Acta Mech 228, 3581–3593 (2017)
DOI 10.1007/s00707-017-1894-4

ORIGINAL PAPER

Arman Shojaei · Farshid Mossaiby · Mirco Zaccariotto · Ugo Galvanetto

The meshless finite point method for transient elastodynamic problems

Received: 11 October 2016 / Revised: 22 February 2017 / Published online: 28 June 2017

Abstract In this paper, the application of the meshless finite point method (FPM) to solve elastodynamic problems through an explicit velocity–Verlet time integration method is investigated. Strong form-based methods, such as the FPM, are generally less stable and accurate in terms of satisfaction of Neumann boundary conditions than weak form-based methods. This is due to the fact that in such types of methods, Neumann boundary conditions must be imposed by a series of equations which are different from the governing equations in the problem domain. In this paper, keeping all the advantages of FPM in terms of simplicity and efficiency, a new simple strategy for proper satisfaction of Neumann boundary conditions in time for elastodynamic problems is investigated. The method is described in detail, and several numerical examples are presented. Moreover, the accuracy of the method with reference to the solution of some 3D problems is discussed.

1 Introduction

During the past few decades, the computational mechanics community has given specific attention to so-called mesh reduction methods. Thereupon, there is a fast growing interest in developing meshless (or mesh-free) methods as an alternative to conventional mesh-based methods such as the finite element method (FEM). Although FEM has been developed thoroughly and applied successfully to a variety of engineering problems, the task of mesh generation can be very costly and burdensome for three-dimensional problems especially those with complicated domains. The key idea of meshless methods is to provide numerical solutions on a set of arbitrarily distributed nodes instead of elements, and replacing meshing or re-meshing procedures with adding or eliminating nodes at desired parts. The reader may refer to [14,17] for development history of meshless methods. Depending on how equations are discretized, meshless methods can be classified into two major categories. The first category constitutes meshless methods based on weak form such as the element-free Galerkin method [3]. Most of them are only meshless in terms of the numerical approximation of field variables, and they are involved with numerical integration using a background mesh over the problem domain, which makes them computationally expensive and not "truly" meshless.

The second category is meshless methods based on the strong form such as the finite point method (FPM) [19]. These methods often use the point collocation method to satisfy the governing differential equations such

A. Shojaei · M. Zaccariotto · U. Galvanetto
Industrial Engineering Department, University of Padova, v. Venezia 1, 35131 Padua, Italy

A. Shojaei · M. Zaccariotto · U. Galvanetto
Center of Studies and Activities for Space (CISAS)-"G. Colombo", v. Venezia 15, 35131 Padua, Italy

F. Mossaiby (✉)
Department of Civil Engineering, University of Isfahan, Isfahan 81744-73441, Iran
E-mail: mossaiby@eng.ui.ac.ir

as in [8], or they do the approximation using base functions which satisfy the governing differential equation as in [16,24,25]. Since they do not need any background mesh, they are truly meshless, simpler to implement, and computationally less expensive than meshless methods based on weak form.

FPM proposed by Oñate et al. [19] is a notable method within the context of truly meshless methods. The extension of the method to the solution of static elasticity problems can be found in [20]. FPM uses a weighted least squares (WLS) scheme for approximating the unknown field function. The approximation can be easily constructed to have a consistency of a desired order, and by adopting the point collocation method the discrete equations can be obtained. FPM has received considerable attention in different studies and applications; see for example [2,4,7,9,21,28,30].

However, strong form-based methods such as FPM, in comparison with weak form-based methods, are most often less stable and accurate for problems governed by partial differential equations with Neumann (derivative) boundary conditions such as solid mechanics problems with stress (natural) boundary conditions. These methods fall within a category of methods called direct meshless collocation methods [23]. In such type of methods, Neumann boundary conditions should be imposed directly through a series of independent equations, which are different from the governing equations in the problem domain. This contributes to a poor accuracy on Neumann boundaries, and it may be taken as the main source of instability for collocation methods [11,31]. In this regard, several studies by different researchers have been proposed to circumvent this deficiency. Oñate [18] and Oñate et al. [20] proposed a stabilized version of FPM using finite calculus (FIC). A simple modification to stabilize FPM using FIC was proposed by Boroomand et al. [5]. Shu et al. [27] used several layers of orthogonal grids near and on the boundaries for proper satisfaction of Neumann conditions. La Rocca and Power [10] introduced a double-boundary collocation Hermitian technique in which at the boundary collocation points the governing differential equations and the boundary conditions are coupled and satisfied simultaneously. Lie and Gu [12] and Liu et al. [13] proposed a combined formulation that benefits from both the local weak form and strong form equations. In this sense, the strong form formulation is applied to all nodes whose local support domains do not intersect with Neumann boundaries, while the application of weak form formulation is restricted to nodes on or near the Neumann boundaries. Pursuing such an idea, Sadeghirad and Mohammadi [22,23] proposed the equilibrium on line method (ELM) for imposition of Neumann boundary conditions in FPM using straight line integration domains.

Dynamic analysis of elastic structures is an important issue in various areas of engineering. The solution of such problems with meshless methods is still the subject of different studies in the literature as in [6,15]. In this paper, the application of FPM to elastodynamic problems is investigated. Here the main attempt is to extend the solution of FPM in time for elasticity problems while keeping all its main advantages in terms of efficiency and simplicity of implementation. In this way, for the nodes in the body the time marching is performed by using and explicit velocity–Verlet time integration method. Moreover, for the nodes located on Neumann boundaries a simple technique is introduced to update the nodal displacements of the boundary nodes by the solution of a set of linear equations in time. This system of equations includes Neumann boundary nodes as well as all their adjacent nodes that fall within their support domains. The displacements of the nodes on Neumann boundaries are updated at each time step, simultaneously, by a set of equilibrated equations that correspond to body nodes that are consistent with the governing equation of body itself. This strategy makes the solution to proceed in time appropriately. The similar technique has been used by a recent study of the authors in [26] for some 2D problems, yet a detailed formulation about the satisfaction of dynamic boundary conditions was not provided. In the present paper, the formulation is thoroughly explained; moreover, a short technical description on implementation of the work in an Open Multi-Processing application programming interface (OpenMP) is discussed. We shall show that the proposed technique preserves the appealing features of FPM from the implementation point of view. We shall assess the accuracy of the method through some benchmark examples including some 3D dynamic problems. To the best of the authors' knowledge, this is the first study on the application of FPM to the solution of 3D elastodynamic problems.

The organization of this paper is as follows: In Sect. 2, the mathematical description of the elastodynamic problems is provided, Sect. 3 is devoted to the explanation of the methodology applied in the present study, in Sect. 4 the capability of the proposed method through some benchmark problems is discussed, and in Sect. 5 the conclusions are summarized.

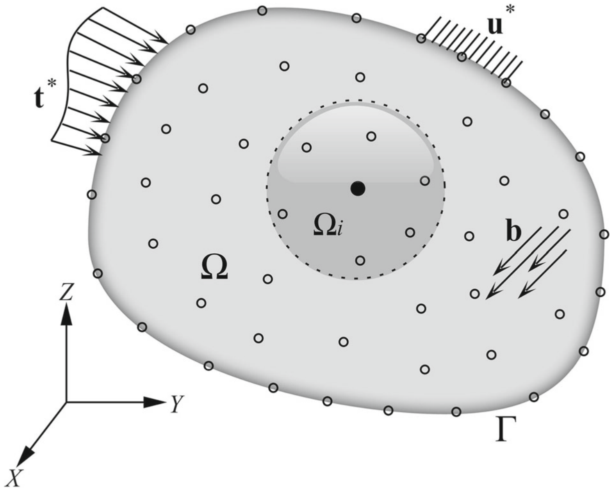

Fig. 1 Domain representation of a general three-dimensional problem in a meshless style

2 Problem description

Consider a linear elastic body in a three-dimensional domain Ω bounded by a boundary Γ consisting of a Dirichlet part Γ_D constrained by prescribed displacements $\mathbf{u}^*$, and a Neumann part Γ_N where prescribed tractions $\mathbf{t}^*$ are imposed (see Fig. 1).

In an isotropic homogenous medium for any point of the domain Ω, with coordinates $\mathbf{X} = (X, Y, Z)$, the governing equations of motion, at time t, can be written in the following form:

$$\mathbf{S}^T\mathbf{DSu}(\mathbf{X}, t) + \mathbf{b}(\mathbf{X}, \mathbf{t}) = c\dot{\mathbf{u}}(\mathbf{X}, t) + \rho\ddot{\mathbf{u}}(\mathbf{X}, t), \quad \mathbf{X} \in \Omega \tag{1}$$

with prescribed boundary conditions:

$$\tilde{\mathbf{n}}\mathbf{DSu}(\mathbf{X}, t) + \mathbf{b} = \mathbf{t}^*(\mathbf{X}, t), \quad \mathbf{X} \in \Gamma_N, \tag{2a}$$

$$\mathbf{u}(\mathbf{X}, t) = \mathbf{u}^*(\mathbf{X}, t), \quad \mathbf{X} \in \Gamma_D, \tag{2b}$$

and initial displacement and velocity conditions:

$$\mathbf{u}(\mathbf{X}, 0) = \mathbf{u}_0(\mathbf{X}), \quad \mathbf{X} \in \Omega, \tag{3a}$$

$$\dot{\mathbf{u}}(\mathbf{X}, 0) = \mathbf{v}_0(\mathbf{X}), \quad \mathbf{X} \in \Omega, \tag{3b}$$

where $\mathbf{u}$ is the displacement vector with $\langle u, v, w\rangle$ components, respectively, along the X, Y and Z directions in the global coordinate system. $\dot{\mathbf{u}}$, $\ddot{\mathbf{u}}$ and $\mathbf{b}$ are the vectors of velocity, acceleration and body forces, respectively. ρ is the mass density, and c represents the damping coefficient. $\mathbf{S}$ is the well-known elasticity operator defined as:

$$\mathbf{S}^T = \begin{bmatrix} \partial/\partial X & 0 & 0 & \partial/\partial Y & 0 & \partial/\partial Z \\ 0 & \partial/\partial Y & 0 & \partial/\partial X & \partial/\partial Z & 0 \\ 0 & 0 & \partial/\partial Z & 0 & \partial/\partial Y & \partial/\partial X \end{bmatrix} \tag{4}$$

and $\mathbf{D}$ is the matrix of material constants defined as:

$$\mathbf{D} = \begin{bmatrix} D_1 & D_2 & D_2 & 0 & 0 & 0 \\ & D_1 & D_2 & 0 & 0 & 0 \\ & & D_1 & 0 & 0 & 0 \\ & & & D_3 & 0 & 0 \\ & & & & D_3 & 0 \\ \text{Sym.} & & & & & D_3 \end{bmatrix} \tag{5}$$

and the parameters in the above equation are defined as:

$$\langle D_1, D_2, D_3\rangle = \frac{E}{(1+\nu)(1-2\nu)}\langle 1-\nu, \nu, (1-2\nu)/2\rangle, \tag{6}$$

where ν is Poisson's ratio and E is Young's modulus. $\tilde{\mathbf{n}}$ is a matrix containing n_X, n_Y and n_Z which are the components of the outward unit vector normal to the boundary, defined as:

$$\tilde{\mathbf{n}} = \begin{pmatrix} n_X & 0 & 0 & n_Y & 0 & n_Z \\ 0 & n_Y & 0 & n_X & n_Z & 0 \\ 0 & 0 & n_Z & 0 & n_Y & n_X \end{pmatrix}. \tag{7}$$

3 The solution strategy

3.1 Approximation scheme

Here, we recall the weighted least square (WLS) approximation scheme employed for FPM, and one may consult [20] to get more insight into the details of the formulation. Let $\mathbf{X}_i \in \Omega, i = 1, 2, \ldots$ be a collection of nodes scattered within the solution domain and on the boundaries (see Fig. 1). Accordingly, the time is discretized into stances as $t_1, t_2, \ldots, t_n, t_{n+1}, \ldots$. Around each of the nodes, a sub-domain Ω_i, so-called *cloud*, is considered. Each sub-domain Ω_i has a local coordinate system, with origin on the node and parallel to the global coordinate system. Ω_i contains the neighboring nodes of $\mathbf{X}_i$ as $\mathbf{x}_j$, $j = 1, 2, \ldots, n_i$ over which a displacement variable, to exemplify u at time step t_n, could be approximated by $\hat{u}^n$ locally as:

$$\hat{u}^n(\mathbf{x}) = \sum_{j=1}^{m} p_j(\mathbf{x})\alpha_j^n = \mathbf{p}^T(\mathbf{x})\boldsymbol{\alpha}^n, \quad \mathbf{x} \in \Omega_i, \tag{8}$$

where $\mathbf{x} = (x, y, z)$ stands for the local coordinate system, $\mathbf{p}(\mathbf{x})$ indicates a vector consisting of monomial bases and $\boldsymbol{\alpha}$ is a vector of unknown coefficients to be found in terms of nodal values. Considering a complete set of monomials, for a 3D problem one can specify $\mathbf{p}$; for instance,

$$\mathbf{p} = \langle 1, x, y, z, x^2, xy, y^2, yz, zx, z^2 \rangle^T, \quad m = 10. \tag{9}$$

To proceed with the approximation, u can be sampled at the n_i nodes of Ω_i as:

$$\bar{\mathbf{u}}^n = \begin{Bmatrix} \bar{u}_1^n \\ \bar{u}_2^n \\ \vdots \\ \bar{u}_{n_i}^n \end{Bmatrix} \approx \begin{Bmatrix} \hat{u}_1^n \\ \hat{u}_2^n \\ \vdots \\ \hat{u}_{n_i}^n \end{Bmatrix} = \begin{bmatrix} \mathbf{p}^T(\mathbf{x}_1) \\ \mathbf{p}^T(\mathbf{x}_2) \\ \vdots \\ \mathbf{p}^T(\mathbf{x}_{n_i}) \end{bmatrix} \boldsymbol{\alpha}^n = \mathbf{C}\boldsymbol{\alpha}^n. \tag{10}$$

$\mathbf{C}$ is regarded as the moment matrix associated with the local approximation within Ω_i. Assuming $n_i > m$, then $\mathbf{C}$ is not a square matrix; in turn, the approximation cannot fit to all the values of $\bar{\mathbf{u}}^n$. Likewise, the approximation entails a WLS procedure that results in a minimization of a norm J as follows:

$$J = \sum_{j=1}^{n_i} w(r_j)\left(\bar{u}_j^n - \hat{u}(\mathbf{x}_j^n)\right)^2 = \sum_{j=1}^{n_i} w(\mathbf{x}_j)\left(\bar{u}_j^n - \mathbf{p}^T(\mathbf{x}_j)\boldsymbol{\alpha}^n\right)^2, \tag{11}$$

where $r_j = \left|\mathbf{x}_j\right|$ and w is a weight function that should be taken suitably for Ω_i. In this work, we take as suggested in [5]:

$$w(r_j) = \frac{1 - \exp(64 - 16r_j^2/\delta^2)}{1 - \exp(64)}, \tag{12}$$

where δ indicates the distance of the most remote node of the cloud from the central node. Minimization of the norm J in Eq. (11) with respect to $\boldsymbol{\alpha}^n$ yields the following system of equations:

$$\mathbf{A}\boldsymbol{\alpha}^n = \mathbf{B}\bar{\mathbf{u}}^n, \tag{13}$$

where

$$\mathbf{A} = \sum_{j=1}^{n_i} w(r_j)\mathbf{p}(\mathbf{x}_j)\mathbf{p}^T(\mathbf{x}_j) \tag{14}$$

and

$$\mathbf{B} = \left[\, w(r_1)\mathbf{p}(\mathbf{x}_1)\ w(r_2)\mathbf{p}(\mathbf{x}_2)\ \cdots\ w(r_{n_i})\mathbf{p}(\mathbf{x}_{n_i})\,\right]. \tag{15}$$

Solving Eq. (13) for $\boldsymbol{\alpha}^n$ results in:

$$\boldsymbol{\alpha}^n = \mathbf{A}^{-1}\mathbf{B}\bar{\mathbf{u}}^n. \tag{16}$$

Finally, $\hat{u}^n(\mathbf{x})$ can be obtained in terms of nodal values, by substitution of $\boldsymbol{\alpha}^n$ from Eq. (16) into (8) which gives:

$$\hat{u}^n(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}\mathbf{B}\bar{\mathbf{u}}^n = \sum_{j=1}^{n_i} N_j(\mathbf{x})\bar{u}^n_j. \tag{17}$$

3.2 Discretization of governing equations

The discretized system of equations in FPM can be easily obtained by substituting the approximated functions of displacements $\mathbf{u}^n(\mathbf{x}) = \langle \bar{u}^n(\mathbf{x}), \bar{v}^n(\mathbf{x}), \bar{w}^n(\mathbf{x})\rangle^T$, from Eq. (17), in Eqs. (1) and (2) which results in

$$(\mathbf{S}^T\mathbf{DSN})\Big|_{\mathbf{X}_i} \mathbf{u}^n_R + \mathbf{b}^n_i = \rho\ddot{\mathbf{u}}^n_i + c\dot{\mathbf{u}}^n_i, \quad \mathbf{X}_i \in \Omega, \tag{18}$$

where $\mathbf{u}^n_R = \langle \bar{u}^n_1, \bar{v}^n_1, \bar{w}^n_1, \bar{u}^n_2, \bar{v}^n_2, \bar{w}^n_2, \ldots, \bar{u}^n_{n_i}, \bar{v}^n_{n_i}, \bar{w}^n_{n_i}\rangle^T$ is the vector of nodal displacements in the cloud and $\mathbf{N}$ is the matrix of shape functions as follows:

$$\mathbf{N} = \begin{pmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & \cdots \\ 0 & N_1 & 0 & 0 & N_2 & 0 & \cdots \\ 0 & 0 & N_1 & 0 & 0 & N_2 & \cdots \end{pmatrix}. \tag{19}$$

Consequently, for the Neumann boundary nodes one can conclude:

$$\mathbf{t}^*(\mathbf{X}_i, t^n) = (\tilde{\mathbf{n}}\mathbf{DSN}^T)\Big|_{\mathbf{X}_i} \mathbf{u}^n_R, \quad \mathbf{X}_i \in \Gamma_N. \tag{20}$$

3.3 Time integration and boundary condition satisfaction

In the present work, the time integration in Eq. (18) is carried out based on an explicit velocity–Verlet scheme:

$$\dot{\mathbf{u}}^{n+1/2} = \dot{\mathbf{u}}^n + \frac{\Delta t}{2}\ddot{\mathbf{u}}^n, \tag{21a}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\dot{\mathbf{u}}^{n+1/2}, \tag{21b}$$

$$\dot{\mathbf{u}}^{n+1} = \dot{\mathbf{u}}^{n+1/2} + \frac{\Delta t}{2}\ddot{\mathbf{u}}^{n+1}, \tag{21c}$$

in which Δt is the constant time step which must be taken less than the critical time step Δt_c:

$$\Delta t_c = \Delta_{\min}/c_k, \tag{22}$$

where $\Delta_{\min}$ is the minimum nodal distance in the discretized domain and c_k is the maximum speed of sound in the material. In turn, Δt must be chosen within the following range:

$$\Delta t \leq \Delta_{\min}\sqrt{\frac{3(1-\nu)\rho}{E}}. \tag{23}$$

Having known the displacement and velocity vectors of each node at time step n, the displacement vector at the next time step, on the basis of Eq. (21), can be found as:

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\dot{\mathbf{u}}^n + \frac{\Delta t^2}{2}\ddot{\mathbf{u}}^n, \tag{24}$$

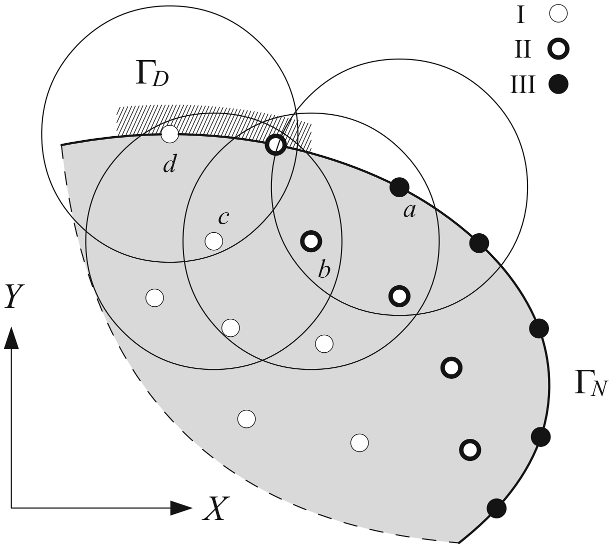

Fig. 2 Boundary of a generic discretized problem domain

and hence, for the degrees of freedom (DOFs) inside the domain one may easily determine $\ddot{\mathbf{u}}^{n+1}$ at each time step from Eq. (18) and find $\mathbf{u}^{n+1}$. However, for nodes exposed to Neumann boundary conditions progression in time cannot be reached by using Eq. (24). This is due to the fact that for these nodes the acceleration term in is not appeared directly in Eq. (20); therefore, a suitable strategy to manage such a problem should be devised.

We explain the solution strategy for a general 2D problems as shown in Fig. 2. “c” and “b” are two body nodes, and “a” and “d” are two boundary nodes with Neumann and Dirichlet condition, respectively. For the sake of brevity, we assume that at each boundary node prescribed boundary conditions are of either Dirichlet or Neumann type; the generalization to 3D problems as well as nodes with mixed boundary condition can be done easily. Figure 2 represents a small portion of a generic discretized body close to its boundary. The distributed nodes, based on their position in the solution domain, are classified into three types as I, II and II. Type I stands for the nodes whose cloud does not contain any node with Neumann boundary condition (“c” and “d”). Type II (“b”) represents the nodes whose clouds contain at least a node with Neumann boundary condition, and Type III (node “a”) are the nodes located on Neumann boundaries. It can be concluded that Type II nodes play the role of an interface layer of nodes between Neumann boundaries and the other parts of the body.

The displacement vectors for Types I and II nodes (excluding Dirichlet nodes) at time step $n+1$ can be found explicitly. It suffices to obtain the vector of acceleration at time step n by using Eq. (18) and advancing to the next step using Eq. (24). To exemplify, for nodes “c” and “b” one can write:

$$\mathbf{u}_b^{n+1} = \mathbf{u}_b^n + \cdots + \boldsymbol{\psi}_{ba}\mathbf{u}_a^n + \boldsymbol{\psi}_{bb}\mathbf{u}_b^n + \boldsymbol{\psi}_{bc}\mathbf{u}_c^n + \cdots + \boldsymbol{\theta}_b^n, \tag{25a}$$

$$\mathbf{u}_c^{n+1} = \mathbf{u}_c^n + \cdots + \boldsymbol{\psi}_{cb}\mathbf{u}_b^n + \boldsymbol{\psi}_{cc}\mathbf{u}_c^n + \boldsymbol{\psi}_{cd}\mathbf{u}_d^n + \cdots + \boldsymbol{\theta}_c^n, \tag{25b}$$

where $\boldsymbol{\psi}_{ij} = \frac{\Delta t^2}{2\rho}(\mathbf{S}^T\mathbf{DSN}_k)\Big|_{\mathbf{X}_i}\mathbf{u}_j^n$, and $\boldsymbol{\theta}_i^n = (\Delta t - \frac{c\Delta t^2}{2\rho})\dot{\mathbf{u}}_i^n + \frac{\Delta t^2}{2\rho}\mathbf{b}_i^n$ supposing that $\mathbf{X}_j$ is the kth family node of $\mathbf{X}_i$. Updating the displacement vector of “d,” as a node with Dirichlet condition, can be made easily as:

$$\mathbf{u}_d^{n+1} = \mathbf{u}^*(\mathbf{X}_d, t^{n+1}). \tag{26}$$

It can be concluded that for Type I and II nodes the solution procedure deals with an uncoupled set of equations which can be assembled in matrix form as

$$\mathbf{U}_{\mathrm{I,II}}^{n+1} = \mathbf{K}_{\mathrm{I,II}}\mathbf{U}^n + \mathbf{F}_{\mathrm{I,II}}^n \tag{27}$$

in which the above equation, in the view of Fig. 2, can be derived as:

$$\begin{Bmatrix} \vdots \\ \mathbf{u}_b^{n+1} \\ \vdots \\ \mathbf{u}_c^{n+1} \\ \mathbf{u}_d^{n+1} \\ \vdots \end{Bmatrix} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \boldsymbol{\psi}_{ba} & \cdots & \mathbf{I}+\boldsymbol{\psi}_{bb} & \cdots & \boldsymbol{\psi}_{bc} & \mathbf{0} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \mathbf{0} & \cdots & \boldsymbol{\psi}_{cb} & \mathbf{I}+\boldsymbol{\psi}_{cc} & \boldsymbol{\psi}_{cd} & \cdots & \cdots \\ \cdots & \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{0} & \mathbf{0} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{Bmatrix} \vdots \\ \mathbf{u}_a^n \\ \vdots \\ \mathbf{u}_b^n \\ \vdots \\ \mathbf{u}_c^n \\ \mathbf{u}_d^n \\ \vdots \end{Bmatrix} + \begin{Bmatrix} \vdots \\ \boldsymbol{\theta}_b^n \\ \vdots \\ \boldsymbol{\theta}_c^n \\ \mathbf{u}^*(\mathbf{X}_d, t^{n+1}) \\ \vdots \end{Bmatrix}, \tag{28}$$

where $\mathbf{I}$ stands for an identity matrix. According to Eq. (20), for “a” as a node with Neumann condition one may conclude:

$$\cdots + \bar{\boldsymbol{\psi}}_{aa}\mathbf{u}_a^{n+1} + \bar{\boldsymbol{\psi}}_{ab}\mathbf{u}_b^{n+1} + \cdots = \mathbf{t}^*(\mathbf{X}_a, t^{n+1}), \tag{29}$$

where $\bar{\boldsymbol{\psi}}_{ij} = (\tilde{\mathbf{n}}\mathbf{DSN}_k)|_{\mathbf{X}_i}\,\mathbf{u}_j^n$ supposing that $\mathbf{X}_j$ is the kth family node of $\mathbf{X}_i$. It is clear that for updating the displacements of “d” the updated displacements of all its family nodes are required. Some of these nodes are of Type II, and thus, they have so far been updated by using Eq. (27), yet for the rest family nodes (Type III nodes) their updated displacement is not available. In a nutshell, a coupled system of equations for a narrow layer of nodes close to the Neumann boundaries is obtained. Therefore, at each time step the solution of a linear system of equations is required, which can be written as follows:

$$\mathbf{U}_{\mathrm{II,III}}^{n+1} = \mathbf{K}_{\mathrm{II,III}}^{-1}\mathbf{F}_{\mathrm{II,III}}^{n+1}. \tag{30}$$

It is noteworthy that as the matrix $\mathbf{K}_{\mathrm{II,III}}$ is constant during the analysis, its factorization can be done once, leading to a huge saving in computational time. In the view of Fig. 2, the above equation can be expressed as follows:

$$\begin{Bmatrix} \vdots \\ \mathbf{u}_a^{n+1} \\ \vdots \\ \mathbf{u}_b^{n+1} \\ \vdots \end{Bmatrix} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \bar{\boldsymbol{\psi}}_{aa} & \cdots & \bar{\boldsymbol{\psi}}_{ab} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \mathbf{0} & \cdots & \mathbf{I} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}^{-1} \begin{Bmatrix} \vdots \\ \mathbf{t}^*(\mathbf{X}_a, t^{n+1}) \\ \vdots \\ \mathbf{u}_b^{n+1} \\ \vdots \end{Bmatrix}. \tag{31}$$

Having solved the above equation, the displacements of all the nodes with Neumann conditions can be found. It should be pointed out that in contrary to direct imposition of Neumann boundary conditions, which is the case for the static solution of conventional FPM, here the discretized Neumann equations, associated with Type III nodes, at each time step are satisfied with a set of *equilibrated* equations, associated with Type II nodes, consistent with equations governing the body itself. In Sect. 4, we shall show the suitability of the proposed approach for some benchmark problems.

3.4 Implementation

The formulation presented in the previous section is implemented in a C++ program. The program was compiled using Microsoft Visual Studio 2015. To take advantage of available multi-core CPUs, all possible parts of the program were parallelized using the Open Multi-Processing (OpenMP) directives. OpenMP is an open standard for shared memory parallelization adopted by a large number of C, C++ and Fortran compilers. In this parallelization scheme, the blocks of code to be parallelized (usually `for` loops) are marked by special `#pragma omp` directives. The compiler generates the appropriate parallel code for blocks. Special care must be taken to avoid *race conditions*, i.e., where the result of an operation depends on the order execution of different threads (which is indeterministic). OpenMP uses system threads for its parallel execution model. Each thread is an independent path of execution, which if possible is scheduled on a separate core in multi-core CPUs by the operating system to leverage full power of such hardware platform.

An important step in the solution procedure is solution of a linear system of equations in Eq. (30). To this end, we employed the “AMGCL” library [1]. AMGCL is a light, header-only, templated C++ library for

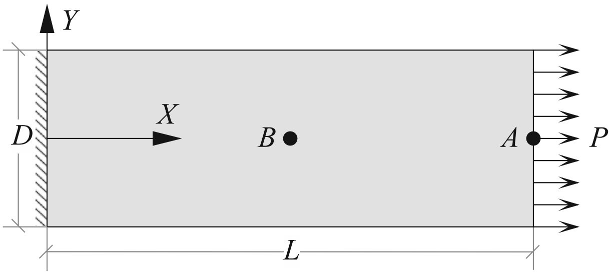

Fig. 3 Problem domain in Example 1

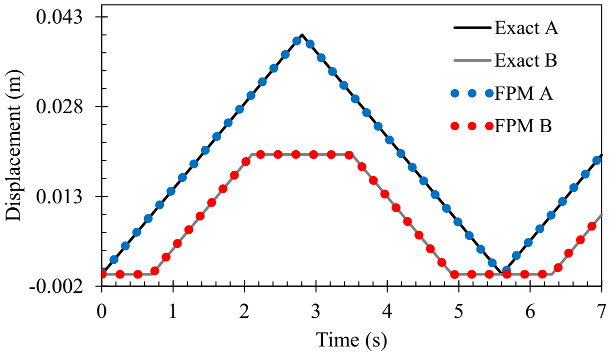

Fig. 4 Variations of horizontal displacement at points A and B in Example 1; $\Delta x = 0.125$ m

solution of sparse linear systems of equations with several options for solver, pre-conditioner and parallelization strategy. It includes implementation of algebraic multi-grid (AMG) methods which enhances the performance of solution. As the coefficient matrix in Eq. (30) is constant during the solution, setting up the required data structures and factorizing the matrix is done only once and the computational cost is greatly reduced.

4 Numerical examples

In this section, three numerical examples are described to illustrate the potentialities of the presented approach. In Example 1, the results are compared with the exact solution through a norm of error defined at node $\mathbf{X}_i$ as

$$e = \sqrt{\sum_{j=1}^{\eta}(u_{\text{ex}}(\mathbf{X}_i, t_j) - u_i^n)^2 / \sum_{j=1}^{\eta}(u_{\text{ex}}(\mathbf{X}_i, t_j))^2}, \tag{32}$$

where u_{ex} indicates the exact solution and η stands for the number of time steps based on which the norm is calculated.

Example 1 In this example, we consider a rectangular plate fixed rigidly at its base and subjected to an impulsive load at the free edge, as shown in Fig. 3. The geometric, mechanical and loading parameters are: length $L = 8$ m, height $D = 2$ m, $E = 8 \times 10^4$ Pa, $\nu = 0$, $\rho = 2450\,\text{kg/m}^3$, $c = 0$, and traction $P = 200$ Pa. The exact solution of this example is given by [29]:

$$u(X, t) = \frac{8PL}{\pi^2 E} \sum_{j=1}^{\infty} \frac{(-1)^{j-1}}{(2j-1)^2} \sin \frac{(2j-1)\pi X}{2L} (1 - \cos \omega_j t) \tag{33}$$

in which $\omega_j = \frac{(2j-1)\pi}{2L}\sqrt{E/\rho}$. The solution of this example using the present approach for a time duration of 7 seconds is considered. The domain is discretized through a Cartesian uniform grid of nodes with equal nodal spacing of Δx both in horizontal and vertical directions. To show the convergence of the solution obtained by the present approach, we solve the problem for three different grid sizes taking Δx equal to 0.5, 0.125, and 0.03125 m. Meanwhile, in all the cases Δt is taken to be 0.0005 s. The horizontal displacements of node A and

Table 1 Obtained norms of error in Example 1 at nodes A and B

Δx	0.5 m	0.125 m	0.03125 m
A	0.011934	0.004388	0.003270
B	0.015454	0.005628	0.004152

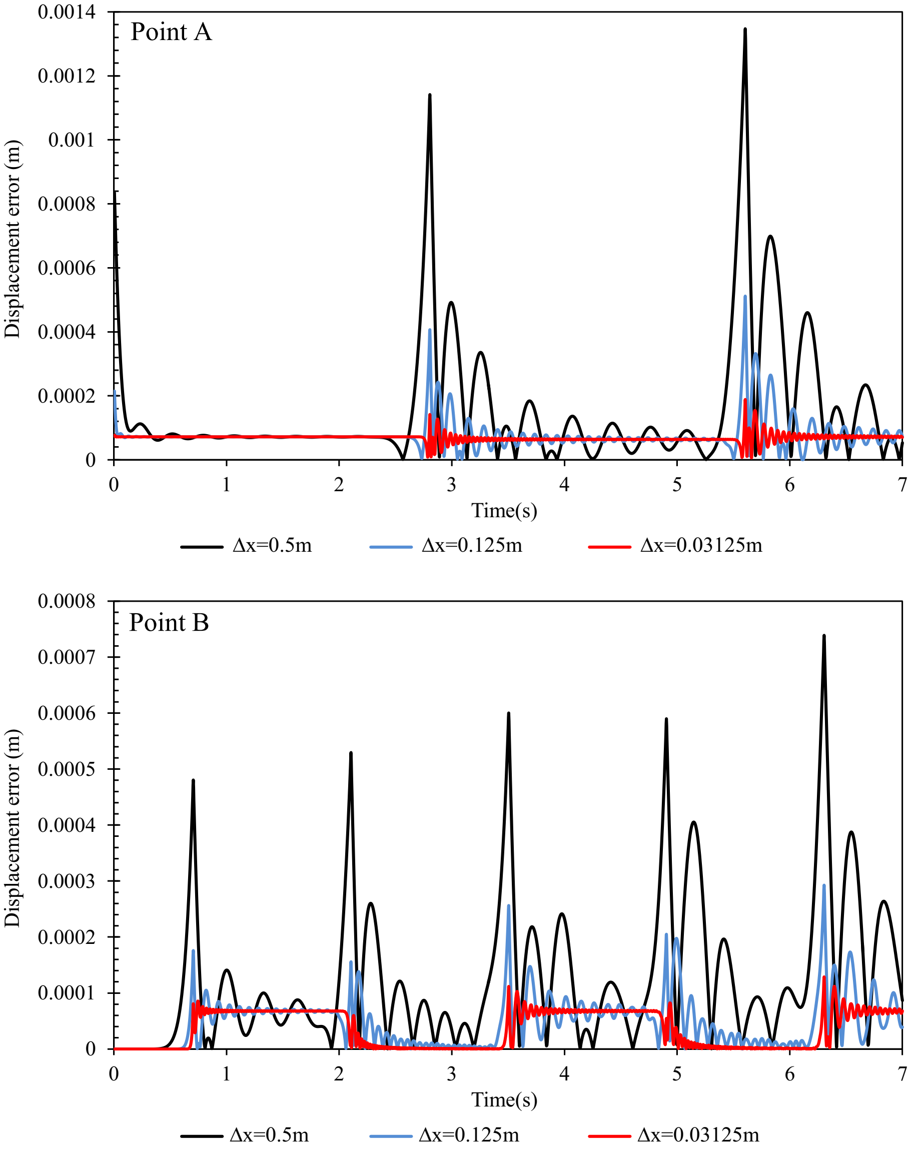

Fig. 5 Variations of displacement error at points A and B in Example 1

B, as shown in Fig. 3, are presented in Fig. 4. As a consequence, the results obtained by FPM are in perfect agreement with the exact solution.

The obtained norm of error, based on Eq. (32), for the horizontal displacement of nodes A and B is reported in Table 1. The variation of displacement error at points A and B, for different nodal spacings, is shown in Fig. 5. It can be concluded that the method performs well and its solution converges to the exact solution by refining the solution domain. That is also true for point A where Neumann boundary conditions are applied.

Example 2 In this example, we consider a 3D cantilever beam with a rectangular cross section as shown in Fig. 6. The beam is subjected to a periodic shear stress distribution $q(t) = \tau \sin \omega_f t$ at the free end. The basic parameters are: length $L = 48$ m, depth $D = 12$ m, width $W = 6$ m, $E = 3 \times 10^4$ Pa, $\nu = 1/3$, $\rho = 1$ kg/m^3,

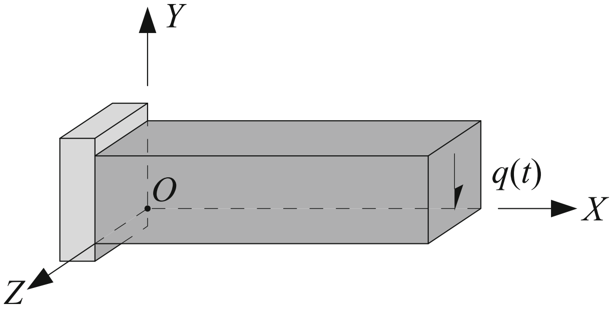

Fig. 6 Problem domain in Example 2

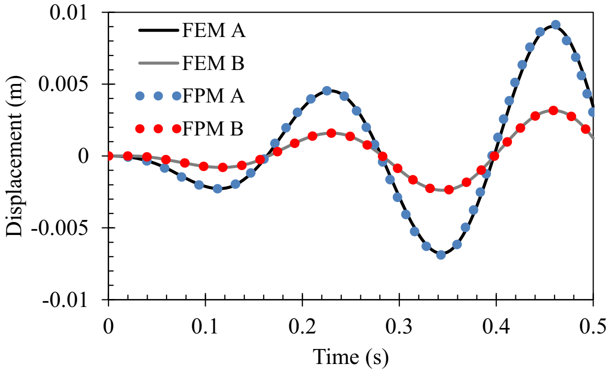

Fig. 7 Variations of horizontal displacement at points A and B in Example 2; $\Delta x = 0.6\,\mathrm{m}$

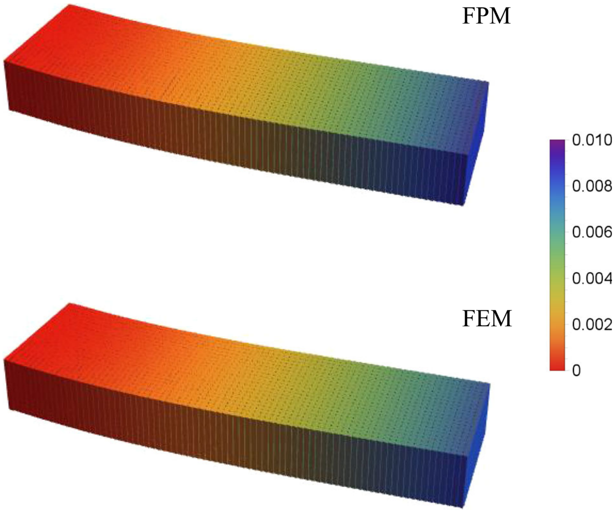

Fig. 8 Contour plot of displacement (m) along Y axis obtained by both FPM and FEM methods, at $t = 0.44\,\mathrm{s}$, in Example 2

and $c = 0$. The magnitude and frequency of the excitation are, respectively, as $\tau = -13.89\,\mathrm{N/m^2}$ and $\omega_f = 27\,\mathrm{rad/s}$.

To proceed with the solution of FPM, the domain is discretized by a Cartesian uniform grid of nodes with a nodal spacing of $\Delta x = 0.6\,\mathrm{m}$. To check the suitability of the results, the solution of a standard/explicit FEM model using 16,000 linear cubic elements is taken into account, as well. The nodal spacing in both models is identical which results in 18,711 number of nodes. We consider the solution for a duration of 0.5 s, and in both models Δt is taken to be 0.00002 s. The variation of displacements along Y axis at point A at the free end, $\mathbf{X}_A = (48, 6, 3)$, and point B at the center of the beam, $\mathbf{X}_B = (24, 6, 3)$, is shown in Fig. 7. The results illustrate that the results obtained by FPM are in excellent agreement with those of FEM. The contour plot

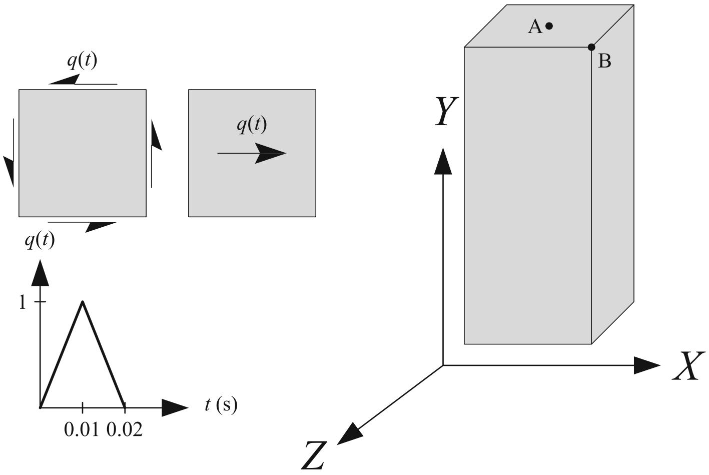

Fig. 9 Problem domain in Example 3

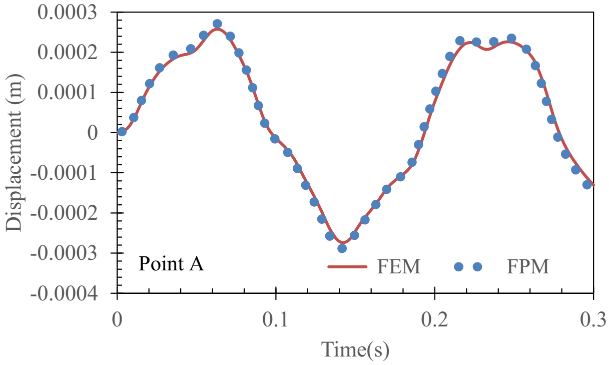

Fig. 10 Displacement for point A in Example 3

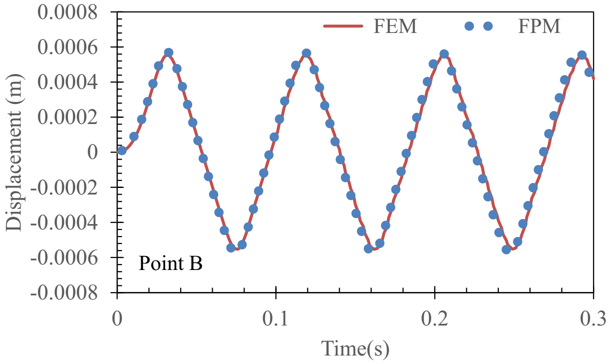

Fig. 11 Displacement for point B in Example 3

of displacement along the Y axis (for both models), at $t = 0.44$ s is reported in Fig. 8. As can be seen, the solution of FPM resembles that of FEM using the same nodal spacing.

Example 3 This example concerns the transient behavior of a prismatic body with a rectangular 2 m × 2 m cross section and with a height of 4 m as shown in Fig. 9. The material properties of the body are $E = 10^5$ N/m^2, $\nu = 0.25$, $\rho = 1$ kg/m^3, and $c = 0$. The body is fixed at its end and subjected to two different impulsive flexural and torsional loading conditions (viz. Fig. 9). For both cases, the surface traction is imposed with an identical triangular time variation illustrated in the same figure, and the solution of the problem for a time duration of 0.3 seconds is sought. In this example, the FPM solution is performed taking $\Delta t = 0.00015$ s using a uniform discretization of the domain with an average nodal spacing of $\Delta x = 0.125\,\text{m}$ which results in 9537 nodes. Again for the sake of verification, the solution of FEM by using linear cubic elements, and the same discretization size which results in 8912 elements, is taken into account. The variation of displacement along the X axis for point A, in the flexural loading case, and point B, in the torsional loading case, is shown in Figs. 10 and 11. Comparing the obtained results to those obtained by FEM, one can conclude that the agreement between the two numerical approaches is very good.

5 Conclusion

In this paper, the meshless finite point method (FPM) is extended and applied to elastodynamic problems. The time integration is performed through an explicit velocity–Verlet approach. A simple technique is introduced to satisfy Neumann type boundary conditions (tractions) in time. This technique preserves the originality and advantages of the FPM in terms of simplicity and efficiency of the approach. The system of equations is formed so that the major parts of the solution domain are governed by uncoupled equations; only for a layer of nodes close to Neumann boundaries the equations become coupled. The detailed formulation for advancing in time is presented, and the accuracy of the approach via several numerical examples, including some 3D problems, is investigated. The obtained results show that the method is capable of yielding proper results with an excellent agreement with those of reference solutions.

Acknowledgements The support of the 'Visiting Scientist' scheme of Padua university is gratefully acknowledged.

References

1. AMGCL C++ library. https://github.com/ddemidov/amgcl. Accessed 11 Feb 2017
2. Bajko, J., Čermák, L., Jícha, M.: High order finite point method for the solution to the sound propagation problems. Comput. Methods Appl. Mech. Eng. **280**, 157–175 (2014)
3. Belytschko, T., Lu, Y.Y., Gu, L.: Element-free Galerkin methods. Int. J. Numer. Methods Eng. **37**(2), 229–256 (1994)
4. Boroomand, B., Najjar, M., Oñate, E.: The generalized finite point method. Comput. Mech. **44**(2), 173–190 (2009)
5. Boroomand, B., Tabatabaei, A.A., Oñate, E.: Simple modifications for stabilization of the finite point method. Int. J. Numer. Methods Eng. **63**(3), 351–379 (2005)
6. Dai, B., Wang, Q., Zhang, W., Wang, L.: The complex variable meshless local Petrov–Galerkin method for elastodynamic problems. Appl. Math. Comput. **243**, 311–321 (2014). doi:10.1016/j.amc.2014.05.123
7. Fang, J., Parriaux, A.: A regularized Lagrangian finite point method for the simulation of incompressible viscous flows. J. Comput. Phys. **227**(20), 8894–8908 (2008)
8. Giunta, G., Belouettar, S., Ferreira, A.J.M.: A static analysis of three-dimensional functionally graded beams by hierarchical modelling and a collocation meshless solution method. Acta Mech. **227**(4), 969–991 (2016). doi:10.1007/s00707-015-1503-3
9. Hsieh, P.W., Shih, Y., Yang, S.Y.: A tailored finite point method for solving steady MHD duct flow problems with boundary layers. Commun. Comput. Phys. **10**(01), 161–182 (2011)
10. La Rocca, A., Power, H.: A double boundary collocation hermitian approach for the solution of steady state convection–diffusion problems. Comput. Math. Appl. **55**(9), 1950–1960 (2008)
11. Liszka, T., Duarte, C., Tworzydlo, W.: hp-Meshless cloud method. Comput. Methods Appl. Mech. Eng. **139**(1–4), 263–288 (1996)
12. Liu, G., Gu, Y.: A meshfree method: meshfree weak–strong (mws) form method, for 2-d solids. Comput. Mech. **33**(1), 2–14 (2003)
13. Liu, G., Wu, Y., Ding, H.: Meshfree weak–strong (mws) form method and its application to incompressible flow problems. Int. J. Numer. Meth. Fluids **46**(10), 1025–1047 (2004)
14. Liu, G.R.: Mesh Free Methods: Moving Beyond the Finite Element Method. Taylor & Francis, London (2010)
15. Mirzaei, D., Hasanpour, K.: Direct meshless local Petrov–Galerkin method for elastodynamic analysis. Acta Mech. **227**(3), 619–632 (2016). doi:10.1007/s00707-015-1494-0
16. Mossaiby, F., Ghaderian, M.: A preliminary study on the meshless local exponential basis functions method for nonlinear and variable coefficient pdes. Eng. Comput. **33**(8), 2238–2263 (2016)
17. Nguyen, V.P., Rabczuk, T., Bordas, S., Duflot, M.: Meshless methods: a review and computer implementation aspects. Math. Comput. Simul. **79**(3), 763–813 (2008)
18. Oñate, E.: Derivation of stabilized equations for numerical solution of advective–diffusive transport and fluid flow problems. Comput. Methods Appl. Mech. Eng. **151**(1–2), 233–265 (1998)
19. Oñate, E., Idelsohn, S., Zienkiewicz, O.C., Taylor, R.L.: A finite point method in computational mechanics. Applications to convective transport and fluid flow. Int. J. Numer. Methods Eng. **39**(22), 3839–3866 (1996)
20. Oñate, E., Perazzo, F., Miquel, J.: A finite point method for elasticity problems. Comput. Struct. **79**(22–25), 2151–2163 (2001)
21. Ortega, E., Oñate, E., Idelsohn, S.: An improved finite point method for tridimensional potential flows. Comput. Mech. **40**(6), 949–963 (2007)

22. Sadeghirad, A., Kani, I.M.: Modified equilibrium on line method for imposition of neumann boundary conditions in meshless collocation methods. Int. J. Numer. Methods Biomed. Eng. **25**(2), 147–171 (2009)
23. Sadeghirad, A., Mohammadi, S.: Equilibrium on line method (ELM) for imposition of Neumann boundary conditions in the finite point method (FPM). Int. J. Numer. Methods Eng. **69**(1), 60–86 (2007)
24. Shojaei, A., Boroomand, B., Mossaiby, F.: A simple meshless method for challenging engineering problems. Eng. Comput. **32**(6), 1567–1600 (2015)
25. Shojaei, A., Boroomand, B., Soleimanifar, E.: A meshless method for unbounded acoustic problems. J. Acoust. Soc. Am. **139**(5), 2613–2623 (2016)
26. Shojaei, A., Mudric, T., Zaccariotto, M., Galvanetto, U.: A coupled meshless finite point/peridynamic method for 2d dynamic fracture analysis. Int. J. Mech. Sci. **119**, 419–431 (2016)
27. Shu, C., Ding, H., Yeo, K.: Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier–Stokes equations. Comput. Methods Appl. Mech. Eng. **192**(7), 941–954 (2003)
28. Tatari, M., Kamranian, M., Dehghan, M.: The finite point method for the p-Laplace equation. Comput. Mech. **48**(6), 689–697 (2011)
29. Weaver Jr., W., Timoshenko, S.P., Young, D.H.: Vibration Problems in Engineering. Wiley, New York (1990)
30. Wu, N.J., Chen, B.S., Tsay, T.K.: A review on the modified finite point method. Math. Probl. Eng. **2014**, 1–29 (2014)
31. Zhang, X., Song, K.Z., Lu, M.W., Liu, X.: Meshless methods based on collocation with radial basis functions. Comput. Mech. **26**(4), 333–343 (2000)