Strong-form approach to elasticity: Hybrid finite difference-meshless collocation method (FDMCM)

G.C. Bourantas\textsuperscript{a}, K.A. Mountris\textsuperscript{b}, V.C. Loukopoulos\textsuperscript{c}, L. Lavier\textsuperscript{d}, G.R. Joldes\textsuperscript{a,e}, A. Wittek\textsuperscript{a}, K. Miller\textsuperscript{a,f,*}

\textsuperscript{a}Intelligent Systems for Medicine Laboratory, The University of Western Australia, 35 Stirling Highway, Perth, WA 6009, Australia
\textsuperscript{b}LUTIM, INSERM, UMR 1101, CHRU Brest, Brest, France
\textsuperscript{c}Department of Physics, University of Patras, Patras, 26500 Rion, Greece
\textsuperscript{d}Institute for Geophysics, University of Texas, J. J. Pickle Research Campus Bldg. 196, 10100 Burner Rd., Austin, TX 78758-0000, USA
\textsuperscript{e}School of Engineering and Information Technology, Murdoch University, 90 South St, Murdoch, Australia
\textsuperscript{f}Institute of Mechanics and Advanced Materials, Cardiff School of Engineering, Cardiff University, Wales, UK

\textbf{A R T I C L E   I N F O}

\textbf{Article history:}
Received 7 October 2016
Revised 11 August 2017
Accepted 11 September 2017
Available online 21 September 2017

\textbf{Keywords:}
Meshless method
Strong form
Cartesian grid embedded
Finite difference method
Discretization Correction Particle Strength Exchange (DC PSE)
Elastostatics

\textbf{A B S T R A C T}

We propose a numerical method that combines the finite difference (FD) and strong form (collocation) meshless method (MM) for solving linear elasticity equations. We call this new method FDMCM. The FDMCM scheme uses a uniform Cartesian grid embedded in complex geometries and applies both methods to calculate spatial derivatives. The spatial domain is represented by a set of nodes categorized as (i) boundary and near boundary nodes, and (ii) interior nodes. For boundary and near boundary nodes, where the finite difference stencil cannot be defined, the Discretization Corrected Particle Strength Exchange (DC PSE) scheme is used for derivative evaluation, while for interior nodes standard second order finite differences are used. FDMCM method combines the advantages of both FD and DC PSE methods. It supports a fast and simple generation of grids and provides convergence rates comparable to weak formulations. We demonstrate the appropriateness and robustness of the proposed scheme through various benchmark problems in 2D and 3D. Numerical results show good accuracy and h-convergence properties. The ease of computational grid generation makes the method particularly suited for problems where geometries are very complicated and known only imperfectly from images, frequently occurring in e.g. geomechanics and patient-specific biomechanics, where the proposed FDMCM method, after its extension to non-linear regime, appears to be a promising alternative to the traditional weak form-based numerical schemes used in the field.

© 2017 Elsevier Inc. All rights reserved.

1. Introduction

In this paper, we introduce a hybrid finite difference-meshless collocation method (FDMCM) for solving steady state linear elasticity problems. Our approach is motivated by the need; arising frequently e.g. in patient-specific biomechanics and geomechanics, for the very rapid generation of computational grids with complicated geometries that are only known from
imperfect images. Here, we propose a numerical scheme that eliminates the burden of time consuming mesh generation, by using Cartesian grids embedded in the geometry to treat irregular domains.

FDMCM can be considered as a generalization of the finite difference (FD) method. It utilizes both FD and meshless methods (MM) and the newly developed Discretization Corrected Particle Strength Exchange (DC PSE) interpolation scheme [1,2]. DC PSE uses Taylor expansions on irregularly distributed point clouds to compute derivatives. A variety of finite difference schemes applied on irregular nodal distributions has been proposed [3]. For incompressible flows on moving domains the immersed boundary method (IBM) [4] has been frequently used. IBM uses discrete delta functions on domain boundaries to enforce no-slip boundary conditions. Another example is the closely related immersed interface method [5], which uses a rotated coordinate system along with interface jump conditions to define a stencil. Additionally, boundary integral techniques have been introduced and explored in [6–8]. Authors in [9] presented a technique for Cartesian grid generation applied to irregular two-dimensional (2D) geometries. The method works by constructing a Cartesian grid, which has the boundaries nodes as regular nodes of the grid.

An alternative numerical approach to, both, classical and Generalized FD (GFD) methods are meshless methods (MMs). Although the initial MMs were developed over forty years ago, a focused research effort devoted entirely to MMs has been limited until recently. A handful of meshless methods have been developed, categorized as weak and strong form [10–13]. Weak MMs formulations for linear elasticity have been developed: Element-Free Galerkin (EFG) method [14–17], Meshless Local Petrov–Galerkin (MLPG) [18], Local Boundary Integral Equation (LBIE) [19–21] and Maximum Entropy [22]. In the context of strong formulation, MMs mostly use collocation, utilizing several interpolating/approximation schemes, such as Moving Least Squares (MLS) [23], Modified Moving Least Squares (MMLS) [24], Interpolating Moving Least Squares (IMLS) [25], Integrated Ral Bas Functions (RBF) [26] and Maximum Entropy Approximation [27]. More details can be found in textbooks [10–13]. Among collocation methods, RBFs are rapidly gaining popularity [28]. RBF collocation methods are attractive due to their straightforward implementation, high convergence rates, and flexibility regarding of enforcement of arbitrary boundary conditions. The most common way to implement RBFs is in a global sense, by using the nodes in the entire spatial domain to compute shape functions and derivatives. However, the use of globally supported basis functions leads to fully populated matrices, which have the tendency to become ill-conditioned and computationally expensive when the number of nodes increases [28]. Global type RBFs drawbacks motivated researchers to investigate local type RBFs, like RBF-generated finite differences (RBF–FD). Local type RBFs can provide local type, sparse differentiation matrices. Tolstykh introduced RBF–FD method [29], but efforts in [30–32] gave the method its real start. The RBF–FD methodologically is similar to classical FD, with the difference that weights are computed using radial basis functions instead of polynomials. RBF–FD share advantages with global RBF methods since they are truly meshless (work properly without an underlying mesh) and they can easily extend to higher dimensions. One of the main drawbacks of RBF–FD is that the spectral accuracy of global RBF is lost [28]. However, RBF–FD interpolation scheme possesses high-order accuracy: increasing the stencil size increases the accuracy of the approximation (6th to 10th order accuracy is quite common). RBF–FD are also extremely fast to compute, giving N the total number of nodes, RBF–FD preprocessing complexity is dominated by O(N), while for the global RBF method is of O(N^3)) [28]. Additionally, like all local type meshless methods, they are inherently parallel. The other path in the evolution of meshless methods has been the development of the generalized finite difference (GFD) method, also called meshless finite difference (FD) method. Early contributors were Perron and Kao [33], while Jensen [34] was the first to introduce fully arbitrary mesh, by considering Taylor series expansions in order to derive the FD approximation of derivatives (up to the second order). These two very formulations were later improved and extended by many authors [35,36].

In this paper, we propose a hybrid finite difference-meshless collocation (FDMCM) scheme for the numerical solution of linear steady state elasticity equations in their strong form. The scheme inherits the advantages of both methods; the robustness and accuracy of the FD and the flexibility of MM handling irregular geometries. FD methods are extremely fast and accurate, especially when uniform Cartesian grids are used. On the other hand, meshless methods do not require a mesh and, they are more appropriate than finite element (FE) methods in the cases of very large mesh deformation, moving discontinuities and local refinement. The method is similar to that presented in [37], although the current scheme accounts for elastostatic cases and makes use of a sophisticated and a newly developed interpolating scheme (DC PSE), instead of an approximating one (MLS). Furthermore, the method is extended to 3D test cases. The proposed scheme utilizes a Cartesian embedded grid, having the boundaries represented by point clouds (usually given as STL file).

The paper is organized as follows. In Section 2, the governing equations are presented. Our FD–DC PSE interpolating scheme is briefly described in Section 3. In Section 4, we verify the method on a range of 2D and 3D problems and demonstrate its robustness. Finally, the conclusions and suggestions for future work are given in Section 5.

2. Governing equations

We apply our strong-form solution scheme to linear elasticity problems. We consider the boundary value problem defined as

$$\sigma_{ij,j} + b_i = 0 \text{ in } \Omega$$  \hspace{1cm} (1)

$$u_i = f_i \text{ on } \partial \Omega^{2D}$$  \hspace{1cm} (2)
with the isotropic constitutive law given by
\[ \sigma_{ij} = \lambda \varepsilon_{kk} \delta_{ij} + 2\mu \varepsilon_{ij} \]  

(4)

where \( \sigma_{ij} \) is the Cauchy stress tensor, \( \varepsilon_{ij} = 0.5(u_{ij} + u_{ji}) \) is the (linear) strain tensor, \( \lambda \) and \( \mu \) are Lamé constants; \( b_i \) is the body force; \( u_i \) is the displacement; \( \Omega \) is the open domain with a closed boundary \( \partial \Omega \); \( \partial \Omega^D \) and \( \partial \Omega^N \) are the Dirichlet and Neumann boundaries, respectively; \( t_i \) is the surface traction on \( \partial \Omega^N \); \( n_i \) is the surface outward normal on \( \partial \Omega^N \);

In the context of FDM and DC PSE only the derivatives are computed (field values are known on the point cloud). Thus, for the displacement field \( u_i \) defined on a set of \( N \) points (nodes) in the domain spatial \( \Omega \) we can write
\[ u_i \approx u_i^N = \sum_{l=1}^{N} g_i d_l \]  

(5)

and in matrix notation
\[ u^N = \text{Id} \]  

(6)

where \( \text{Id} \) is the identity matrix. The derivatives of the unknown field function can be obtained in a straightforward manner by \( D^nu^N(\mathbf{x}) = \sum_{l=1}^{N} D^ng_i d_l \), where \( D^nu_i = \partial^{|a|} u_i/\partial x_1^{a_1} \ldots \partial x_d^{a_d} \), \( |a| = \sum_{l=1}^{d} a_i \) is the differential operator.

In the FDM framework, derivatives are introduced into Eqs. (1)-(3) and by using collocation the residuals are enforced to be zero on the points \((\zeta_j)_{j=1}^N\) set used to represent the spatial domain. Applying collocation on the elasticity equations, we obtain
\[ L(1(\zeta))d = R(\zeta) \quad \forall (\zeta) \in \Omega \]  

(7)

In the two-dimensional elasticity the operator matrices \( L, B^D \) and \( B^N \) can be written as
\[ L = \begin{bmatrix} \lambda + 2\mu & \partial^2 & \partial^2 \\ \partial^2 & \lambda + \mu & \partial^2 \\ \partial^2 & \partial^2 & \lambda + 2\mu \end{bmatrix} \]  

(8a)

\[ B^D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \]  

(8b)

\[ B^N = \begin{bmatrix} \lambda n_x & \mu n_y & \lambda n_y + \mu n_x \\ \lambda n_y & \mu n_x & \lambda n_x + \mu n_y \end{bmatrix} \]  

(8c)

while in three dimensions, we get
\[ L = \begin{bmatrix} \lambda + 2\mu & \partial^2 & \partial^2 & \partial^2 & \partial^2 & \partial^2 \\ \partial^2 & \lambda + \mu & \partial^2 & \partial^2 & \partial^2 & \partial^2 \\ \partial^2 & \partial^2 & \lambda + 2\mu & \partial^2 & \partial^2 & \partial^2 \\ \partial^2 & \partial^2 & \partial^2 & \lambda + \mu & \partial^2 & \partial^2 \\ \partial^2 & \partial^2 & \partial^2 & \partial^2 & \lambda + 2\mu & \partial^2 \end{bmatrix} \]  

(9a)

\[ B^D = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \]  

(9b)

\[ B^N = \begin{bmatrix} \lambda n_x & \mu n_y & \lambda n_y + \mu n_x & \lambda n_x & \mu n_y & \lambda n_y + \mu n_x \\ \lambda n_y & \mu n_x & \lambda n_x + \mu n_y & \lambda n_y & \mu n_x & \lambda n_x + \mu n_y \\ \lambda n_x & \mu n_y & \lambda n_y + \mu n_x & \lambda n_x & \mu n_y & \lambda n_y + \mu n_x \end{bmatrix} \]  

(9c)
3. Interpolation method

Spatial derivatives for the unknown field function \( u \) at the point \((x, y) = (x_i, y_j) = (i, j)\) are approximated by using the following expressions [37]

\[
\left( \frac{\partial u}{\partial x} \right)_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2h} \tag{10a}
\]

\[
\left( \frac{\partial u}{\partial y} \right)_{i,j} = \frac{u_{i,j+1} - u_{i,j-1}}{2h} \tag{10b}
\]

\[
\left( \frac{\partial^2 u}{\partial x^2} \right)_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} \tag{10c}
\]

\[
\left( \frac{\partial^2 u}{\partial y^2} \right)_{i,j} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2} \tag{10d}
\]

\[
\left( \frac{\partial^2 u}{\partial x \partial y} \right)_{i,j} = \frac{u_{i+1,j+1} - u_{i+1,j-1} + u_{i-1,j-1} - u_{i-1,j+1}}{4h^2} \tag{10e}
\]

In our FDMCM scheme, FD expressions are applied on the interior nodes (where the 5 and 7 nodes stencil can be defined in 2D and 3D, respectively), while for the boundary and the interior nodes located close to the boundary (where the stencil cannot be defined), the DC PSE interpolation scheme is used instead.

DC PSE was originally formulated as a correction to the Particle Strength Exchange (PSE) method [38,39] on irregular nodal distributions. PSE is used for the evaluation of spatial derivatives of any order of a sufficiently smooth function discretized over scattered colocation points. Their main drawback is the introduction of an overlap condition [1], which requires a large number of particles for small kernel sizes. In the DC PSE method, the overlap condition can be relaxed by directly satisfying discrete moment conditions over the collocation points. In fact, the discrete moment conditions for DC PSE can be considered as a direct analogue to the continuous moment conditions for PSE.

For simplicity, we present the DC PSE formulation in 2D, with standard kernel functions. However, the formulation of the DC PSE operators in \( n \) dimensions with arbitrary kernel functions is straightforward [40,41]. We begin by considering a differential operator, of arbitrary order, for a sufficiently smooth field \( f(x) = f(x,y) \) at point \( x_p = (x_p,y_p) \) on a point set

\[
D^m f(x_p) = \frac{\partial^{m+n} f(x,y)}{\partial x^m \partial y^n} f(x,y)|_{x=x_p,y=y_p} \tag{11}
\]

where \( m \) and \( n \) are integers that determine the order of the differential operator. The DC PSE operator for the spatial derivative \( D^{m,n} f(x_p) \) is

\[
Q^{m,n} f(x_p) = \frac{1}{\varepsilon(x_p)^{m+n}} \sum_{\xi \in N(x_p)} (f(x_p) \pm f(x_{\xi})) \eta(\frac{x_p-x_{\xi}}{\varepsilon(x_p)}) \tag{12}
\]

where \( \varepsilon(x) \) is a spatially dependent scaling or resolution function, \( \eta(x,\varepsilon(x)) \) a kernel function, and \( N(x_p) \) is the set of points in the support of the kernel function. The sign in Eq. (12) is positive for \((m+n)\) odd, and negative if even. We will construct the DC PSE operators so that as we decrease the average spacing between nodes, \( \varepsilon(x_p) \rightarrow 0 \), the operator converges to the spatial derivative \( D^{m,n} f(x_p) \) with an asymptotic rate \( r \)

\[
Q^{m,n} f(x_p) = D^{m,n} f(x_p) + O(\varepsilon(x_p)^r) \tag{13}
\]

where it is convenient to explicitly define the component-wise average neighbour spacing as

\[
h(x_p) = \frac{1}{N} \sum_{x_{\xi} \in N(x)} |x_p-x_{\xi}| + |y_p-y_{\xi}| \tag{14}
\]

where \( N \) is the number of nodes in the support of \( x_p \). We need to define a kernel function \( \eta(x) \) and a scaling relation \( \varepsilon(x_p) \). To achieve this, we replace the terms \( f(x_{\xi}) \) in Eq. (12) with their Taylor series expansions around the point \( x_p \). This substitution gives:

\[
Q^{m,n} f(x_p) = \left( \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \varepsilon(x_p)^{i+j-m-n} \frac{(-1)^i j!}{i! j!} D^i j f(x_p) Z^{i,j}(x_p) \right)

\[
\pm Z^{0,0} (x_p) \varepsilon(x_p)^{-m-n} f(x_p) \tag{15}
\]
where
\[ Z_{i,j}^{l}(x_p) = \sum_{x_k \in \Lambda(x)} \frac{(x_p - x_k)^j(x_q - y_q)^j}{\epsilon(x_p)^{i+j}} \eta \left( \frac{x_p - x_q}{\epsilon(x_p)} \right) \]  

are the discrete moments of \( \eta \). Now if we restrict the scaling parameter \( \epsilon(x_p) \) to converge at the same rate as the average spacing between points \( h(x_p) \), that is
\[ \frac{h(x_p)}{\epsilon(x_p)} \in O(1) \]  
we find that the discrete moments \( Z_{i,j}^{l} \) are \( O(1) \) as \( h(x_p) \to 0 \) and \( \epsilon(x_p) \to 0 \). Given Eq. (17), the convergence rate \( r \) of the DC PSE operator \( Q^{m,n} \) is determined by the coefficients of the terms \( \epsilon(x_p)^{i+j-m-n} \) in Eq. (15). The coefficient of \( \epsilon(x_p)^{i+j-m-n} \) in Eq. (13) is required to be 1 when \( i = m \) and \( j = n \), and 0 when \( i + j + m + n < r \). This results in the following set of conditions for the discrete moments,
\[ Z_{i,j}^{l}(x_p) = \left\{ \begin{array}{ll} \frac{i!}{i!} \eta^{i+j} & \text{if } i, j \geq 0, \min \{i+j\} < r \\frac{n}{\epsilon(x_p)^{i+j}} & \text{otherwise} \end{array} \right. \]  

where \( \alpha_{\min} \) is 1 if \( m + n \) is even and 0 if odd. This is due to the zeroth moment \( Z_{0,0}^{l} \) cancelling out for odd \( i + j \). The choice of the factor \( \epsilon(x_p)^{i+j-m-n} \) in Eq. (12) acts as to simplify the expression of the moment conditions. For the kernel function \( \eta(x) \) to be able to satisfy the moment conditions given in Eq. (18) for arbitrary neighbourhood node distributions, the operator must have \( l \) degrees of freedom. This leads to the requirement that the support \( N(x) \) of the kernel function has to include at least \( l \) nodes. In this paper, we use kernel functions of the form
\[ \eta(x) = \left\{ \begin{array}{ll} \sum_{i,j} a_{i,j} x^{i} y^{j} e^{-x^2-y^2} & \sqrt{x^2+y^2} < r_c \\ 0 & \text{otherwise} \end{array} \right. \]  

This is a monomial basis multiplied by an exponential window function, where \( r_c \) sets the kernel support and the \( \alpha_{i,j} \) are scalars to be determined to satisfy the moment conditions in Eq. (18). The cut-off radius \( r_c \) should be set to include at least \( l \) collocation nodes in the support \( N(x) \). In this paper, \( r_c \) is set implicitly by using the \( l - 1 \) nearest neighbors of each node. Alternatively, adaptive methods can be used [42]. If \( \alpha_{\min} = 1 \), the \( \alpha_{0,0} \) coefficient is a free parameter and can be used to increase the numerical robustness of solving the linear system of equations for the remaining \( \alpha_{i,j} \).

To construct the operator \( Q^{m,n}(x_p) \) at node \( x_p \), the coefficients are found by solving a linear system of equations from Eqs. (19) and (18). With our choice of kernel function, we have,
\[ Q^{m,n} f(x_p) = \frac{1}{\epsilon(x_p)^{m+n}} \sum_{x_k \in \Lambda(x)} \left( f(x_k) - f(x_p) \right) p \left( \frac{x_p - x_k}{\epsilon(x_p)} \right) a^T(x_p) e^{-\frac{(x_p-x_q)^2+(y_p-y_q)^2}{\epsilon(x_p)^2}} \]  

where \( p(x) = p_1(x), p_2(x), \ldots, p_l(x) \) and \( a(x) \) are vectors of the terms in the monomial basis and of their coefficients in Eq. (19), respectively. Using this formulation, the operator system becomes trivial to solve. For example, if we set \( m = 2 \) and approximate the first spatial derivative in the \( x \) direction, \( D^{1,0} \), we have \( l = 6 \) moment conditions and the monomial basis is \( p(x,y) = \{1, x, y, x^2, y^2, x^3 y^2\} \). The linear system for the kernel coefficients then is:
\[ A(x_p) a(x_p) = b \]  
where
\[ A(x_p) = B(x_p)^T B(x_p) \in \mathbb{R}^{l \times l} \]  
\[ B(x_p) = E(x_p)^T V(x_p) \in \mathbb{R}^{k \times l} \]  
\[ b = (-1)^{m+n} D^{m,n} p(x)|_{x=0} \in \mathbb{R}^{l \times 1} \]  

The scalar number \( k \geq l \) is the number of nodes in the support of the operator, \( l \) is the number of moment conditions to be satisfied, and \( V(x_p) \) the Vandermonde matrix constructed from the monomial basis \( p(x_p) \). \( E(x_p) \) is a diagonal matrix containing the square roots of the values of the exponential window function at the neighboring nodes in the operator support. Further, for node \( x_p \) we define \( \left\{ x_q(x_p) \right\}_{q=1}^k = (x_p - x_q)_{x_q \in \Lambda(x)} \) the set of vectors pointing to \( x_p \) from all neighboring nodes.
nodes \( x_q \) in the support of \( x_p \). So then explicitly

\[
V(x_p) = \begin{bmatrix}
    p_1 \left( \frac{z_1(x_p)}{\epsilon(x_p)} \right) & p_2 \left( \frac{z_1(x_p)}{\epsilon(x_p)} \right) & \cdots & p_l \left( \frac{z_1(x_p)}{\epsilon(x_p)} \right) \\
    p_1 \left( \frac{z_2(x_p)}{\epsilon(x_p)} \right) & p_2 \left( \frac{z_2(x_p)}{\epsilon(x_p)} \right) & \cdots & p_l \left( \frac{z_2(x_p)}{\epsilon(x_p)} \right) \\
    \vdots & \vdots & \ddots & \vdots \\
    p_1 \left( \frac{z_k(x_p)}{\epsilon(x_p)} \right) & p_2 \left( \frac{z_k(x_p)}{\epsilon(x_p)} \right) & \cdots & p_l \left( \frac{z_k(x_p)}{\epsilon(x_p)} \right)
\end{bmatrix} \in \mathbb{R}^{k \times l}
\]  

(25)

\[
E(x_p) = \text{diag} \left( \frac{1}{\epsilon(x_p)} \right)
\]

(26)

Once the matrix \( A(x_p) \) is constructed at each node \( x_p \), the linear systems can be solved for the coefficients \( \alpha(x_p) \) used in the DC PSE operators at each node as in Eq. (20). The matrix \( A(x_p) \) only depends on the number of moment conditions \( l \) and the local distribution of nodes in \( N(x_p) \). Therefore, if the system in Eq. (21) is solved using a decomposition (such as LU), of \( A(x_p) \). This form can be re-used for multiple right-hand sides, i.e., for different differential operators (albeit with different convergence rates \( r \)). The matrix \( A \) has an analogue in MLS where it is often called moment matrix. This contains information about the spatial distribution of the collocation nodes around the center point \( x_p \). The invertibility of \( A \) depends entirely on that of the Vandermonde matrix \( V \), due to \( E \) being a diagonal matrix with non-zero entries. The condition number of \( A \) depends on both \( V \) and \( E \) and determines the robustness of the numerical inversion.

Using the DC PSE method the spatial derivatives \( D^0 = D^{0,n} \) up to second order are given as

\[
D^{1,0} = \frac{\partial}{\partial x}, \quad D^{0,1} = \frac{\partial}{\partial y}
\]

(27)

and

\[
D^{1,1} = \frac{\partial^2}{\partial x \partial y} = \frac{\partial^2}{\partial y \partial x}, \quad D^{2,0} = \frac{\partial^2}{\partial x^2}, \quad D^{0,2} = \frac{\partial^2}{\partial y^2}
\]

(28)

4. Numerical results

In this section, we present numerical examples that demonstrate the accuracy, efficiency and robustness of the FDMCM scheme. FDMCM numerical effectiveness heavily relies on the robust computation (in terms of accuracy and computational cost) of strain and stress (spatial derivatives of the displacement field). Derivatives are computed through the values of the neighboring nodes (nodes in the support domain), influenced only by the nodes in the vicinity in a way that only nodes at a distance shorter than a specified length (radius of the support domain) are to be included in the evaluation.

The process of finding nodes inside the support domain requires the computation of all pair-wise distances, a task that is computationally demanding (order of \( N^2 \), with \( N \) being the number of nodes), especially for large number of nodes [43]. The FDMCM method reveals important locality properties; for the 2D case 4 and 9 neighbors are defined for the interior and boundary (and near boundary) nodes, respectively, while for the 3D case 7 and 12 (we use the twelve closest neighboring nodes), accordingly. Taking advantage of these local properties, existing methodologies provide approximate solutions in much lower complexity bounds equal to \( O(\text{NlogN}) \) and \( O(N) \) using fast \( N \)-body solvers such as the Barnes–Hut algorithm [44] and Fast Multipole Methods [45]. Additionally, interactions that only involve local neighborhood can efficiently be computed in \( O(N) \) time using fast neighbor lists, such as cell lists [46] or Verlet lists [47].

In the FDMCM method computational efficiency is achieved mainly due to the symmetry of the Cartesian grid, which naturally provides the nearest neighbors. The pre-processing step introduces a computational overhead, which is linearly proportional to the number of nodes (\( O(N) \) complexity). It is obvious that computing of the nearest neighbors stands only for the case of DC PSE, consequently only for a small portion of nodes (around 3%) defining the point cloud the neighbors need to be determined. For those nodes, located on the boundary and close to it, the nearest neighbors are the nodes with distance \( r_{\text{cut}} \) less or equal to \( r_{\text{cut}} = 3h \) (with \( r_{\text{cut}} \) being the cut-off radius). Table 1 lists the overall computational cost (in seconds) needed for defining the neighbors and constructing the derivatives obtained after the discretization of the governing equations for the entire number of nodes using the DC PSE scheme (the point cloud used for the computations refers to the turbine blade problem).

Numerical examples presented depict the efficiency and the accuracy of the proposed scheme. 2D benchmark problems were studied to highlight both the accuracy and the convergence of the proposed method, while the 3D examples show its applicability to real world applications. The results obtained with the proposed scheme are compared to analytical solutions, where available, and to results obtained with linear finite elements. In these comparisons, different error measures are used:
Table 1
Computational time needed for the nearest neighbors definition and construction of the derivatives in the turbine blade problem.

<table>
<thead>
<tr>
<th># of nodes</th>
<th>Nearest neighbors (s)</th>
<th>Derivatives (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10,000</td>
<td>0.018</td>
<td>0.21028</td>
</tr>
<tr>
<td>250,000</td>
<td>0.419</td>
<td>5.09179</td>
</tr>
<tr>
<td>1,000,000</td>
<td>1.594</td>
<td>20.2145</td>
</tr>
</tbody>
</table>

Fig. 1. Geometrical configuration and boundary conditions for the pressurized cylinder test case.

- Energy and $L_2$ error norms [48]

$$E = \sqrt{\int \left( (\epsilon - \epsilon^h)^T (\sigma - \sigma^h) \right) d\Omega}$$

$$L_2 = \sqrt{\int \left( (\mathbf{u} - \mathbf{u}^h)^T (\mathbf{u} - \mathbf{u}^h) \right) d\Omega}$$

where $\epsilon$, $\epsilon^h$, $\sigma$, $\sigma^h$, $\mathbf{u}$, $\mathbf{u}^h$ are the analytical and numerical strains, stresses and displacements (in vector form), respectively.

- The normalized root mean square error (NRMSE) defined as [49]

$$NRMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left( \frac{u_i^{\text{analytical}} - u_i^{\text{numerical}}}{u_{\text{max}}^{\text{analytical}} - u_{\text{min}}^{\text{analytical}}} \right)^2}$$

where $N$ is the total number of nodes.

4.1. Problems in 2D

4.1.1. Infinite cylinder subject to internal and external pressure

As a first example, we examined the displacement and stress fields within an infinite cylindrical annulus subjected to internal and external pressure [49] (Fig. 1). The infinite cylindrical annulus benchmark problem has an analytical solution.

For an annulus with internal and external radius $a$ and $b$, respectively, subjected to internal pressure $p_1$ and external pressure $p_0$, the exact solution for plane-strain in Cartesian coordinates is given by [49]:

$$u_i = \frac{x_i}{2\mu(b^2 - a^2)} \left[ (1 - 2\nu)(p_1a^2 - p_0b^2) + (p_1 - p_0) \frac{a^2b^2}{r^2} \right]$$

(32)
\[
\sigma_{ij} = \frac{a^2b^2}{(b^2-a^2)} \left[ \left( \frac{p_1}{b^2} - \frac{p_0}{a^2} + \frac{p_1-p_0}{r^2} \right) \delta_{ij} - 2 \left( p_1 - p_0 \right) \frac{x_i x_j}{r^4} \right]
\]  
(33)

where \( \mu \) is the second Lamé constant, \( \nu \) the Poisson ratio and \( r \) the polar coordinate (\( a \leq r \leq b \)). Due to the symmetry, for the simulations only one quarter of the spatial domain has been considered. Pressure-traction boundary conditions at the inner and outer surface were applied

\[
\tau_i = \sigma_{ij} n_j = \begin{cases} 
-p_1 n_i & r = a \\
-p_0 n_i & r = b 
\end{cases}
\]  
(34)

with \( n_i \) being the outward surface normal. At \( x = 0 \) and \( y = 0 \), symmetry boundary conditions (i.e., zero normal displacement, zero tangential traction) have been enforced. We considered material with Young’s modulus of \( E = 125 \text{ GPa} \), and \( \nu = 0.33 \), corresponding to \( \mu = \frac{E}{2(1+\nu)} = 4.699 \times 10^{10} \) and \( \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} = 9.1220 \times 10^{10} \). The internal and external radii were \( a = 0.2 \text{ m} \) and \( b = 0.3 \text{ m} \). The external pressure was set to \( p_0 = 0.1 \text{ MPa} \), with a pressurized interior being \( p_1 = 50 \text{ MPa} \).

In the FDMCM method, the domain is represented by a set of nodes, with nodes representing the boundary (Fig. 2(a)) provide the average spacing \( h \) (in the pressurized cylinder the nodes on the inner and outer cylinder are used to compute the spacing \( h \)) for the Cartesian grid (Fig. 2(b)). The latter uses only the nodes located in the interior of the domain.

Special care is taken for the nodes located close to the boundary nodes (degenerated nodes, those located in distance less than \( 0.2 \text{ h} \) are deleted) as it can be seen in Fig. 2(c). The built-in MATLAB function inpolygon has been used to define the Cartesian grid nodes that are located inside the spatial domain. Special care is taken for the nodes (degenerated nodes) that are close to the boundary (with distance less than the Cartesian grid step size \( h \)); nodes where the finite difference stencil cannot be defined. Degenerated nodes, when included in the point cloud, give rise to ill-conditioned Vandermonde matrices. Consequently, they are removed and not included in the final point cloud. Following, through the FDMCM scheme spatial derivatives are computed; for the interior nodes, where the 5 nodes stencil is defined, the FD discretization is used (Eqs. (10)), while for the nodes on the boundary and interior nodes close to it (there the stencil cannot be defined) DC PSE is used Eqs. (27) and (28). An important issue is to assign boundary conditions at the corner nodes. In the strong formulation, we can decide the type (Dirichlet or Neumann) of boundary conditions applied on the corner nodes. For the nodes located at corners, traction boundary conditions were applied.

To obtain a grid independent solution successively denser grids were considered. The Cartesian grid spacing \( h \) is defined by the average spacing of the interior boundary (cylinder) nodes. The number \( n \) of nodes on each of the circular boundaries (inner and outer) were \( n = 45, 90, 180, 360, 720 \) and \( 1440 \) (\( h = 0.009, 0.0045, 0.00225, 0.001125, 0.0005625 \) and \( 0.00028125 \) resulting in \( 580, 2131, 8138, 31803, 125647 \) and \( 499502 \) nodes, respectively).

We should notice that even if the same number of nodes were used to represent the inner and the outer boundary, the spacing of the inner boundary nodes \( h_{\text{in}} \) and the spacing of the outer boundary nodes \( h_{\text{out}} \) would be different. The Cartesian grid step size \( h_c \), which is based on the boundary nodes spacing, is defined as the average of the inner and outer boundary nodes spacing, reading \( h_{\text{ave}} = (h_{\text{in}} + h_{\text{out}})/2 \). This way, we can ensure that the number of nodes in the support domain of the boundary nodes and those close to the boundary lead to well-conditioned Vandermonde matrix and, consequently to accurate and stable numerical results. The numerical results obtained were compared against the analytical solution.

Fig. 3(a) shows the normalized root mean square error (Eq. (31)) for the displacements field using FDMCM, Moving Least Squares (MLS) and DC PSE method to compute derivatives (and shape functions in the case of MLS) while Fig. 3(b) shows the normalized root mean square error for stress field components \( \sigma_{xx}, \sigma_{xy}, \sigma_{yy} \). For the displacement field error, as the number of nodes increases the error is decreasing having an erratic behavior, which tends to stabilize as the number of nodes increases. As the number of nodes used to represent the spatial domain increases, derivatives tend to be more accurately computed with the FDMCM method, leading to more accurate computation of the displacement and stress fields. On the other hand, stress errors converge smoothly showing a behavior that is less dependent on the number of nodes used. It can be observed that the numerical results computed by the FDMCM and DC PSE method are in excellent agreement, which can be explained by the fact that DC SPE is as a generalization of FD methods in complex geometries. All the meshless approximation/interpolation methods used result in the same accuracy as the number of nodes increases. FDMCM method produces linear systems having smaller bandwidth (less nearest neighbors in the support domain) than those defined in the DC PSE and MLS. Eventually, robust solvers can be used, decreasing drastically the computational cost.

Additionally, \( L^2 \) and energy error norms (Eqs. (29) and (30)) were computed, as shown in Fig. 4. The \( L^2 \) and energy norms numerical results computed are mesh independent; the number of triangular elements and, subsequently the number of Gauss points used, ensure the convergence of the numerical computations. Stress field contour plots are presented in Fig. 5.

4.1.2. Infinite plate with a circular hole

The second benchmark problem considered is the thin infinite plate with a circular hole subjected to uniform traction [48], as shown in Fig. 6.
The analytical solution for the displacement field in $x$- and $y$-directions is given as [48]:

$$u_x(r, \theta) = \frac{\sigma_0}{8\mu} \left[ \frac{r}{a} (k+1)\cos\theta + \frac{2a}{r} \left( (1+k)\cos\theta + \cos3\theta \right) - \frac{2a^3}{r^3} \cos3\theta \right]$$

$$u_y(r, \theta) = \frac{\sigma_0}{8\mu} \left[ \frac{r}{a} (k-3)\sin\theta + \frac{2a}{r} \left( (1-k)\sin\theta + \sin3\theta \right) - \frac{2a^3}{r^3} \sin3\theta \right]$$

Fig. 2. (a) Boundary nodes (b) Cartesian grid and (c) Cartesian grid boundary embedded with the nodes located inside the spatial domain (square symbol refers to boundary nodes, circular to near boundary interior nodes (DC PSE is applied) and cross corresponds to interior nodes (FD is applied)).
Fig. 3. Normalized root mean square error (NRMSE) for (a) displacement and (b) stress field. Slope is computed by using the last two points.
Fig. 4. Convergence plot for the L2 and energy norms. Slope is computed by using the last two points in the graph.

where \( r \) and \( \theta \) are the polar coordinates, \( a \) is the radius of the hole, \( \sigma_0 \) is the far-field traction (applied in the x-direction), \( \mu \) is the Lamé constant and \( \kappa \) is the Kolosov constant, which is equal to \( \kappa = 3-4\nu \) for plane strain. The stress field is given by

\[
\sigma_{xx} = \sigma_0 \left[ 1 - \frac{a^2}{r^2} \left( \frac{3\cos 2\theta + \cos 4\theta}{4} \right) + \frac{3a^4}{2r^4} \cos 4\theta \right] \\
\sigma_{xy} = -\sigma_0 \left[ \frac{a^2}{r^2} \left( \frac{\sin 2\theta + \sin 4\theta}{4} \right) - \frac{3a^4}{2r^4} \sin 4\theta \right] \\
\sigma_{yy} = -\sigma_0 \left[ \frac{a^2}{r^2} \left( \frac{\cos 2\theta - \cos 4\theta}{4} \right) + \frac{3a^4}{2r^4} \cos 4\theta \right]
\]

(37a)

(37b)

(37c)

For the numerical simulations, we used a material with Young modulus \( E = 210 \) GPa and Poisson ratio \( \nu = 0.33 \), \( \mu = \frac{E}{2(1+\nu)} = 7.895 \times 10^{10} \), \( \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} = 1.5325 \times 10^{11} \) radius being \( a = 1 \) m and applied a far-field traction of \( \sigma_0 = 100 \) MPa. Based on the symmetry of the problem we used a single quadrant, computing the numerical solution in a domain of \( 4 \times 4 \) m. On the hole boundary, we enforce zero-traction condition. On the left horizontal and on the bottom vertical walls symmetric conditions were imposed. For the remaining boundaries we used the analytical surface traction field (defined by \( \tau_1 = \sigma_0 n_1 \)).

A grid independent solution is obtained by considering successively denser grids. The spacing \( h \) is computed by the average spacing of the nodes used to represent the boundary of the circular inclusion. The number \( n \) of nodes used on the circular inclusion were \( n = 45, 90, 180 \) and 360 nodes, with spacing being \( h = 0.035, 0.0175, 0.012 \) and 0.0088 spacing (resulting to 12,590, 50,330, 106,143 and 197,950 number of nodes).

Fig. 7(a) shows the convergence plot for the normalized root mean square error norm for the displacement components, Fig. 7(b) the stress components \( \sigma_{xx}, \sigma_{xy} \) and \( \sigma_{yy} \), while Fig. 7(c) plots the L2 and Energy error norms. Numerical results obtained by the FDMCM were projected on the Gauss points defined at the triangular elements (in total 97,828 Gauss points were used for the integration ensuring a grid independent solution). Contour plots for the stress field are presented in Fig. 8.

4.1.3. Perforated plate under bending

In the third example, we examine the bending of a perforated plate under shearing stress, as it is shown in Fig. 9.

The beam has length \( L = 10 \) m, width \( D = 2 \) m and thirty circular inclusions (having radius between 0.16 and 0.2 m), distributed randomly over the spatial domain. The material properties used are \( E = 210 \) GPa and \( \nu = 0.3 \) corresponding to
The left wall ($x = 0$) has zero displacements while on the right wall ($x = L$) a bending load is enforced, with the applied end load being a shearing stress given as $\tau_{xy} = \frac{P}{4} \left( \frac{D^2}{4} - y^2 \right)$. Over the remaining boundaries we apply zero-traction condition.

To ensure a grid independent solution, computations were performed for successively denser grid configurations. The number of nodes $n$ used to represent each inclusion was $n = 45, 90, 180$ and $360$, resulting in spacing $h = 0.025, 0.0125, 0.01$ and $0.005$, respectively (the total number of nodes is 30,874, 122,594, 191,321 and 761,951, respectively). Since there is no analytical solution, the numerical results obtained by the FDMCM scheme were compared against those obtained by the commercial software ABAQUS. For a direct comparison, we used a point-wise comparison of computed displacements and stresses. We chose to conduct this comparison on the mesh nodes were the FEM solution from ABAQUS is computed. Thus, FDMCM displacement and stress field values were projected on the dense triangular mesh (for the projection we used the interpolating DC PSE method), used by ABAQUS (a dense triangular grid has been used to ensure a grid independent solution).

\[ \mu = \frac{E}{2(1 + \nu)} = 8.077 \times 10^{10} \] and \[ \lambda = \frac{E\nu}{(1 + \nu)(1 - 2\nu)} = 1.2115 \times 10^{11}. \]
Fig. 6. Geometrical configuration and boundary conditions for the infinite plate with circular hole.

Fig. 7. Convergence plots for the (a) normalized root mean square error norm for the stress components $\sigma_{xx}$, $\sigma_{xy}$ and $\sigma_{yy}$ and (b) L2 and Energy error norms. Slope is computed by using the last two points in the graph.
The finite element mesh consists of 886,878 elements and 446,634 nodes. The normalized root mean square error for the coarse distribution (of 30,874 nodes) was equal to 4.2% while for the denser one (of 761,951) was 0.084%. Displacement and stress field contour plots are shown in Fig. 10 and Fig. 11, respectively.

4.2. Case studies in 3D

4.2.1. Cantilever beam under shear load

As a benchmark 3D example, we considered a rectangular beam with length $L = 5$ m, width $2a = 1$ m and height $2b = 1$ m, as shown in Fig. 12. The beam is fixed on one end while having bending loading conditions due to an end-shear load ($F = 1000$ Pa, along the $z$-direction) [50].

The material is taken to be homogeneous and isotropic with Young’s modulus $E = 10^7$ Pa and Poisson’s ratio $\nu = 0.3$, corresponding to $\mu = \frac{E}{2(1+\nu)} = 3.846 \times 10^6$ and $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} = 5.769 \times 10^6$. The analytical solution [51] for the displacement...
Fig. 9. Geometry of the perforated plate with length \( L = 10 \) m and width \( D = 2 \) m, with thirty circular inclusions.

Fig. 10. Contour plots for (a) \( u \) and (b) \( v \) displacement field for the perforated plate.

Fig. 11. Stress field contour plots for (a) \( \sigma_{xx} \) (b) \( \sigma_{xy} \) and (c) \( \sigma_{yy} \) for the perforated plate.

Fig. 12. Geometrical configuration and boundary conditions for the rectangular beam.
field is given by

\[
\begin{align*}
    u_x &= \frac{F}{EI} \left[ \frac{z}{2} (vy^2 + x^2) + \frac{vz^3}{6} + (1 + v) \left( b^2 z - \frac{z^3}{3} \right) - \frac{\alpha^2 vz}{3} - \frac{4\alpha^3 \nu}{\pi^3} \sum_{n=1}^{\infty} \frac{(-1)^n}{n^3} \cos\left( \frac{n\pi y}{\alpha} \right) \sinh\left( \frac{n\pi z}{\alpha} \right) \right] \\
    u_y &= -\frac{Fv}{EI} xyz \\
    u_z &= \frac{F}{EI} \left[ \frac{v}{2} (y^2 - z^2) x - \frac{x^3}{6} \right]
\end{align*}
\]  

(38a)

while for the stress field, the analytical solution is given as,

\[
\begin{align*}
    \sigma_{yy} &= \sigma_{zz} = \sigma_{yx} = 0 \\
    \sigma_{xx} &= \frac{F}{I} xz \\
    \sigma_{xz} &= \frac{F}{I} \left[ \frac{2\alpha^2}{\pi^2} \frac{\nu}{1 + \nu} \sum_{n=1}^{\infty} \frac{(-1)^n}{n^2} \sin\left( \frac{n\pi y}{\alpha} \right) \frac{\sinh\left( \frac{n\pi z}{\alpha} \right)}{\cosh\left( \frac{n\pi b}{\alpha} \right)} \right] \\
    \sigma_{yz} &= \frac{F}{I} \left[ \frac{b^2 - z^2}{2} + \frac{\nu}{1 + \nu} \sum_{n=1}^{\infty} \frac{(-1)^n}{n^2} \cos\left( \frac{n\pi y}{\alpha} \right) \frac{\cosh\left( \frac{n\pi z}{\alpha} \right)}{\cosh\left( \frac{n\pi b}{\alpha} \right)} \right]
\end{align*}
\]  

(39a)

(39b)

(39c)

(39d)

where \( \nu \) is Poisson’s ratio, and \( I = 4ab^3/3 \) is the second moment of area about the \( x \)-axis.

A Cartesian grid is used to represent the spatial domain. A grid independent solution is obtained by using successively refined grids considering grid spacing \( h = 0.2, 0.1, 0.05 \) and 0.025, resulting in 936, 6171, 44,541 and 337,881 number of nodes, respectively. Considering the boundary conditions applied, the exact displacements are prescribed on the face \( x = L \), surface tractions consistent with the exact stress field are prescribed on the face \( x = 0 \), while the four remaining faces are traction free.

Fig. 13 shows the convergence plot for the normalized root mean square error norm for both displacement and stress components, while in Fig. 14 the contour plots for the displacements are shown.

4.2.2. Blade with applied external pressure

For a complex geometry test case, we considered the deformation of a turbine blade under external pressure loads, as shown in Fig. 15.

The turbine blade geometry is given as an STL file (downloaded from GrabCAD (https://grabcad.com/)) (Fig. 16(a)). The STL file provides the point cloud used to represent the blade’s boundary. The average spacing \( h \) of boundary nodes determines the Cartesian grid spacing. The spatial domain is represented by a Cartesian grid (Fig. 16(b)), which is then embedded in the geometry. Special care is taken for the nodes located close to the boundary nodes (degenerated nodes, those located in distance less than 0.2\( h \) are deleted). FDMCM is used to compute derivatives up to second order; for those nodes where the FD stencil can be defined derivatives are computed using the FD central difference scheme (first and second order derivatives), while for the nodes located on the boundary and close to it (there the FD stencil cannot be defined) the DC PSE scheme is used to compute derivatives up to second order.

We chose material properties of \( E = 125 \text{ GPa} \) and \( \nu = 0.33 \), corresponding to \( \mu = \frac{E}{2(1 + \nu)} = 4.699 \times 10^{10} \text{ and } \lambda = \frac{E\nu}{(1 + \nu)(1 - 2\nu)} = 9.122 \times 10^{10} \), representing mild steel. In the simulations we applied constant pressure of \( P_0 = 200 \text{ MPa} \). The blade has height \( H = 1.8 \text{ m} \) and its surface is represented by 38,000 triangular elements. By extending the methodology presented for 2D geometries in 3D, the spacing \( h \) for the uniform Cartesian grid is correlated to the surface mesh resolution (triangulation of the surface) and, is defined by the average radius of all circumference circles of the boundary facets of the STL file (this ensures constant density definition of the nearest neighbors of each node).

The numerical results obtained by the FDMCM scheme were compared against those obtained by the commercial software ABAQUS. We compared computed values on the mesh nodes where the ABAQUS numerical solution was computed. FDMCM displacement and stress field values were projected on the dense tetrahedral mesh (for the projection we used the interpolating DC PSE method), used by ABAQUS.

The normalized root mean square error norm has been computed and the results are listed in Table 2, considering 523,882 nodes (surface and interior nodes). Finally, Fig. 17 shows the displacement field plots.
Fig. 13. Convergence plot of the normalized root mean square error norm for (a) displacement and (b) stress components.
Fig. 14. Contour plots of the displacements for the cantilever beam test case.

Fig. 15. Geometry of the turbine blade along with the applied pressure boundary conditions.
4.2.3. Femur bone with applied external loading

As the final example, we consider the case of pressure applied on the femur bone; a test case used as an example of a geometry imperfectly described by remote sensing images. Since femur bone takes the largest percentage of the body weight, it is crucial to compute the mechanical loads that femur can be subjected and withstand.

The geometry used in the simulations along with boundary conditions applied can be seen in Fig. 18(a). The femur geometry has been reconstructed applying image analysis techniques and, is given as a triangulated surface (STL file). The average spacing $h$ of the boundary nodes determines the Cartesian grid spacing $h_c$, which when embedded in the femur geometry represents the spatial domain (Fig. 18(b)). Special care is taken for the nodes (degenerated nodes) located close to the boundary nodes (located in distance less than $0.2h$), which are removed and not included in the point cloud. The FDMCM computes derivatives up to the second order; for nodes where the FD stencil can be defined derivatives are computed using the FD central difference scheme (first and second order derivatives), while for the nodes located on the boundary and close to it (there the FD stencil cannot be defined) the DC PSE scheme is used to compute derivatives up to second order.

Three STL input files were used, having successively denser surface triangulation. As previously explained, in the context of FDMCM the Cartesian grid step size is computed based on the average radius of all circumference circles of the boundary facets of the STL files. For the STL files considered, the Cartesian grid step size was set to $h = 0.12$, $0.1$ and $0.075$, respectively, resulting to 15,000 (2008 belong to the boundaries), 26,321 nodes (2999 belong to the boundaries), 63,925 (6003 belong to the boundaries) nodes of nodes. At the femur base zero displacements were applied, while the rest of the femur bone surface has been subjected to uniform external pressure. The Young modulus and the Poisson ratio of the femur bone were set to $E = 19$ GPa and $v = 0.3$, respectively, while the applied pressure set to 20 MPa. The numerical results obtained with the proposed scheme were compared against those computed by the Finite Element method using the FEBio software. The normalized root mean square error norm for the displacement field components, considering the denser grid used of 63,925 nodes is listed in Table 3, while Fig. 19 shows the von Mises stress computed by the proposed scheme.

![Diagram](image.png)

**Fig. 16.** (a) STL file of the turbine blade (b) the Cartesian grid used for the representation of the interior domain.

**Table 2**

<table>
<thead>
<tr>
<th></th>
<th>$u$</th>
<th>$v$</th>
<th>$w$</th>
<th>$\sigma_{xx}$</th>
<th>$\sigma_{yy}$</th>
<th>$\sigma_{zz}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NRMSE</td>
<td>3.2e−02</td>
<td>4.5e−02</td>
<td>6.2e−02</td>
<td>5.2e−02</td>
<td>8.7e−02</td>
<td>4.4e−02</td>
</tr>
</tbody>
</table>

![Diagram](image.png)
Fig. 17. Contour plots of the (a) $u$ (b) $v$ and (c) $w$ displacements for the cantilever turbine blade test case.

Fig. 18. (a) Femur geometry along with the applied pressure boundary conditions (b) the Cartesian grid used for the representation of the interior domain.

<table>
<thead>
<tr>
<th>Table 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normalized root mean square error norm for displacement field.</td>
</tr>
<tr>
<td>$u$</td>
</tr>
<tr>
<td>$NRMSE$</td>
</tr>
</tbody>
</table>
5. Conclusions

We have presented a scheme aimed at providing the numerical solution for the strong form of the static momentum balance for linear elasticity in two- and three-dimension by using a hybrid finite difference-meshless collocation (FDMCM) scheme. Combining the numerical stability of FD methods with the flexibility of MM to discretize differential operators on point clouds, we efficiently and accurately solved, for the deformation and stress fields, for static linear elasticity problems using the proposed FDMCM numerical scheme.

We have demonstrated the appropriateness and effectiveness of FDMCM method on five test problems; three of them being well-known benchmark problems, two others having no known analytical solutions. For the test problems with known analytical solution (pressurized cylinder, infinite plate with hole and 3D cantilever beam), we have shown that the FDMCM numerical solution converges to the analytical solution with increasing spatial resolution. For problems having no analytical solution (blade test case, femur bone), we verified the accuracy of our numerical results by comparing with those obtained with the reliable commercial code ABAQUS.

Calculating derivatives using the FDMCM formulation results in linear systems with constant conditioning properties. The condition number of the Vandermonde matrix remains low (∼250) and independent of the smallest nodal spacing (related to the spacing h of the Cartesian grid). These are essentially the same parameters as those for a problem without complex boundaries and with the same rectangular mesh spacing [1]. This way, the decrease in the convergence and accuracy and the increase in the computational cost observed when using meshless collocation methods is compensated by using a very efficient FD method in the majority of the nodes used to represent the spatial domain. The robustness of the proposed scheme is ensured by combining FD and meshless methods. This gives rise to linear systems with smaller bandwidth compared to those obtained by pure meshless methods. Additionally, the linear systems are diagonally dominant and positive definite and can be efficiently solved by using a direct solver or a simple iterative solver [52] since the structure of the system resembles that of classical FD. Where our method differs is at the nodes located at and in close proximity to the boundaries, of which there are not more than ∼3% of the total number of nodes in typical cases. Furthermore, we can confirm that the entire procedure is relatively easy to implement. The proposed FDMCM method offers the (obvious) potential of local mesh refinement, stemming from the fact that increasing the density of a regular, Cartesian finite difference grid is straightforward. We will endeavor to exploit this potential in our future work.

The main limitation of the FDMCM method is that it has been applied only to linear problem. Before it finds its use in real-life problems in the fields of geomechanics and biomechanics, it must be extended to fully non-linear problems including finite deformations, non-linear materials and contacts. In our future work we will incorporate the FDMCM method...
into the Total Lagrangian Explicit Dynamics Scheme that we successfully used for efficient finite element [53] and weak form-based meshless computations [54–57]. The defining advantage of the proposed FDMMC method is that it uses a computational grid that can be automatically generated in a straightforward way. Therefore, when demonstrated applicable and effective in the non-linear regime the FDMMC method will bring us closer to the ideal of “an image as a model” [58].

References


