J Sci Comput (2017) 71:1197–1211
DOI 10.1007/s10915-016-0337-z

An Element Free Galerkin Method Based on the Modified Moving Least Squares Approximation

Habibullah Amin Chowdhury[1] · Adam Wittek[1] · Karol Miller[1] · Grand Roman Joldes[1]

Received: 19 August 2016 / Revised: 7 December 2016 / Accepted: 9 December 2016 /
Published online: 28 December 2016

Abstract This paper demonstrates that the recently developed modified moving least squares (MMLS) approximation possess the necessary properties which allow its use as an element free Galerkin (EFG) approximation method. Specifically, the consistency and invariance properties for the MMLS are proven. We demonstrate that MMLS shape functions form a partition of unity and the MMLS approximation satisfies the patch test. The invariance properties are important for the accurate computation of the shape functions by using translation and scaling to a canonical domain. We compare the performance of the EFG method based on MMLS, which uses quadratic base functions, to the performance of the EFG method which uses classical MLS with linear base functions, using both 2D and 3D examples. In 2D we solve an elasticity problem which has an analytical solution (bending of a Timoshenko beam) while in 3D we solve an elasticity problem which has an exact finite element solution (unconstrained compression of a cube). We also solve a complex problem involving complicated geometry, non-linear material, large deformations and contacts. The simulation results demonstrate the superior performance of the MMLS over classical MLS in terms of solution accuracy, while shape functions can be computed using the same nodal distribution and support domain size for both methods.

Keywords Meshless method · Modified moving least squares · Element free Galerkin · Numerical methods

JEL Classification 65C20

✉ Grand Roman Joldes
grand.joldes@uwa.edu.au

[1] Intelligent Systems for Medicine Laboratory, School of Mechanical and Chemical Engineering, The University of Western Australia, 35 Stirling Highway, Crawley/Perth, WA 6009, Australia

1 Introduction

Meshless methods (MMs) have been developed as alternative solution methods which address some of the shortcomings of the finite element method (FEM). MMs use only a set of scattered nodes in the problem domain and on its boundary to discretise the problem space. Therefore, the need for creating a high quality mesh is eliminated. MMs are suitable for solving large deformation problems, where traditional FEM has difficulties in obtaining a solution due to element distortion. Furthermore, changes to the discretisation can be easily made in MMs, making them suitable for solving problems which involve topology changes and discontinuities, such as crack propagation and cutting [1–4].

The element free Galerkin (EFG) method is an effective meshless method for nonlinear problems proposed by Belytschko et al. [5], based on the diffuse elements method (DEM) originated by Nayroles et al. [6]. The solution procedure of the EFG method is similar to that used in FEMs. However, in EFG the problem domain discretisation is achieved using nodes arbitrarily distributed within and on the boundary of the problem domain. A Galerkin weak form is employed to develop the discretised system of equations and background cells are used for numerical integration. Moving Least Squares (MLS) is used to approximate the displacement field at a point of interest using the nodal parameters of the nodes that fall into the support domain of that point. The MLS shape functions used in the EFG method are 'meshless' as they are constructed entirely based on the nodes. In FEM, the supports of the shape functions are also used as integration cells. However, in EFG-based MMs the background integration cells are constructed independently of the shape functions [7]. Although the simplest way to define the background integration cells is by creating a mesh, such mesh does not need to be defined using the discretising nodes, and each integration cell can be handled independently during integration (for example when applying adaptive integration procedures [8]) without having to ensure mesh compatibility.

A number of ways have been proposed over the years to construct meshless shape functions [7,9,10]. According to the literature, shape functions used in meshless methods should satisfy some basic requirements [7]: they should be sufficiently robust for reasonably arbitrarily distributed nodes; should be numerically stable; should have a certain order of consistency; should be compactly supported (should be zero outside a bounded region called the support domain); should be computationally efficient and should ideally possesses the Kronecker delta function property. Not having the Kronecker delta function property means that the shape function associated with a particle does not vanish at all other particles; this makes difficult the imposition of essential boundary conditions. The point interpolation method (PIM), proposed by Liu and Gu [7], possesses the Kronecker delta function property. However, PIM requires the number of basis functions to be equal to the number of nodes in the support domain, increasing the chance of a singular moment matrix for arbitrarily distributed nodes. PIM shape functions are also not compatible over the problem domain [7].

Due to its continuity and smoothness properties, the Moving Least Squares (MLS) approximation has been the preferred choice in EFG. However, in order to compute the shape functions, the classical MLS places strict requirements on the nodal distributions inside each support domain. The practical use of higher order polynomial basis, which can generate more accurate approximations of complex deformation fields, is not trivial in classical MLS for randomly distributed nodes. In this context, a modified moving least squares (MMLS) approximation has been recently developed [11,12].

The MMLS shape functions must also satisfy the requirements stated above in order to be used in EFG [7]. Some of these properties (acceptable node distribution, continuity) have already been derived in the paper describing the MMLS approximation [11]. Other properties (compatibility of the field approximation, compact support) are ensured by selecting appropriate weight functions [7]. In this paper we will demonstrate that the MMLS approximation has the necessary order of consistency to satisfy the patch test. We discuss the computational efficiency and derive the invariance properties of MMLS, necessary for an accurate computation of the shape functions.

The paper is organised as follows. A short introduction to MMLS is presented in Sect. 2; the consistency and invariance properties of the approximation are derived in Sect. 3; numerical examples are presented in Sect. 4 and the conclusions are presented in Sect. 5.

2 The Modified Moving Least Squares (MMLS) Approximation Method

The procedure for constructing classical MLS shape functions Ø starts with the approximation of a function u(x), denoted by u^h(x), which is defined by a combination of m monomials (also called basis functions) [7]:

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

where m is the number of terms in the basis $\mathbf{p}(\mathbf{x})$, and $a_i(\mathbf{x})$ are coefficients that depend on the spatial coordinates $\mathbf{x}$. These coefficients are computed by minimizing an error functional defined based on the weighted least squares errors:

$$\bar{J}(\mathbf{x}) = \sum_{j=1}^{n} \left[\left(u^h\left(\mathbf{x}_j\right) - u_j \right)^2 \, w\left(\left\| \mathbf{x} - \mathbf{x}_j \right\| \right) \right] \tag{2}$$

where n is the number of nodes in the support domain of $\mathbf{x}$. After minimization and solving the resulting systems of equations, the classical MLS approximation is obtained as:

$$u^h(\mathbf{x}) = \mathbf{P}^T \left(\mathbf{P}^T \mathbf{W} \mathbf{P} \right)^{-1} \mathbf{P}^T \mathbf{W} \mathbf{u} = \sum_{j=1}^{n} \emptyset_j(\mathbf{x}) u_j = \emptyset^T(\mathbf{x}) \mathbf{u} \tag{3}$$

As can be seen from Eq. (3), the classical MLS shape functions construction is depended on the non-singularity of the moment matrix defined by $\mathbf{P}^T\mathbf{W}\mathbf{P}$. The necessary conditions for the moment matrix to be non-singular depend on the type of basis functions used; for higher order approximation, it requires more nodes to be included inside the support domain, resulting in higher computational cost. Although higher order polynomial basis results in better approximation and convergence properties, these restrictions prevent the practical use of such basis.

In this context, Joldes et al. [11] developed a modified MLS (MMLS) with second order polynomial basis, by including in the error functional additional constraints on the coefficients $\boldsymbol{a}$ corresponding to the second degree monomials in the basis. When the classical MLS moment matrix is singular, the minimization problem solved to compute the coefficients has multiple solutions. The additional constraints ensure that, for such cases, a unique solution can be obtained. This solution corresponds to the coefficients of the second degree monomials in the basis being close or equal to zero, and is therefore close to the solution obtained using linear basis. Furthermore, choosing the weights for the additional constraints as small positive numbers ensures that the classical MLS shape functions are altered only very slightly when the moment matrix in not singular.

In 2D, the error functional used to derive the MMLS shape functions is:

$$\bar{J}(\mathbf{x}) = \sum_{j=1}^{n} \left[\left(u^h\left(\mathbf{x}_j\right) - u_j \right)^2 \ w\left(\| \ \mathbf{x} - \mathbf{x}_j \ \| \right) \right] + \mu_{x^2} a_{x^2}^2 + \mu_{xy} a_{xy}^2 + \mu_{y^2} a_{y^2}^2 \tag{4}$$

The MMLS approximation in obtained as:

$$\bar{u}^h(\mathbf{x}) = \ \mathbf{p}^T \left(\mathbf{P}^T \mathbf{W} \mathbf{P} + \mathbf{H} \right)^{-1} \mathbf{P}^T \mathbf{W} \mathbf{u} = \sum_{j=1}^{n} \bar{\emptyset}_j(\mathbf{x}) u_j = \ \bar{\boldsymbol{\Phi}}^{\boldsymbol{T}}(\mathbf{x}) \mathbf{u} \tag{5}$$

with the new shape functions:

$$\bar{\boldsymbol{\Phi}} = \left[\bar{\emptyset}_1(\mathbf{x}) \cdots \ \bar{\emptyset}_n(\mathbf{x}) \right] = \mathbf{p}^T \left(\mathbf{P}^T \mathbf{W} \mathbf{P} + \mathbf{H} \right)^{-1} \mathbf{P}^T \mathbf{W} \tag{6}$$

where $\mathbf{H}$ is a 6×6 matrix with all elements zeros except the last three diagonal entries, which are equal to the positive weights of the additional constraints $\boldsymbol{\mu} = [\mu_{x^2} \ \mu_{xy} \ \mu_{y^2}]$

$$\boldsymbol{H} = \begin{bmatrix} \mathbf{O}_{33} & \mathbf{O}_{33} \\ \mathbf{O}_{33} & diag\left(\boldsymbol{\mu}\right) \end{bmatrix} \tag{7}$$

For the 3D case, the error functional is defined as:

$$\begin{aligned} \bar{J}(\mathbf{x}) = & \sum_{j=1}^{n} \left[\left(u^h\left(\mathbf{x}_j\right) - u_j \right)^2 \ w\left(\|\mathbf{x} - \mathbf{x}_j\| \right) \right] + \mu_{x^2} a_{x^2}^2 + \mu_{y^2} a_{y^2}^2 + \mu_{z^2} a_{z^2}^2 \\ & + \mu_{xy} a_{xy}^2 + \mu_{xz} a_{xz}^2 + \mu_{yz} a_{yz}^2 \end{aligned} \tag{8}$$

where $\boldsymbol{\mu}$ are the positive weights for the additional constraints:

$$\boldsymbol{\mu} = \left[\mu_{x^2} \ \mu_{y^2} \ \mu_{z^2} \ \mu_{xy} \ \mu_{xz} \ \mu_{yz} \right] \tag{9}$$

Using the same minimization procedure, the shape functions are derived as:

$$\bar{\boldsymbol{\Phi}}(\mathbf{x}) = \left[\bar{\emptyset}_1(\mathbf{x}) \cdots \bar{\emptyset}_n(\mathbf{x}) \right] = \mathbf{P}^T \left(\mathbf{P}^T \mathbf{W} \mathbf{P} + \mathbf{H} \right)^{-1} \mathbf{P}^T \mathbf{W} \tag{10}$$

where $\mathbf{H}$ is a 10×10 matrix with all elements zeros except the last six diagonal entries, which are equal to $\boldsymbol{\mu}$:

$$\boldsymbol{H} = \begin{bmatrix} \mathbf{O}_{44} & \mathbf{O}_{46} \\ \mathbf{O}_{64} & diag\left(\boldsymbol{\mu}\right) \end{bmatrix} \tag{11}$$

It has been shown in [11] that the nodal distributions which are admissible for the classical MLS with linear basis functions are also admissible for the MMLS which uses second order quadratic basis. This requires at least 3 non-collinear nodes in 2D and 4 non-coplanar nodes in 3D to be found in the support domain of any point where the shape functions are computed.

In the next section we demonstrate that the MMLS approximation has the necessary order of consistency to be used as approximation in a meshless method and study the invariance of the MMLS shape functions to translation and scaling.

3 Properties of the Modified Moving Least Squares (MMLS) Approximation

3.1 Consistency of Shape Functions

A certain order of consistency is a prerequisite to ensure the convergence of the numerical results when nodal spacing is reduced; convergence implies that the numerical solution should approach the exact solution when the nodal spacing approaches zero [7]. In case of MLS, the consistency of the approximation depends on the complete order of the monomials used in the basis. It has been shown in [7] that, if pure polynomial basis is assumed, the MLS shape function will possess C^k consistency (can represent exactly polynomial fields up to order k), where k is the complete order of the monomials in the basis. For MMLS we will demonstrate the following consistency property:

Lemma 1 *If the weight function satisfies the positivity requirement,*

$$w\left(\|\mathbf{x}-\mathbf{x}_j\|\right) > 0, \quad \mathbf{x}_j \in \Omega_s \tag{12}$$

where Ω_s is the support domain of point $\mathbf{x}$*, the MMLS shape functions based on quadratic basis* $(m=2)$ *will possess C^1 consistency.*

Proof Since the weight function and the components of vector $\boldsymbol{\mu}$ are positive, the energy functional $\bar{J}$ in Eq. (4) is semi-positive definite. Therefore, its minimum has to be non-negative. Consider a function $u(\mathbf{x}) \in span\left\{p_j(\mathbf{x})\right\}$given by

$$u(\mathbf{x}) = \sum_j^r p_j(\mathbf{x})a_j, \quad r < m \tag{13}$$

The approximation field can always be rewritten as

$$u(\mathbf{x}) = \sum_j^m p_j(\mathbf{x})a_j(\mathbf{x}) \tag{14}$$

by simply assigning

$$a_j(\mathbf{x}) = \begin{cases} a_j, & j \le r \\ 0, & r < j \le m \end{cases} \tag{15}$$

For such a field $\bar{J}$ will vanish and it will necessarily be a minimum, and therefore

$$\bar{u}^h(\mathbf{x}) = \sum_j^r p_j(\mathbf{x})a_j(\mathbf{x}) = u(\mathbf{x}) \tag{16}$$

This proves that any field given by Eq. (13) will be reproduced exactly by the MMLS approximation. □

Corollary 1 *The shape functions created by the MMLS approximation form a Partition of Unity:*

$$\sum_j^n \bar{\boldsymbol{\Phi}}_j(\mathbf{x}) = 1 \tag{17}$$

This is a necessary condition for the shape functions to be able to represent any rigid motion of the problem domain.

In the application of approximation methods to numerical solutions of partial differential equations, practitioners often judge an interpolation method by its ability to pass the patch test, meaning it can reproduce exactly linear functions. In engineering applications this often applies to exact calculation of constant stress and strain.

Corollary 2 *The shape functions created by the modified MLS approximation possess the linear field reproduction property:*

$$\sum_{j}^{n} \bar{\mathbf{\Phi}}_j(\mathbf{x}) x_j = x \tag{18}$$

The above two Corollaries are a direct result of Lemma 1. C^0 consistency, which means constant fields can be exactly reproduced by the approximation, is a particular case of partition of unity. The linear field reproduction property (patch test) is identical to the C^1 consistency condition.

3.2 Invariance of Shape Functions

The invariance to translation and scaling is an important property for meshless shape functions which indicates that the shape functions and the approximation do not depend on the global nodal positions. In other words, any translation or scaling of the nodal positions to a local coordinate system should not change the values of the shape functions. This allows the use of a canonical domain in the computation of shape functions, which reduces the rounding errors which may occur due to the inclusion of too large or too small entries in the moment matrix. For the sake of simplicity and without loss of generality, the following proofs are presented only for the 2D case.

Lemma 2 *The modified MLS shape functions are invariant to the translation of nodes—the shape functions for nodes* $\mathbf{x}_j^* = \left(\mathbf{x}_j + \mathbf{d}\right)$ *have the same values as the shape functions for nodes* $\mathbf{x}_j$*:*

$$\bar{\mathbf{\Phi}}^* \left(\mathbf{x} + \mathbf{d}\right) = \bar{\mathbf{\Phi}}(\mathbf{x}) \tag{19}$$

Proof Based on Eq. (6), the shape functions for nodes $\mathbf{x}_j^*$ can be written as:

$$\bar{\mathbf{\Phi}}^* = \mathbf{p}^T \left(\mathbf{P}^{*T}\mathbf{W}\mathbf{P}^* + \mathbf{H}\right)^{-1} \mathbf{P}^{*T}\mathbf{W} \tag{20}$$

with

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}\left(\mathbf{x}_1^*\right)^T \\ \mathbf{p}\left(\mathbf{x}_2^*\right)^T \\ \vdots \\ \mathbf{p}\left(\mathbf{x}_n^*\right)^T \end{bmatrix} \tag{21}$$

The weight functions $\mathbf{W}$ do not change since they are functions of relative distances, which are not affected by translations. Therefore,

$$\bar{\mathbf{\Phi}}^* \left(\mathbf{x} + \mathbf{d}\right) = \mathbf{p}\left(\mathbf{x} + \mathbf{d}\right)^T \left(\mathbf{P}^{*T}\mathbf{W}\mathbf{P}^* + \mathbf{H}\right)^{-1} \mathbf{P}^{*T}\mathbf{W} \tag{22}$$

We note that

$$\mathbf{p}\left(\mathbf{x}+\mathbf{d}\right)=\begin{bmatrix}1\\ x+d_x\\ y+d_y\\ \left(x+d_x\right)^2\\ \left(x+d_x\right)\left(y+d_y\right)\\ \left(y+d_y\right)^2\end{bmatrix}=\begin{bmatrix}1\\ x+d_x\\ y+d_y\\ x^2+2d_xx+d_x^2\\ xy+d_yx+d_xy+d_xd_y\\ y^2+2d_yy+d_y^2\end{bmatrix}=\mathbf{T}\begin{bmatrix}1\\ x\\ y\\ x^2\\ xy\\ y^2\end{bmatrix}=\mathbf{Tp}(\mathbf{x}) \tag{23}$$

with

$$\mathbf{T}=\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0\\ d_x & 1 & 0 & 0 & 0 & 0\\ d_y & 0 & 1 & 0 & 0 & 0\\ d_x^2 & 2d_x & 0 & 1 & 0 & 0\\ d_xd_y & d_y & d_x & 0 & 1 & 0\\ d_y^2 & 0 & 2d_y & 0 & 0 & 1\end{bmatrix} \tag{24}$$

Combining Eqs. (21) and (24), we obtain:

$$\mathbf{P}^*=\begin{bmatrix}\mathbf{p}\left(\mathbf{x_1}+\mathbf{d}\right)^T\\ \mathbf{p}\left(\mathbf{x_2}+\mathbf{d}\right)^T\\ \vdots\\ \mathbf{p}\left(\mathbf{x}_n+\mathbf{d}\right)^T\end{bmatrix}=\begin{bmatrix}\mathbf{p}\left(\mathbf{x_1}\right)^T\mathbf{T}^T\\ \mathbf{p}\left(\mathbf{x_2}\right)^T\mathbf{T}^T\\ \vdots\\ \mathbf{p}\left(\mathbf{x}_n\right)^T\mathbf{T}^T\end{bmatrix}=\begin{bmatrix}\mathbf{p}\left(\mathbf{x_1}\right)^T\\ \mathbf{p}\left(\mathbf{x_2}\right)^T\\ \vdots\\ \mathbf{p}\left(\mathbf{x}_n\right)^T\end{bmatrix}\mathbf{T}^T=\mathbf{PT}^T \tag{25}$$

Replacing Eqs. (25) and (23) in Eq. (22), we obtain:

$$\begin{aligned}\bar{\mathbf{\Phi}}^*\left(\mathbf{x}+\mathbf{d}\right)&=\mathbf{p}(\mathbf{x})^T\mathbf{T}^T\left(\mathbf{TP}^T\mathbf{WPT}^T+\mathbf{H}\right)^{-1}\mathbf{TP}^T\mathbf{W}\\ &=\mathbf{p}(\mathbf{x})^T\left(\mathbf{P}^T\mathbf{WP}+\mathbf{T}^{-1}\mathbf{HT}^{-T}\right)^{-1}\mathbf{P}^T\mathbf{W}\end{aligned} \tag{26}$$

We notice that if

$$\mathbf{T}^{-1}\mathbf{HT}^{-T}=\mathbf{H} \tag{27}$$

then Eq. (19) is satisfied and the lemma is proved.

Equation (27) can be re-ordered into:

$$\mathbf{H}=\mathbf{THT}^{-T} \tag{28}$$

By rewriting **T** from Eq. (24) using 3 × 3 sub-matrices and considering the definition of **H** in Eq. (7), we can see that:

$$\mathbf{THT}^{-T}=\begin{bmatrix}\mathbf{T}_{11} & \mathbf{O}_{33}\\ \mathbf{T}_{21} & \mathbf{I}_{33}\end{bmatrix}\begin{bmatrix}\mathbf{O}_{33} & \mathbf{O}_{33}\\ \mathbf{O}_{33} & diag\left(\boldsymbol{\mu}\right)\end{bmatrix}\begin{bmatrix}\mathbf{T}_{11}^T & \mathbf{T}_{21}^T\\ \mathbf{O}_{33} & \mathbf{I}_{33}\end{bmatrix}=\begin{bmatrix}\mathbf{O}_{33} & \mathbf{O}_{33}\\ \mathbf{O}_{33} & diag\left(\boldsymbol{\mu}\right)\end{bmatrix}=\mathbf{H} \tag{29}$$

and therefore the lemma is proved. □

Lemma 3 *Consider the scaling of nodal positions* $\mathbf{x}_j^*=\mathbf{Sx}_j$ *with* **S** *a scaling matrix:*

$$\mathbf{S}=\begin{bmatrix}s_x & 0\\ 0 & s_y\end{bmatrix}, \tag{30}$$

If the following conditions are met:

(1) The influence domains are scaled in such a way that the weight functions remain unchanged

$$w\left(\|\mathbf{Sx}-\mathbf{Sx}_j\|\right)=w\left(\|\mathbf{x}-\mathbf{x}_j\|\right) \tag{31}$$

(2) The weights for the additional constrains are changed to:

$$\boldsymbol{\mu}^*=\left[s_x^4\mu_{x^2},\, s_x^2 s_y^2\mu_{xy},\, s_y^4\mu_{y^2}\right] \tag{32}$$

then the MMLS shape functions are invariant to the scaling of nodal positions—the shape functions for nodes $\mathbf{x}_j^*$ have the same values as the shape functions for nodes $\mathbf{x}_j$:

$$\bar{\boldsymbol{\Phi}}^*\left(\mathbf{Sx}\right)=\bar{\boldsymbol{\Phi}}(\mathbf{x}) \tag{33}$$

Proof From Eq. (31), the shape functions for nodes $\mathbf{x}_j^*$ are:

$$\bar{\boldsymbol{\Phi}}^*=\mathbf{p}^T\left(\mathbf{P}^{*T}\mathbf{W}\mathbf{P}^*+\mathbf{H}^*\right)^{-1}\mathbf{P}^{*T}\mathbf{W} \tag{34}$$

with

$$\mathbf{P}^*=\begin{bmatrix}\mathbf{p}\left(\mathbf{x}_1^*\right)^T\\ \mathbf{p}\left(\mathbf{x}_2^*\right)^T\\ \vdots\\ \mathbf{p}\left(\mathbf{x}_n^*\right)^T\end{bmatrix} \tag{35}$$

Therefore,

$$\bar{\boldsymbol{\Phi}}^*\left(\mathbf{Sx}\right)=\mathbf{p}\Big(\left(\mathbf{Sx}\right)^T\left(\mathbf{P}^{*T}\mathbf{W}\mathbf{P}^*+\mathbf{H}^*\right)^{-1}\mathbf{P}^{*T}\mathbf{W} \tag{36}$$

We note that

$$\mathbf{p}\left(\mathbf{Sx}\right)=\begin{bmatrix}1\\ xs_x\\ ys_y\\ x^2s_x^2\\ xs_xys_y\\ y^2s_y^2\end{bmatrix}=\mathbf{R}\begin{bmatrix}1\\ x\\ y\\ x^2\\ xy\\ y^2\end{bmatrix}=\mathbf{Rp}(\mathbf{x}) \tag{37}$$

with

$$\mathbf{R}=\begin{bmatrix}1&0&0&0&0&0\\ 0&s_x&0&0&0&0\\ 0&0&s_y&0&0&0\\ 0&0&0&s_x^2&0&0\\ 0&0&0&0&s_xs_y&0\\ 0&0&0&0&0&s_y^2\end{bmatrix} \tag{38}$$

Combining Eqs. (35) and (38), we obtain:

$$\mathbf{P}^*=\begin{bmatrix}\mathbf{p}\left(\mathbf{Sx_1}\right)^T\\ \mathbf{p}\left(\mathbf{Sx_2}\right)^T\\ \vdots\\ \mathbf{p}\left(\mathbf{Sx}_n\right)^T\end{bmatrix}=\begin{bmatrix}\mathbf{p}\left(\mathbf{x_1}\right)^T\mathbf{R}^T\\ \mathbf{p}\left(\mathbf{x_2}\right)^T\mathbf{R}^T\\ \vdots\\ \mathbf{p}\left(\mathbf{x}_n\right)^T\mathbf{R}^T\end{bmatrix}=\begin{bmatrix}\mathbf{p}\left(\mathbf{x_1}\right)^T\\ \mathbf{p}\left(\mathbf{x_2}\right)^T\\ \vdots\\ \mathbf{p}\left(\mathbf{x}_n\right)^T\end{bmatrix}\mathbf{R}^T=\mathbf{PR}^T \tag{39}$$

Replacing Eqs. (39) and (37) in Eq. (36):

$$\begin{aligned}\bar{\mathbf{\Phi}}^*(\mathbf{Sx}) &= \mathbf{p}(\mathbf{x})^T\mathbf{R}^T\left(\mathbf{RP}^T\mathbf{WPR}^T+\mathbf{H}^*\right)^{-1}\mathbf{RP}^T\mathbf{W}\\ &= \mathbf{p}(\mathbf{x})^T\left(\mathbf{P}^T\mathbf{WP}+\mathbf{R}^{-1}\mathbf{H}^*\mathbf{R}^{-T}\right)^{-1}\mathbf{P}^T\mathbf{W}\end{aligned} \tag{40}$$

We notice that if

$$\mathbf{R}^{-1}\mathbf{H}^*\mathbf{R}^{-T}=\mathbf{H} \tag{41}$$

then Eq. (33) is satisfied and the lemma is proved. Given the definition of **H**, it is easy to show that relation in Eq. (41) is satisfied as long as the weights for the additional constrains given by Eq. (32) are used in $\mathbf{H}^*$. □

4 Numerical Examples

4.1 Bending of a Timoshenko Beam

In this section, the analytical solution for the bending of a Timoshenko beam is compared to the numerical results obtained using an EFG meshless implementation based on the classical MLS and MMLS approximations. The example allows us to compare the accuracy of MMLS based EFG to that of the EFG method based on classical MLS (with linear and quadratic basis) in calculating the displacement field.

A beam of length L and characteristic height D is considered which is subjected to a parabolic varying traction at the free end as shown in Fig. 1. The beam is assumed to be in a state of plane stress. The exact solution of this kind of problem is discussed by Timoshenko and Goodier [13]:

$$u_x = -\frac{Py}{6EI}\left[(6L-3x)x+(2+\nu)\left(y^2-\frac{D^2}{4}\right)\right] \tag{42}$$

$$u_y = \frac{P}{6EI}\left[3\nu y^2(L-x)+(4+5\nu)\frac{D^2x}{4}+(3L-x)x^2\right] \tag{43}$$

The problem is solved for the plane stress case with Young's modulus E $= 3.0\times 10^4$, Poisson's ratio $\nu = 0.3$, L $= 48$, D $= 12$, and P $= 1000$.

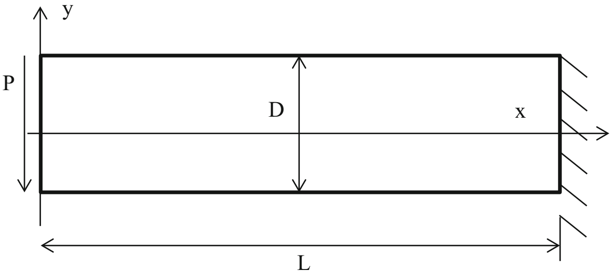

Fig. 1 Bending of a cantilever beam—problem definition

Table 1 Difference in nodal displacements between the meshless and analytical solutions

Approximation method	Average difference	Maximum difference
Classical MLS (linear basis)	1.23×10^{-3}	2.74×10^{-3}
MMLS (quadratic basis)	0.92×10^{-3}	2.34×10^{-3}
Classical MLS (quadratic basis)	Fails to compute due to singular moment matrices	

The domain is discretized using 65 regularly distributed nodes. In order to avoid integration errors, a large number of background integration cells (100 × 25) are used for integration, with a 10 × 10 Gaussian quadrature applied in each cell. The solutions are obtained using a quadratic spline weight function and circular influence domains having the same radius (R = 8). Essential boundary conditions are imposed exactly by using additional unknown tractions on the essential boundary and adding the equations describing the essential boundary conditions to the system of equations [8]. The same weights were used for all the additional MMLS constraints ($\mu_x^2 = \mu_{xy} = \mu_y^2 = 10^{-2}$).

Table 1 shows the maximum and average differences between the analytical solution and the solutions obtained by classical MLS and MMLS. The differences are computed at the nodes. For the given nodal influence domain radius, the classical MLS with quadratic basis failed due to singular moment matrices. Figure 2 shows the difference in the computed deformation field between MMLS and the analytical solution. The results demonstrate that MMLS outperforms classical MLS in terms of solution accuracy.

4.2 Unconstrained Compression of a Cube

In this experiment we simulate the unconstrained compression of a cube with an edge length of 0.1 m. The meshless discretization of the problem domain and the boundary conditions are shown in Fig. 3. A hyper-elastic Neo-Hookean material model with Young′s modulus of 3000 Pa and Poisson's ratio of 0.49 is used and a maximum displacement of 0.02 m is applied to the top surface. The chosen deformation mode and material model result in uniform strains in the cube, and the finite element solution should be exact and independent of discretisation.

For the meshless simulations, the cube is discretised using 260 nodes and 5070 integration points are created using tetrahedral background integration cells with five integration points per tetrahedron. A constant influence domain radius ($R = 0.0325$) is used for all nodes. The same weights were used for all the additional MMLS constraints ($\mu = 10^{-7}$). The essential boundary conditions are imposed by coupling the FEM and MLS shape functions near the essential boundaries. In this way, like in FEM, the prescribed displacement values can be directly imposed to the field variables on the essential boundaries [14]. The meshless simulations are performed using the Meshless Total Lagrangian Explicit Dynamics (MTLED) algorithm [15], with dynamic relaxation used in order to ensure fast convergence to the steady state solution [16,17].

The obtained meshless simulation results are compared with the finite element solution obtained using the non-linear static solver from the commercial finite element software ABAQUS and the same discretization. The average and maximum differences between the nodal displacements obtained using the classical MLS and MMLS meshless methods and the reference solution obtained using ABAQUS are presented in Table 2 and Fig. 4. Similar to the previous example, MMLS outperforms classical MLS in terms of solution accuracy.

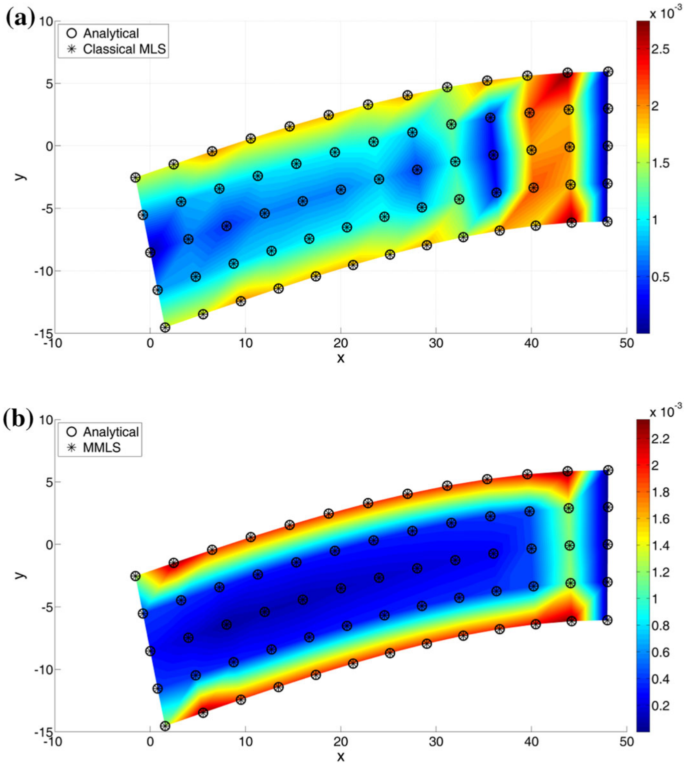

Fig. 2 Difference in the computed deformation field for the Timoshenko beam bending between: **a** classical MLS (linear basis) and analytical solutions; **b** MMLS and analytical solutions

4.3 Craniotomy-Induced Brain Shift

This example, a craniotomy induced brain deformation simulation in 2D, illustrates the use of the proposed method in a case involving complicated geometry, non-linear material, large deformations and contacts. Due to a number of physical and physiological reasons, the brain deforms after craniotomy, a phenomenon known as 'brain shift' [18]. High quality pre-operative images are used for identifying the areas of interest (e.g. a tumour) and creating a biomechanical model, which is then used for computing the position of the tumour and critical healthy tissues within the brain after craniotomy.

In order to simulate brain deformation, based on experimental data [19] and previous modelling experience [20,21], the Young's modulus for the brain parenchyma and the tumour was set to 3000 and 6000 Pa respectively. Because the brain tissue is almost incompressible [18,21], a Poisson's ratio of 0.49 was assigned for both parenchyma and tumour. The ventricles are modelled as a cavity as the cerebrospinal fluid can freely move in and out of them. The skull is assumed to be rigid and the interaction between the skull and the brain is modelled as finite sliding, frictionless contact. Displacements, measured intra-operatively, are applied on the brain surface exposed by craniotomy. The brain model is discretised with 707 nodes, and 4988 integration points were created from a triangular background grid with four integration points per cell. A constant influence domain ($R = 8$) and same weights for the additional constraints ($\mu = 10^{-7}$) were used in the meshless computation. For easy

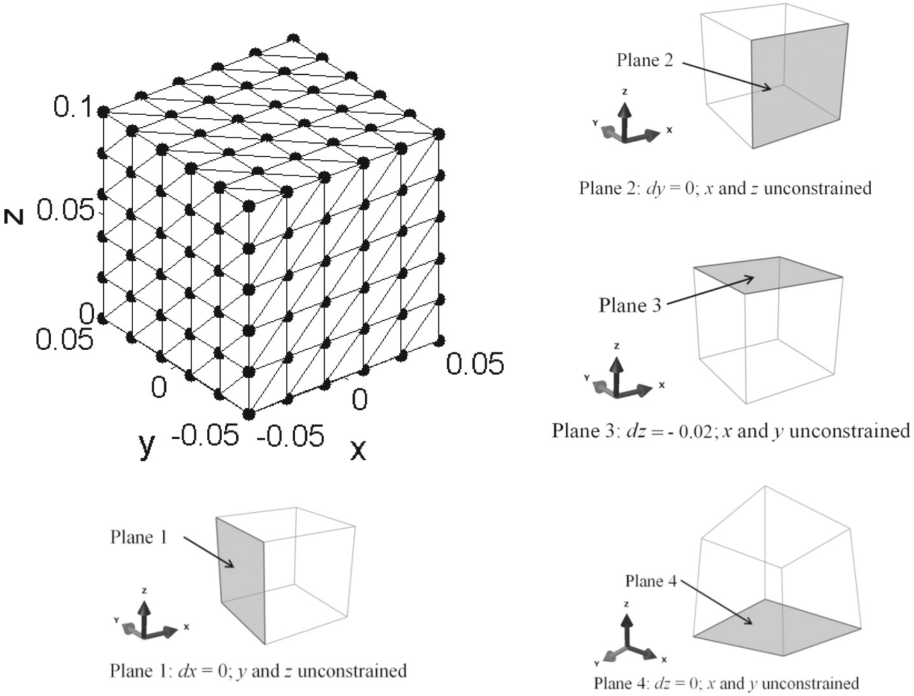
0.1
N 0.05
0
0.05
0
y -0.05 -0.05 x
0
0.05
Plane 2
z
Y
x
Plane 2: dy = 0; x and z unconstrained
Plane 3
Plane 3: dz = - 0.02; x and y unconstrained
Plane 1
Plane 1: dx = 0; y and z unconstrained
Plane 4
Plane 4: dz = 0; x and y unconstrained

Fig. 3 The discretization (260 nodes, 1014 integration cells) and the boundary conditions for the cube used in the unconstrained compression simulation

Table 2 Difference in nodal displacements between the meshless and ABAQUS solutions for the cube compression simulation

Approximation method	Average difference (m)	Maximum difference (m)
Classical MLS (linear basis)	2.48×10^{-5}	9.70×10^{-5}
MMLS (quadratic basis)	1.65×10^{-5}	5.35×10^{-5}
Classical MLS (quadratic basis)	Fails to compute due to singular moment matrices	

imposition of the essential boundary conditions, a regularized weight function [22] was used, resulting in shape functions with almost interpolating properties.

The solutions obtained using the MMs were compared with a reference solution obtained using ABAQUS. High order plain strain elements with hybrid formulation were used in ABAQUS to handle the incompressibility of the soft tissues. The constitutive material laws (Neo-Hookean), loading and boundary conditions were identical in both meshless and ABAQUS computations. The differences of the computed deformation field between classical MLS and modified MLS in comparison with ABAQUS are shown in Fig. 5. Numerical details of the comparison are presented in Table 3. We note that, unlike in the previous examples, the reference solution obtained using ABAQUS is not the exact solution.

For the given support domain radius, the classic MLS with quadratic basis failed due to the singularity of moment matrix, whereas the modified MLS with quadratic basis had no problem in computing the shape functions. As shown in Table 3, the maximum and average

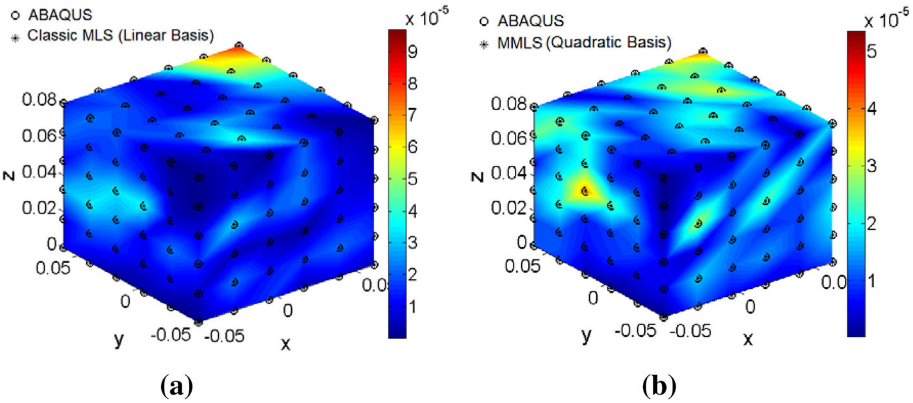

Fig. 4 **a** Differences of computed deformation fields between classical MLS (linear basis, $R = 0.0325$) and ABAQUS. **b** Differences of computed deformation fields between MMLS ($R = 0.0325, \mu = 10^{-7}$) and ABAQUS. All dimensions are in meters

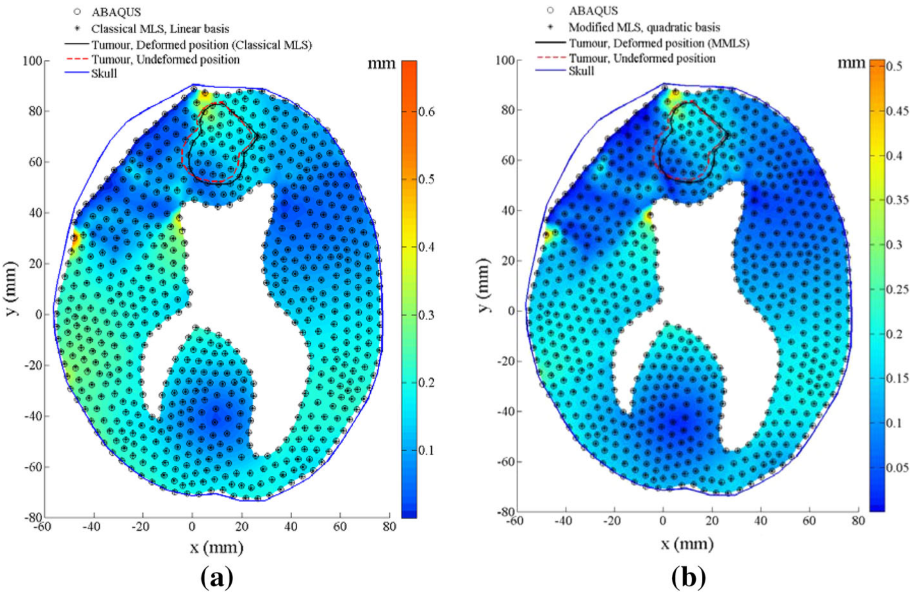

Fig. 5 Differences of computed deformation fields in the brain: **a** between classical MLS (linear basis, $R = 8$) and ABAQUS, **b** between MMLS ($R = 8, \mu = 10^{-7}$) and ABAQUS. Reproduced from [23]

Table 3 Difference in nodal displacements between the meshless and ABAQUS solutions for the brain shift simulation

Approximation method	Average difference (mm)	Maximum difference (mm)
Classical MLS (linear basis)	0.14509	0.67531
MMLS (quadratic basis)	0.12332	0.50729
Classical MLS (quadratic basis)	Fails to compute due to singular moment matrices	

differences between MMLS and ABAQUS are found to be lower compared to those between classic MLS with linear basis and ABAQUS.

5 Conclusions

In this paper we prove that the MMLS approximation presented in [11] has the required order of consistency to be used as an approximation in a meshless method (the resulting shape functions form a partition of unity and the approximation has the linear field reproducing property).

The computation of the MMLS shape functions involves a slight modification of the moment matrix; therefore, the computation efficiency is similar to the classical MLS with quadratic basis for the same number of nodes in a given support domain. Nevertheless, the MMLS has better computation efficiency because it does not need the radius of influence to be as large as for the classical MLS with quadratic basis (requires less nodes in the support domain of each integration point).

The invariance of the shape functions against translation and scaling have been derived. These properties are important for a robust numerical implementation of the method, allowing scaling of the domain which contains the nodes involved in the shape functions computation to a canonical domain, in order to avoid rounding errors in the inversion of the moment matrix.

The 2D and 3D numerical examples demonstrate that MMLS produces more accurate results than classical MLS with linear basis, for the same support domain size. This behaviour stems from the fact that MMLS has better approximation capability compared to classical MLS with linear basis, as shown in [11], therefore reducing the discretisation error. For the chosen support domain size, the classic MLS with quadratic basis fails due to singular moment matrices. To further increase the accuracy of the approximation, the method can be extended to use higher order base functions in a similar fashion, by including in the error functional additional constraints on the coefficients corresponding to the higher degree monomials in the basis.

Acknowledgements The first author is a recipient of the SIRF scholarship and acknowledges the financial support of the University of Western Australia. The financial support of Australian Research Council (Discovery Grants No. DP120100402 and DP160100714) is gratefully acknowledged. We wish to acknowledge the Raine Medical Research Foundation for funding G. R. Joldes through a Raine Priming Grant.

References

1. Miller, K., Horton, A., Joldes, G.R., Wittek, A.: Beyond finite elements: a comprehensive, patient-specific neurosurgical simulation utilizing a meshless method. J. Biomech. **45**, 2698–2701 (2012)
2. Horton, A., Wittek, A., Joldes, G.R., Miller, K.: A meshless total Lagrangian explicit dynamics algorithm for surgical simulation. Int. J. Numer. Method Biomed. Eng. **26**, 977–998 (2010)
3. Li, M., Miller, K., Joldes, G.R., Kikinis, R., Wittek, A.: Biomechanical model for computing deformations for whole-body image registration: a meshless approach. Int. J. Numer. Methods Biomed. Eng. (2016). doi:10.1002/cnm.2771
4. Jin, X., Joldes, G.R., Miller, K., Yang, K.H., Wittek, A.: Meshless algorithm for soft tissue cutting in surgical simulation. Comput. Methods Biomech. Biomed. Eng. **17**, 800–811 (2014)
5. Belytschko, T., Lu, Y.Y., Gu, L.: Element-free Galerkin methods. Int. J. Numer. Methods Eng. **37**, 229–256 (1994)
6. Nayroles, B., Touzot, G., Villon, P.: Generalizing the finite element method: diffuse approximation and diffuse elements. Comput. Mech. **10**, 307–318 (1992)

7. Liu, G.R.: Meshfree Methods: Moving Beyond the Finite Element Method. CRC Press, Boca Raton (2010)
8. Joldes, G.R., Wittek, A., Miller, K.: Adaptive numerical integration in element-free Galerkin methods for elliptic boundary value problems. Eng. Anal. Bound. Elem. **51**, 52–63 (2015)
9. Liu, G.-R., Gu, Y.-T.: An Introduction to Meshfree Methods and Their Programming. Springer, Berlin (2005)
10. Fasshauer, G.E.: Meshfree Approximation Methods with MATLAB. World Scientific Publishing Co. Inc, Singapore (2007)
11. Joldes, G.R., Chowdhury, H.A., Wittek, A., Doyle, B., Miller, K.: Modified moving least squares with polynomial bases for scattered data approximation. Appl. Math. Comput. **266**, 893–902 (2015)
12. Chowdhury, H., Joldes, G., Wittek, A., Doyle, B., Pasternak, E., Miller, K.: Implementation of a modified moving least squares approximation for predicting soft tissue deformation using a meshless method. In: Doyle, B., Miller, K., Wittek, A., Nielsen, P.M.F. (eds.) Computational Biomechanics for Medicine, pp. 59–71. Springer, Berlin (2015)
13. Timoshenko, S., Goodier, J.N.: Theory of Elasticity, 3rd edn. McGraw-Hill, New York (1970)
14. Belytschko, T., Organ, D., Krongauz, Y.: A coupled finite element—element-free Galerkin method. Comput. Mech. **17**, 186–195 (1995)
15. Zhang, G., Wittek, A., Joldes, G.R., Jin, X., Miller, K.: A three-dimensional nonlinear meshfree algorithm for simulating mechanical responses of soft tissue. Eng. Anal. Bound. Elem. **42**, 60–66 (2014)
16. Joldes, G.R., Wittek, A., Miller, K.: An adaptive dynamic relaxation method for solving nonlinear finite element problems. Application to brain shift estimation. Int. J. Numer. Methods Biomed. Eng. **27**, 173–185 (2011)
17. Joldes, G.R., Wittek, A., Miller, K.: Computation of intra-operative brain shift using dynamic relaxation. Comput. Methods Appl. Mech. Eng. **198**, 3313–3320 (2009)
18. Miller, K.: Biomechanics of the Brain. Springer, Berlin (2011)
19. Miller, K., Chinzei, K., Orssengo, G., Bednarz, P.: Mechanical properties of brain tissue in-vivo: experiment and computer simulation. J. Biomech. **33**, 1369–1376 (2000)
20. Joldes, G.R., Wittek, A., Couton, M., Warfield, S.K., Miller, K.: Real-time prediction of brain shift using nonlinear finite element algorithms. In: Medical Image Computing and Computer-Assisted Intervention—Miccai 2009, Pt II, Proceedings vol. 5762, pp. 300–307 (2009)
21. Zhang, J.Y., Joldes, G.R., Wittek, A., Miller, K.: Patient-specific computational biomechanics of the brain without segmentation and meshing. Int. J. Numer. Methods. Biomed. Eng. **29**, 293–308 (2013)
22. Most, T., Bucher, C.: A moving least squares weighting function for the element-free Galerkin method which almost fulfills essential boundary conditions. Struct. Eng. Mech. **21**, 315–332 (2005)
23. Chowdhury, H.A., Joldes, G.R., Wittek, A., Doyle, B., Pasternak, E., Miller, K.: Implementation of a modified moving least squares approximation for predicting soft tissue deformation using a meshless method. In: Doyle, B.J., Miller, K., Wittek, A., Nielsen, P.M.F. (eds.) Computational Biomechanics for Medicine: New Approaches and New Applications, pp. 59–71. Springer, New York (2015)