VIII International Conference on Particle-Based Methods
PARTICLES 2023
U. Perego, M. Cremonesi, A. Franci (Eds)

COMPARATIVE STUDY OF METHODS TO GENERATE $\mathcal{C}^2$ CONTINUOUS MESH-FREE BASIS USING THE MOVING LEAST SQUARE METHOD

GAURAV CHANDRA BRIDHANI[1] AND U. SARAVANAN[2]

[1] Department of Civil Engineering
Indian Institue of Technology Madras (IIT Madras)
Chennai, Tamil Nadu, India
e-mail: gbridhani@gmail.com, https://sites.google.com/view/gbridhani/home

[2] Department of Civil Engineering
Indian Institue of Technology Madras (IIT Madras)
Chennai, Tamil Nadu, India
email: saran@iitm.ac.in, https://civil.iitm.ac.in/faculty/saran/

Abstract: This study investigates the ability of three different approaches for constructing smooth and continuous mesh-free basis functions, namely: (i) assuming one degree of freedom per node with the moving least squares method, (ii) assuming six degrees of freedom per node with moving least squares method, and (iii) assuming six degrees of freedom per node with Hermite-type moving least squares method. Further, it provides evidence that all three approaches can generate continuous mesh-free basis functions; however, the first and third approach results in a $\mathcal{C}^2$ continuous mesh-free basis, where the function and its first and second-order derivatives are related. Finally, a comparative study is performed among the three approaches using fourth-order polynomial basis and fifth-order spline weight function.

Keywords. Mesh-free methods, Moving Least Squares method, Hermite-type moving least squares method

1 INTRODUCTION

While solving different boundary value problems (BVPs), the field variable generally varies over the problem domain in a non-linear manner. The numerical solution for such BVPs is obtained by approximating the field variable using continuous functions over the domain. The finite difference and the finite element method are mesh-based methods used to numerically solve ordinary/partial differential equations subjected to some boundary conditions. However, in mesh-based methods, the interpolation functions used for the approximation depend on the shape and size of the element. Also, in FEM, it is impossible to construct continuous interpolation functions of $\mathcal{C}^2$ and above for arbitrary-shaped elements when the geometric mapping is not isoparametric. On the other hand, meshless approximations do not require topological mapping, can handle problems with changing geometry and provide easier formulation of higher-order approximation functions [1].

Therefore, meshless approximation theories are used to formulate higher-order continuous basis functions.

Recently, [2]proposed a numerical scheme to address some difficulties in solving plane BVPs involving implicit constitutive relations. A relation in which the kinematic and kinetic variables are related through an implicit functional in the implicit constitutive framework [3]. In the scheme of [2], the stress and the displacement fields are taken as unknowns, and the physical requirements, such as balance laws and strain-displacement relationship are met point-wise. Finally, the stress and displacement fields are obtained by satisfying the constitutive relation between the stress and strain in a weak integral sense subject to the satisfaction of the traction and displacement boundary condition. The scheme is advantageous over the classical displacement-based finite element method as it provides better stress estimate and faster convergence rate for non-linear material response. This scheme expresses the divergence-free stress field in terms of Airy's stress potential function [4]. Therefore, the stress components are related to the second derivative of Airy's potential. Furthermore, the basis functions must be $\mathcal{C}^2$ continuous over the domain to satisfy the traction continuity over the domain. Hence, this study proposes to perform a comparative study of three different approaches to generate $\mathcal{C}^2$ continuous meshless basis functions.

The paper is organized as follows. Section 2 briefly discusses the Moving Least square and Hermite-type MLS methods. In section 3, the MLS basis functions are obtained. Sections 4 and 5 derive basis functions for MLS with six degrees of freedom (DOF) per node and the Hermite-type MLS with six DOF per node, respectively. The basis functions for all three approaches are plotted in section 6, and finally, in section 7 the conclusion is presented.

2 MOVING LEAST SQUARES METHOD (MLS)

The Moving Least Squares method approximates the unknown variable using polynomials in meshless approximation. [5] developed a method to generate a smooth approximation surface by fitting two-dimensional irregularly spaced field data. Later, [6]used the moving least squares approximation to generate a smooth and interpolating surface for the scattered data and demonstrated that Shepard's method is a special case of MLS approximation. The seminal work of [7] and others [8, 9, 10, 11, 12, 13] uses MLS to approximate displacement fields. The method provides smoother approximation with reasonable accuracy. However, in problems where the stress boundary condition is prescribed, the gradient of the field variable is required to be continuously approximated. The Hermite-type MLS method approximates function and its derivatives over the domain [14]. Although Hermite-type MLS increases the DOF, the accuracy of the field variable and its derivative is improved. [15] presented the mathematical background of Hermite-type moving least squares (HMLS) approximation. In a recent work, [17] proposes a mixed approximation using MLS and the Hermite-type MLS to approximate rotation and displacement, respectively, for studying behavior of thin plates in pre and post-buckling. Therefore, MLS and HMLS approximation approaches are explored in this work to generate $\mathcal{C}^2$ continuous basis functions. Next, the MLS method and its first and

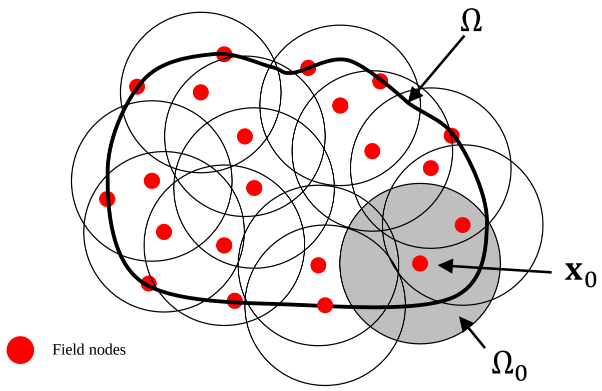


Figure 1: Problem domain with scattered field nodes.

second-order derivatives are derived.

3 FIRST APPROACH: MLS WITH ONE DOF PER NODE

In this first approach, the Moving Least Squares method is derived considering one-degree of freedom for each node to approximate any scalar function. Subsequently, the higher-order derivatives of the basis functions are obtained. The choice of polynomial and weight function is decided depending on the required order of continuity for the basis functions.

Let a scalar-valued vector function $\Phi\colon \Omega \to \mathbb{R}$ with its local approximation $\Phi^h\colon \Omega \to \mathbb{R}$ is represented around a point $\mathbf{x}_0$, such that $\mathbf{x}_0 \in \Omega_0$ and evaluated at $\mathbf{x} \in \Omega$ as

$$\Phi^h(\mathbf{x}, \mathbf{x}_0) = \sum_{j=1}^{m} p_j a_j = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}_0), \tag{1}$$

where $\mathbf{a}$ is a vector of non-constant coefficients, and $\mathbf{p}$ is the polynomial function of order m. The unknown coefficient vector $\mathbf{a}$ is computed at node point $\mathbf{x}_0$ by minimizing a discrete weighted L_2 norm J defined as:

$$J = \sum_{i=1}^{n} \hat{W}(\mathbf{x}_0 - \mathbf{x}_i) \left[\mathbf{p}^T(\mathbf{x}_i)\mathbf{a}(\mathbf{x}_0) - \phi_i\right]^2, \tag{2}$$

where $\hat{W}(\mathbf{x}_0 - \mathbf{x}_i)$ is the value of the weighting function at the i^{th} node in the neighborhood of point $\mathbf{x}_0$, such that $\mathbf{x}_i \in \Omega$ and the shaded region around $\mathbf{x}_0$ is known as the support domain as depicted in Fig.1. The weight function $\hat{W}(\mathbf{x}_0 - \mathbf{x}_i) \neq 0$ for all n nodes inside the support domain, and ϕ_i is the unknown nodal parameter. [14] suggests that support domain size influences the accuracy and efficiency of the approximation. This study employs a circular support domain.

The value of coefficient vector $\mathbf{a}$ is obtained by minimizing J in Eq.(2) with respect to $\mathbf{a}$ as:

$$\sum_{i=1}^{n} \hat{W}(\mathbf{x}_0 - \mathbf{x}_i)\mathbf{p}(\mathbf{x}_i)\mathbf{p}^T(\mathbf{x}_i)\mathbf{a}(\mathbf{x}_0) = \sum_{i=1}^{n} \hat{W}(\mathbf{x}_0 - \mathbf{x}_i)\mathbf{p}(\mathbf{x}_i)\phi_i, \tag{3}$$

where $\mathbf{x}_i = (x_i, y_i)$ and $||\mathbf{x}_0 - \mathbf{x}_i|| = \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2}$. Further, Eq.(3) can be compactly written as:

$$\mathbf{A}(\mathbf{x_0})\mathbf{a}(\mathbf{x_0}) = \mathbf{B}(\mathbf{x_0})\boldsymbol{\phi}, \tag{4}$$

where the *moment matrix* $\mathbf{A}$ and matrix $\mathbf{B}$ are of dimension $m \times m$ and $m \times n$, respectively and expressed as:

$$\begin{aligned} \mathbf{A}(\mathbf{x_0}) &= \sum_{i=1}^{n} \hat{W}(\mathbf{x}_0 - \mathbf{x}_i)\mathbf{p}(\mathbf{x}_i)\mathbf{p}^T(\mathbf{x}_i); \\ \mathbf{B}(\mathbf{x_0}) &= \begin{bmatrix} \hat{W}(\mathbf{x}_0 - \mathbf{x}_1)\mathbf{p}(\mathbf{x}_1) & \hat{W}(\mathbf{x}_0 - \mathbf{x}_2)\mathbf{p}(\mathbf{x}_2) & \cdots & \hat{W}(\mathbf{x}_0 - \mathbf{x}_n)\mathbf{p}(\mathbf{x}_n) \end{bmatrix}, \end{aligned} \tag{5}$$

where the vector $\boldsymbol{\phi} = [\phi_1 \quad \phi_2 \quad \phi_3 \quad \cdots \quad \phi_n]^T$ is defined for the support domain of each node. The value of $\mathbf{a}$ is obtained using Eq.(4) as

$$\mathbf{a}(\mathbf{x_0}) = \mathbf{A}^{-1}(\mathbf{x_0})\mathbf{B}(\mathbf{x_0})\boldsymbol{\phi}, \tag{6}$$

only when the moment matrix $\mathbf{A}$ is invertible. A necessary condition for the matrix to be invertible is that when $n \geq m$, i.e. the number of nodes inside the support domain must be greater than or equal to the number of terms in the polynomial basis such that the nodes in the support domain do not form a pattern. Finally, the approximation Φ^h evaluated at $\mathbf{x}$ is obtained by substituting the value of $\mathbf{a}$ in Eq.(1) and represented as

$$\Phi^h(\mathbf{x}, \mathbf{x}_0) = \mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x}_0)\mathbf{B}(\mathbf{x}_0)\boldsymbol{\phi} = \boldsymbol{\Psi}(\mathbf{x})\boldsymbol{\phi} = \sum_{i=1}^{N} \Psi_i(\mathbf{x})\phi_i, \tag{7}$$

with $\boldsymbol{\Psi}(\mathbf{x})$ as the vector of Moving Least Squares basis function for the point $\mathbf{x}_0$ and represented as

$$\boldsymbol{\Psi}(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x}_0)\mathbf{B}(\mathbf{x}_0). \tag{8}$$

where N is the total number of nodes in the domain. Further, the first derivative of the MLS basis functions with respect to spatial coordinates (x, y) is given as

$$\begin{aligned} \frac{\partial \boldsymbol{\Psi}}{\partial x} &= \frac{\partial \mathbf{p}^T}{\partial x}\mathbf{A}^{-1}\mathbf{B} + \mathbf{p}^T \left(\frac{\partial \mathbf{A}^{-1}}{\partial x}\mathbf{B} + \mathbf{A}^{-1}\frac{\partial \mathbf{B}}{\partial x} \right); \\ \frac{\partial \boldsymbol{\Psi}}{\partial y} &= \frac{\partial \mathbf{p}^T}{\partial y}\mathbf{A}^{-1}\mathbf{B} + \mathbf{p}^T \left(\frac{\partial \mathbf{A}^{-1}}{\partial y}\mathbf{B} + \mathbf{A}^{-1}\frac{\partial \mathbf{B}}{\partial y} \right), \end{aligned} \tag{9}$$

and the second derivative is given as

$$\begin{aligned}
\frac{\partial^2 \boldsymbol{\Psi}}{\partial x^2} &= \left[\frac{\partial^2 \mathbf{p}^T}{\partial x^2}\mathbf{A}^{-1}\mathbf{B} + 2\frac{\partial \mathbf{p}^T}{\partial x}\left(\frac{\partial \mathbf{A}^{-1}}{\partial x}\mathbf{B} + \mathbf{A}^{-1}\frac{\partial \mathbf{B}}{\partial x}\right) + \mathbf{p}^T\left(\frac{\partial^2 \mathbf{A}^{-1}}{\partial x^2}\mathbf{B} + 2\frac{\partial \mathbf{A}^{-1}}{\partial x}\frac{\partial \mathbf{B}}{\partial x} + \mathbf{A}^{-1}\frac{\partial^2 \mathbf{B}}{\partial x^2}\right)\right] \\
\frac{\partial^2 \boldsymbol{\Psi}}{\partial y^2} &= \left[\frac{\partial^2 \mathbf{p}^T}{\partial y^2}\mathbf{A}^{-1}\mathbf{B} + 2\frac{\partial \mathbf{p}^T}{\partial y}\left(\frac{\partial \mathbf{A}^{-1}}{\partial y}\mathbf{B} + \mathbf{A}^{-1}\frac{\partial \mathbf{B}}{\partial y}\right) + \mathbf{p}^T\left(\frac{\partial^2 \mathbf{A}^{-1}}{\partial y^2}\mathbf{B} + 2\frac{\partial \mathbf{A}^{-1}}{\partial y}\frac{\partial \mathbf{B}}{\partial y} + \mathbf{A}^{-1}\frac{\partial^2 \mathbf{B}}{\partial y^2}\right)\right] \\
\frac{\partial^2 \boldsymbol{\Psi}}{\partial x \partial y} &= \left[\frac{\partial^2 \mathbf{p}^T}{\partial x \partial y}\mathbf{A}^{-1}\mathbf{B} + \frac{\partial \mathbf{p}^T}{\partial y}\left(\frac{\partial \mathbf{A}^{-1}}{\partial x}\mathbf{B} + \mathbf{A}^{-1}\frac{\partial \mathbf{B}}{\partial x}\right) + \frac{\partial \mathbf{p}^T}{\partial x}\left(\frac{\partial \mathbf{A}^{-1}}{\partial y}\mathbf{B} + \mathbf{A}^{-1}\frac{\partial \mathbf{B}}{\partial y}\right)\right. \\
&\quad \left. + \mathbf{p}^T\left(\frac{\partial^2 \mathbf{A}^{-1}}{\partial x \partial y}\mathbf{B} + \frac{\partial \mathbf{A}^{-1}}{\partial y}\frac{\partial \mathbf{B}}{\partial x} + \frac{\partial \mathbf{A}^{-1}}{\partial x}\frac{\partial \mathbf{B}}{\partial y} + \mathbf{A}^{-1}\frac{\partial^2 \mathbf{B}}{\partial x \partial y}\right)\right],
\end{aligned} \tag{10}$$

where the expressions for derivatives of *moment matrix* $\mathbf{A}$ and matrix $\mathbf{B}$ can be obtained in terms of $\hat{W}$ and $\mathbf{p}$.

3.1 Weight function

The form of weight function used influences the continuity and shape of the basis functions [14, 16]. This work employs fifth-order (quintic) spline weight function to generate the MLS basis functions. The fifth-order spline weight function corresponding to the i^{th} node can be represented as:

$$\hat{W}(d_i) = \begin{cases} 1 - 10\left(\frac{d_i}{r_i}\right)^2 + 20\left(\frac{d_i}{r_i}\right)^3 - 15\left(\frac{d_i}{r_i}\right)^4 + 4\left(\frac{d_i}{r_i}\right)^5, & 0 \leq \left(\frac{d_i}{r_i}\right) \leq 1 \\ 0, & \left(\frac{d_i}{r_i}\right) > 1 \end{cases} \tag{11}$$

where, $d_i = |\mathbf{x} - \mathbf{x}_i|$ is the distance between support node $\mathbf{x}_i$ and chosen point $\mathbf{x}$, and r_i is the radius of the circular support domain for the node $\mathbf{x}_i$. The fifth-order spline weight function is constructed such that it is $\mathcal{C}^3$ over the support domain following the suggestions of [14].

4 SECOND APPROACH: MLS WITH SIX DOF PER NODE

In this approach, the MLS method is used to approximate the field variable and its derivatives, namely: Φ, $\Phi_{,x}$, $\Phi_{,y}$, $\Phi_{,xx}$, $\Phi_{,yy}$ and $\Phi_{,xy}$ at a node such that the unknown coefficients vary over the domain. In other words, the MLS method derived in Sec.3 is applied to all six degrees of freedom. The construction of the basis functions using this approach is illustrated next.

Let the approximation of all six degrees of freedom at an arbitrary point $\mathbf{x} \in \Omega$ is written in the form

$$\hat{\boldsymbol{\Phi}}^h(x, y) = \hat{\mathbf{P}}(x, y)\hat{\mathbf{a}}(x_0, y_0), \tag{12}$$

where the approximation vector $\hat{\boldsymbol{\Phi}}^h$ is of dimension 6 × 1 and represented as:

$$\hat{\boldsymbol{\Phi}}^h(\mathbf{x}) = \{\Phi(\mathbf{x}) \;\; \Phi_{,x}(\mathbf{x}) \;\; \Phi_{,y}(\mathbf{x}) \;\; \Phi_{,xx}(\mathbf{x}) \;\; \Phi_{,yy}(\mathbf{x}) \;\; \Phi_{,xy}(\mathbf{x})\}^T, \tag{13}$$

with the non-constant unknown coefficients vector $\hat{\mathbf{a}}$ of dimension $6m \times 1$ is defined as

$$\hat{\mathbf{a}} = \{\hat{a}_1 \cdots \hat{a}_m \cdots \hat{a}_{2m} \cdots \hat{a}_{3m} \cdots \hat{a}_{4m} \cdots \hat{a}_{5m} \cdots \hat{a}_{6m}\}^T, \tag{14}$$

and the polynomial matrix $\hat{\mathbf{P}}$ is of dimension $6 \times 6m$ and is defined as:

$$\hat{\mathbf{P}}(\mathbf{x}) = \begin{bmatrix} \mathbf{p}(\mathbf{x}) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{p}_{,x}(\mathbf{x}) & \mathbf{p}(\mathbf{x}) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{p}_{,y}(\mathbf{x}) & \mathbf{0} & \mathbf{p}(\mathbf{x}) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{p}_{,xx}(\mathbf{x}) & 2\mathbf{p}_{,x}(\mathbf{x}) & \mathbf{0} & \mathbf{p}(\mathbf{x}) & \mathbf{0} & \mathbf{0} \\ \mathbf{p}_{,yy}(\mathbf{x}) & \mathbf{0} & 2\mathbf{p}_{,y}(\mathbf{x}) & \mathbf{0} & \mathbf{p}(\mathbf{x}) & \mathbf{0} \\ \mathbf{p}_{,xy}(\mathbf{x}) & \mathbf{p}_{,y}(\mathbf{x}) & \mathbf{p}_{,x}(\mathbf{x}) & \mathbf{0} & \mathbf{0} & \mathbf{p}(\mathbf{x}) \end{bmatrix}, \tag{15}$$

where $\mathbf{p}$ is a general complete-order polynomial function of order m. Now, to find the vector of unknown coefficients $\hat{\mathbf{a}}$ at $\mathbf{x}_0$, the discrete weighted L_2 norm of the error between approximation and nodal parameters for all the degrees of freedom is calculated. The function J can be written as:

$$J = \sum_{i=1}^{n} \hat{W}(\mathbf{x}_i - \mathbf{x}_0) \Big[\mathbf{P}(\mathbf{x}_i)\hat{\mathbf{a}}(\mathbf{x}_0) - \hat{\boldsymbol{\phi}}(\mathbf{x}_i)\Big]^T \Big[\mathbf{P}(\mathbf{x}_i)\hat{\mathbf{a}}(\mathbf{x}_0) - \hat{\boldsymbol{\phi}}(\mathbf{x}_i)\Big], \tag{16}$$

where n is the number of nodes in the support domain, $\hat{W}$ is the weight function such that its value is non-zero inside the support domain, and the vector of nodal parameter $\hat{\boldsymbol{\phi}}(\mathbf{x}_i)$ is of dimension 6×1 is defined as:

$$\hat{\boldsymbol{\phi}}(\mathbf{x}_i) = \Big\{\hat{\phi}(\mathbf{x}_i) \;\; \hat{\phi}_{,x}(\mathbf{x}_i) \;\; \hat{\phi}_{,y}(\mathbf{x}_i) \;\; \hat{\phi}_{,xx}(\mathbf{x}_i) \;\; \hat{\phi}_{,yy}(\mathbf{x}_i) \;\; \hat{\phi}_{,xy}(\mathbf{x}_i)\Big\}. \tag{17}$$

Minimizing J with respect $\hat{\mathbf{a}}$ the following relation is obtained:

$$\hat{\mathbf{A}}(\mathbf{x}_0)\hat{\mathbf{a}}(\mathbf{x}_0) = \hat{\mathbf{B}}(\mathbf{x}_0)\hat{\boldsymbol{\phi}}(\mathbf{x}_i), \tag{18}$$

where the *moment matrix* $\hat{\mathbf{A}}$ and matrix $\hat{\mathbf{B}}$ are of dimension $6m \times 6m$ and $6m \times 6n$ respectively, and defined as:

$$\hat{\mathbf{A}}(\mathbf{x}_0) = \sum_{i=1}^{n} \hat{W}(\mathbf{x}_i - \mathbf{x}_0)\hat{\mathbf{P}}^T(\mathbf{x}_i)\hat{\mathbf{P}}(\mathbf{x}_i); \quad \hat{\mathbf{B}}(\mathbf{x}_0) = \sum_{i=1}^{n} \hat{W}(\mathbf{x}_i - \mathbf{x}_0)\hat{\mathbf{P}}^T(\mathbf{x}_i), \tag{19}$$

where n is the number of neighboring nodes in the support domain. Finally, the coefficient vector $\hat{\mathbf{a}}$ is written in terms of matrices $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ using Eq.(18) and substituted in Eq.(12), resulting in the approximation as

$$\hat{\boldsymbol{\Phi}}^h(\mathbf{x}) = \hat{\mathbf{P}}(\mathbf{x})\hat{\mathbf{A}}^{-1}(\mathbf{x}_0)\hat{\mathbf{B}}(\mathbf{x}_0)\hat{\boldsymbol{\phi}}, \tag{20}$$

where the inversion of *moment matrix* $\hat{\mathbf{A}}$ is ensured by the requirement of $6n > m$ which is governed by the size of the support domain for a given distribution of nodes. Using Eq.(20), the basis function can be written as:

$$\boldsymbol{\Psi}(\mathbf{x}) = \hat{\mathbf{P}}(\mathbf{x})\hat{\mathbf{A}}^{-1}(\mathbf{x}_0)\hat{\mathbf{B}}(\mathbf{x}_0), \tag{21}$$

where $\boldsymbol{\Psi}$ is represented as

$$\boldsymbol{\Psi}(\mathbf{x}) = \{\Psi(\mathbf{x}) \;\; \Psi_{,x}(\mathbf{x}) \;\; \Psi_{,y}(\mathbf{x}) \;\; \Psi_{,xx}(\mathbf{x}) \;\; \Psi_{,yy}(\mathbf{x}) \;\; \Psi_{,xy}(\mathbf{x})\}^T \tag{22}$$

which encompasses six basis functions at each node corresponding to six degrees of freedom. Therefore, this approach results in a total of $6N$ basis functions, where N is the total number of nodes in the domain.

5 THIRD APPROACH: HERMITE-TYPE MLS WITH SIX DOF PER NODE

The Hermite-type MLS is previously used to approximate displacement field as discussed in Sec.2. However, the notion of developing $\mathcal{C}^2$ continuous basis functions using Hermite-type MLS is novel. Therefore, this work attempts to develop $\mathcal{C}^2$ continuous meshless basis function using Hermite-type MLS with six degrees of freedom per node. In this study, the gradient of the unknown field variable is required to be approximated. Therefore, the ability of the Hermite-type MLS method to approximate the field variable Φ along with its derivatives is utilized to obtain the basis functions.

Let the MLS approximation of a scalar valued vector function Φ about any node $\mathbf{x}$ is given as:

$$\Phi^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\tilde{\mathbf{a}}(\mathbf{x}_0) = \sum_{j=1}^{m} p_j(\mathbf{x})\tilde{a}_j(\mathbf{x}_0), \tag{23}$$

where $\mathbf{p}$ is a polynomial function, and $\tilde{\mathbf{a}}$ is the vector of unknown coefficient. The approximation is evaluated at least at $n > m$ points within the support domain of point $\mathbf{x}_0$ to determine the unknown coefficients in the least square sense. Therefore, the approximation function in Eq.(23) is evaluated at n neighborhood point as:

$$\begin{Bmatrix} \Phi^h(\mathbf{x}_1) \\ \Phi^h(\mathbf{x}_2) \\ \vdots \\ \Phi^h(\mathbf{x}_n) \end{Bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & \cdots & p_m(x_1, y_1) & \\ 1 & x_2 & y_2 & \cdots & p_m(x_2, y_2) & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & y_n & \cdots & p_m(x_n, y_n) & \end{bmatrix} \begin{Bmatrix} \tilde{a}_1 \\ \tilde{a}_2 \\ \vdots \\ \tilde{a}_m \end{Bmatrix}, \tag{24}$$

which can be rewritten in a compact form as:

$$\boldsymbol{\Phi}^h(\mathbf{x}) = \mathbf{P}(\mathbf{x})\tilde{\mathbf{a}}(\mathbf{x}_0), \tag{25}$$

where $\tilde{\mathbf{P}}$ is a non-square matrix and $\tilde{\boldsymbol{\Phi}}^h$ is the vector of the field variable to be approximated at each node. Further, the first and second-order derivatives of the function $\boldsymbol{\Phi}^h$ in Eq.(25) with respect to x and y can obtained considering $\tilde{\mathbf{a}}$ as a constant unknown coefficient vector and written as:

$$\begin{aligned} &\boldsymbol{\Phi}^h_{,x}(\mathbf{x}_i) = \frac{\partial \mathbf{P}}{\partial x}\tilde{\mathbf{a}}(\mathbf{x}_0); \quad \boldsymbol{\Phi}^h_{,y}(\mathbf{x}_i) = \frac{\partial \mathbf{P}}{\partial y}\tilde{\mathbf{a}}(\mathbf{x}_0); \quad \boldsymbol{\Phi}^h_{,xx}(\mathbf{x}_i) = \frac{\partial^2 \mathbf{P}}{\partial x^2}\tilde{\mathbf{a}}(\mathbf{x}_0), \\ &\boldsymbol{\Phi}^h_{,yy}(\mathbf{x}_i) = \frac{\partial^2 \mathbf{P}}{\partial y^2}\mathbf{a}(\mathbf{x}_0); \quad \boldsymbol{\Phi}^h_{,xy}(\mathbf{x}_i) = \frac{\partial^2 \mathbf{P}}{\partial x \partial y}\tilde{\mathbf{a}}(\mathbf{x}_0); \end{aligned} \tag{26}$$

which can be combined together to form the approximation of all six degrees of freedom at n nodes in the support domain of point $\mathbf{x}_0$ as

$$\tilde{\mathbf{\Phi}}^h(\mathbf{x}_i) = \tilde{\mathbf{P}}(\mathbf{x}_i)\tilde{\mathbf{a}}(\mathbf{x}_0), \tag{27}$$

where the vector $\tilde{\mathbf{\Phi}}$ and the non-square matrix $\tilde{\mathbf{P}}$ is of dimension $6n \times 1$ and $6n \times m$ respectively, are defined as

$$\tilde{\mathbf{\Phi}}^h(\mathbf{x}_i) = \left\{\mathbf{\Phi}^h(\mathbf{x}_i)\ \mathbf{\Phi}^h_{,x}(\mathbf{x}_i)\ \mathbf{\Phi}^h_{,y}(\mathbf{x}_i)\ \mathbf{\Phi}^h_{,xx}(\mathbf{x}_i)\ \mathbf{\Phi}^h_{,yy}(\mathbf{x}_i)\ \mathbf{\Phi}^h_{,xy}(\mathbf{x}_i)\right\}^T,\ \tilde{\mathbf{P}}(\mathbf{x}_i) = \begin{Bmatrix} \mathbf{P}(\mathbf{x}_i) \\ \mathbf{P}_{,x}(\mathbf{x}_i) \\ \mathbf{P}_{,y}(\mathbf{x}_i) \\ \mathbf{P}_{,xx}(\mathbf{x}_i) \\ \mathbf{P}_{,yy}(\mathbf{x}_i) \\ \mathbf{P}_{,xy}(\mathbf{x}_i) \end{Bmatrix}, \tag{28}$$

where $(\cdot)_{,i}$ denotes the partial derivative with respect to the spatial coordinate. To find $\tilde{\mathbf{a}}$, the discrete weighted L_2 norm with respect to all degrees of freedom is determined as

$$J = \sum_{i=1}^{n} \hat{W}(\mathbf{x}_i - \mathbf{x}_0)\left[\tilde{\mathbf{P}}(\mathbf{x}_i)\tilde{\mathbf{a}}(\mathbf{x}_0) - \boldsymbol{\phi}(\mathbf{x}_i)\right]^T \left[\tilde{\mathbf{P}}(\mathbf{x}_i)\tilde{\mathbf{a}}(\mathbf{x}_0) - \boldsymbol{\phi}(\mathbf{x}_i)\right], \tag{29}$$

where $\hat{W}_i$ is the weight function value at the ith node inside the support domain, n is the number of nodes inside the support domain for which $\hat{W}(\mathbf{x} - \mathbf{x}_i) \neq 0$, and $\boldsymbol{\phi}(\mathbf{x}_i)$ is the nodal parameter defined as

$$\begin{aligned}\boldsymbol{\phi}(\mathbf{x}_i) = \{&\phi(\mathbf{x}_1)\ \cdots\ \phi(\mathbf{x}_n)\ \phi_{,x}(\mathbf{x}_1)\cdots\ \phi_{,x}(\mathbf{x}_n)\ \phi_{,y}(\mathbf{x}_1)\cdots\ \phi_{,y}(\mathbf{x}_n)\ \phi_{,xx}(\mathbf{x}_1)\cdots\ \phi_{,xx}(\mathbf{x}_n) \\ &\phi_{,yy}(\mathbf{x}_1)\cdots\ \phi_{,yy}(\mathbf{x}_n)\ \phi_{,xy}(\mathbf{x}_1)\cdots\ \phi_{,xy}(\mathbf{x}_n)\}^T\end{aligned}, \tag{30}$$

with six degrees of freedom per node. The Eq.(29) is minimized with respect to $\tilde{\mathbf{a}}$ to get

$$\tilde{\mathbf{A}}(\mathbf{x}_0)\tilde{\mathbf{a}} = \tilde{\mathbf{B}}(\mathbf{x}_0)\boldsymbol{\phi}, \tag{31}$$

where *moment matrix* $\tilde{\mathbf{A}}$ and matrix $\tilde{\mathbf{B}}$ are of dimension $m \times m$ and $m \times 6n$ respectively, are defined as

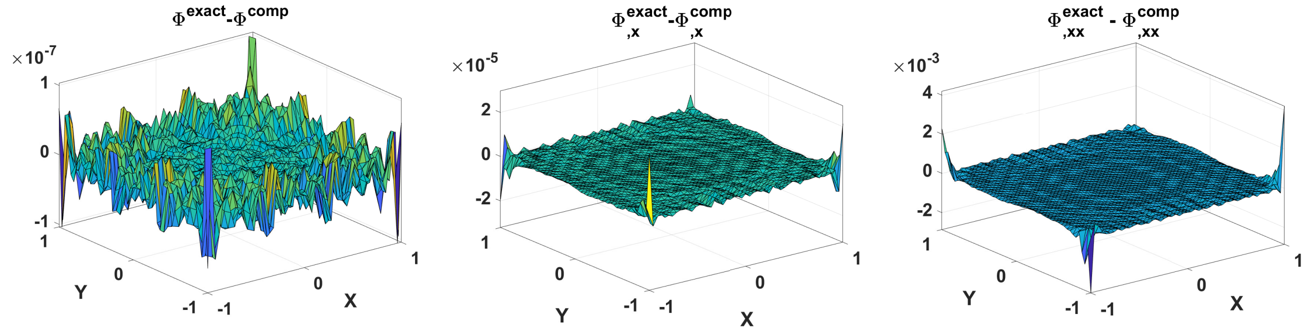
$$\tilde{\mathbf{A}}(\mathbf{x}_0) = \sum_{i=1}^{n} \hat{W}(\mathbf{x}_i - \mathbf{x}_0)\tilde{\mathbf{P}}^T(\mathbf{x}_i)\tilde{\mathbf{P}}(\mathbf{x}_i); \qquad \tilde{\mathbf{B}}(\mathbf{x}_0) = \sum_{i=1}^{n} \hat{W}(\mathbf{x}_i - \mathbf{x}_0)\tilde{\mathbf{P}}^T(\mathbf{x}_i), \tag{32}$$

which are summed over all the support nodes. The approximation in Eq.(23) is obtained by substituting the value of $\tilde{\mathbf{a}}$ from Eq.(31) as,

$$\Phi^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\tilde{\mathbf{A}}^{-1}(\mathbf{x}_0)\tilde{\mathbf{B}}(\mathbf{x_0})\boldsymbol{\phi} = \mathbf{\Psi}(\mathbf{x})\boldsymbol{\phi}, \tag{33}$$

subject to the condition that matrix $\tilde{\mathbf{A}}$ always remain invertible. This is enforced by selecting the size of the support domain such that $6n > m$. The Hermite-type MLS approximation basis function $\mathbf{\Psi}$ is defined as:

$$\mathbf{\Psi}(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\tilde{\mathbf{A}}^{-1}(\mathbf{x}_0)\tilde{\mathbf{B}}(\mathbf{x}_0), \tag{34}$$



(a) $\Phi^{exact} - \Phi^{comp}$ (b) $\Phi^{exact}_{,x} - \Phi^{comp}_{,x}$ (c) $\Phi^{exact}_{,xx} - \Phi^{comp}_{,xx}$

Figure 2: Error in approximation using first approach.

where $\mathbf{\Psi}$ is a matrix of dimension $1 \times 6N$ containing six basis functions for each field node and N is the total number of nodes in the domain. The Hermite approximation of $\mathbf{\Phi}$ given by Eq.(33) can alternatively be written as:

$$\Phi^h(\mathbf{x}) = \sum_{i=1}^{N} \left(\psi(\mathbf{x})\phi + \psi_x(\mathbf{x})\phi^x + \psi_y(\mathbf{x})\phi^y + \psi_{xx}(\mathbf{x})\phi^{xx} + \psi_{yy}(\mathbf{x})\phi^{yy} + \psi_{xy}(\mathbf{x})\phi^{xy}\right); \quad (35)$$

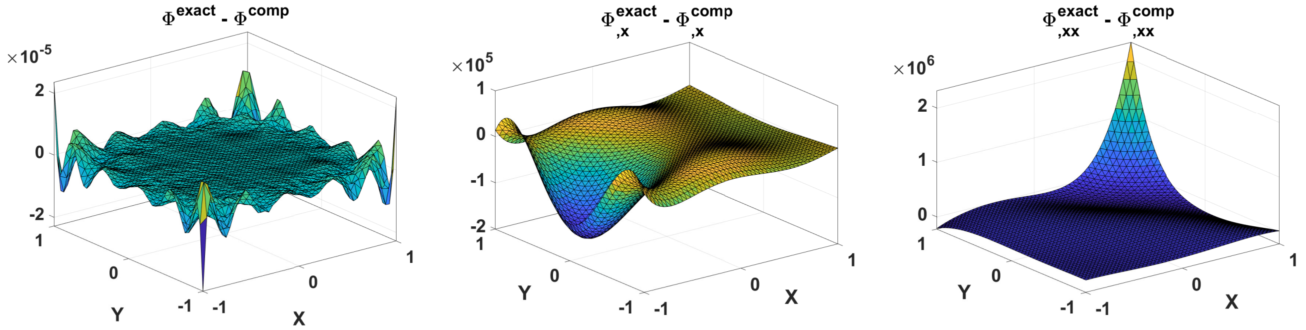
a linear combination of the contribution from each degree of freedom. Subsequently, the first and second-order derivatives of the basis function $\mathbf{\Psi}$ in Eq.(34) are obtained in a similar manner as demonstrated in Eq.(10). Next, the basis functions developed using three approaches are plotted.

6 RESULTS

The approximation function in the first, second and third approaches is given by Eq.(7), Eq.(20) and Eq.(33). The polynomial and weight functions are chosen in such a way that the basis functions are at least $\mathcal{C}^2$ continuous in the domain. This work uses a complete fourth-order polynomial and fifth-order spline weight function to construct the basis functions. The three approximations are validated by fitting a function, chosen randomly of the form as:

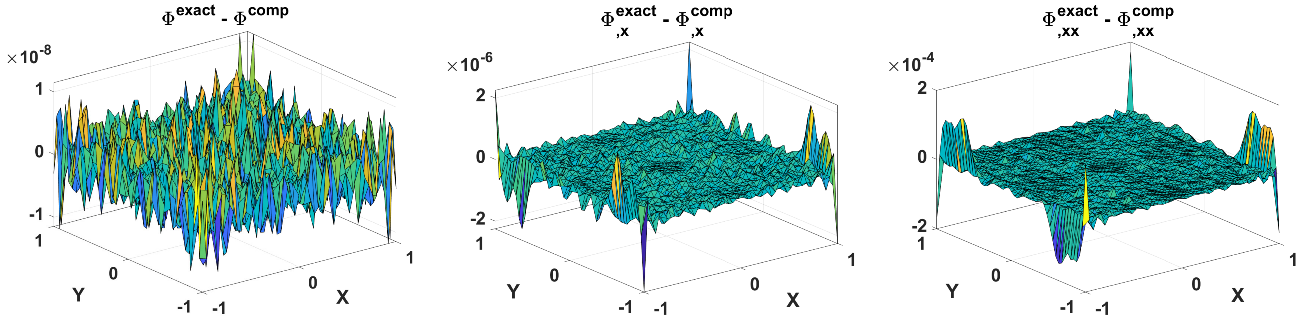
$$f(x, y) = \sin(x)\sin(y) \quad (36)$$

over a domain $\mathfrak{R} = \{(x, y)| -1 \leq x \leq 1, -1 \leq y \leq 1\}$ and plotting on a grid of 51 × 51. The size of the support domain in each approach is obtained using numerical experiments. The function, its first and second derivative with respect to x, is approximated using all three approaches for the function given in Eq.(36). The first approach uses the MLS method with 1 DOF per node to approximate Φ. The error in approximating Φ, $\Phi_{,x}$ and $\Phi_{,xx}$ as obtained for the first approach, is shown in Fig.2. Here, the error in approximating the function is of order $\approx 10^{-7}$. The error increases while approximating first and second-order derivatives. The second approach employs the MLS approximation with six DOFs



(a) $\Phi^{exact} - \Phi^{comp}$ (b) $\Phi^{exact}_{,x} - \Phi^{comp}_{,x}$ (c) $\Phi^{exact}_{,xx} - \Phi^{comp}_{,xx}$

Figure 3: Error in approximation using second approach.



(a) $\Phi^{exact} - \Phi^{comp}$ (b) $\Phi^{exact}_{,x} - \Phi^{comp}_{,x}$ (c) $\Phi^{exact}_{,xx} - \Phi^{comp}_{,xx}$

Figure 4: Error in approximation using third approach.

per node. The error incurred in approximating the function, its first and second derivative with respect to x for the same function form as given in Eq.(36) is depicted in Fig.3. The error in approximating the function is of order $\approx 10^{-5}$. However, the first and second derivatives show higher errors between the computed and exact values. The Hermite-type MLS approximation is employed in the third approach for all six DOFs. The error in approximating the value of the function, its first and second-order derivative is shown in Fig.4. Contrary to the second approach, the approximation of first and second-order derivatives in the third approach results in lower error of order $\approx 10^{-4}$. Also, in the third approach, the error in approximating the function, as given in Eq.(36), is the least among the three approaches.

7 CONCLUSION

This paper presents three different approaches for constructing $\mathcal{C}^2$ continuous meshless basis functions. In the first approach, one DOF is considered at each node, and the higher-order derivatives are obtained through differentiation. The approximation error, as shown in Fig.2, indicates that the function and its first and second derivatives are well approximated.

In the second approach, the MLS method approximates six DOFs at each node, and the unknown field $\hat{\mathbf{a}}$ is taken as a variable. The second approach results in six basis functions at each field node corresponding to each DOF. The error in approximating the first and second-order derivative of the function is higher. This is attributed to the fact that the first and second derivatives are not derived from the function; the form of the basis function is the same for both the function and its derivatives. Hence, it can be concluded that the second approach is not appropriate to generate $\mathcal{C}^2$ continuous functions.

The last approach approximates all six DOFs using the Hermite-type MLS method. The error in the approximation of function, its first and second-order derivative, is lower compared to the other approaches. Therefore, it can be concluded that among the three approaches studied, the Hermite-type MLS method can be employed where accuracy is more important than computational effort, and the Moving Least Squares method can be considered where computational power is also of concern.

ACKNOWLEDGEMENT

The authors acknowledge Prime Minister Research Fellowship (PMRF) contingency fund for providing funding for conference travel. The research leading to these results received funding from the Science and Engineering Research Board (SERB), India under Grant Agreement No. CRG/2019/001090.

REFERENCES

[1] Nguyen, Vinh Phu and Rabczuk, Timon and Bordas, Stéphane and Duflot, Mar. Meshless methods: a review and computer implementation aspects. *Mathematics and Computers in Simulation* (2008) **79**(3), 763-813.

[2] Shankar, LS and Rajthilak, S and Saravanan, U. Numerical technique for solving truss and plane problems for a new class of elastic bodies. *Acta Mechanica* (2016) **227**, 3147-3176.

[3] Rajagopal, Kumbakonam R. On implicit constitutive theories. *Applications of Mathematics* (2003) **48**, 279-319.

[4] Truesdell, Clifford and Toupin, Richard. *The classical field theories.* Springer (1960).

[5] Shepard, Donald. A two-dimensional interpolation function for irregularly-spaced data. *Proceedings of the 1968 23rd ACM national conference* (1968) **23**, 517-524.

[6] P. Lancaster, K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of computation* (1981) **37** (155), 141–158.

[7] Belytschko, Ted and Lu, Yun Yun and Gu, Lei. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering* (1994) **37**(2), 229-256.

[8] Belytschko, T and Lu, YY and Gu, L and Tabbara, M. Element-free Galerkin methods for static and dynamic fracture. *International Journal of Solids and Structures* (1995) **32**(17-18), 2547–2570.

[9] Belytschko, Ted and Organ, Daniel and Krongauz, Yury. A coupled finite element-element-free Galerkin method. *Computational Mechanics* (1995) **17**(3), 186-195.

[10] Belytschko, Ted and Lu, YY and Gu, L. Crack propagation by element-free Galerkin methods. *Engineering Fracture Mechanics* (1995) **51**(2), 295-15.

[11] Belytschko, Ted and Black, Tom. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering* (1999) **45**(5), 601-620.

[12] Atluri, Satya N and Zhu, Tulong. A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics* (1998) **22**(2), 117-127.

[13] Liszka, TJ and Duarte, CAM and Tworzydlo. hp-Meshless cloud method. *Computer Methods in Applied Mechanics and Engineering* (1996) **139**(1-4), 263-288.

[14] Liu, G. R. and Gu, Y. T. *An introduction to meshfree methods and their programming.* Springer Science & Business Media, (2005).

[15] Komargodski, Z. and D. Levin. Hermite type moving-least-squares approximations. *Computers & Mathematics with Applications* (2006) **51**(8), 1223–1232.

[16] Liu, Wing-Kam and Li, Shaofan and Belytschko, Ted. Moving least-square reproducing kernel methods (I) methodology and convergence. *Computer Methods in Applied Mechanics and Engineering* (1997) **143**, 113-154.

[17] Hilali, Y. and O. Bourihane . A mixed mls and hermite-type mls method for buckling and postbuckling analysis of thin plates. *Structures* (2021) **33**, 2349–2360.