Engineering Analysis with Boundary Elements 37 (2013) 1576–1584

Contents lists available at ScienceDirect

Engineering Analysis with Boundary Elements

journal homepage: www.elsevier.com/locate/enganabound

The improved element-free Galerkin method for two-dimensional elastodynamics problems

Zan Zhang [a], S.Y. Hao [b], K.M. Liew [c], Y.M. Cheng [d,*]

[a] School of Science, East China University of Science and Technology, Shanghai 200237, China
[b] School of Traffic and Transportation, Lanzhou Jiaotong University, Lanzhou 730070, China
[c] Department of Civil and Architectural Engineering, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong Special Administrative Region
[d] Shanghai Institute of Applied Mathematics and Mechanics, Shanghai University, Shanghai 200072, China

ARTICLE INFO

Article history:
Received 7 April 2012
Accepted 28 August 2013
Available online 15 October 2013

Keywords:
Weighted orthogonal function
Improved moving least squares (IMLS) approximation
Improved element-free Galerkin (IEFG) method
Elastodynamics
Penalty method
Newmark-β algorithm

ABSTRACT

In this paper, we derive an improved element-free Galerkin (IEFG) method for two-dimensional linear elastodynamics by employing the improved moving least-squares (IMLS) approximation. In comparison with the conventional moving least-squares (MLS) approximation function, the algebraic equation system in IMLS approximation is well-conditioned. It can be solved without having to derive the inverse matrix. Thus the IEFG method may result in a higher computing speed. In the IEFG method for two-dimensional linear elastodynamics, we employed the Galerkin weak form to derive the discretized system equations, and the Newmark time integration method for the time history analyses. In the modeling process, the penalty method is used to impose the essential boundary conditions to obtain the corresponding formulae of the IEFG method for two-dimensional elastodynamics. The numerical studies illustrated that the IEFG method is efficient by comparing it with the analytical method and the finite element method.

1. Introduction

Structures experience vibration under the action of dynamic forces, which vary with time. In such cases, stress, strain, and displacement are functions of time, and the principles and theories of dynamics apply. These dynamic problems are typically described by linear or nonlinear partial differential equations associated with boundary and initial conditions; however, it is usually very difficult to obtain analytical solutions for them. Therefore, for many decades, researchers have devoted considerable effort to develop efficient numerical methods to solve such problems. The finite element method (FEM) and boundary element method (BEM) are presently the dominant approaches for modeling transient dynamic problems [1]. They are numerical methods based on meshes, and extremely large deformations of the mesh may encounter re-meshing.

Meshless methods constitute an interesting numerical technique for solving many engineering problems that are not suited to conventional numerical methods, with a minimum of meshing or no meshing at all [2]. Belytschko et al. used the element-free Galerkin (EFG) method to solve dynamic fracture problems [3–6], and Liu et al. [7] proposed the reproducing kernel particle method (RKPM) for structural dynamic analysis. The meshless local Petrov–Galerkin (MLPG) method has been applied to solve two-dimensional linear elastodynamics by Ching [8], and to analyze the free and forced vibration of solids by Gu et al. [9]. Bueche et al. [10] applied the natural element method (NEM) to solve two-dimensional linear elastodynamics problems, and Li and Cheng [11] presented a meshless manifold method to analyze both transient elastodynamic deformations and dynamic fractures. Cheng and Liew [12–14] used a direct meshless boundary integral equation method, which coined the boundary element-free method (BEFM), to solve two-dimensional linear elastodynamics. Chen and Cheng presented a complex variable reproducing kernel particle method for two-dimensional elastodynamics [15]. Cheng et al. proposed a complex variable element-free Galerkin (CVEFG) method for two-dimensional elastodynamics problems [16].

One of the major features of the EFG method is that the moving least-square (MLS) approximation is employed to construct shape functions. Based on the weighted orthogonal basis function, the improved moving least-square (IMLS) approximation was adopted to formulate different meshless methods, such as new implementation of the EFG method [17] and the BEFM [14,15]. The MLS approximation was developed from the conventional least-squares method, and in practical numerical processes, it essentially involved the application of the conventional method to every selected point. A disadvantage of the conventional least-squares method is that the final system of algebraic equations is sometimes ill-conditioned. Hence, in MLS approximation, the ill-conditioned system of algebraic equations must be solved. However, we are unable to locate which

* Corresponding author. Tel.: +86 21 56331043; fax: +86 21 36033287.
E-mail address: ymcheng@shu.edu.cn (Y.M. Cheng).

0955-7997/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.enganabound.2013.08.017

algebraic equation system is ill-conditioned. In mathematical theories, there are no methods for judging whether an algebraic equation system is ill-conditioned or not before the equation is solved. Thus we sometimes cannot obtain a good solution or even a correct numerical solution. Using the weighted orthogonal basis function, the IMLS approximation may overcome this disadvantage of the MLS approximation. The algebraic equations system in the IMLS approximation is well-conditioned, and it can be solved without deriving the inverse matrix [14,15]. This is because there are fewer coefficients in the IMLS than in the MLS approximation. In the IEFG method that is formed with the IMLS approximation, fewer nodes are selected in the entire domain than those are selected in the conventional EFG method. Thus the IEFG method should result in a higher computing speed [18–20].

In this paper, based on the IMLS approximation, an improved element-free Galerkin (IEFG) method for two-dimensional linear elastodynamics is proposed. In the IEFG technique, the Galerkin weak form is employed to obtain the discretized system equations, and the Newmark time integration method is used for the time history analyses. In the solution process, the penalty method is employed to impose the essential boundary conditions so as to obtain the corresponding formulae of the IEFG method for 2D elastodynamics.

2. Improved moving least-square approximation

In the EFG method, MLS approximation is employed to construct the shape function. A useful property of this approximation is that its continuity is equal to the continuity of the weight function, and thus highly continuous approximations can be generated when the appropriate weight function is chosen. This means that the post-processing to generate smooth stress fields, which is mandatory for C^0 finite element methods, is unnecessary [21,22].

In the IMLS approximation, the trial function is

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

where $p_i(x),\ i=1,2,\ldots,m$, are monomial basis functions, m is the number of terms in the basis, and $a_i(x)$ are the coefficients of the basis functions.

The local approximation, as described by Lancaster and Salkauskas [23], is

$$u^h(x,\bar{\boldsymbol{x}}) = \sum_{i=1}^{m} p_i(\bar{\boldsymbol{x}})a_i(\boldsymbol{x}) = \boldsymbol{p}^{\mathrm{T}}(\bar{\boldsymbol{x}})\boldsymbol{a}(\boldsymbol{x}). \tag{2}$$

For obtaining the local approximation of the function $u(\boldsymbol{x})$, the difference between it and the local approximation $u^h(\boldsymbol{x})$ must be minimized by a weighted least-squares method.

Define a function

$$J = \sum_{I=1}^{n} w(\boldsymbol{x}-\boldsymbol{x}_I)[u^h(\boldsymbol{x},\boldsymbol{x}_I)-u(\boldsymbol{x}_I)]^2$$

$$= \sum_{I=1}^{n} w(\boldsymbol{x}-\boldsymbol{x}_I)\left[\sum_{i=1}^{m} p_i(\boldsymbol{x}_I)\cdot a_i(\boldsymbol{x})-u(\boldsymbol{x}_I)\right]^2, \tag{3}$$

where $w(\boldsymbol{x}-\boldsymbol{x}_I)$ is a weight function with a domain of influence, and $\boldsymbol{x}_I$ $(I=1,2,\ldots,n)$ are the nodes with domains of influence that cover the point $\boldsymbol{x}$.

Eq. (3) can be written as

$$J = (\boldsymbol{P}\boldsymbol{a}-\boldsymbol{u})^{\mathrm{T}}\boldsymbol{W}(\boldsymbol{x})(\boldsymbol{P}\boldsymbol{a}-\boldsymbol{u}), \tag{4}$$

where

$$\boldsymbol{u}^T = (u_1, u_2, \ldots, u_n), \tag{5}$$

$$\boldsymbol{P} = \begin{bmatrix} p_1(\boldsymbol{x}_1) & p_2(\boldsymbol{x}_1) & \cdots & p_m(\boldsymbol{x}_1) \\ p_1(\boldsymbol{x}_2) & p_2(\boldsymbol{x}_2) & \cdots & p_m(\boldsymbol{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(\boldsymbol{x}_n) & p_2(\boldsymbol{x}_n) & \cdots & p_m(\boldsymbol{x}_n) \end{bmatrix} \tag{6}$$

and

$$\boldsymbol{W}(\boldsymbol{x}) = \begin{bmatrix} w(\boldsymbol{x}-\boldsymbol{x}_1) & 0 & \cdots & 0 \\ 0 & w(\boldsymbol{x}-\boldsymbol{x}_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w(\boldsymbol{x}-\boldsymbol{x}_n) \end{bmatrix}. \tag{7}$$

The minimization condition requires that

$$\frac{\partial J}{\partial \boldsymbol{a}} = 0, \tag{8}$$

which results in the equation system:

$$\boldsymbol{A}(\boldsymbol{x})\boldsymbol{a}(\boldsymbol{x}) = \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u}, \tag{9}$$

where matrices $\boldsymbol{A}(\boldsymbol{x})$ and $\boldsymbol{B}(\boldsymbol{x})$ are

$$\boldsymbol{A}(\boldsymbol{x}) = \boldsymbol{P}^{\mathrm{T}}\boldsymbol{W}(\boldsymbol{x})\boldsymbol{P} \tag{10}$$

and

$$\boldsymbol{B}(\boldsymbol{x}) = \boldsymbol{P}^{\mathrm{T}}\boldsymbol{W}(\boldsymbol{x}). \tag{11}$$

In the MLS approximation, Eq. (9) is sometimes ill-conditioned, even in the presence of a singular phenomenon. In addition, it is difficult to obtain the correct numerical solution. Using the weighted orthogonal basis functions, the IMLS approximation was presented [24,25].

For $\forall f(\boldsymbol{x}),\ g(\boldsymbol{x}) \in span(\boldsymbol{p})$, define

$$(f,g) = \sum_{I=1}^{n} w(\boldsymbol{x}-\boldsymbol{x}_I)f(\boldsymbol{x}_I)g(\boldsymbol{x}_I), \tag{12}$$

and then $(f,\ g)$ is an inner product, and $span\ (\boldsymbol{p})$ is a Hilbert space.

In the Hilbert space $span\ (\boldsymbol{p})$, for the set of points $\{\boldsymbol{x}_i\}$ and the weight functions $\{w_i\}$, if the functions $p_1(\boldsymbol{x}), p_2(\boldsymbol{x}), \ldots, p_m(\boldsymbol{x})$ satisfy the conditions

$$(p_k,p_j) = \sum_{i=1}^{n} w_i p_k(\boldsymbol{x}_i)p_j(\boldsymbol{x}_i) = \begin{cases} 0, & k\neq j \\ A_k, & k=j \end{cases} \quad (k,j=1,2,\ldots,m), \tag{13}$$

then the function set $p_1(\boldsymbol{x}), p_2(\boldsymbol{x}), \ldots, p_m(\boldsymbol{x})$ is called a weighted orthogonal function set with a weight function $\{w_i\}$ about points $\{\boldsymbol{x}_i\}$. If $p_1(\boldsymbol{x}), p_2(\boldsymbol{x}), \ldots, p_m(\boldsymbol{x})$ are polynomials, the function set $p_1(\boldsymbol{x})$, $p_2(\boldsymbol{x}), \ldots, p_m(\boldsymbol{x})$ is called a weighted orthogonal polynomial set with the weight function $\{w_i\}$ about points $\{\boldsymbol{x}_i\}$.

From Eqs. (12) and (9) can be written as

$$\begin{bmatrix} (p_1,p_1) & (p_1,p_2) & \cdots & (p_1,p_m) \\ (p_1,p_1) & (p_2,p_2) & \cdots & (p_2,p_m) \\ \vdots & \vdots & \ddots & \vdots \\ (p_1,p_1) & (p_m,p_2) & \cdots & (p_m,p_m) \end{bmatrix} \begin{bmatrix} a_1(\boldsymbol{x}) \\ a_2(\boldsymbol{x}) \\ \vdots \\ a_m(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} (p_1,u_I) \\ (p_2,u_I) \\ \vdots \\ (p_m,u_I) \end{bmatrix}. \tag{14}$$

When the basis function set $p_i(\boldsymbol{x}) \in span(\boldsymbol{x})$, $i=1,2,\ldots,m$, is used as the basis function, Eq. (14) becomes

$$\begin{bmatrix} (p_1,p_1) & 0 & \cdots & 0 \\ 0 & (p_2,p_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (p_m,p_m) \end{bmatrix} \begin{bmatrix} a_1(\boldsymbol{x}) \\ a_2(\boldsymbol{x}) \\ \vdots \\ a_m(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} (p_1,u_I) \\ (p_2,u_I) \\ \vdots \\ (p_m,u_I) \end{bmatrix} \tag{15}$$

and

$$\boldsymbol{a}(\boldsymbol{x}) = \overline{\boldsymbol{A}}(\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x})\boldsymbol{u}, \tag{16}$$