Acta Mech 227, 619–632 (2016)
DOI 10.1007/s00707-015-1494-0

ORIGINAL PAPER

Davoud Mirzaei · Kourosh Hasanpour

Direct meshless local Petrov–Galerkin method for elastodynamic analysis

Received: 2 September 2015 / Published online: 22 October 2015

Abstract This article describes a new and fast meshfree method based on a generalized moving least squares (GMLS) approximation and the local weak forms for vibration analysis in solids. In contrast to the meshless local Petrov–Galerkin method, GMLS directly approximates the local weak forms from meshless nodal values, which shifts the local integrations over the low-degree polynomial basis functions rather than over the complicated MLS shape functions. Besides, if the method is set up properly, all local integrals have the same value if all local subdomains have the same shape. These features reduce the computational costs, remarkably. The new technique is called *direct meshless local Petrov–Galerkin* (*DMLPG*) method. In DMLPG, the stiff and mass matrices are constructed by integration against polynomials. This overcomes the main drawback of meshfree methods in comparison with the finite element methods (FEM). The Newmark scheme is adapted as a time integration method, and numerical results are presented for various dynamic problems. The results are compared with the exact solutions, if available, and the FEM solutions.

1 Introduction

Although meshless methods promise to overcome some disadvantages of the traditional finite element methods (FEM), they still have some limitations that sometimes prevent their acceptance among researchers and engineers. High computational cost seems to be the most significant disadvantage of meshfree methods in comparison with FEM. This becomes more serious in weak-based methods where numerical integrations against meshfree shape functions should be performed. Moreover, as pointed out in [1,4] by numerical examples and then in [2,33] by theoretical justifications, numerical integration plays an important role in the convergence of numerical solutions of meshless methods. To obtain the full rates of convergence in meshless methods, the energy and mass in the weak formulation need to be evaluated accurately. The numerical integration in the usual finite element method is not a serious problem, because the integrands are simple polynomials and the stiffness and mass matrices may be exactly integrated by using a quadrature formula with few integration points. However, the integration is much difficult in meshfree methods due to complexity of the integrand. In some meshfree methods, we face with non-close form shape functions which often have complex features and different forms in each small subregions. Besides, the derivatives of shape functions have an oscillation or an indentation, a peak and a discontinuity in a local sense. These features pose a serious challenge in the use of numerical integration to compute the elements of the stiffness and mass matrices. The moving least

D. Mirzaei (✉)
Department of Mathematics, University of Isfahan, P.O.Box 81746-73441, Isfahan, Iran
E-mail: d.mirzaei@sci.ui.ac.ir
Tel.: +98-31-37934642

K. Hasanpour
Department of Mechanical Engineering, Faculty of Engineering, University of Isfahan, P.O.Box 81746-73441, Isfahan, Iran
E-mail: hasanpour@eng.ui.ac.ir

square (MLS) shape functions can be highlighted as an example, and therefore, all meshfree methods based on the MLS approximation suffer from this drawback. Thus, special care should be taken for performing numerical quadratures in meshfree methods. These challenges have been addressed in various engineering papers [1,3,5,8,16,26]. Several approaches to implementing numerical integration have been proposed in the literature. A brief review of these approaches is presented in Sect. 3 of [2].

Some weak-based meshfree methods are called “truly” meshless, because triangulation is not again required for numerical integration. In fact they are meshfree in both trial and test sides. The meshless local Petrov–Galerkin (MLPG) method can be addressed as an example of such methods. MLPG is based on *local weak forms*, and it uses no global background mesh to evaluate integrals, and everything breaks down to some regular, well-shaped and independent subdomains. *But* due to the above discussions, MLPG still suffers from the cost of numerical integration.

This disadvantage might be overcome by a simple and useful modification which has been proposed in [22]. This modification uses the concept of generalized moving least squares (GMLS) approximation [24] and shifts the numerical integrations over low-degree polynomial basis functions. Since we have a *direct* approximation of local weak functionals, the new technique is called direct MLPG (DMLPG). Ignoring the costs of mesh generation and mesh refinement, we can roughly say DMLPG reduces the computational cost of MLPG to the level of that in classical FEM, because both FEM and DMLPG lead to sparse final linear systems, and in both methods, numerical integrations are performed over low-degree polynomial basis functions. As we mentioned before, the latter is not the case in MLPG itself.

Theoretically, DMLPG is different from MLPG because it rules out the trial space completely. In fact, in DMLPG the derivative or integral operators are directly approximated without a detour via the shape functions.

The DMLPG has been applied to the heat conduction analysis in [23] and elastostatic problems in [20,31], and it has been numerically investigated for 2D and 3D potential problems in [17,18]. An application to fractional advection–diffusion problems can be found in [27].

The MLPG method and its variations have been applied for elastodynamic problems by several authors. For example see [7,9,10,25,28–30]. On the other hand, such problems have been well studied by variations of finite element and boundary element methods. See for example [12,13] and the references therein. The aim of this paper is just to introduce the new DMLPG method which overcomes one of the major disadvantages of the previous meshfree techniques. The numerical test problems of this paper might be solved by available FEM or BEM routines; however, we hope that the result of this paper makes an essential contribution in its own direction and helps the meshless community to solve more complicated elasticity problems.

2 Test discretization via local weak forms for elastodynamic equation

Let $\Omega \subset \mathbb{R}^2$ be a bounded domain with boundary Γ. We consider the following two-dimensional elastodynamic problem:

$$\sigma_{ij,j} + b_i = c\dot{u}_i + \rho\ddot{u}_i, \quad \text{in } \Omega, \tag{1}$$

where $[\sigma_{11}, \sigma_{22}, \sigma_{12}]^T =: \boldsymbol{\sigma}$ is the stress tensor, which corresponds to the displacement field $[u_1, u_2]^T =: \boldsymbol{u}$, and $[b_1, b_2]^T =: \boldsymbol{b}$ is the body force, c is the damping coefficient, ρ is the mass density, $\dot{\boldsymbol{u}} = \partial \boldsymbol{u}/\partial t$ is the velocity, and $\ddot{\boldsymbol{u}} = \partial^2\boldsymbol{u}/\partial t^2$ is the acceleration in which t denotes the time variable. In the above formulation, $\boldsymbol{\sigma} = DL\boldsymbol{u}$, where the derivative matrix L is defined as

$$L = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}$$

and for a problem of isotropic material, the stress–strain matrix D is defined by

$$D = \frac{\overline{E}}{1-\overline{\nu}^2}\begin{bmatrix} 1 & \overline{\nu} & 0 \\ \overline{\nu} & 1 & 0 \\ 0 & 0 & (1-\overline{\nu})/2 \end{bmatrix},$$

where

$$\overline{E} = \begin{cases} E & \text{for plane stress} \\ \frac{E}{1-\nu^2} & \text{for plane strain} \end{cases} \qquad \overline{\nu} = \begin{cases} \nu & \text{for plane stress} \\ \frac{\nu}{1-\nu} & \text{for plane strain} \end{cases},$$

in which E and ν are Young's modulus and Poisson's ratio, respectively. The corresponding boundary and initial conditions for (1) are given by

$$\begin{aligned} \boldsymbol{u} &= \overline{\boldsymbol{u}}, \quad \text{on } \Gamma_u, \\ t_i &= \sigma_{ij} n_j = \overline{t}_i, \quad \text{on } \Gamma_t, \\ \boldsymbol{u}(x,0) &= \boldsymbol{u}_0(x), \quad x \in \Omega, \\ \dot{\boldsymbol{u}}(x,0) &= \dot{\boldsymbol{u}}_0(x), \quad x \in \Omega, \end{aligned}$$

where $\overline{\boldsymbol{u}}$, $[\overline{t}_1, \overline{t}_2]^{\mathrm{T}} =: \overline{\boldsymbol{t}}$, $\boldsymbol{u}_0$ and $\dot{\boldsymbol{u}}_0$ are the prescribed displacement, traction, initial displacement and initial velocity, respectively, and $[n_1, n_2] =: \boldsymbol{n}$ is the unit outward normal to the boundary Γ.

Suppose that $X = \{x_1, x_2, \ldots, x_N\} \subset \overline{\Omega}$ is a set of scattered meshless points. The same set X will be used for both trial and test point sets in our numerical simulation.

Instead of a global weak form over entire Ω, we use local weak forms over small subdomains Ω_k around the test points $x_k \in \operatorname{int}(\Omega) \cup \Gamma_t$. Taking integration with respect to the spatial variable from both sides of Eq. (1) against some proper test functions v_i, $i = 1, 2$ (usually $v_1 = v_2 =: v$), and then applying the Gauss divergence theorem, the local weak forms

$$\begin{aligned} \int_{\partial\Omega_k \setminus \Gamma_t} v \mathcal{N} D L \boldsymbol{u} \,\mathrm{d}\Gamma - \int_{\Omega_k} \boldsymbol{\varepsilon}_v D L \boldsymbol{u} \,\mathrm{d}\Omega + \frac{\partial^2}{\partial t^2} \int_{\Omega_k} \rho v \boldsymbol{u} \,\mathrm{d}\Omega + \frac{\partial}{\partial t} \int_{\Omega_k} c\, v \boldsymbol{u} \,\mathrm{d}\Omega \\ = \int_{\Omega_k} v \boldsymbol{b} \,\mathrm{d}\Omega + \int_{\partial\Omega_k \cap \Gamma_t} v \overline{\boldsymbol{t}} \,\mathrm{d}\Gamma \end{aligned} \tag{2}$$

are derived for $k = 1, 2, \ldots, N'$ (N' is the number of points inside Ω and on Neumann part of the boundary), where

$$\mathcal{N} = \begin{bmatrix} n_1 & 0 & n_2 \\ 0 & n_2 & n_1 \end{bmatrix}, \quad \boldsymbol{\varepsilon}_v = \begin{bmatrix} v_{,1} & 0 & v_{,2} \\ 0 & v_{,2} & v_{,1} \end{bmatrix}.$$

Note that in (2) the natural boundary conditions $\sigma_{ij} n_j = \overline{t}_i$ have been imposed on $\partial\Omega_k \cap \Gamma_t$. The essential boundary condition $u_i = \overline{u}_i$ can be imposed using a proper collocation method.

Since we are going to perform a Petrov–Galerkin method, test function v can be chosen from an arbitrary space independent of the trial space.

Now we define

$$\begin{aligned} \lambda_k^S(\boldsymbol{u}) &:= -\int_{\Omega_k} \boldsymbol{\varepsilon}_v D L \boldsymbol{u} \,\mathrm{d}\Omega + \int_{\partial\Omega_k \setminus \Gamma_t} v \mathcal{N} D L \boldsymbol{u} \,\mathrm{d}\Gamma, \\ \lambda_k^M(\boldsymbol{u}) &:= \int_{\Omega_k} \rho v \boldsymbol{u} \,\mathrm{d}\Omega, \\ \lambda_k^C(\boldsymbol{u}) &:= \int_{\Omega_k} c v \boldsymbol{u} \,\mathrm{d}\Omega, \\ \boldsymbol{\beta}_k &:= \int_{\Omega_k} v \boldsymbol{b} \,\mathrm{d}\Omega + \int_{\partial\Omega_k \cap \Gamma_t} v \overline{\boldsymbol{t}} \,\mathrm{d}\Gamma, \end{aligned}$$

where the superscripts S, M and C stand for stiff, mass and damping matrices, respectively. Now Eq. (2) can be read as

$$\lambda_k^S(\boldsymbol{u}) + \frac{\partial^2}{\partial t^2} \lambda_k^M(\boldsymbol{u}) + \frac{\partial}{\partial t} \lambda_k^C(\boldsymbol{u}) = \boldsymbol{\beta}_k, \quad k = 1, 2, \ldots, N'. \tag{3}$$

The above semi-discrete equation will be addressed in Sect. 4 where we will present the formulation of DMLPG method. In the next section, we will discuss an approximation technique which finally converts (3) to a full-discrete linear system of equations.

3 Generalized moving least squares

We refer the readers to [24] for a complete presentation of the generalized moving least squares (GMLS) approximation and to [21] for error analysis in Sobolev spaces. However, here we outline a brief discussion.

Let Ω be a bounded subset of $\mathbb{R}^d$, $d \in \mathbb{Z}_+$, and $X = \{x_1, x_2, \dots, x_N\} \subset \Omega$ be a set of meshless points scattered over Ω. The MLS method approximates the function $u \in U$ (with certain smoothness) by its values at points x_j, $j = 1, \dots, N$. The MLS is a meshless (meshfree) approximation method because it writes the approximant entirely in terms of scattered points. The formulation can be written as

$$u(x) \approx \widehat{u}(x) = \sum_{j=1}^{N} a_j(x) u(x_j), \quad x \in \Omega, \tag{4}$$

where $a_j(x)$ are MLS shape functions obtained in such a way that $\widehat{u}$ be the best approximation of u in polynomial subspace $\mathbb{P}_m(\mathbb{R}^d) = \operatorname{span}\{p_1, \dots, p_Q\}$, $Q = \binom{m+d}{d}$, with respect to a weighted, discrete and *moving* ℓ^2 norm. In fact, for every evaluation point x, a local least square problem with a different inner product should be solved. The role of the weight function makes such approximation possible. The weight function governs the influence of the data points and is assumed to be a compactly supported function $w : \Omega \times \Omega \to \mathbb{R}$ which vanishes for arguments $x, y \in \Omega$ with $\|x - y\|_2$ greater than a certain threshold, say δ. For example we can define $w(x, y) := \varphi(\|x - y\|_2/\delta)$ where φ is a compactly supported function supported in $[0, 1]$. On the other hand, if we define

$$\begin{aligned} P &:= \big(p_k(x_j)\big) \in \mathbb{R}^{N \times Q}, \\ W &= W(x) := \operatorname{diag}\{w(x_j, x)\} \in \mathbb{R}^{N \times N}, \end{aligned} \tag{5}$$

then a simple calculation gives the *shape functions*

$$\boldsymbol{a}(x) := [a_1(x), \dots, a_N(x)] = \boldsymbol{p}(x)(P^{\mathrm{T}} W P)^{-1} P^{\mathrm{T}} W, \tag{6}$$

where $\boldsymbol{p} = [p_1, \dots, p_Q]$. For every $x \in \Omega$, if $\{x_j : \|x - x_j\| \leqslant \delta\}$ is $\mathbb{P}_m(\mathbb{R}^d)$-unisolvent, then $A(x) = P^{\mathrm{T}} W(x) P$ is positive definite [32] and the MLS approximation is well-defined.

Derivatives of u are usually approximated by derivatives of $\widehat{u}$,

$$D^\alpha u(x) \approx D^\alpha \widehat{u}(x) = \sum_{j=1}^{N} D^\alpha a_j(x) u(x_j), \quad x \in \Omega, \quad \alpha = (\alpha_1, \dots, \alpha_d) \in \mathbb{N}_0^d. \tag{7}$$

These derivatives are called *standard* or *full* derivatives. Since in this paper we do not consider the standard MLS derivatives further, we refer the readers to [4,14,19] for details of formulations and error bounds for derivative approximations.

In a more general situation, suppose that λ is a linear functional from the dual space U^*. As examples, λ can describe the stiff, mass or damping functionals defined in (3). The value of λu is approximated by $\lambda \widehat{u}$,

$$(\lambda u)(x) \approx (\lambda \widehat{u})(x) = \sum_{j=1}^{N} \lambda a_j(x) u(x_j), \quad x \in \Omega.$$

As we can see, λ should operate on MLS shape functions a_j. The non-close form shape functions a_j should be constructed during the MLS algorithm. Thus, the computation of $\lambda \widehat{u}$ is a time-consuming task, specially when λ has a complex structure.

The GMLS approximation was introduced in [24] to overcome the above drawback. In GMLS, λu is *directly* approximated from nodal values $u(x_j)$, $j = 1, \dots, N$, say

$$(\lambda u)(x) \approx (\widehat{\lambda u})(x) = \sum_{j=1}^{N} a_{j,\lambda}(x) u(x_j), \tag{8}$$

where $a_{j,\lambda}$ are shape functions associated with the functional λ. If λ is chosen to be the *point evaluation* functional δ_x, where $\delta_x(u) := u(x)$, then the classical MLS approximation (4) is resulted. Since λ is finally evaluated at sample point x, the same weight function w as in the classical MLS can be used independent of the choice of λ. Using this assumption, analogous to (6), [24] proves,

$$\boldsymbol{a}_\lambda := [a_{1,\lambda}, \ldots, a_{N,\lambda}] = \lambda(\boldsymbol{p})(P^{\mathrm{T}}WP)^{-1}P^{\mathrm{T}}W, \tag{9}$$

where $\lambda(\boldsymbol{p}) = [\lambda(p_1), \ldots, \lambda(p_Q)]$. As we can immediately see, λ acts only on polynomial basis functions $p_1, \ldots, p_Q$. This is the central idea in this GMLS approximation which finally speeds up our numerical algorithms.

In particular, if $\lambda(u) = D^\alpha(u)$, then derivatives of u are recovered. They are different from the standard derivatives (7), and in meshless literatures, they are sometimes called *diffuse* or *uncertain* derivatives. But [24] and [21] prove the optimal rate of convergence for them toward the exact derivatives, and thus, there is nothing diffuse or uncertain about them. As suggested in [24], they can be called *GMLS derivative* approximations.

Finally, as a comment for numerical implementation, a shifted and scaled polynomial basis leads to a more stable MLS/GMLS approximation. As we discussed, the MLS/GMLS solves a local ℓ_2 best approximation problem for every selected evaluation point $\widehat{x} \in \Omega$. The polynomial basis for this local problem can be defined as

$$\left\{ \frac{(x-\widehat{x})^\alpha}{h_{X,\Omega}^{|\alpha|}} \right\}_{0\leqslant|\alpha|\leqslant m}, \tag{10}$$

where $h_{X,\Omega}$ is the mesh size (fill distance) of point set X in Ω, and $\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}_0^d$ is a multi-index, and $|\alpha| = \alpha_1 + \cdots + \alpha_d$, here $d = 2$. In fact we change the basis as the evaluation point is changed. We can do this because the formulation of MLS/GMLS approximation is independent of the choice of the basis functions. The effect of this type of basis on the stability and convergence of (G)MLS algorithm is studied in [19,24].

4 Going from MLPG to DMLPG

Equation (3) is a semi-discrete formulation where the discretization is only done on the *test* side. Different choices for test function lead to variations of MLPG/DMLPG. Couple of them are much interesting in numerical simulation. In DMLPG1 the first and in DMLPG5 the second integral in the stiff functional λ_k^S is eliminated by applying a compactly supported and constant test functions in Ω_k, respectively. The formulations of the other DMLPG methods can be found in [22,23].

Up to here, MLPG and DMLPG have the same structures. In MLPG, to apply the *trial* discretization, first the displacement vector $\boldsymbol{u}$ is approximated by MLS (or any trial approximation method) in the spatial domain by

$$\boldsymbol{u}(x,t) \approx \widehat{\boldsymbol{u}}(x,t) = \sum_{j=1}^N \begin{bmatrix} a_j(x) & 0 \\ 0 & a_j(x) \end{bmatrix} \boldsymbol{u}(x_j,t),$$

where $a_j(x)$ are corresponding shape functions. Then functionals $\lambda_k(\boldsymbol{u})$ are approximated by $\lambda_k(\widehat{\boldsymbol{u}})$. Thus, λ_k should operate on a_j, leading to numerical derivatives and then numerical integrations against shape functions. This part is the most important stage of the algorithm where both accuracy and complexity (computational cost) of MLPG are relied thereon. To evaluate the elements of stiffness and mass matrices accurately, a quadrature formula with many integration points is required. Unfortunately, this leads to an expensive algorithm due to the complexity of shape functions a_j and their derivatives. To scape from this stage, fortunately there is a bypass via GMLS approximation leading to DMLPG method.

In DMLPG, the functionals $\lambda_k(\boldsymbol{u})$ are directly approximated by GMLS as

$$\lambda_k(\boldsymbol{u}) \approx \widehat{\lambda_k}(\boldsymbol{u}) = \sum_{j=1}^N A_{kj}\boldsymbol{u}(x_j,t),$$

where A_{kj} is a 2×2 matrix depending on functional λ_k and trial point x_j via indices k and j, respectively. Let A is a $2N'\times 2N$ matrix with block elements A_{kj} for $k=1,\ldots,N'$ and $j=1,2,\ldots,N$. According to (9), if $A_{k,:}$ represents the kth block row of A, then

$$A_{k,:}=\lambda_k(\boldsymbol{p})\Phi\in\mathbb{R}^{2\times 2N},$$

where $\Phi\in\mathbb{R}^{2Q\times 2N}$ is a block matrix obtained from $\phi:=(P^{\mathrm{T}}WP)^{-1}WP^{\mathrm{T}}\in\mathbb{R}^{Q\times N}$ by

$$\Phi_{ij}=\begin{bmatrix}\phi_{ij} & 0\\ 0 & \phi_{ij}\end{bmatrix}\in\mathbb{R}^{2\times 2}.$$

The matrices P and W are defined in (5), and here $\boldsymbol{p}$ is defined by

$$\boldsymbol{p}=\begin{bmatrix}p_1(x) & p_2(x) & \cdots & p_Q(x)\\ p_1(x) & p_2(x) & \cdots & p_Q(x)\end{bmatrix}\in\mathbb{R}^{2\times Q}.$$

Thus, $\lambda_k(\boldsymbol{p})\in\mathbb{R}^{2\times 2Q}$. Rows of $\boldsymbol{p}$ are the same because we use the same trial space for both u_1 and u_2. To distinguish between the notations, for $\lambda_k=\lambda_k^S,\lambda_k^M,\lambda_k^C$, matrices S, M, C are used instead of A, respectively.

One can immediately see that the functionals λ_k act only on polynomial basis functions $p_1,p_2,\ldots,p_Q$, not on a trial space spanned by shape functions. Besides, if the shifted polynomial basis functions (10) are employed and if the same test function v is used for all local subdomains, then $\lambda_k(\boldsymbol{p})=\lambda_j(\boldsymbol{p})$ provided that $\Omega_k=x+\Omega_j=\{x+y: y\in\Omega_j\}$, $x\in\mathbb{R}^d$. As an example, for all interior test points only one integral should be computed if all interior local subdomains have the same shape. Consequently, the cost of DMLPG is in order of the cost of an MLS collocation method. These fundamental features highly accelerate the algorithm and make DMLPG absolutely faster than the original MLPG.

It remains to discuss the way of enforcing the essential boundary condition $\boldsymbol{u}=\overline{\boldsymbol{u}}$ on Γ_u. It is done for any point $x_k\in\Gamma_k$ by choosing $\lambda_k(\boldsymbol{u})=\delta_{x_k}(\boldsymbol{u})$ (the point evaluation functional) in GMLS approximation. This is identical to the classical MLS approximation and leads to a collocation method by the equations

$$\sum_{j=1}^{N}B_{kj}\boldsymbol{u}(x_j,t)=\overline{\boldsymbol{u}}(x_k,t),\quad x_k\in\Gamma_u,\tag{11}$$

where

$$B_{kj}=\begin{bmatrix}a_j(x_k) & 0\\ 0 & a_j(x_k)\end{bmatrix},\quad x_k\in\Gamma_u,\ 1\leqslant j\leqslant N.$$

If we define

$$U(t):=[\boldsymbol{u}(x_1,t),\boldsymbol{u}(x_2,t),\ldots,\boldsymbol{u}(x_N,t)]^{\mathrm{T}}\in\mathbb{R}^{2N},$$

then (11) can be written in matrix form as $BU(t)=\overline{U}(t)$, where B is a $2(N-N')\times 2N$ matrix.

If (without loss of generality) we assume that the last $N-N'$ points are located on Γ_u, then the final time-dependent linear system of equations can be written as

$$\begin{aligned}M\ddot{U}(t)+C\dot{U}(t)+SU(t)&=F(t),\\ BU(t)&=\overline{U}(t),\end{aligned}\tag{12}$$

where $F(t)=[\boldsymbol{\beta}_1(t),\boldsymbol{\beta}_2(t),\ldots,\boldsymbol{\beta}_{N'}(t)]^{\mathrm{T}}$. Equation (12) is a $2N\times 2N$ system of wave equations which can be abbreviated as

$$\widetilde{M}\ddot{U}(t)+\widetilde{C}\dot{U}(t)+\widetilde{S}U(t)=\widetilde{F}(t),\tag{13}$$

where

$$\widetilde{M}=\begin{bmatrix}M\\ \mathbf{0}\end{bmatrix},\quad \widetilde{C}=\begin{bmatrix}C\\ \mathbf{0}\end{bmatrix},\quad \widetilde{S}=\begin{bmatrix}S\\ B\end{bmatrix},\quad \widetilde{F}=\begin{bmatrix}F\\ \overline{U}\end{bmatrix},$$

in which **0** is zero matrix of size B. Equation (13) should be accompanied with the initial conditions

$$U(0) = U_0, \quad \dot{U}(0) = \dot{U}_0. \tag{14}$$

In the next section we describe a time integration procedure for solving the equilibrium Eq. (13) together with the initial conditions (14).

5 The time integration scheme

Various finite difference schemes can be developed for solving the second-order dynamic Eqs. (13), (14). Examples are the explicit central finite difference method and Newmark time integration methods [6,11]. The former is conditionally stable, and the size of the time step should be adjusted to the spatial mesh size. Moreover, in some longtime examples, the numerical solutions obtained by the central difference scheme start damping even when the damping parameter is zero. The reason is that the numerical wave amplitude number of this scheme is less than unity, while the exact amplitude number is 1. Thus, here we apply the Newmark method which is easy to implement and sufficiently versatile to address a broad class of time-dependent problems.

Suppose that the time interval $[0, t_f]$ is uniformly partitioned to T subintervals where $\Delta t = t_f/T$ and $t_n = n\Delta t$ for $n = 0, 1, \ldots, T$. Let $U_n = U(t_n)$, $V_n = \dot{U}(t_n)$ and $A_n = \ddot{U}(t_n)$ are displacement, velocity and acceleration vectors in time instance t_n, respectively. By exploiting Taylor expansion theorem, they can be expressed with respect to their first time derivatives as follows:

$$U_{n+1} = U_n + \Delta t V_n + \frac{\Delta t}{2} A_{n+2\beta}, \quad \text{where} \quad A_{n+2\beta} = (1-2\beta)A_n + 2\beta A_{n+1},$$
$$V_{n+1} = V_n + \Delta t A_{n+\alpha}, \quad \text{where} \quad A_{n+\alpha} = (1-\alpha)A_n + \alpha A_{n+1},$$

where α and β are user-defined constants. Now we can adapt the following Newmark time integration algorithm where quantities U^p_{n+1} and V^p_{n+1} are the predictor terms depending on the previous time step n [11].

Algorithm 1 The Newmark time integration algorithm

```
1: Inputs: $\alpha, \beta, U_0, V_0, \widetilde{M}, \widetilde{C}, \widetilde{S}, \widetilde{F}_n (n = 0, \ldots, T), \Delta t$
2: $K \leftarrow \widetilde{M} + \alpha \Delta t \widetilde{C} + \beta(\Delta t)^2 \widetilde{S}$
3: $b_0 \leftarrow \widetilde{F}_0 - \widetilde{C} V_0 - \widetilde{S} U_0$
4: $A_0 \leftarrow \text{Linsolve}(K, b_0)$
5: for $n = 0$ to $T - 1$ do
6:     $U^p_{n+1} \leftarrow U_n + \Delta t V_n + \frac{(\Delta t)^2}{2}(1-2\beta)A_n$
7:     $V^p_{n+1} \leftarrow V_n + \Delta t(1-\alpha)A_n$
8:     $b_{n+1} \leftarrow \widetilde{F}_{n+1} - \widetilde{C} V^p_{n+1} - \widetilde{S} U^p_{n+1}$
9:     $A_{n+1} \leftarrow \text{Linsolve}(K, b_{n+1})$
10:    $U_{n+1} \leftarrow U^p_{n+1} + (\Delta t)^2 \beta A_{n+1}$
11:    $V_{n+1} \leftarrow V^p_{n+1} + \alpha \Delta t A_{n+1}$
12: end for
13: Outputs: $U_n,\ n = 1, 2, \ldots, T$
```

Of course we should perform an LU decomposition for matrix K in the beginning of calculations and use forward and backward substitutions in our "Linsolve" subroutine for all iterations.

The properties of Newmark scheme depend on parameters α and β. It can be either explicit or implicit and conditionally or unconditionally stable. The most common choices are listed in Table 1. For more details, see [6].

Table 1 Some types of Newmark scheme

Method	α	β	Stability
Central difference	1/2	0	Explicit and conditionally stable
Fox-Goodwin	1/2	1/2	Implicit and conditionally stable
Linear acceleration	1/2	1/6	Implicit and conditionally stable
Average acceleration	1/2	1/4	Implicit and unconditionally stable

Note that for $\alpha = 1/2$, the Newmark method is second-order accurate and this is the reason why this value is often assigned to α. In this paper, we use the implicit and unconditionally stable average acceleration scheme where $\alpha = 1/2$ and $\beta = 1/4$.

6 Numerical results

The following compactly supported Gaussian weight function is used:

$$w(x, y) = \varphi(r) = \frac{\exp(-(\epsilon r)^2) - \exp(-\epsilon^2)}{1 - \exp(-\epsilon^2)}, \quad 0 \leqslant r = \frac{\|x - y\|_2}{\delta} \leqslant 1,$$

where the shape parameter ϵ is taken to be 6 in this paper. Here δ is the radius of circular support of weight function w at point x in question. This radius should be large enough to ensure the regularity of the moment matrix $P^{\mathrm{T}} W P$ in MLS/GMLS approximation. Thus δ is chosen to be proportional to h (mesh size) and m, say $\delta = cmh$. In this study we use $\delta = 1.2$ mh. The polynomial degree $m = 2$ and both circular and square subdomains are used for Ω_k. As we pointed before, in DMLPG5 the simple test function $v = 1$ is used on Ω_k. In DMLPG1 for circular subdomians, the above Gaussian weight function with δ being replaced by the radius r_0 of the local domain Ω_k is used as a test function, while for square subdomains the test function

$$v = v(x; x_k) = \begin{cases} \prod_{i=1}^{2} \left(1 - \frac{4}{s_0^2}(x^i - x_k^i)^2\right), & x \in S(x_k, s_0) = \Omega_k, \\ 0, & \text{otherwise} \end{cases}$$

is applied where $x = (x^1, x^2)$ and $x_k = (x_k^1, x_k^2)$. Here s_0 is the side length of the square. It is clear that in both cases v vanishes on $\partial\Omega_k$. As pointed in [22], a 2-point Gauss quadrature in each axis is enough to get the exact numerical integrations in DMLPG, provided that squares are used as local subdomains and the above polynomial test function is applied. In the other cases, we use a 10-point Gaussian formula. By the way, as pointed out in Sect. 4, numerical integration is not a crucial task in DMLPG because only few integrals against low-degree polynomials must be evaluated.

In the time domain, $\Delta t = 0.005$ is assigned for all examples. A plane stress problem is considered in all numerical simulations.

DMLPG routines are written using MATLAB© and run on a Pentium 4 PC with 8 GB of Memory and a 7-core 2.40 GHz CPU.

6.1 A rectangular plane under a uniaxial tension

Consider a rectangular plane fixed rigidly at its base and subjected to an impact load at the free end, as in Fig. 1. Parameters are length $L = 8$ m, height $D = 2$ m, Young's modulus $E = 8 \times 10^4$ Pa, Poisson's ratio $\nu = 0$, mass density $\rho = 2450$ kg/m^3, damping parameter $c = 0$, and face force $P = 200$ Pa. The exact solutions are given by

$$u_1(x, t) = \frac{8PL}{\pi^2 E} \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{(2n-1)^2} \sin \frac{(2n-1)\pi x}{2L} (1 - \cos \omega_n t),$$

$$\sigma(x, t) = E \frac{\partial u_1}{\partial x} = \frac{4P}{\pi} \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{2n-1} \cos \frac{(2n-1)\pi x}{2L} (1 - \cos \omega_n t),$$

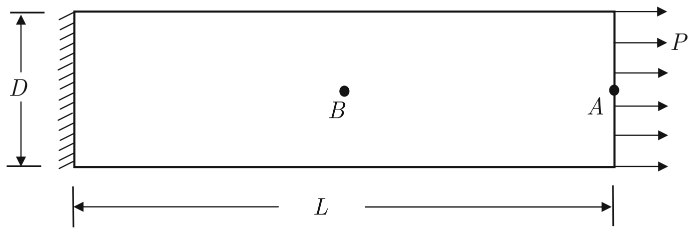


Fig. 1 Rectangular plane subjected to uniform load

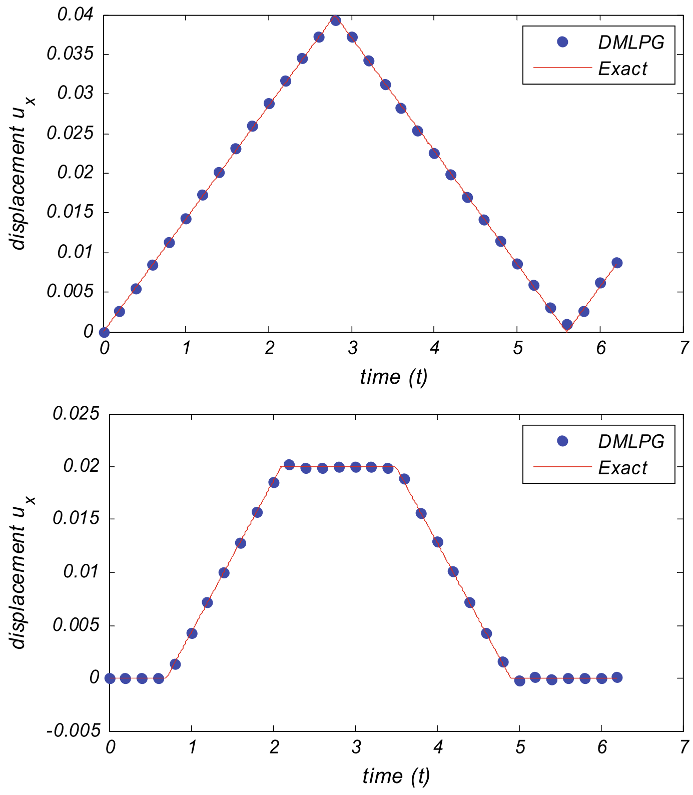


Fig. 2 Time variations of displacement u_1 at point A (*up*) and point B (*down*)

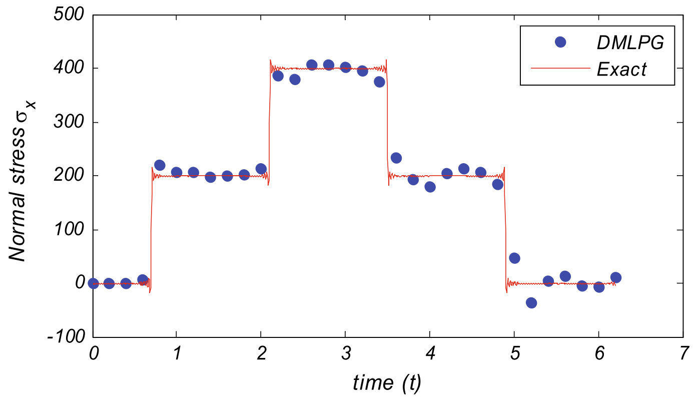


Fig. 3 Time variation of normal stress (axial force) at point B

where $\omega_n = \frac{(2n-1)\pi}{2L}\sqrt{\frac{E}{\rho}}$. DMLPG1 with square subdomains and 33×9 regular points are used for numerical simulation. In Fig. 2, we compare DMLPG and exact solutions of horizontal displacement at points A and B, and in Fig. 3, the same is done for normal stress at point B. MLPG produces approximately the same results. But in Fig. 4, the computational costs of both methods for constructing the final stiff and mass matrices are compared. As we can see, DMLPG is absolutely faster than MLPG. The reason was well discussed in previous sections.

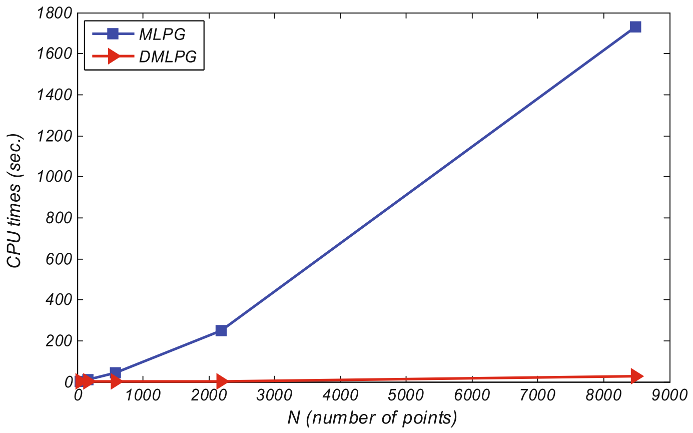


Fig. 4 Comparing the computational costs of MLPG and DMLPG

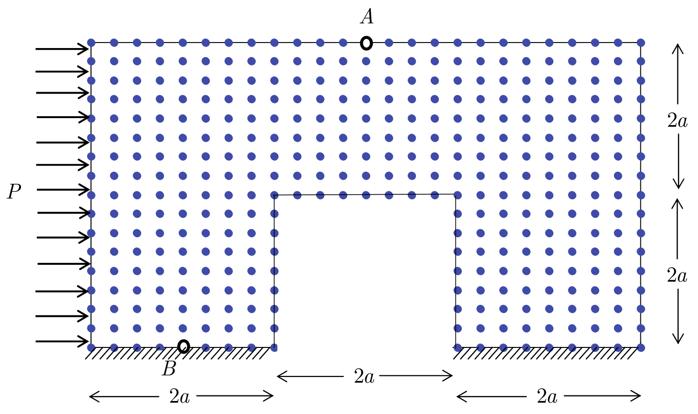


Fig. 5 Frame structure loaded by wind-like loading, and meshless points

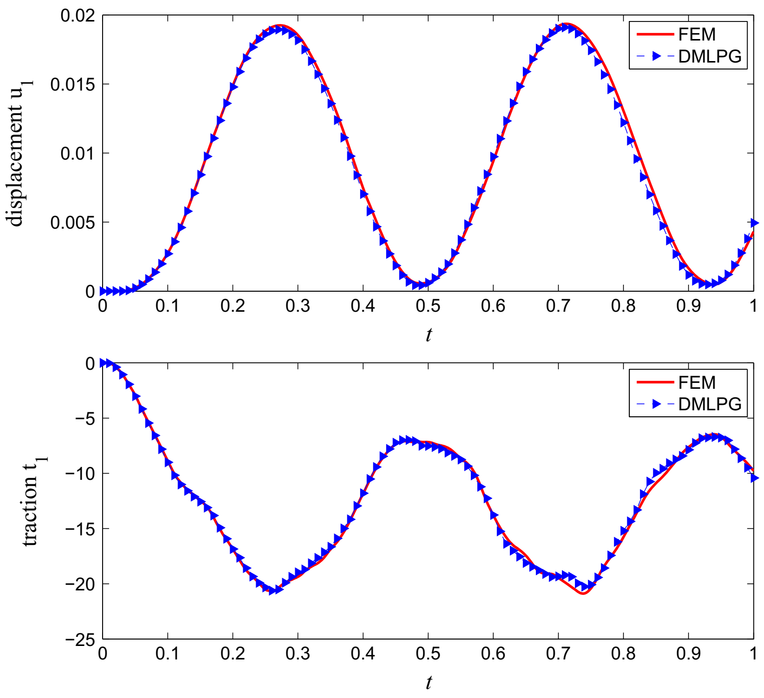


Fig. 6 Comparing FEM and DMLPG solutions; time variations of displacement u_1 at point A (*top*) and traction t_1 at point B (*bottom*)

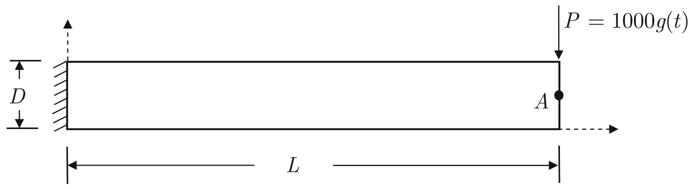


Fig. 7 A cantilever beam subjected to a parabolic traction

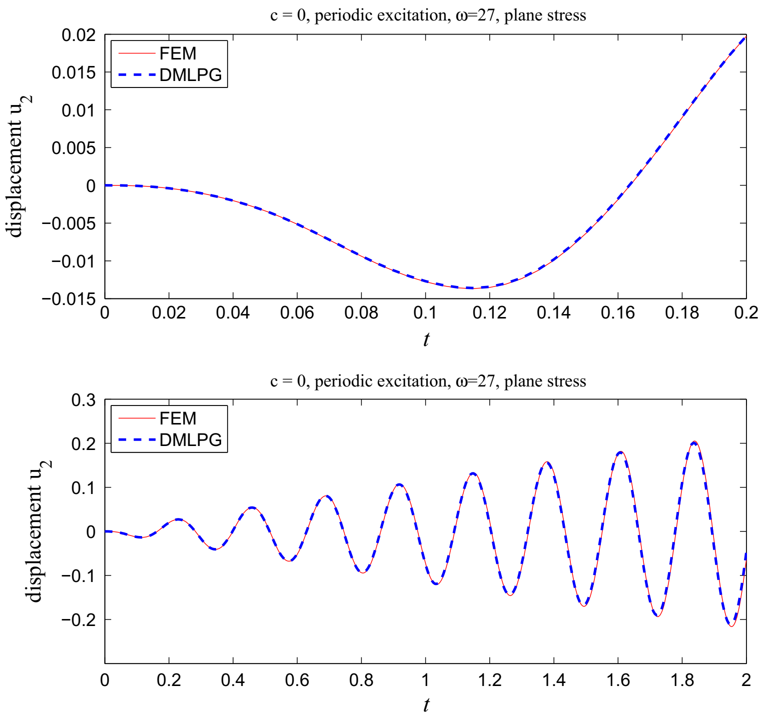


Fig. 8 Comparing FEM and DMLPG solutions; time variations of displacement u_2 at point A for periodic load, $c = 0$

6.2 Frame structure

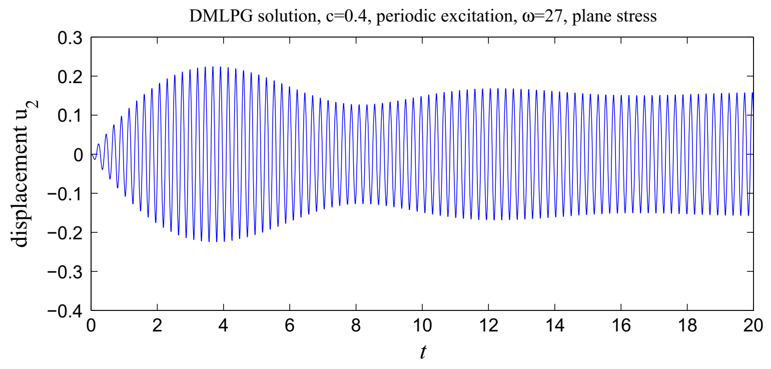
As second example, we analyze a frame structure as shown in Fig. 5. Material parameters are the same as those given in [28]: $E = 10^4$ Pa, $\rho = 1$ kg/m^3, $\nu = 0.2$ and $a = 1$. The base of the frame is fixed, and the structure is loaded by wind-like loading where $P = 10g(t)$ and

$$g(t) = \begin{cases} 10t, & 0 \leqslant t \leqslant 0.1, \\ 0, & t > 0.1. \end{cases}$$

A finite element solution, obtained by meshing the frame to 500 quadrilateral plane stress elements in ABAQUS, is used as a reference solution. DMLPG5 with circular subdomains and 369 meshless points (see Fig. 5) are used for numerical simulation. Time variations of displacement u_1 at point A and traction t_1 at point B are depicted and compared with the FEM solutions in Fig. 6. A good agreement with the reference solution can be observed.

6.3 Vibration analysis of a cantilever beam

Consider a cantilever beam loaded by a tangential and parabolic traction on the free end, as shown in Fig. 7. Basic parameters are $L = 48$ m, $D = 12$ m, $E = 3 \times 10^7$ Pa, $\nu = 0.3$, $\rho = 1$ kg/m^3 and the external excitation force $P = 1000g(t)$.



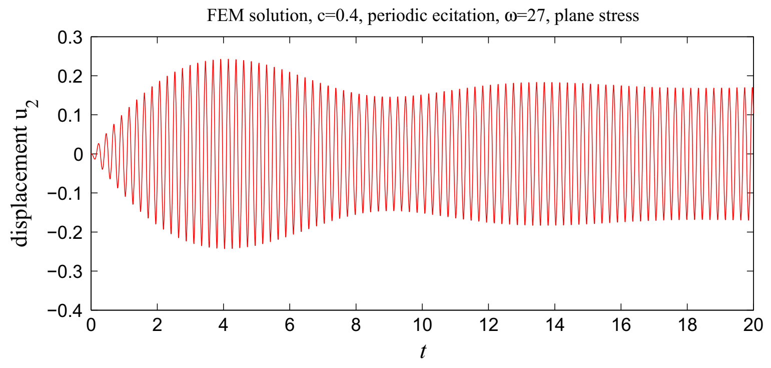


Fig. 9 Time variations of displacement u_2 at point A for periodic load, $c = 0.4$, DMLPG (*up*) and FEM (*down*) solutions

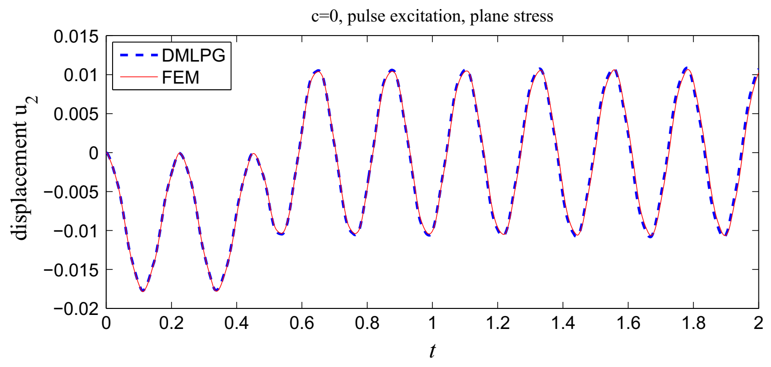


Fig. 10 Comparing FEM and DMLPG solutions; time variations of displacement u_2 at point A for transient load, $c = 0$

As before, the finite element results are used for reference solutions. The FEM solutions are obtained by ABAQUS software with 400 quadrilateral plane stress elements. DMLPG1 with circular subdomains is applied, and results are compared with the reference solutions.

First, we consider a simple periodic load

$$g(t) = \sin(\omega t),$$

where ω is the frequency of dynamic load, and $\omega = 27$ rad/s is used in this paper [10,15].

In Fig. 8 displacement u_2 at point A is drawn and compared with FEM in the time intervals $[0, 0.2]$ and $[0, 2]$ for $c = 0$. In Fig. 9 the time evolutions of displacement u_2 at point A by both DMLPG and FEM are shown in the time interval $[0, 20]$ for $c = 0.4$.

Second, we consider the transient response of the beam subjected to a suddenly loaded and suddenly vanishing force $P = 1000g(t)$, where

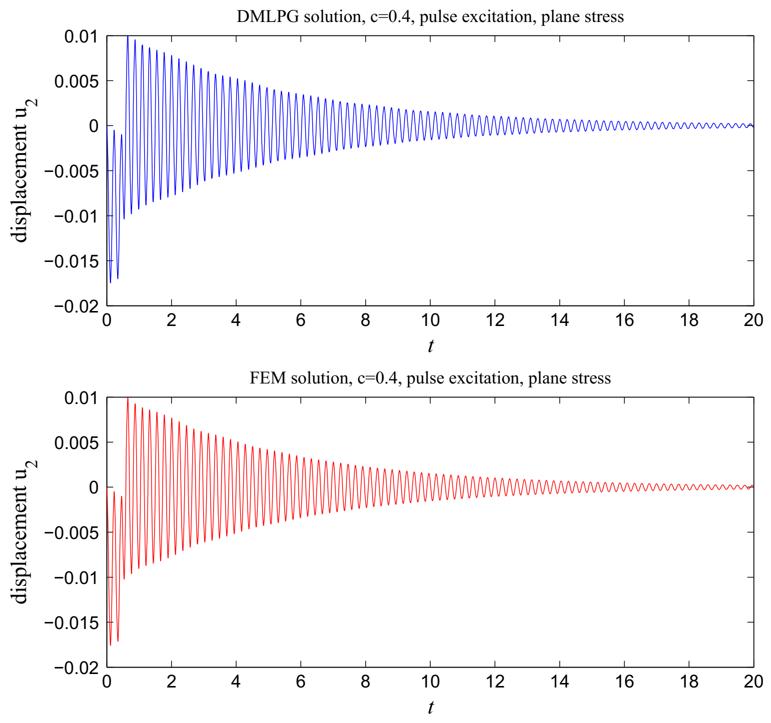


Fig. 11 Time variations of displacement u_2 at point A for transient load, $c = 0.4$, DMLPG (*up*) and FEM (*down*) solutions

$$g(t) = \begin{cases} 1, & 0 \leqslant t \leqslant 0.5, \\ 0, & t > 0.5. \end{cases}$$

Transient responses are given and compared with FEM solutions in Figs. 10 and 11 with and without damping, respectively.

In both cases, one can see the good agreement between DMLPG and FEM solutions.

References

1. Atluri, S.N., Cho, J.Y., Kim, H.G.: Analysis of thin beams, using the meshless local Petrov–Galerkin method, with generalized moving least squares interpolations. Comput. Mech. **24**, 334–347 (1999)
2. Babuska, I., Banerjee, U., Osborn, J., Zhang, Q.: Effect of numerical integration on meshless methods. Comput. Methods Appl. Mech. Eng. **198**, 27–40 (2009)
3. Beissel, S., Belytschko, T.: Nodal integration of the element-free Galerkin method. Comput. Methods Appl. Mech. Eng. **139**, 49–74 (1996)
4. Belytschko, T., Krongauz, Y., Organ, D., Fleming, M., Krysl, P.: Meshless methods: an overview and recent developments. Comput. Methods Appl. Mech. Eng. **139**, 3–47 (1996)
5. Carpinteri, A., Ferro, G., Ventura, G.: The partition of unity quadrature in meshless methods. Int. J. Numer. Methods Eng. **54**, 987–1006 (2002)
6. Chiba, F., Kako, T.: Stability and error analyses by energy estimate for Newmarks method. Natl. Inst. Fusion Sci. **17-18, 40**, 82–91 (1999)
7. Dai, B., Cheng, J., Zheng, B.: A moving kriging interpolation-based meshless local Petrov–Galerkin method for elastodynamic analysis. Int. J. Appl. Mech. **5**, 135,001 (21 pages) (2013)
8. Dolbow, J., Belytschko, T.: Numerical integration of the galerkin weak form in meshfree methods. Comput. Mech. **23**, 219–230 (1999)
9. Ghadiri Rad, M.H., Shahabian, F., Hosseini, S.M.: A meshless local Petrov-Galerkin method for nonlinear dynamic analyses of hyper-elastic FG thick hollow cylinder with Rayleigh damping. Acta Mech. p. In press (2014). doi:10.1007/s00707-014-1266-2
10. Gu, Y.T., Liu, G.R.: A meshless local Petrov-Galerkin (MLPG) method for free and forced vibration analyses for solids. Comput. Mech. **27**, 188–198 (2001)
11. Hoghes, T., Pister, K., Taylor, R.: Implicit–explicit finite elements in nonlinear transient analysis. Comput. Methods Appl. Mech. Eng. **17–18, Part 1**, 159–182 (1979)

12. Idesman, A., Pham, D.: Finite element modeling of linear elastodynamics problems with explicit time-integration methods and linear elements with the reduced dispersion error. Comput. Methods Appl. Mech. Eng. **271**, 86–108 (2014)
13. Kandilas, C.B.: Transient elastodynamic analysis of nonhomogeneous anisotropic plane bodies. Acta Mech. **223**, 861–878 (2012)
14. Lancaster, P., Salkauskas, K.: Surfaces generated by moving least squares methods. Math. Comput. **37**, 141–158 (1981)
15. Long, S.Y., Liu, K.Y., Hu, D.A.: A new meshless method based on MLPG for elastic dynamic problems. Eng. Anal. Bound. Elem. **30**, 43–48 (2006)
16. Mazzia, A., Pini, G.: Product Gauss quadrature rules vs. cubature rules in the meshless local Petrov–Galerkin method. J. Complex. **26**, 82–101 (2010)
17. Mazzia, A., Pini, G., Sartoretto, F.: Numerical investigation on direct MLPG for 2D and 3D potential problems. Comput. Model. Simul. Eng. **88**, 183–209 (2012)
18. Mazzia, A., Pini, G., Sartoretto, F.: Meshless techniques for anisotropic diffusion. Appl. Math. Comput. **236**, 54–66 (2014)
19. Mirzaei, D.: Analysis of moving least squares approximation revisited. J. Comput. Appl. Math. **282**, 237–250 (2015)
20. Mirzaei, D.: A new low–cost meshfree method for two and three dimensional problems in elasticity. Appl. Math. Model. p. In press (2015)
21. Mirzaei, D.: Error bounds for GMLS derivatives approximations of sobolev functions. J. Comput. Appl. Math. **294**, 93–101 (2016)
22. Mirzaei, D., Schaback, R.: Direct Meshless Local Petrov–Galerkin (DMLPG) method: a generalized MLS approximation. Appl. Numer. Math. **33**, 73–82 (2013)
23. Mirzaei, D., Schaback, R.: Solving heat conduction problem by the Direct Meshless Local Petrov-Galerkin (DMLPG) method. Numer. Algorithms **65**, 275–291 (2014)
24. Mirzaei, D., Schaback, R., Dehghan, M.: On generalized moving least squares and diffuse derivatives. IMA J. Numer. Anal. **32**, 983–1000 (2012)
25. Moosavi, M.R., Khelil, A.: Finite volume meshless local Petrov-Galerkin method in elastodynamic problems. Eng. Anal. Bound. Elem. **33**, 1016–1021 (2009)
26. Pecher, R.: Efficient cubature formulae for MLPG and related methods. Int. J. Numer. Methods Eng. **65**, 566–593 (2006)
27. Ramezani, M., Mojtabaei, M., Mirzaei, D.: DMLPG solution of the fractional advection–diffusion problem. Eng. Anal. Bound. Elem. **59**, 36–42 (2015)
28. Sladek, J., Sladek, V., Van Keer, R.: Meshless local boundary integral equation method for 2D elastodynamic problems. Int. J. Numer. Methods Eng. **57**, 235–249 (2003)
29. Sladek, J., Sladek, V., Zhang, C.: Application of meshless local Petrov-Galerkin (MLPG) method to elastodynamic problems in continuously nonhomogeneous solids. CMES Comput. Model. Eng. Sci. **4**, 637–648 (2000)
30. Soares, D. Jr., Sladek, V., Sladek, J.: Modified meshless local Petrov–Galerkin formulations for elastodynamics. Int. J. Numer. Methods Eng. **90**, 1508–1828 (2012)
31. Taleei, A., Dehghan, M.: Direct meshless local Petrov-Galerkin method for elliptic interface problems with applications in electrostatic and elastostatic. Comput. Methods Appl. Mech. Eng. **278**, 479–498 (2014)
32. Wendland, H.: Scattered Data Approximation. Cambridge University Press, Cambridge (2005)
33. Zhang, Q., Banerjee, U.: Numerical integration in Galerkin meshless methods, applied to elliptic Neumann problem with non-constant coefficients. Adv. Comput. Math. **37**, 453–492 (2012)