial and test functions can belong to distinct functional spaces, making this method very flexible (Atluri & Shen, 2002). This attracted lots of researchers in the past few years (Han & Atluri, 2004; Zhang et al., 2006; Liu et al., 2011; Kamranian et al., 2017).

Atluri & Shen (2002) did a convergence study on how the radius sizes influence the error, however they used the linear Poisson's equation instead of a 2D elasto-static problem. Atluri & Zhu (2000) solved elasto-static problems, but they did not use the MLPG5 approach, neither analyzed the radius' influence. Xiao (2004) do some tests in the MLPG method using compactly supported radial basis functions (CSRBF). Hu et al. (2006) do some similar analysis as we do here, though the step used in the tests is spaced, losing some important patterns. The influence of the radii in the results is also observed in the result of elasto-dynamic problems (Long et al., 2006).

In this work, we explain the MLPG method for elastostatics, using the Moving Least Square (MLS) to interpolate the global domain as trial function and the Heaviside step function as test

function. The essential boundary conditions are applied using the MLS collocation method (Liu, 2009). Some analytical results are used to validate the technique.

The paper is organized as follows: In Section 2, we introduce the MLPG method, starting from the strong formulation of the problem to the local weak form. We also explain the trial and test functions, and how to apply the displacement boundary conditions. In Section 3, we present some tests using a cantilever beam as an example for comparing the numerical results with the analytical one. Some conclusions are discussed in Section 4.

2 MESHLESS LOCAL PETROV-GALERKIN

In this section, we explain each part of the MLPG formulation, including: how to calculate the local weak form from the strong formulation, how to create the trial and test functions and how to enforce the essential boundary conditions. The overview is depicted in Fig. 1.

The model is defined by: the object's material; the structure and shape of the object; the essential boundary condition, which defines the prescribed displacements at subregions of the object's boundary; and the natural boundary condition, which defines the prescribed tractions at certain subregions of the object's boundary. The object's domain is represented by a set of scattered nodes, which, through predefined influence functions, contribute to the determination of the field variables at any point of the domain. For example, in Fig. 1, the model represents a steel cantilever beam with length L, height H, thickness b, and a load P applied to its right end. The circular support of the test function is defined by its radius, r_s, and the circular support of the trial function is defined by its radius, r_w. The integration is calculated in the support area of a node's test function, and, since MLPG5 only needs integration on the border of that support region, only that border's integration points are created.

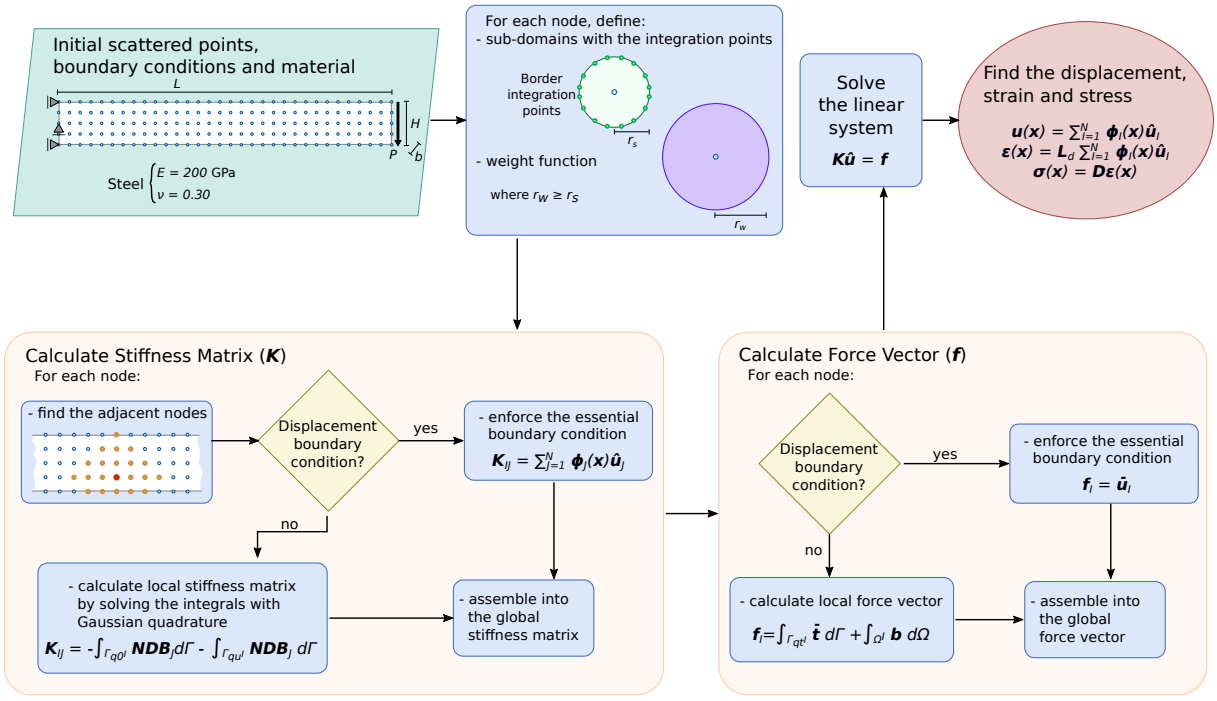


Figure 1: Overview of the technique.

2.1 Local Weak Form

Consider a linear elastic body with domain $\Omega \subset \mathbb{R}^d$ (where $d = 2$ or $d = 3$) and boundary Γ. We assume that the body undergoes infinitesimal deformation, then the equation of equilibrium is given by:

$$\sigma_{ij,j} + b_i = 0 \quad \text{in } \Omega, \tag{1}$$

where σ_{ij} are the components of the Cauchy stress tensor; b_i are the components of the body force; $()_{,j}$ denotes $\frac{\partial()}{\partial x_j}$; and $1 \leq i, j \leq d$. The corresponding boundary conditions are:

$$u_i = \bar{u_i} \quad \text{on } \Gamma_u \tag{2a}$$
$$\sigma_{ij} n_j = \bar{t_i} \quad \text{on } \Gamma_t, \tag{2b}$$

where $\bar{u_i}$ and $\bar{t_i}$ are the prescribed components of displacement and tractions respectively, on the displacement boundary Γ_u and on the traction boundary Γ_t, and n_j is the component of the unit outward normal vector on the boundary Γ.

The components σ_{ij} of the stress tensor relates to the components of the strain tensor ε_{kl} by the generalized Hooke's law (Liu, 2009):

$$\sigma_{ij} = c_{ijkl}\varepsilon_{kl}, \tag{3}$$

where c_{ijkl} denotes the material's elasticity coefficients, which, due to the inherent symmetries of σ and ε, process the following properties:

$$c_{ijkl} = c_{jikl} = c_{ijlk} = c_{klij}.$$

The strain displacement relation is written as:

$$\varepsilon_{ij} = \frac{u_{i,j} + u_{j,i}}{2}, \tag{4}$$

where u_i and u_j are the displacement's components of a point $\mathbf{x}$ in the domain Ω. Thus, we want to find the displacement of any point in the domain that satisfies (1)-(2).

The MLPG approach receives as input a set of scattered nodes, where each node I has a subdomain Ω_q^I and a boundary Γ_q^I, as shown in Fig. 2. A subdomain can have an arbitrary shape, it must be inside the domain Ω and the union of all subdomains should cover the global domain (Atluri & Zhu, 1998). The generalized local weak form of the differential equation (Eq. (1)) over a local subdomain Ω_q, can be written as:

$$\int_{\Omega_q} (\sigma_{ij,j} + b_i) v_i d\Omega = 0, \tag{5}$$

where v_i is the component of the test function, which was chosen from a space that is different the one in which the trial function lies. Applying the divergence theorem to Eq. (5) yields:

$$\int_{\Gamma_q} \sigma_{ij} n_j v_i d\Gamma - \int_{\Omega_q} \sigma_{ij} v_{i,j} d\Omega + \int_{\Omega_q} b_i v_i d\Omega = 0. \tag{6}$$

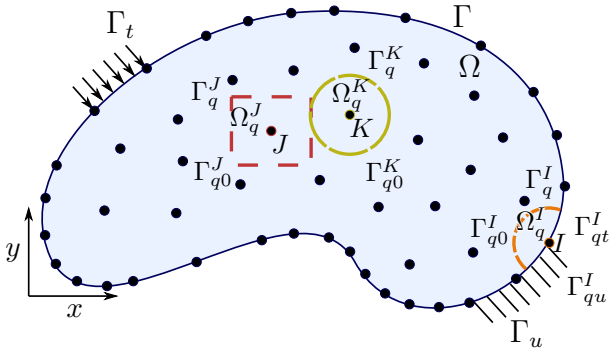


Figure 2: Two-dimensional body with domain Ω and boundary $\Gamma = \Gamma_u \cup \Gamma_t$. Each node has a subdomain Ω_q^I and a boundary $\Gamma_q^I = \Gamma_{qu} \cup \Gamma_{qt} \cup \Gamma_{q0}$, where Γ_{qu} and Γ_{qt} have the prescribed values; and Γ_{q0} is inside the domain.

Separating the boundary Γ_q, that is the union of Γ_{q0}, Γ_{qu} and Γ_{qt}, and using the boundary condition of Eq. (2b), Eq. (6) can be rewritten as:

$$\int_{\Omega_q} \sigma_{ij} v_{i,j} d\Omega - \int_{\Gamma_{q0}} \sigma_{ij} n_j v_i d\Gamma - \int_{\Gamma_{qu}} \sigma_{ij} n_j v_i d\Gamma = \int_{\Gamma_{qt}} \bar{t}_i v_i d\Gamma + \int_{\Omega_q} b_i v_i d\Omega. \tag{7}$$

This is the local weak form of the problem. Notice that, when Ω_q is entirely inside Ω, the integrals of Γ_{qu} and Γ_{qt} vanish, since $\Gamma_q = \Gamma_{q0}$, and Eq. (7) can be simplified to:

$$\int_{\Omega_q} \sigma_{ij} v_{i,j} d\Omega - \int_{\Gamma_{q0}} \sigma_{ij} n_j v_i d\Gamma = \int_{\Omega_q} b_i v_i d\Omega. \tag{8}$$

The local weak form (7) can also be rewritten in matrix form, taking into consideration that the repeated indices imply summation:

$$\int_{\Omega_q} \mathbf{w}^T \boldsymbol{\sigma} d\Omega - \int_{\Gamma_{q0}} \mathbf{v}^T \mathbf{N} \boldsymbol{\sigma} d\Gamma - \int_{\Gamma_{qu}} \mathbf{v}^T \mathbf{N} \boldsymbol{\sigma} d\Gamma = \int_{\Gamma_{qt}} \mathbf{v}^T \bar{\mathbf{t}} d\Gamma + \int_{\Omega_q} \mathbf{v}^T \mathbf{b} d\Omega, \tag{9}$$

where, $\boldsymbol{\sigma}$ is the stress tensor as vector, $\mathbf{u}$ is the displacement vector, $\mathbf{v}$ is the test function vector, $\mathbf{L}_d$ is the differential operators matrix, $\mathbf{N}$ is the unit outward normal vector on the boundary Γ, $\bar{\mathbf{t}}$ is the prescribed traction vector on Γ_{qt}, $\bar{\mathbf{u}}$ is the prescribed displacement vector on Γ_{qu} and $\mathbf{b}$ is the body force vector. For example, in a two-dimensional case:

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{Bmatrix}, \quad \mathbf{u} = \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} \quad \mathbf{v} = \begin{Bmatrix} v_1 \\ v_2 \end{Bmatrix}, \quad \mathbf{L}_d = \begin{bmatrix} \frac{\partial}{\partial x_1} & 0 \\ 0 & \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} \end{bmatrix},$$

$$\mathbf{w} = \mathbf{L}_d \mathbf{v} = \begin{Bmatrix} v_{1,1} \\ v_{2,2} \\ v_{1,2} + v_{2,1} \end{Bmatrix}, \quad \mathbf{N} = \begin{bmatrix} n_1 & 0 & n_2 \\ 0 & n_2 & n_1 \end{bmatrix},$$

$$\bar{\mathbf{t}} = \left\{ \begin{array}{c} \bar{t}_1 \\ \bar{t}_2 \end{array} \right\}, \ \bar{\mathbf{u}} = \left\{ \begin{array}{c} \bar{u}_1 \\ \bar{u}_2 \end{array} \right\} \text{ and } \mathbf{b} = \left\{ \begin{array}{c} b_1 \\ b_2 \end{array} \right\}.$$

Equations (3)-(4) can also be rewritten in a matrix form:

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\varepsilon} \tag{10a}$$

$$\boldsymbol{\varepsilon} = \mathbf{L}_d\mathbf{u}, \tag{10b}$$

where $\mathbf{D}$ is a matrix of elastic stiffnesses of the material and $\boldsymbol{\varepsilon}$ is the infinitesimal strain tensor as a vector:

$$\boldsymbol{\varepsilon} = \left\{ \begin{array}{c} \varepsilon_{11} \\ \varepsilon_{22} \\ \gamma_{12} \end{array} \right\}.$$

If we consider an isotropic material:

$$\mathbf{D} = \frac{\bar{E}}{1-\bar{\nu}^2} \left[\begin{array}{ccc} 1 & \bar{\nu} & 0 \\ \bar{\nu} & 1 & 0 \\ 0 & 0 & \frac{1-\bar{\nu}}{2} \end{array} \right].$$

with

$$\bar{E} = \left\{ \begin{array}{cc} E & \text{for plane stress} \\ \frac{E}{1-\nu^2} & \text{for plane strain} \end{array} \right., \ \bar{\nu} = \left\{ \begin{array}{cc} \nu & \text{for plane stress} \\ \frac{\nu}{1-\nu} & \text{for plane strain} \end{array} \right..$$

Substituting Eq. (10b) and Eq. (10a) into Eq. (9), the local symmetric weak form can be written using the displacement values:

$$\int_{\Omega_q} \mathbf{w}^T\mathbf{D}\mathbf{L}_d\mathbf{u}d\Omega - \int_{\Gamma_{q0}} \mathbf{v}^T\mathbf{N}\mathbf{D}\mathbf{L}_d\mathbf{u}d\Gamma - \int_{\Gamma_{qu}} \mathbf{v}^T\mathbf{N}\mathbf{D}\mathbf{L}_d\mathbf{u}d\Gamma = \int_{\Gamma_{qt}} \mathbf{v}^T\bar{\mathbf{t}}d\Gamma + \int_{\Omega_q} \mathbf{v}^T\mathbf{b}d\Omega. \tag{11}$$

2.2 Trial Function

To solve Eq. (11) an approximated distribution of the function u in the domain must be obtained. Although several methods can be used for that, in this work, we adopted Moving Least Squares (MLS) method, which is generally used because it can interpolate data with a reasonable accuracy (Atluri & Shen, 2002). The drawback of this method is the lack of Kronecker Delta property because the approximating functions are not interpolation functions, i.e., they do not vanish at the neighboring nodes.

Let $u(\mathbf{x})$ be a function defined in the domain Ω that represents a degree of freedom. The approximation using MLS is:

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}), \tag{12}$$

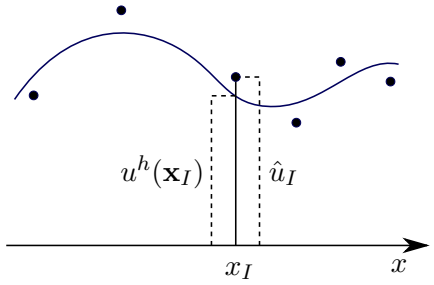


Figure 3: A one-dimensional example of the lack of Kronecker Delta property. The approximated function $u^h(\mathbf{x}_I)$ and the nodal calculated value $\hat{u}_I$ using MLS for a node I.

where $\mathbf{p}(\mathbf{x}) = [p_1(\mathbf{x}) \quad p_2(\mathbf{x}) \quad \cdots \quad p_m(\mathbf{x})]^T$ is a complete monomial basis and $\mathbf{a}(\mathbf{x}) = [a_1(\mathbf{x}) \quad a_2(\mathbf{x}) \quad \cdots \quad a_m(\mathbf{x})]^T$ is the coefficient vector. If t is the highest-order of the basis and d is the problem's dimension, then m can be determined as (Atluri & Shen, 2002):

$$m = \frac{(t+1)(t+2)\cdots(t+d)}{d!}. \tag{13}$$

For instance, if $t = 1$ and $d = 2$, from Eq. (13), $m = 3$ and $\mathbf{p}^T(\mathbf{x}) = [1 \quad x \quad y]$; or if $t = 2$, then $m = 6$ and $\mathbf{p}^T(\mathbf{x}) = [1 \quad x \quad y \quad x^2 \quad xy \quad y^2]$.

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

$$J(\mathbf{x}) = \sum_{I=1}^{N} w_I(\mathbf{x})[\mathbf{p}^T(\mathbf{x}_I)\mathbf{a}(\mathbf{x}) - \hat{u}_I]^2. \tag{14}$$

where $w_I(\mathbf{x})$ is the weight function associated with node I, with $w_I(\mathbf{x}) > 0$ for all $\mathbf{x}$ in the support of $w_I(\mathbf{x})$ and $\mathbf{x}_I$ is the position of node I (Fig. 4).

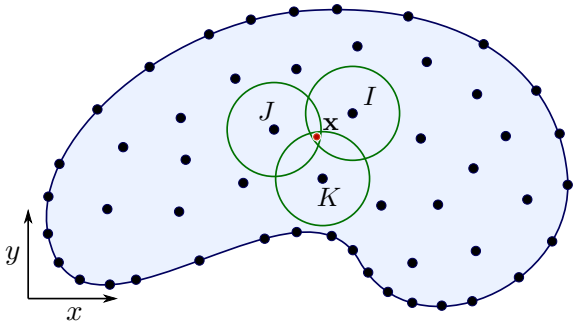


Figure 4: The support domains of nodes I, J and K are shown and they are the only nodes which influence the interpolation at point $\mathbf{x}$.

Minimizing J to find the values of $\mathbf{a}$ and then replacing into Eq. (12), we obtain the approximation at each node:

$$u^h(\mathbf{x}) = \sum_{I=1}^{N} \phi_I(\mathbf{x})\hat{u}_I, \tag{15}$$

where the shape function can be written as:

$$\phi_I(\mathbf{x}) = \sum_{j=1}^{m} p_j(\mathbf{x})[\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})]_{jI}. \tag{16}$$

with

$$\mathbf{A}(\mathbf{x}) = \sum_{I=1}^{N} w_I(\mathbf{x})\mathbf{p}(\mathbf{x}_I)\mathbf{p}^T(\mathbf{x}_I) \tag{17a}$$

$$\mathbf{B}(\mathbf{x}) = [w_1(\mathbf{x})\mathbf{p}(\mathbf{x}_1) \;\; w_2(\mathbf{x})\mathbf{p}(\mathbf{x}_2) \;\; \cdots \;\; w_N(\mathbf{x})\mathbf{p}(\mathbf{x}_N)], \tag{17b}$$

The partial derivative of Eq. (16) can be calculated as:

$$\phi_{I,k}(\mathbf{x}) = \sum_{j=1}^{m}(p_{j,k}(\mathbf{x})[\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}]_{jI}(\mathbf{x}) + p_j(\mathbf{x})[\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}_{,k}(\mathbf{x}) + \mathbf{A}^{-1}_{,k}(\mathbf{x})\mathbf{B}]_{jI}(\mathbf{x})), \tag{18}$$

where $\phi_{I,k}(\mathbf{x})$, $p_{j,k}(\mathbf{x})$, $\mathbf{A}^{-1}_{,k}(\mathbf{x})$ and $\mathbf{B}_{,k}(\mathbf{x})$ are the partial derivatives of $\phi_I(\mathbf{x})$, $p_j(\mathbf{x})$, $\mathbf{A}^{-1}(\mathbf{x})$ and $\mathbf{B}(\mathbf{x})$ with respect to the coordinate k of point $\mathbf{x}$, respectively. The matrix $\mathbf{A}^{-1}_{,k}$ can be obtained as:

$$\mathbf{A}^{-1}_{,k} = -\mathbf{A}^{-1}\mathbf{A}_{,k}\mathbf{A}^{-1}, \tag{19}$$

where $\mathbf{A}_{,k}$ is the derivative of $\mathbf{A}$ with respect to de coordinate k of $\mathbf{x}$.

Notice from Eq. (16) that the shape functions can be calculated only if matrix $\mathbf{A}$ is non-singular. A necessary condition for this to happen is: the number N of non-zero weight functions on $\mathbf{x}$ must be greater than the number m of monomial basis (i.e. $N \geq m$), and they cannot be arranged in a special pattern such as on a straight line (collinear) (Atluri & Zhu, 2000).

The consistency of the MLS approximation depends on the complete order of the monomial basis used. If the basis has monomials of order k, then the MLS will have the C^k consistency Liu (2009).

Weight function

In this paper, we use the forth order spline with compact support as the weight function used in Eq. (14), that is written as (Atluri & Shen, 2002; Quak et al., 2011; Liu et al., 2011):

$$w_I(\mathbf{x}) = \begin{cases} 1 - 6\left(\dfrac{d_I}{r_{wI}}\right)^2 + 8\left(\dfrac{d_I}{r_{wI}}\right)^3 - 3\left(\dfrac{d_I}{r_{wI}}\right)^4, & 0 \leq d_I \leq r_{wI} \\ 0, & d_I > r_{wI} \end{cases} \tag{20}$$

where $d_I = \|\mathbf{x} - \mathbf{x}_I\|$ is the distance from position $\mathbf{x}_I$ of node I to point $\mathbf{x}$ and r_{wI} is the size of the support for the weight function of node I. The size of the radius, r_{wI}, must be large enough to assure that the matrix $\mathbf{A}$ is not-singular ($N \geq m$). However, it should be small enough so the MLS still be a local method (Atluri & Zhu, 2000).

After substituting Eq. (15) into Eq. (11), the local weak form can be rewritten as:

$$\begin{aligned} \int_{\Omega_q} \mathbf{w}^T\mathbf{D}\sum_{J=1}^{M}\mathbf{B}_J\hat{\mathbf{u}}_J d\Omega - \int_{\Gamma_{q0}} \mathbf{v}^T\mathbf{N}\mathbf{D}\sum_{J=1}^{M}\mathbf{B}_J\hat{\mathbf{u}}_J d\Gamma \\ - \int_{\Gamma_{qu}} \mathbf{v}^T\mathbf{N}\mathbf{D}\sum_{J=1}^{M}\mathbf{B}_J\hat{\mathbf{u}}_J d\Gamma = \int_{\Gamma_{qt}} \mathbf{v}^T\overline{\mathbf{t}} d\Gamma + \int_{\Omega_q} \mathbf{v}^T\mathbf{b} d\Omega \end{aligned} \tag{21}$$

where $\mathbf{B}_J$ is the matrix of shape function derivatives of node J and $\hat{\mathbf{u}}_J$ is the vector of unknown fictitious displacements. In a two-dimensional case, we have:

$$\mathbf{\Phi}_J = \begin{bmatrix} \phi_J & 0 \\ 0 & \phi_J \end{bmatrix}, \ \hat{\mathbf{u}}_J = \begin{Bmatrix} \hat{u}_{1J} \\ \hat{u}_{2J} \end{Bmatrix} \text{ and } \mathbf{B}_J = \mathbf{L}_d \mathbf{\Phi}_J = \begin{bmatrix} \phi_{J,1} & 0 \\ 0 & \phi_{J,2} \\ \phi_{J,2} & \phi_{J,1} \end{bmatrix}.$$

2.3 Test function

The MLPG method uses a Petrov-Galerkin formulation, which allows the test and trial functions to be chosen from two different function spaces. Atluri & Shen (2002) proposed six types of test functions. However, in this paper, we are going to use the MLPG5, that uses the Heaviside step function.

The Heaviside step function (Fig. 5) is defined as:

$$v(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \Omega_v \\ 0, & \text{otherwise} \end{cases}. \tag{22}$$

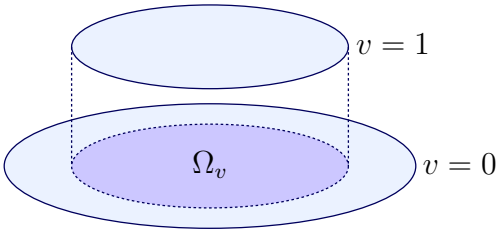


Figure 5: Heaviside step function in a two-dimensional case.

The reason for the low costs of the MLPG5 method is that there are no domain integrals in the construction of the stiffness matrix, instead there are only boundary integrals, as shown in the updated local weak form:

$$\begin{aligned} -\int_{\Gamma_{q0}} \mathbf{ND} \sum_{J=1}^{M} \mathbf{B}_J \hat{\mathbf{u}}_J d\Gamma - \int_{\Gamma_{qu}} \mathbf{ND} \sum_{J=1}^{M} \mathbf{B}_J \hat{\mathbf{u}}_J d\Gamma \\ = \int_{\Gamma_{qt}} \bar{\mathbf{t}} d\Gamma + \int_{\Omega_q} \mathbf{b} d\Omega. \end{aligned} \tag{23}$$

Equation (23) can be simplified into the following linear algebraic equations:

$$\sum_{J=1}^{M} \mathbf{K}_{IJ} \hat{\mathbf{u}}_J = \mathbf{f}_I, \tag{24}$$

where $\mathbf{K}_{IJ}$ is the $(d \times d)$ nodal stiffness matrix and $\mathbf{f}_I$ is the $(d \times 1)$ nodal force vector:

$$\mathbf{K}_{IJ} = -\int_{\Gamma^I_{q0}} \mathbf{NDB}_J d\Gamma - \int_{\Gamma^I_{qu}} \mathbf{SNDB}_J d\Gamma, \tag{25a}$$

$$\mathbf{f}_I = \int_{\Gamma^I_{qt}} \bar{\mathbf{t}} d\Gamma + \int_{\Omega^I_q} \mathbf{b} d\Omega. \tag{25b}$$

where in a two-dimensional case:

$$\mathbf{S} = \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix}, \tag{26}$$

with

$$s_i = \begin{cases} 1, & \text{if } u_i \text{ is prescribed on } \Gamma_u \\ 0, & \text{otherwise} \end{cases}. \tag{27}$$

This S matrix is used in cases where a node is fixed in one direction and free in the other. So, it eliminates the traction part of the integration in the stiffness, which should not be included in the calculation since is the free part.

After applying Eq. (24) for all nodes in the domain (without displacement boundary condition), the global system needs to be solved, resulting in the factious unknown values, $\hat{\mathbf{u}}$. The approximated solution at each dimension in any point $\mathbf{x}$ in the domain can be calculated using the Eq. (15). The integrals in Eq. (25) are usually solved using gauss quadrature in the subdomain's border. The stiffness matrix (Eq. (25a)) is non-symmetric, because of the formulation of Petrov-Galerkin.

2.4 Enforcement of the essential boundary conditions

Since the MLS method has no Kronecker Delta property, the displacement boundary condition can not be applied easily. The penalty method and Lagrange multipliers are widely used, however we use the MLS collocation method, which is applied to nodes located on Γ_t (Liu, 2009; Mirzaei, 2015; Li et al., 2003). Replacing Eq. (15) into Eq. (2a), we obtain:

$$\sum_{J=1}^{N} \phi_J(\mathbf{x})\hat{u}_J = \bar{u}_I, \tag{28}$$

where $\bar{u}_I$ is the prescribed essential boundary condition on node I.

3 NUMERICAL EXAMPLE

In this section, through the analysis of a cantilever beam, we discuss how the radii r_s and r_w of the support regions for the test and trial functions, respectively, influence the simulation's results. In this work, those values are proportional to the minimum distance (d_{min}) between a node and its adjacent nodes:

$$r_s(d_{min}, \alpha) = \alpha d_{min}, \tag{29a}$$
$$r_w(d_{min}, \beta) = \beta d_{min}. \tag{29b}$$

Since $r_s < r_w$ (Atluri & Shen, 2002), then $\alpha < \beta$.

The implementation of MLPG5 was done in C++ and used the Eigen Library (2017) to handle matrix manipulation and solve the linear system. In all examples, the body force, $\mathbf{b}$,

is set to zero. The relative displacement and von Mises stress errors are used to measure the errors, which are defined as:

$$error_u = \frac{\|\mathbf{u}^{exact} - \mathbf{u}^{numerical}\|}{\|\mathbf{u}^{exact}\|} \quad \text{and} \quad error_v = \frac{\|\boldsymbol{\sigma}_v^{exact} - \boldsymbol{\sigma}_v^{numerical}\|}{\|\boldsymbol{\sigma}_v^{exact}\|},$$

where $\|\cdot\|$ is ta discrete L_2-norm on a very dense scattered points in the domain Ω (Mirzaei, 2015).

To solve the integrals of Eq. (25) some Gaussian quadrature points are created on the sub-domains' border, as illustrated in Fig. 6. In the following examples, we split the arcs with $\theta = 60°$, and create $k = 3$ integration points.

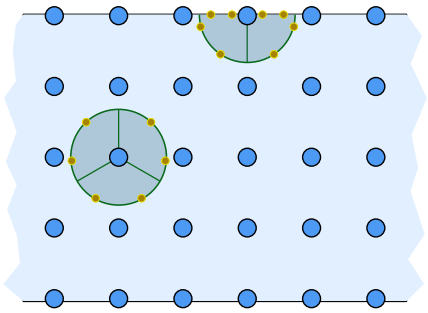


Figure 6: Each subdomain is divided into arcs of angle θ. On each arc's border, k integration points are created. If the node is in the domain's border (Γ), k integration points are created on the line at each side. The arc that was left is divided in arcs of angle θ_2, so that $\theta_2 \leq \theta$.

3.1 Cantilever beam

The cantilever beam (Fig. 7) is a benchmark problem in 2D elasticity, which is a beam anchored at one end, with a shear load applied at the free end. The exact solution of this problem is (Liu, 2009; Atluri et al., 2004):

$$u_1(x,y) = -\frac{Py}{6EI}\left[(6L-3x)x + (2+\nu)\left(y^2 - \frac{D^2}{4}\right)\right], \tag{30a}$$

$$u_2(x,y) = \frac{P}{6EI}\left[3\nu y^2(L-x) + (4+5\nu)\frac{D^2 x}{4} + (3L-x)x^2\right], \tag{30b}$$

where $I = \frac{D^3}{12}$ is the moment of inertia for a cantilever with rectangular cross section and unit thickness. The stress's analytical solution can also be calculated as:

$$\sigma_x(x,y) = -\frac{Py}{I}\left[L-x\right], \tag{31a}$$

$$\sigma_y(x,y) = 0, \tag{31b}$$

$$\sigma_{xy}(x,y) = -\frac{P}{2I}\left[\frac{D^2}{4} - y^2\right]. \tag{31c}$$

The problem is solved using dimensionless parameters as in Atluri et al. (2004):

- Loading: $P = 1$;

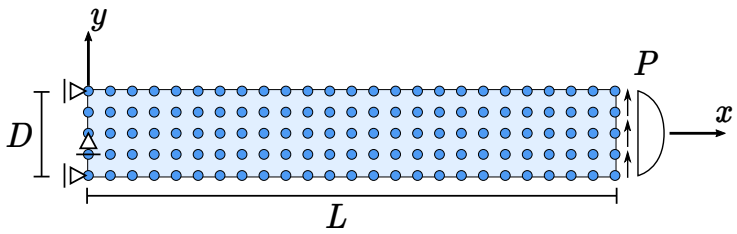


Figure 7: An example of the cantilever beam problem with some discretized nodes for the simulation.

- Young's modulus for the material: $E = 1$;
- Poisson's ratio for the material: $\nu = 0.25$;
- Length of the cantilever: $L = 24$;
- Height of the cantilever: $D = 4$;

Choosing the radius (r_s) of the test function's support.

Atluri & Shen (2002) recommend that the unions of the local domains (Ω_q) of radius r_s, should cover the global domain (Ω), in order to satisfy the equilibrium equation and the boundary conditions in the global domain and on its boundary, respectively (Eqs. (1-2)). Even though they state that the results are satisfactory without covering the global domain. However, our first test will establish a better value for α.

In the first test, the domain was discretized into 259 regularly distributed nodes (37×7). Here, d_{min} can be the distance between adjacent nodes in either the x or the y direction. The simulations' error were analyzed for some values of α (curves) varying the values of β by steps of 0.01 (x-axis) as shown in Fig. 8. The shapes of the first two curves ($\alpha = 0.6$ and 0.65) are different from the others, making the result uncertain, and the last one ($\alpha = 0.9$) starts to show loss of stability. The most interesting values in this examples are $0.7 < \alpha < 0.9$. Notice that $\alpha > 0.7$ covers all the domain.

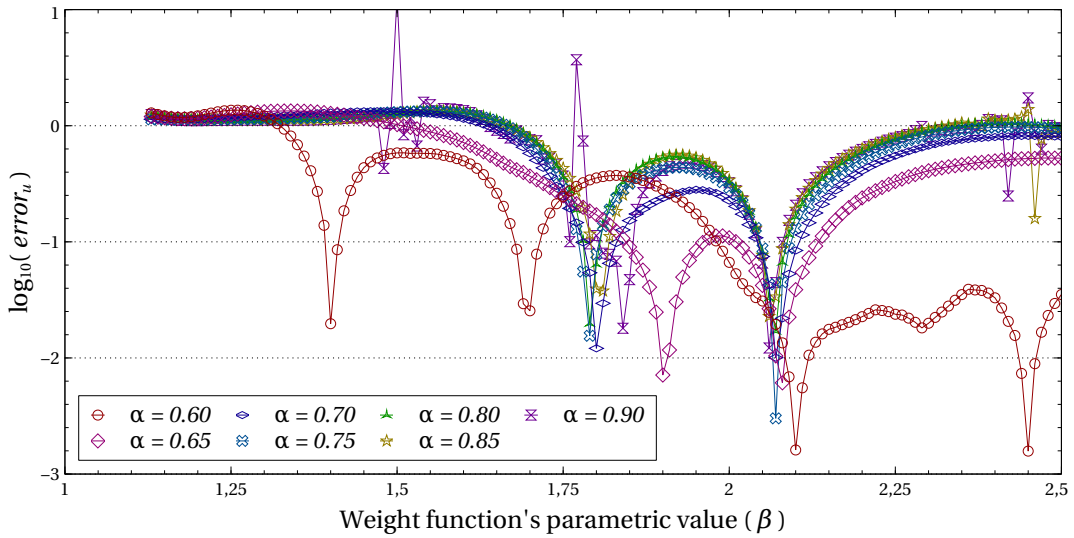
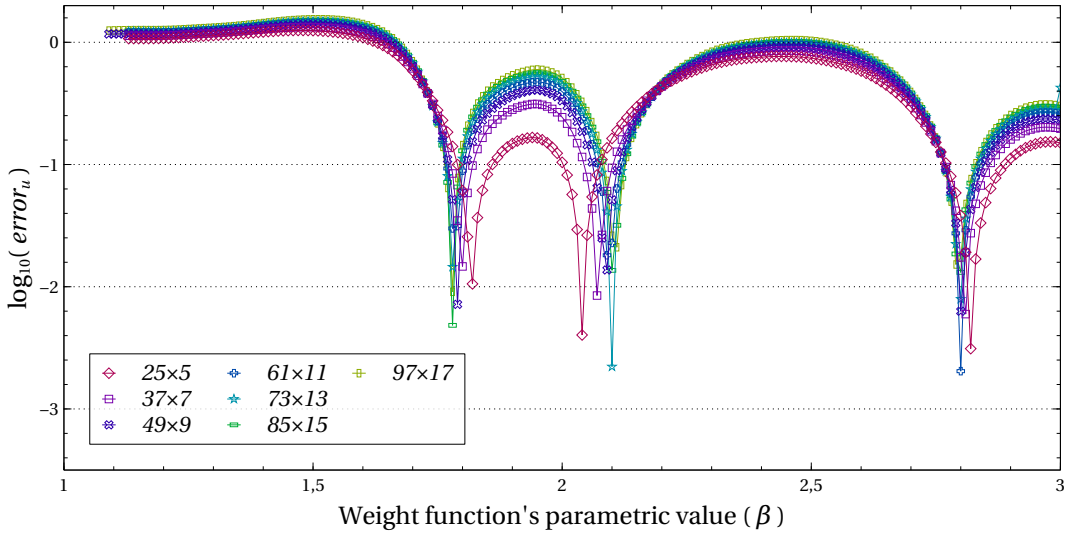


Figure 8: Displacement errors for different values of α which are used to calculate the subdomain's radius r_s as in Eq. (29a).

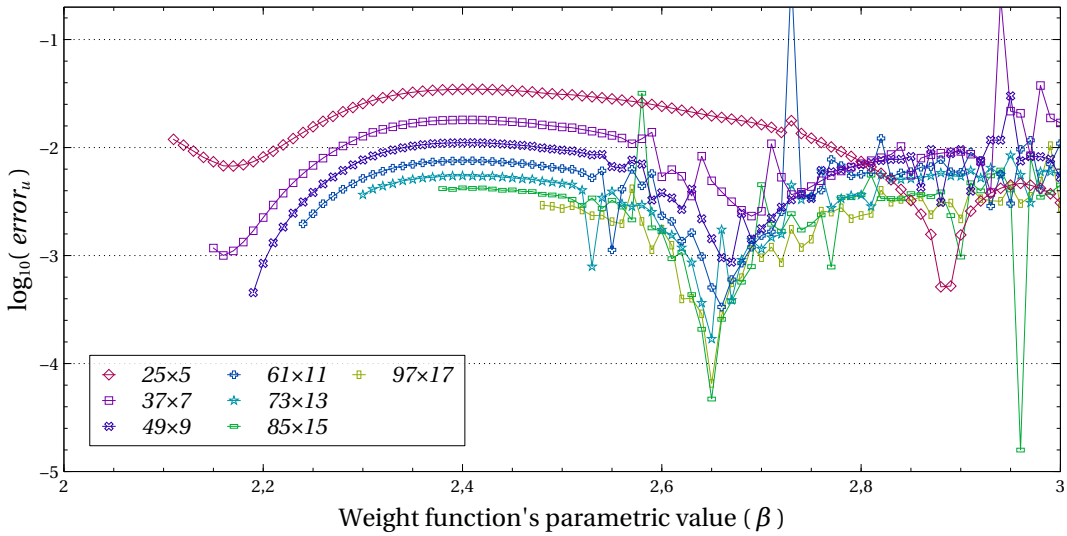
We fix $\alpha = 0.71$ in the following tests to analyze only the influence of β on the accuracy. This value was chosen as the next closest value that covers the entire domain. However, as mentioned before, a lower value could also be used.

Choosing weight function's support radius (r_w)

The radius of the shape function's support, which is the same as the radius of the weight function's, defines the zone of influence of a given node. The analysis' error is greatly influenced by r_w, as can be seen in Fig. 8 from the previous example. In this test, we vary the density of the discrete nodes in the domain (curves) and analyze the error as a function of β (x-axis).



(a) The monomial highest-order is $t = 1$.

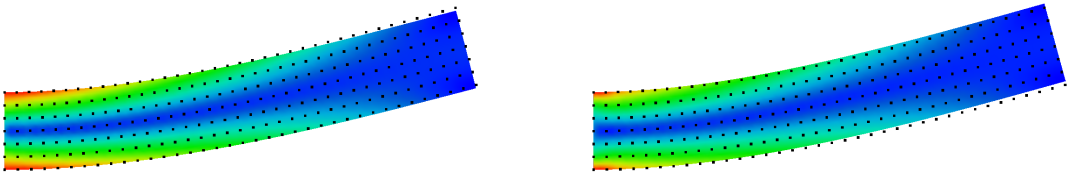


(b) The monomial highest-order is $t = 2$.

Figure 9: Displacement's error with varying node density (Eq. (13))

Although the polynomial basis of higher order yielded more accurate results (Fig. 9), the higher order basis requires more adjacent nodes in a given influence zone in order do obtain an invertible $\mathbf{A}$ matrix (Eq. (17a)). Therefore, for $t = 1$, $\mathbf{A}$ is invertible for $\beta > 1$ while, for $t = 2$, $\mathbf{A}$ is invertible for $\beta > 2$.

From the convergence study shown in Fig. 9, one can see that the results are more consistent and stable for $t = 1$, regardless of the node density. On the other hand, for $t = 2$, the accuracy increases, with a regular pattern, with the increase of node density. However, from around $\beta = 2.5$ marked oscillations occur as β increases. To keep a compromise between locality and accuracy, the value of β has to be as small as possible and as accurate as possible. From Fig. 9a ($t = 1$), β should be around 1.8; and, from Fig. 9b ($t = 2$), β should be approximately 2.65. In Fig. 10, we see the results for two values of β in the case where $t = 1$ and the node density is 37×7. For $\beta = 1.79$ the result approximates the exact solution from below, and, for $\beta = 1.81$ the approximation is from above.



(a) Approximating from the bottom ($\beta = 1.79$).

(b) Approximating from the top ($\beta = 1.81$).

Figure 10: Solution of two different β values for the discretization 37×7. The black dots represents the analytical position of each node. Deformation scale factor is 0.005. The colors represent von Mises stress.

Figure 11 shows a global measurement of the stresses errors using von Mises stress. The discretization 37×7 is used to generalize all the results, showing that both displacement and von Mises stress errors are in the same order of precision.

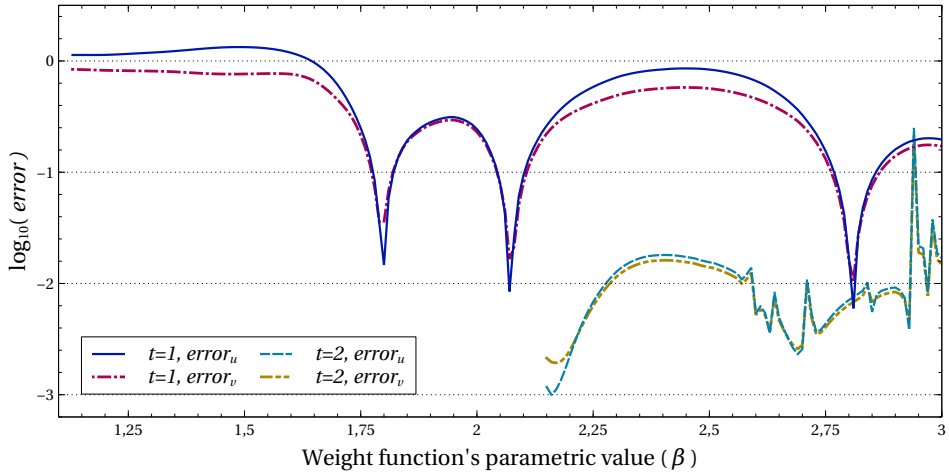
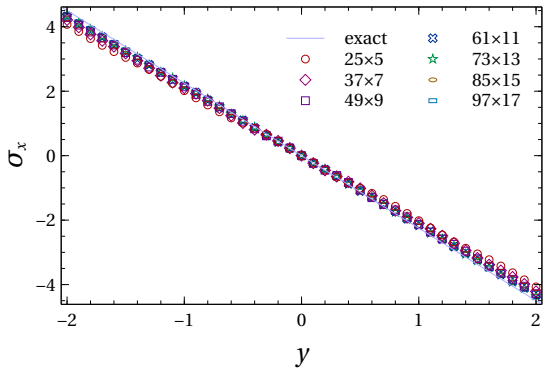
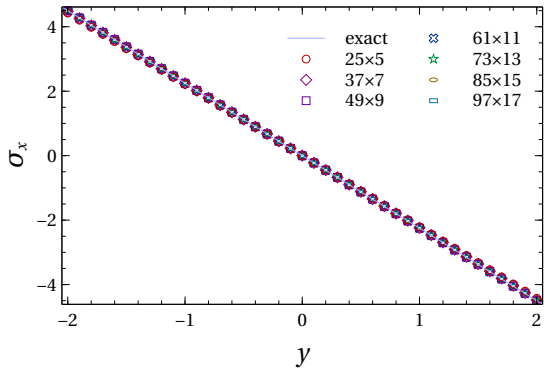


Figure 11: Displacement's and von Mises stress's errors values for the discretization 37×7 with $t = 1$ and $t = 2$.

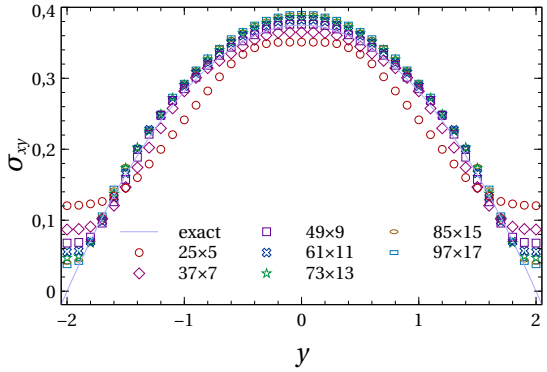
From Fig. 9 we can conclude that as much as the density grows the ideal β value tends to converge to the first local minimum: $t = 1$, $\beta \to 1.78$; $t = 2$, $\beta \to 2.65$. The normal and shear stress at the middle of the beam are illustrated in Fig. 12 using the previously defined β values. All the examples have numerical results close to the analytical. However with $t = 2$, the results are far more accurate, as expected. The results of the coarse examples can also be improved, since we did not use the best β value, for example with 25×5 and $t = 2$, $\beta \to 1.83$ instead of $\beta = 1.79$ as shown in Fig. 9a.



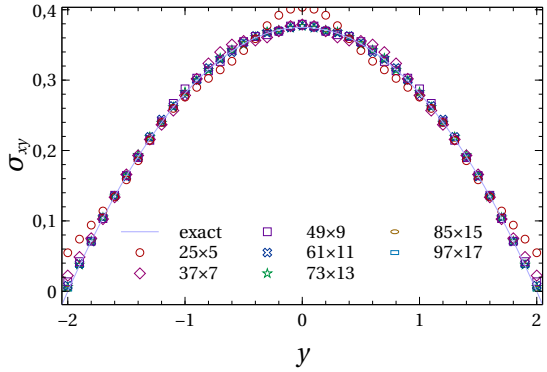
(a) Normal stress with $t = 1$, $\alpha = 0.71$ and $\beta = 1.79$.



(b) Normal stress with $t = 2$, $\alpha = 0.71$ and $\beta = 2.66$.



(c) Shear stress with $t = 1$, $\alpha = 0.71$ and $\beta = 1.79$.



(d) Shear stress with $t = 2$, $\alpha = 0.71$ and $\beta = 2.66$.

Figure 12: Normal and shear stress at $x = 12$ with first and second order monomials.

Choosing the number of integration points

The number Gaussian quadrature points used to integrate the subdomains' borders can cause a small change in the local minimum, as illustrated in Fig. 13. The $\log_{10}(error_u)$ value of each local minimum is almost the same, with a small change in β as showed in Table 1. The experiment used a fine step for β of 0.0001. Each curve tends to have the local minimum moved as much as θ is decreased until it stabilize around $\theta = 30°$. However, we can see that the error is almost the same, meaning that $\theta = 90°$ is already a good angle, even though we used $\theta = 60°$ in the previous examples.

Table 1: Displacement's and Von Misses's errors in some different integration numbers for the discretization 37×7.

θ	β	$\log_{10}(error_u)$		θ	β	$\log_{10}(error_u)$	
180°	1.8154	−2.12539	−1.44705	45°	1.7882	−2.20956	−1.52047
120°	1.7424	−2.17303	−1.48595	30°	1.7896	−2.20890	−1.52060
90°	1.8073	−2.20404	−1.51746	15°	1.7899	−2.20959	−1.52060
60°	1.7971	−2.20869	−1.52232	5°	1.7898	−2.20945	−1.52060

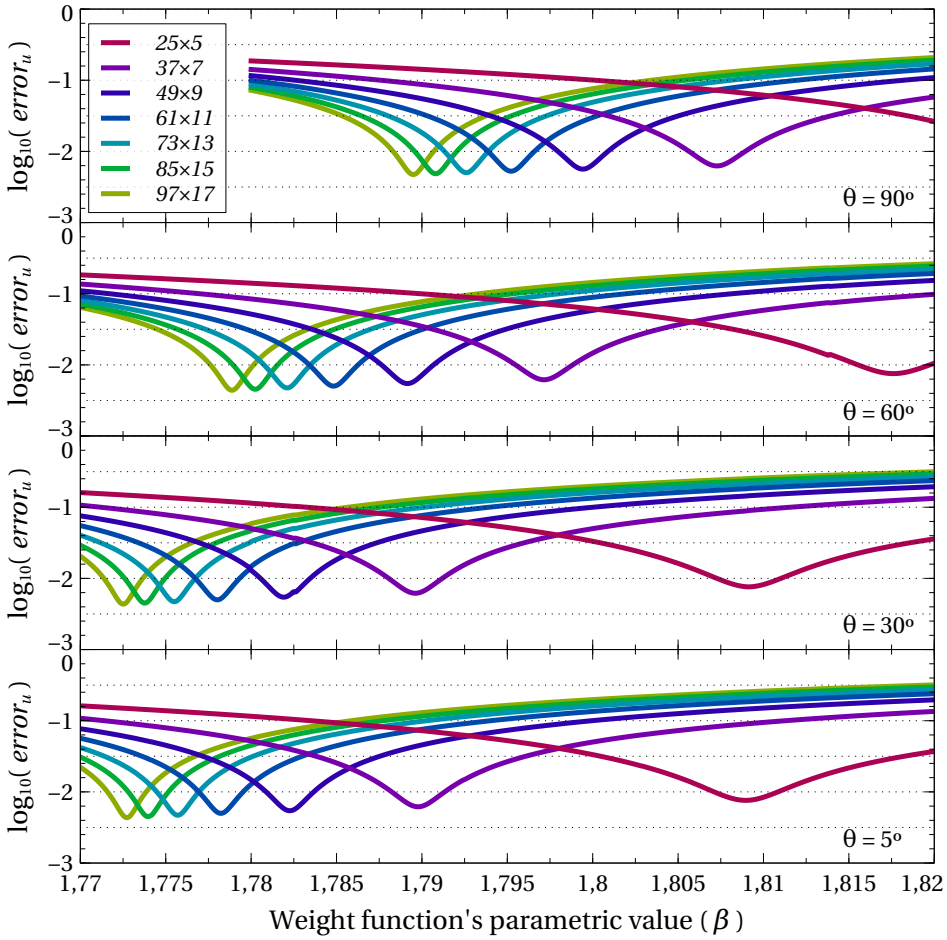


Figure 13: Influence of the number of integration points in the position of the local minimum of the errors' curve.

4 CONCLUSIONS

We investigated the meshless local Petrov-Galerkin method using Heaviside step function as test function, also known as MLPG5, applied to elastostatic problems. The Moving Least Square (MLS) was used to interpolate the global domain (trial function). The MLS collocation method was used to enforce the displacement boundary conditions.

The MLPG5 method has the big advantage of computing the stiffness matrix without expensive domain integration. Besides, if no body force is considered, no domain integral needs to be done at all. The MLS collocation is suitable to be used with the MLPG approach because the equations are established node-by-node, making it possible to use different equations for some of the nodes (Liu, 2009). The boundary conditions are applied in a direct manner, considering that no other variables are added to the problem, making it simple and accurate, but destroying the symmetry of the stiffness matrix. However, since the stiffness matrix of MLPG5 is already non-symmetric, no loss of important properties is imposed.

The numerical examples showed how the sizes of the test and trial functions' supports influence the solution's accuracy of the cantilever beam problem. First we established a good size for the test functions' support. Then, we investigated how the accuracy of the solution varies as a function of the size of the trial functions' supports.

The cantilever beam problem solved with MLPG5, using a linear basis function, created an oscillation phenomenon in the result. A small variation in the radii has great influence on the result. All the local minimum represent a convergence value of the numerical method, which approaches the analytical solution from one side, and then drifts away from the exact solution from the other side. The quadratic basis function does not present this kind of problem, on the other hand it seams more constant, with small instability spikes particularly with a high node density. However, the amplitudes of the spikes are still within good accuracy range. Moreover, in practical simulations, there is no need to use a high node density when a coarse one suffices. The von Mises stress presented, in general, the same order of accuracy as that of the displacements. The normal stresses are more accurate than the shear stresses, however the results improved with increased node density, as expected. The quadratic basis has better results as well. The position (β) of the local minimum is influenced by the number of integration points, however using 12 integration points is already enough in the presented example, since the error is almost the same obtained with 216 integration points.

We suggest that the union of the trial function radius cover all the domain, even though the result can converge without this property. To assure a more accurate solution, the use of a second order monomial is recommended.

ACKNOWLEDGEMENTS

The first author thanks to FUNCAP and CAPES for the financial support. The second and third author would like to thank CNPq for the financial support through the grants: $302726/2016-0$ (PQ) and $307929/2016-7$ (PQ).

REFERENCES

Atluri, S. N., Han, Z. D., & Rajendran, A. M. 2004. A new implementation of the meshless finite volume method, through the MLPG "mixed" approach. *CMES - Computer Modeling in Engineering and Sciences* 6:491–513.

Atluri, S. N. & Shen, S. 2002. The Meshless Local Petrov-Galerkin (MLPG) Method: A Simple & Less-costly Alternative to the Finite Element and Boundary Element Methods. *CMES-Computer Modeling in Engineering & Sciences* .

Atluri, S. N. & Zhu, T. 1998. A new meshless local petrov-galerkin (mlpg) approach in computational mechanics. *Computational Mechanics* 22:117–127.

Atluri, S. N. & Zhu, T. 2000. The Meshless Local Petrov-Galerkin (MLPG) Approach for Solving Problems in Elasto-Statics. *Computational Mechanics* 25:169–179.

Babuska, I. & Melenk, J. M. 1997. The Partition of Unity Method. *International Journal for Numerical Methods in Engineering* 40:727–758.

Belytschko, T., Lu, Y. Y., & Gu, L. 1994. Element-Free Galerkin Methods. *International Journal for Numerical Methods in Engineering* 37:229–256.

Gingold, R. A. & Monaghan, J. J. 1977. Smoothed Particle Hydrodynamics: Theory and Application to Non-Spherical Stars. *Mon. Not. Roy. Astron. Soc.* 181:375.

Guennebaud, G., Jacob, B., et al. 2017. Eigen v3.3. http://eigen.tuxfamily.org.

Han, Z. D. & Atluri, S. N. 2004. Meshless Local Petrov-Galerkin (MLPG) Approaches for Solving 3D Problems in Elasto-Statics. *CMES- Computer Modeling in Engineering & Sciences* 6:169–188.

Hu, D., Long, S., Liu, K., & Li, G. 2006. A modified meshless local Petrov–Galerkin method to elasticity problems in computer modeling and simulation. *Engineering Analysis with Boundary Elements* 30:399–404.

Kamranian, M., Dehghan, M., & Tatari, M. 2017. An adaptive meshless local Petrov–Galerkin method based on a posteriori error estimation for the boundary layer problems. *Applied Numerical Mathematics* 111:181–196.

Li, Q., Shen, S., Han, Z. D., & Atluri, S. N. 2003. Application of meshless local petrov-galerkin (MLPG) to problems with singularities, and material discontinuities, in 3-D elasticity. *CMES - Computer Modeling in Engineering and Sciences* 4:571–585.

Liu, G. 2009. Meshfree Methods: Moving Beyond the Finite Element Method, Second Edition. CRC Press.

Liu, N., He, X., Li, S., & Wang, G. 2011. Meshless Simulation of Brittle Fracture. *Computer Animation and Virtual Worlds* 22:115–124.

Liu, W. K., Jun, S., & Zhang, Y. F. 1995. Reproducing Kernel Particle Methods. *International Journal for Numerical Methods in Fluids* 20:1081–1106.

Long, S. Y., Liu, K. Y., & Hu, D. A. 2006. A new meshless method based on MLPG for elastic dynamic problems. *Engineering Analysis with Boundary Elements* 30:43–48.

Mirzaei, D. 2015. A new low-cost meshfree method for two and three dimensional problems in elasticity. *Applied Mathematical Modelling* 39:7181–7196.

Nguyen, V. P., Rabczuk, T., Bordas, S., & Duflot, M. 2008. Meshless methods: A review and computer implementation aspects. *Mathematics and Computers in Simulation* 79:763–813.

Quak, W., van den Boogaard, A. H., Gonz'alez, D., & Cueto, E. 2011. A comparative study on the performance of meshless approximations and their integration. *Computational Mechanics* 48:121–137.

Sladek, J., Stanak, P., Han, Z., Sladek, V., & Atluri, S. 2013. Applications of the mlpg method in engineering & sciences: a review. *Comput. Model. Eng. Sci* 92:423–475.

Sukumar, N., Moran, B., & Belytschko, T. 1998. The Natural Element Method in Solid Mechanics. *International Journal for Numerical Methods in Engineering* 43:839–887.

Xiao, J. R. 2004. Local Heaviside weighted MLPG meshless method for two-dimensional solids using compactly supported radial basis functions. *Computer Methods in Applied Mechanics and Engineering* 193:117–138.

Zhang, X., Yao, Z., & Zhang, Z. 2006. Application of MLPG in Large Deformation Analysis. *Acta Mechanica Sinica* 22:331–340.