Advances in Engineering Software 41 (2010) 480–488

Contents lists available at ScienceDirect

Advances in Engineering Software

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

Boundary point method for linear elasticity using constant and quadratic moving elements

Hang Ma [a,*], Juan Zhou [a], Qing-Hua Qin [b]

[a] *Department of Mechanics, College of Sciences, Shanghai University, Shanghai 200444, PR China*
[b] *Department of Engineering, Australian National University, ACT 0200, Australia*

ARTICLE INFO

Article history:
Received 18 July 2008
Received in revised form 10 September 2009
Accepted 1 October 2009
Available online 12 November 2009

Keywords:
Boundary integral equation
Boundary-type meshless method
Fundamental solution
One-point computing
Quadratic moving element
Boundary point method
Linear elasticity

ABSTRACT

Based on the boundary integral equations and stimulated by the work of Young et al. [J Comput Phys 2005;209:290–321], the boundary point method (BPM) is a newly developed boundary-type meshless method enjoying the favorable features of both the method of fundamental solution (MFS) and the boundary element method (BEM). The present paper extends the BPM to the numerical analysis of linear elasticity. In addition to the constant moving elements, the quadratic moving elements are introduced to improve the accuracy of the stresses near the boundaries in the post processing and to enhance the analysis for thin-wall structures. Numerical tests of the BPM are carried out by benchmark examples in the two- and three-dimensional elasticity. Good agreement is observed between the numerical and the exact solutions.

1. Introduction

For a large class of physical and engineering problems, boundary-type solution methodologies are now well established as viable alternatives to the prevailing domain-type methods such as the finite difference method, the finite element method and finite volume method, because of the computational advantages they offer and its distinctive feature of requiring only the numerical discretisation of the boundary of the solution domain. In particular, boundary-type methods have the merit of predicting accurate and complete solutions while reducing the dimensionality of any given problem by one and thus simplifying the effort involved in data preparation and computer time. The boundary element method (BEM) is the most popular and efficient boundary-type solution procedure, formulated in terms of boundary integral equations (BIE). In the BIE the governing differential equations are converted into integral identities applied over the boundary of the domain. Then the boundary is discretised into small elements in order to carry out the integration.

For elasto-static problems, the variations of displacements and tractions can be described in terms of values at a number of nodal points associated with each element. Shape functions of linear, quadratic or higher order are used for interpolating between the nodes. The integrations over the boundary are usually performed by the Gauss quadrature technique. Details of the BEM can be found in various publications [1,2]. However, the use of elements in the BEM with integrations especially in three-dimensions still puts bourdon on computing efficiency. For example, the BEM requires polygonisation of the boundary surfaces in general 3D cases, and boundary curves in general 2D cases. The regular, weakly singular, strongly singular, and hyper-singular integrals need be dealt with over boundary segments, which is usually a cumbersome and non-trivial task. The presentation of the boundary contour method [3–6] is an effort to improve the efficiency by transferring surface integrals into line integrals via Stokes' theorem. The boundary node method (BNM) represents a coupling between the BIE and the moving least square approximations [7–9]. Using polynomial or radial basis function (RBF) as basis functions, the point interpolation method has been proposed to construct meshfree shape function with Kronecker delta function properties [10,11] with which the boundary conditions can be easily enforced and coupled with the BIE to construct boundary-type meshfree methods [12]. Remarkable progress has been achieved in solving a wide range of static and dynamic problems for solids and structures.

The Trefftz method is another noteworthy boundary-type meshless method featuring conciseness and ease of performance. The crucial structure of the Trefftz method is the use of a set of trial

[*] Corresponding author.
E-mail address: hangma@staff.shu.edu.cn (H. Ma).

0965-9978/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2009.10.006

functions, singular or non-singular, which *a priori* satisfies the governing differential equation under consideration [13,14]. In the boundary knot method (BKM), a set of non-singular general solutions is employed as trial functions to solve Helmholtz and convection–diffusion equations [15,16]. The merits of using non-singular trial functions such as T-complete functions [17] or general solutions lie in the fact that the collocation and observation points can be coincident and placed on the real boundary of the problem. However, the system matrix thus formed may be ill-conditioned or the condition number of the matrix could become large [17] so that the scale of solvable problems would be limited.

In the method of fundamental solutions (MFS, sometimes called also the F-Trefftz method, charge simulation method, or singularity method), singular functions are taken as the trial functions [18–23]. Just like the BEM, the MFS are best applicable in situations where a fundamental solution to the partial differential equation in question is known. In such cases, the dimensionality of the discretization is reduced. The MFS has certain advantages over the BEM that stems mostly from the fact that the pointisation of the boundary is needed only, which completely avoids any integral evaluations, and makes no principal difference in coding between the 2D and the 3D cases. Because of the singular nature of fundamental solutions, however, the source points must be placed outside the problem domain to avoid the singularity problem, forming a fictitious non-physical boundary. The location of this artificial boundary represents the most serious problem of the MFS and has to be dealt with heuristically, especially for engineering problems with complicated geometry. If the offset distance is too small, the diagonal coefficients of the system matrix will diverge because of the singular nature of fundamental solutions. On the other hand, if the artificial boundary is distant from the real boundary, the system matrices also become ill-conditioned since the condition number of the influence matrix becomes very large. The location of the source and observation points is vital to the accuracy of the solution when implementing the MFS.

A distinct feature in the Trefftz type method including the BKM and the MFS is that each of the coefficients of system matrices is computed on only *one point*, or named integration-free, compared with element-type methods in which the integration must be performed over elements. This feature, which can be termed *one-point* computing, greatly reduces the computing cost. Very recently, by making use of strongly and hyper-singular fundamental solutions of Laplace equation with an indirect formulation, Young et al. proposed a modified MFS for 2D potential problems [24] in which both the source and observation points are located on the real boundary with a singularity removal technique commonly used in the BIE, thus avoiding the inconvenience of using an artificial boundary completely. The only shortcoming of the method seems to be lie in that the equally spaced nodes have to be used along the boundary since the singularity removal technique depends on the divergence-free properties of the kernels [24].

Enlightened by the above-mentioned work, a novel boundary-type meshless method—the boundary point method (BPM) is developed recently for solving the two- and three-dimensional potential problems [25]. The BPM can be looked as something between the MFS and the BEM. In the BPM, the boundary of the problem domain is discretized by boundary nodes, each node having a territory or support where the field variables are defined. By making use of the properties of fundamental solutions, the coefficients of the system matrix in the BPM are computed according to the distances between the two points, the source and observation points. In the cases when the distances are not small, the integrals of kernel functions are evaluated by *one-point* computing, just like that carried out in the MFS, which consist of the most off-diagonal terms of the system matrix. In the cases when the distances are not large, the integrals of kernel functions are evaluated by Gauss quadrature over territories. If the two points coincide, the integrals are treated by the mature techniques available in the BEM [26–28], which constitute the principal diagonal terms of the system matrix. As the adjacent nodes describe the local features of boundary such as position, curvature and direction, the so-called moving elements are introduced in the BPM [25] by organizing the relevant adjacent nodes tentatively, over which the treatment of singularity and Gauss quadrature can be carried out for evaluating the integrals in the latter two cases, i.e., the coincidence or the small distances between the two points.

The current paper extends the BPM to the numerical analysis of linear elasticity. As the field variables are assumed to be constant over each of the territories or supports [25], the accuracy of the field variables in the domain very close to the boundary need to be improved. However, this can be realized by introducing tentatively the quadratic moving element into the BPM in the present work. The basic formulations of the BIE in elasticity are presented in Section 2 as the starting point with the outline of the BPM given briefly in Section 3. The quadratic moving elements are introduced in Section 4. The numerical examples are tested in Section 5, including the comparisons between *one-point* computing and Gauss quadrature and some benchmark examples are presented in the two- and three-dimensional elasticity, showing the feasibility and accuracy of the proposed method.

2. Basic formulations

Considering a linear elastic domain Ω surrounded by the piecewise smooth boundary Γ free of body force, the equilibrium equation is:

$$\sigma_{ij,j} = 0, \quad \text{in } \Omega, \tag{1}$$

where σ_{ij} is the stress tensor. The corresponding boundary conditions are given by:

$$u_i(x) = \bar{u}_i(x), \quad x \in \Gamma_u, \tag{2}$$

$$\tau_i(x) = \sigma_{ij}(x)n_j = \bar{\tau}_i(x) \quad x \in \Gamma_\sigma, \tag{3}$$

where u_i are the displacements, τ_i the tractions, $\bar{u}_i$ the prescribed displacements on the displacement boundary Γ_u, $\bar{\tau}_i$ the prescribed tractions on the traction boundary Γ_σ, and n_i is the outward unit that is normal to the boundary $\Gamma = \Gamma_\sigma \cup \Gamma_u$. From the method of weighted residuals and the constitutive relations of elasticity [1,2], the direct formulations of integral equations can be written as:

$$\gamma(y)\delta_{ij}u_j(y) = \int_\Gamma \tau_j(x)u^*_{ij}(x,y)d\Gamma(x) - \int_\Gamma u_j(x)\tau^*_{ij}(x,y)d\Gamma(x), \tag{4}$$

$$\gamma(y)\delta_{ij}\sigma_{jk}(y) = \int_\Gamma \tau_j(x)u^*_{ikj}(x,y)d\Gamma(x) - \int_\Gamma u_j(x)\tau^*_{ikj}(x,y)d\Gamma(x), \tag{5}$$

where u^*_{ij} and τ^*_{ij} are the Kelvin's displacement and the traction fundamental solutions, u^*_{ikj} and τ^*_{ikj} the derived displacement and traction fundamental solutions, respectively. γ represents the coefficient of the free term of the BIE depending on where the source point y is located. $\gamma(y) = 1$ if $y \in \Omega$, $\gamma(y) = 0$ if $y \in \overline{\Omega \cup \Gamma}$, $\gamma(y) = 0.5$ if $y \in \Gamma$ which is smooth in the neighborhood of the point y. δ_{ij} is the Kronecker symbol. With the Cauchy's relation $\tau_i = \sigma_{ij}n_j$, Eq. (5) can be written as follows:

$$\gamma(y)\delta_{ij}\tau_j(y) = n_k(y)\left\{\int_\Gamma \tau_j(x)u^*_{ikj}(x,y)d\Gamma(x) - \int_\Gamma u_j(x)\tau^*_{ikj}(x,y)d\Gamma(x)\right\}. \tag{6}$$

It is known that, in Eqs. (4)–(6), when $y \in \Gamma$, the integrals with the kernel u^*_{ij} are weakly singular ($O(\log(r^{-1}))$ for 2D or $O(r^{-1})$ for 3D), the integrals with the kernels τ^*_{ij} and u^*_{ikj} are strongly singular

($O(r^{-d})$), and the integrals with the kernel τ^*_{ikj} are hyper-singular ($O(r^{-d-1})$), where $d = 1$ for 2D and $d = 2$ for 3D. r is the distance between the source and the observation points:

$$r = \sqrt{(x_k - y_k)(x_k - y_k)}. \tag{7}$$

Conventionally, when $y \in \Gamma$, Eqs. (4) and (6) are called the conventional and the hyper-singular BIE, abbreviated as the CBIE and the HBIE, respectively. Mathematically, each of the two equations in combination with the boundary conditions fully describes elasticity problems, which are the foundation of boundary-type numerical solutions such as the BEM [1,2], as well as the BPM to be described in the next section.

3. The boundary point method

In the BPM, one of the features of its implementation is its discretization, which is simple to implement but slightly different from that in the BEM using constant boundary elements [25]. Suppose that N nodes are placed on a smooth section of the boundary Γ, each node being the centroid of a territory or support, $\Delta\Gamma_m$ ($m = 1, \ldots, N$), which is locally smooth. A territory is a segment of curved line in 2D or a piece of curved surface in 3D such that the node in the BPM locates on the Γ. All the territories form the total boundary, that is,

$$\Gamma = \bigcup_{m=1}^{N} \Delta\Gamma_m, \quad \Delta\Gamma_m = \begin{cases} l_m & \text{(2D)} \\ a_m & \text{(3D)} \end{cases}, \tag{8}$$

where l_m and a_m denote the curved length and the curved area of territory of the m th node, respectively. The boundary variables are implied to be constant for a territory. However, the difference between a territory in the BPM and a constant element in the BEM can be seen clearly in that an element is a segment of straight line in 2D or a piece of plane surface in 3D so that in general the node in the BEM does not locate on the real Γ. However, the node in the BPM does locate on the real Γ. In addition, the curved line boundary in 2D and the curved surface boundary in 3D can be correctly described by the use of moving elements when necessary.

After the discretization of Γ into territories, Eqs. (4) and (6) can be written in discrete form as follows:

$$\gamma(y^n)\delta_{ij}u_j(y^n) = \sum_{m=1}^{N}\left\{\tau_j^m \int_{\Delta\Gamma_m} u^*_{ij}(x, y^n)d\Gamma(x) - u_j^m \int_{\Delta\Gamma_m} \tau^*_{ij}(x, y^n)d\Gamma(x)\right\}, \tag{9}$$

$$\gamma(y^n)\delta_{ij}\tau_j(y^n) = \sum_{m=1}^{N} n_k(y^n)\left\{\tau_j^m \int_{\Delta\Gamma_m} u^*_{ikj}(x, y^n)d\Gamma(x) - u_j^m \int_{\Delta\Gamma_m} \tau^*_{ikj}(x, y^n)d\Gamma(x)\right\}, \tag{10}$$

where $y^n \in \Gamma$ ($n = 1, \ldots, N$) in above two equations. Defining the integrals of kernel functions as follows:

$$G_{ij}^{nm} = \int_{\Delta\Gamma_m} u^*_{ij}(x, y^n)d\Gamma(x), \tag{11a}$$

$$F_{ij}^{nm} = \int_{\Delta\Gamma_m} \tau^*_{ij}(x, y^n)d\Gamma(x), \tag{11b}$$

$$T_{ij}^{nm} = n_k(y^n)\int_{\Delta\Gamma_m} u^*_{ikj}(x, y^n)d\Gamma(x), \tag{11c}$$

$$H_{ij}^{nm} = n_k(y^n)\int_{\Delta\Gamma_m} \tau^*_{ikj}(x, y^n)d\Gamma(x), \tag{11d}$$

the CBIE (4) and the HBIE (6) can be written in compact form as:

$$\sum_{m=1}^{N} G_{ij}^{nm}\tau_j^m = \sum_{m=1}^{N}\left(F_{ij}^{nm} + \gamma\delta_{ij}\delta_{nm}\right)u_j^m, \tag{12}$$

$$\sum_{m=1}^{N}\left(T_{ij}^{nm} - \gamma\delta_{ij}\delta_{nm}\right)\tau_j^m = \sum_{m=1}^{N} H_{ij}^{nm}u_j^m, \tag{13}$$

Using either of Eqs. (12) and (13) incorporated with the boundary conditions (2) and (3) and rearranging the equation, the system algebraic equations can be obtained as:

$$\mathbf{Ax} = \mathbf{b}. \tag{14}$$

The $2N$ (2D) or $3N$ (3D) boundary unknowns are obtained after the solution of (14) with the linear algebraic solver. The field variables, u_i and σ_{ij}, on the boundary ($y \in \Gamma$) or in the domain ($y \in \Omega$) of interest can be computed by using the discrete form of Eqs. (4) and (5), respectively. When x and y are coincident ($m = n$), the integrals are weakly singular in (11a), strongly singular in (11b,c) and hyper-singular in (11d) which constitute the principal diagonal terms of the system matrix $\mathbf{A}$. The strongly and hyper-singular boundary integrals should be evaluated in the sense of Cauchy principal values and Hadamard finite part values, respectively.

When $m \neq n$ or $y \in \Omega$, all integrals of kernel functions in (11) are regular and therefore easy to evaluate if the distances r are not too small. In this case, Brebbia et al. suggested that four-point Gauss quadrature would have sufficient accuracy for constant elements in the BEM [1]. However, as demonstrated by the successful use of the MFS [24] and illustrated in the present work, the use of *one-point* computing can achieve reasonable accuracy for integrals of kernel functions in (11), that is

$$G_{ij}^{nm} \approx u^*_{ij}(x^m, y^n)\Delta\Gamma_m, \tag{15a}$$
$$F_{ij}^{nm} \approx \tau^*_{ij}(x^m, y^n)\Delta\Gamma_m, \tag{15b}$$
$$T_{ij}^{nm} \approx n_k(y^n)u^*_{ikj}(x^m, y^n)\Delta\Gamma_m, \tag{15c}$$
$$H_{ij}^{nm} \approx n_k(y^n)\tau^*_{ikj}(x^m, y^n)\Delta\Gamma_m. \tag{15d}$$

As these integrals of kernel functions constitute most of the off-diagonal terms of the system matrix $\mathbf{A}$, much computing effort can be saved by the use of *one-point* computing, just like that performed in the MFS. The reasonable accuracy of integrals of kernel functions achieved using *one-point* computing is considered in effect to derive from the properties of the fundamental solutions themselves, just the opposite of the features of singularities, which can be named as the distance effect. For example, if a kernel has a singular order of $O(r^{-s})$, where s is an integer, then the decaying order of the integral values of this kernel function is also s with the increase of the distance r. In addition, the variation of the kernel function will have the order of $O(r^{-s-1})$, higher than that of the kernel itself, which means the decaying order of the variation of the kernel is more quickly than the kernel itself when performing integrations over the boundary elements if r is not too small. In other words, the values of the kernel function can be well represented by the value at the center point of the element, or of the territory in the present work, since the variation along the territory becomes negligibly small in this case. Therefore, if the distance r is not too small, *one-point* computing will achieve reasonable accuracy for integrals of kernel functions using the approximations (15), which constitute most of the off-diagonal terms of the system matrix, and the use of boundary elements is unnecessary.

4. The quadratic moving elements

As discussed above, much computing effort can be saved with the use of *one-point* computing for integrals of kernel functions, which constitute most of the off-diagonal terms of the system matrix in the BPM when the distance r is not too small. However, in cases that the distance between the collocation and field points is small, or they may even be coincident, these integrals of kernel functions constitute the principal and the sub-principal diagonal terms of the system matrix. The treatment of these integrals plays an important role in the computing accuracy of the BPM, although their quantity is fairly small compared with the off-diagonal terms. Noticed that the boundary variables are assumed to be constant on the support and the adjacent boundary nodes describe the local features of boundary such as position, curvature and direction, the problems can be resolved by introducing the constant moving elements [25] where the relevant adjacent nodes are organized tentatively.

With the introduction of the constant moving elements, not only the geometry of curved boundaries can be well described, but also all of the mature techniques in the BEM become applicable in the BPM [26–28]. The treatment of singularity and Gauss quadrature can be carried out over the integration interval if necessary. As the integration interval corresponds only to the territory of the collocation node using the constant moving elements, the algorithm can be changed over readily between the use of Gauss quadrature and *one-point* computing according to the distances between the source and the field points.

However, difficulties arise when computing the field variables in the domain very close to the boundary in the post processing as well as for the analysis of the thin-wall structures since the field variables are assumed to be constant over each of the territories or supports. This is partially owing to the near singularities of the kernels with various orders [29–31], partially owing to that a constant displacement field is locally stress-free at the vicinity near the boundary. Therefore the quadratic moving elements can be introduced tentatively as shown in Fig. 1. The local coordinate system of the three-point quadratic moving element for the nodes of interest, k, is shown in Fig. 1a, where ξ_k represent the dimensionless coordinates of nodes as follows:

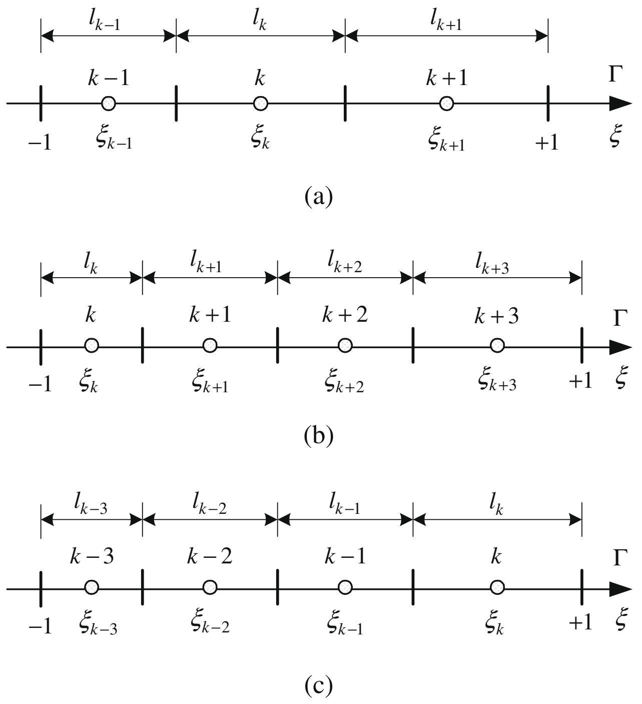

Fig. 1. The local coordinates of the 2D quadratic moving element constructed by (a) three-points, (b) and (c) four-points.

$$\xi_{k-1}=\frac{1}{l}l_{k-1}-1,\quad \xi_k=\frac{1}{l}(2l_{k-1}+l_k)-1,$$
$$\xi_{k+1}=\frac{1}{l}(2l_{k-1}+2l_k+l_{k+1})-1, \tag{16}$$

where l_k denote the territory length of each node and $l = l_{k-1} + l_k + l_{k+1}$. With the local coordinate system defined in Fig. 1a and the definitions (16), the shape functions for each node of the three-point quadratic moving element can be written as follows:

$$\phi^{k-1}=\frac{1}{(\xi_{k-1}-\xi_k)(\xi_{k-1}-\xi_{k+1})}(\xi-\xi_k)(\xi-\xi_{k+1}), \tag{17a}$$
$$\phi^{k}=\frac{1}{(\xi_{k}-\xi_{k-1})(\xi_{k}-\xi_{k+1})}(\xi-\xi_{k-1})(\xi-\xi_{k+1}), \tag{17b}$$
$$\phi^{k+1}=\frac{1}{(\xi_{k+1}-\xi_{k-1})(\xi_{k+1}-\xi_{k})}(\xi-\xi_{k-1})(\xi-\xi_{k}). \tag{17c}$$

The integration interval for the quadratic moving element is [−1, +1] corresponding to the territories of the three nodes. It can be seen that, compared with the constant moving elements, the quadratic moving element so defined belong to the equal-parametric elements which can also describe curved boundaries well. When the location of the source node is near a corner of the boundary, four-point moving elements can be constructed. Two local coordinate systems of four-point moving elements are shown in Fig. 1b and c, respectively. The purpose of using four adjacent nodes to construct an element is to maintain approximately the same accuracy as that of the three-point element, because parts of the integration interval [−1, +1] for the four-point elements around the nodes of interest are determined by extrapolation (Fig. 1b and c). In Fig. 1b, the dimensionless coordinates of nodes are defined as follows:

$$\xi_k=\frac{1}{l}l_k-1,\quad \xi_{k+1}=\frac{1}{l}(2l_k+l_{k+1})-1,\quad \xi_{k+2}=\frac{1}{l}(2l_k+2l_{k+1}+l_{k+2})-1,$$
$$\xi_{k+3}=\frac{1}{l}(2l_k+2l_{k+1}+2l_{k+2}+l_{k+3})-1, \tag{18}$$

where $l = l_k + l_{k+1} + l_{k+2} + l_{k+3}$. With the local coordinate system defined in Fig. 1b and (18), the shape functions for each node of the four-point moving element can be written as follows:

$$\phi^{k}=\frac{1}{(\xi_k-\xi_{k+1})(\xi_k-\xi_{k+2})(\xi_k-\xi_{k+3})}(\xi-\xi_{k+1})(\xi-\xi_{k+2})(\xi-\xi_{k+3}), \tag{19a}$$

$$\phi^{k+1}=\frac{1}{(\xi_{k+1}-\xi_k)(\xi_{k+1}-\xi_{k+2})(\xi_{k+1}-\xi_{k+3})}\times(\xi-\xi_k)(\xi-\xi_{k+2})(\xi-\xi_{k+3}), \tag{19b}$$

$$\phi^{k+2}=\frac{1}{(\xi_{k+2}-\xi_k)(\xi_{k+2}-\xi_{k+1})(\xi_{k+2}-\xi_{k+3})}\times(\xi-\xi_k)(\xi-\xi_{k+1})(\xi-\xi_{k+3}), \tag{19c}$$

$$\phi^{k+3}=\frac{1}{(\xi_{k+3}-\xi_k)(\xi_{k+3}-\xi_{k+1})(\xi_k-\xi_{k+2})}\times(\xi-\xi_k)(\xi-\xi_{k+1})(\xi-\xi_{k+2}). \tag{19d}$$

Similarly, in Fig. 1c, the dimensionless coordinates of nodes are defined as follows:

$$\xi_k=-\frac{1}{l}l_k+1,\quad \xi_{k-1}=-\frac{1}{l}(2l_k+l_{k-1})+1,$$
$$\xi_{k-2}=-\frac{1}{l}(2l_k+2l_{k-1}+l_{k-2})+1, \tag{20}$$
$$\xi_{k-3}=-\frac{1}{l}(2l_k+2l_{k-1}+2l_{k-2}+l_{k-3})+1,$$

where $l = l_{k-3} + l_{k-2} + l_{k-1} + l_k$. With the local coordinate system defined in Fig. 1c and (20), the shape functions for each node of another four-point moving element can be written as follows:

$$\phi^k = \frac{1}{(\xi_k - \xi_{k-1})(\xi_k - \xi_{k-2})(\xi_k - \xi_{k-3})} \times (\xi - \xi_{k-1})(\xi - \xi_{k-2})(\xi - \xi_{k-3}), \tag{21a}$$

$$\phi^{k-1} = \frac{1}{(\xi_{k-1} - \xi_k)(\xi_{k-1} - \xi_{k-2})(\xi_{k-1} - \xi_{k-3})} \times (\xi - \xi_k)(\xi - \xi_{k-2})(\xi - \xi_{k-3}), \tag{21b}$$

$$\phi^{k-2} = \frac{1}{(\xi_{k-2} - \xi_k)(\xi_{k-2} - \xi_{k-1})(\xi_{k-2} - \xi_{k-3})} \times (\xi - \xi_k)(\xi - \xi_{k-1})(\xi - \xi_{k-3}), \tag{21c}$$

$$\phi^{k-3} = \frac{1}{(\xi_{k-3} - \xi_k)(\xi_{k-3} - \xi_{k-1})(\xi_k - \xi_{k-2})} \times (\xi - \xi_k)(\xi - \xi_{k-1})(\xi - \xi_{k-2}), \tag{21d}$$

In a similar way, the surface moving elements can be constructed for the 3D cases along the two directions over the boundary surface as was done in the constant moving elements [25]. It needs to be pointed out that the characteristics of the last two elements in Fig. 1b and c are cubic. However, since the number of those elements is generally small in comparison with those quadratic moving elements in Fig. 1a used along smooth boundaries, the cubic moving elements are not specially dressed in the present paper since they are introduced only for balancing the accuracy with the quadratic moving elements. With the introduction of the quadratic moving elements, the distance transformation techniques can be employed to deal with these near singular integrals [29–31] without difficulty.

5. Numerical examples

In this section, the comparison of accuracy and efficiency is presented first between *one-point* computing and Gauss quadrature. With the comparison, the range or criterion can be determined to change appropriately between the two numerical algorithms during the computation of coefficients of the system matrix. Then some benchmark examples are presented in elasticity including thin-wall structures with both the CBIE and HBIE formulations using the computer program developed based on the BPM. The effect of the quadratic moving elements in the near singular case is shown in the numerical example.

5.1. Comparison between one-point computing and Gauss quadrature

Observation nodes x with territories and distances r_0 from the source point y are schematically shown in Fig. 2 for the comparison of differences between *one-point* computing and Gauss quadrature over a constant moving element. Supposing that K represents any one of the kernels, the integrals over the territories by the *one-point* computing method are computed by

$$I = K(x, y)l, \quad (2D), \tag{22a}$$
$$I = K(x, y)a, \quad (3D). \tag{22b}$$

As a control, the integrals over the territories are computed, respectively, by four-point Gauss quadrature in 2D and 4 × 4-point Gauss quadrature in 3D for comparison:

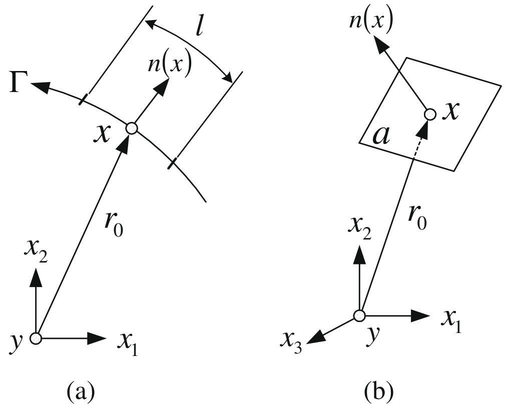

Fig. 2. Schematics in the 2D (a) and the 3D (b) for comparison between *one-point* computing and Gauss quadrature.

$$I = \int_{-1}^{+1} K[x(\xi), y]J(\xi)d\xi, \quad (2D), \tag{23a}$$
$$I = \int_{-1}^{+1}\int_{-1}^{+1} K[x(\xi,\eta), y]J(\xi,\eta)d\xi d\eta, \quad (3D). \tag{23b}$$

The 2D integral values of various kernels computed by *one-point* computing and Gauss quadrature as a function of relative distance, r_0/l, are compared in Fig. 3. It can be seen that the greater the relative distance, the smaller the integral values, a feature stemming from the properties of fundamental solutions, the distance effect. The corresponding relative errors of the 2D integral values are shown in Fig. 4. It can be seen that the greater the relative distance, the smaller the errors between the integral values by the two schemes, a feature also stemming from the properties of fundamental solutions, the distance effect. In the range of relative distances exceeding $r_0/l \geqslant 2$, the differences in the integral values are negligibly small between the two schemes, in which the integrals can be computed suitably by *one-point* computing. The use of Gauss quadrature is necessary only within the range of $r_0/l \leqslant 2$.

The 3D integral values of various kernels computed by *one-point* computing and Gauss quadrature as a function of relative distance, $r_0/a^{1/2}$, are compared in Fig. 5, and the corresponding relative errors are shown in Fig. 6. Similar conclusions can be drawn, that in the range of relative distance exceeding $r_0/a^{1/2} \geqslant 2$, the differences in the integral values are negligibly small between the two schemes, in which the integrals can also be computed satisfactorily by *one-point* computing. The use of Gauss quadrature is necessary

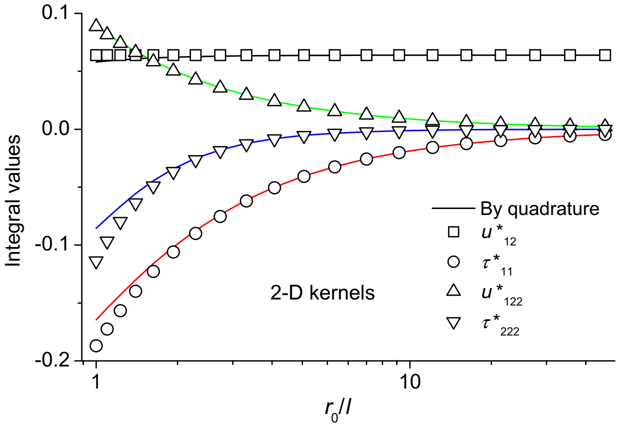

Fig. 3. Comparison of the 2D integral values computed between one-point computing and Gauss quadrature as a function of relative distance, r_0/l.

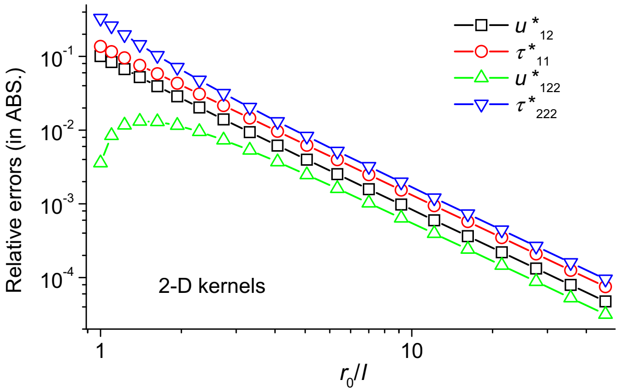

Fig. 4. Relative errors of the 2D integral values between one-point computing and Gauss quadrature as a function of relative distance, r_0/l.

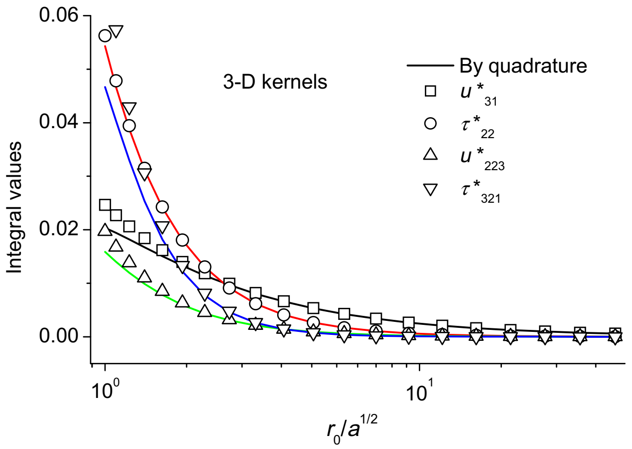

Fig. 5. Comparison of the 3D integral values between one-point computing and Gauss quadrature as a function of relative distance, $r_0/a^{1/2}$.

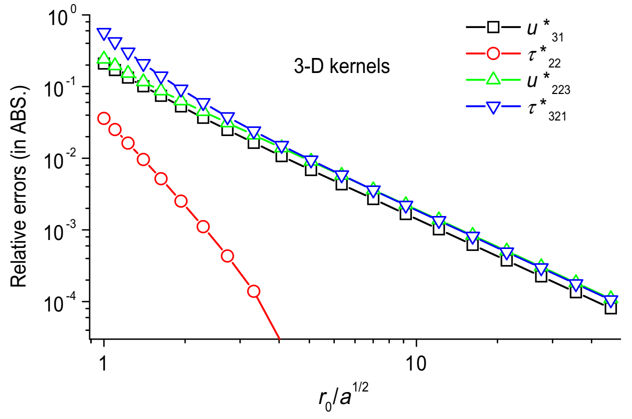

Fig. 6. Relative errors of the 3D integral values between one-point computing and Gauss quadrature as a function of relative distance, $r_0/a^{1/2}$.

only within the range of $r_0/a^{1/2} \leqslant 2$ in the 3D case. It is seen that the comparison in elasticity comes to the same conclusion with that in the potential theory [25] as expected. The use of the algorithms in the present work is summarized in the Table 1 as follows where the conditions for the use of one-point computing are more rigorously than the results shown in Figs. 3–6.

Table 1
The conditions for the algorithms used in the BPM.

Algorithm	Relative distance	Computing stages
One-point computing	>3	All stages
Constant moving elements	⩽3 and >0.5	All stages using a variable number of Gauss points
Constant moving elements	0	Forming system matrix Recovering boundary stresses
Quadratic moving elements	⩽0.5	All stages

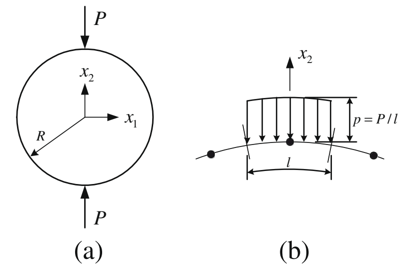

Fig. 7. (a) A circular disk with two equal and opposite forces $P = 1$ acting along the diameter. (b) The point force P treated as traction p distributed over a territory with length l.

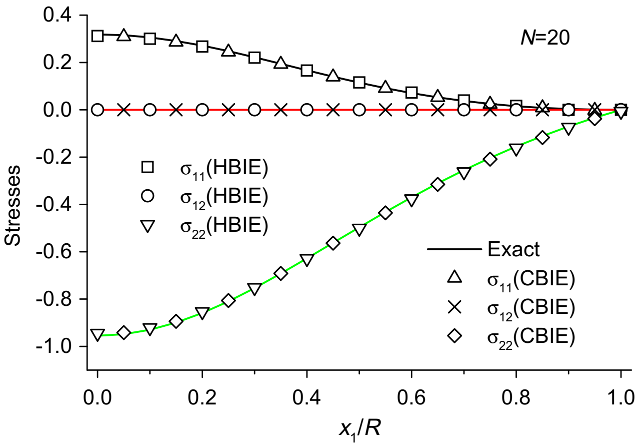

Fig. 8. Computed stresses in domain along $x_2 = 0$ in the circular disk with two equal and opposite forces acting along the vertical diameter.

5.2. Stresses in a circular disk

As shown in Fig. 7a, a circular disk with two equal and opposite forces $P = 1$ acting along the vertical diameter is considered. Because of symmetry, only a quarter of the disk is discretised by uniformly distributed nodes along the boundary and no node is required on the symmetrical axis as the program has the ability to deal with problems with geometrical symmetry. The point force P is treated as traction distributed over a territory with length l as shown in Fig. 7b. The computed results are compared with the analytical solutions [32]. Fig. 8 shows various stress components normalized by the point load computed in the domain along $x_2 = 0$ in the circular disk with a total node number $N = 20$. It can be seen that the computed results are in good agreement with the exact solutions. Fig. 9 gives various normalized stress components computed in the domain along $x_1 = 0$ in the circular disk with the same

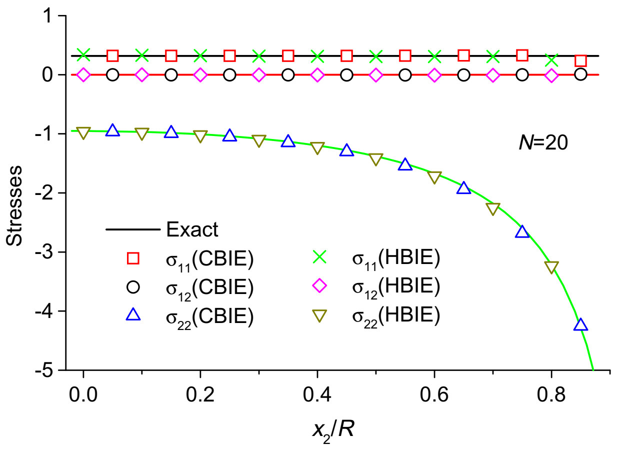

Fig. 9. Computed stresses in domain along $x_1 = 0$ in the circular disk with two equal and opposite forces acting along the vertical diameter.

total node number $N = 20$. It can be seen also that the computed results are in good agreement with the exact solutions except near the point load as it is approximated by the traction as shown in Fig. 7b. The root mean square errors of stresses computed both in the domain and on the boundary nodes of the circular disk are presented in Fig. 10 as a function of the total node number used, showing the convergence behavior of the BPM.

5.3. Infinite plate with an elliptical hole

Another example is an infinite plate with an elliptical hole under far-field uniform tension with $p = 1$, as shown in Fig. 11. a and b are the major and minor axes respectively of the ellipse. Because of symmetry, only a quarter of the hole is discretized along the boundary of the hole using a total node number $N = 20$. The coordinates of nodes are determined using the equations $x = a\text{-}sin(\theta)$ and $y = b\ sin(\theta)$ with equally divided angle θ so that the nodes are distributed uniformly only when $a = b$. The stress concentration factors at the vertex (point A, Fig. 11) of the elliptical hole in the infinite tension plate as a function of b/a are shown in Fig. 12. It can be seen again that the results computed with the BPM compare well with the exact solutions [32]. The computed stresses normalized by the far-field load in the domain along $x_2 = 0$ are presented in Fig. 13. The computed stresses at the places very close to the boundary just following the range in Fig. 13 are compared in Fig. 14 with the two algorithms where one constant and one quadratic moving element are used, respectively, around the source point. It can be seen from Fig. 14 that the results with the constant moving element are unacceptable since the field variables in the territory are assumed to be constant although the distance transformation techniques [29,30] are still employed in the computation. In comparison, the results with the quadratic moving element are shown in good agreement with the exact solutions [32].

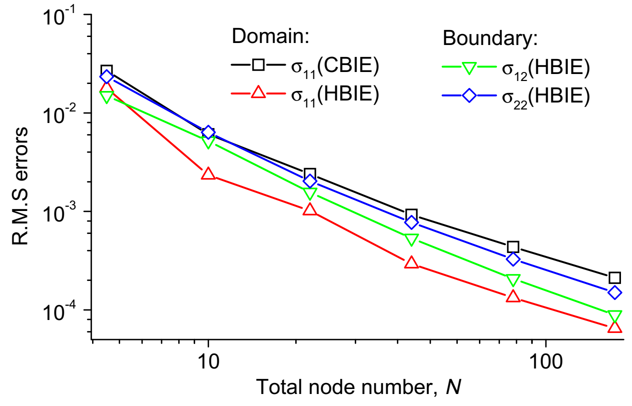

Fig. 10. The root mean square errors of stresses computed in the domain and on the boundary of the circular disk.

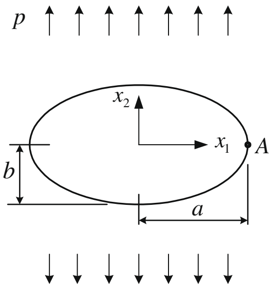

Fig. 11. An infinite plate with an elliptical hole under far-field uniform tension $p = 1$.

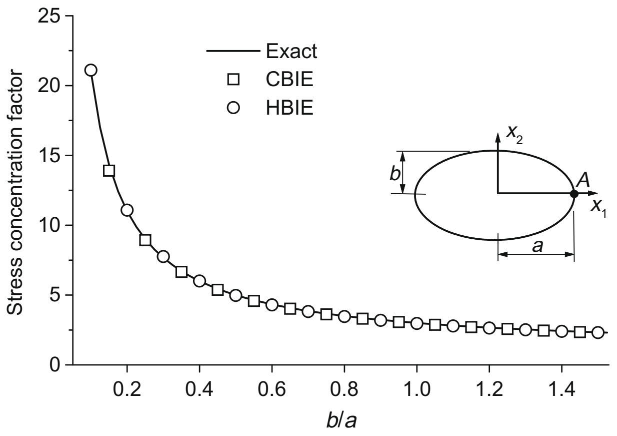

Fig. 12. Stress concentration factors at point A of elliptical hole in the infinite tension plate as a function of b/a.

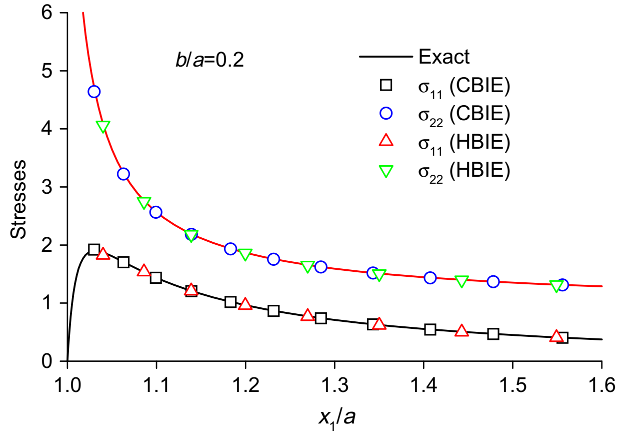

Fig. 13. Computed stresses in domain along $x_2 = 0$ in the infinite tension plate with an elliptical hole.

5.4. The thin-wall cylinder under inner pressure

The next example is the thin-wall cylinder under inner pressure using 96 nodes equally spaced along both the inner and outer

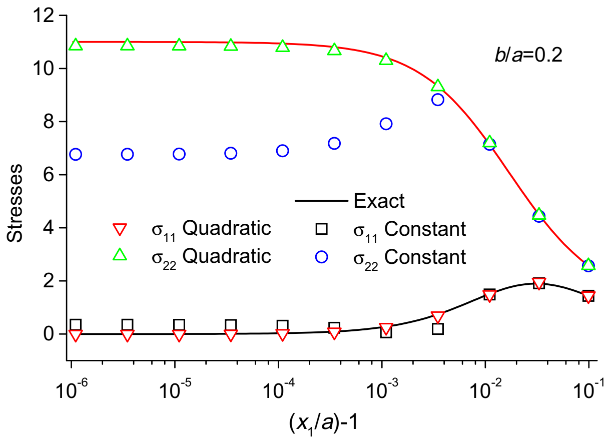

Fig. 14. Computed stresses in domain along $x_2 = 0$ very close to the boundary.

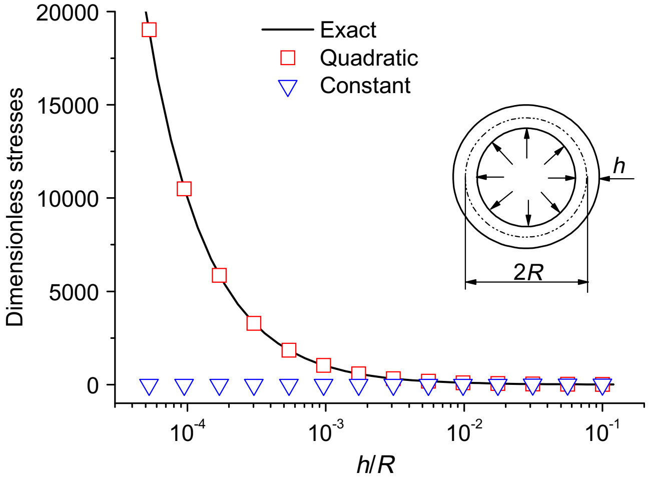

Fig. 15. Circumferential stresses of the thin-wall cylinder under inner pressure.

surfaces. For these problems, the quadratic moving elements should be used not only in the stage of post processing but also in stage of forming the system matrix. In the computation, the thickness h of the cylinder varies while the mean radius R is kept constant. The circumferential stresses at the mean radius normalized by the inner pressure are shown in Fig. 15. It can be seen from Fig. 15 that the results using constant moving elements have no meaning when h/R is small. However, the results using quadratic moving elements are in good agreement with the exact solution in a wide range of h/R. Fig. 16 presents the relative errors of the radial displacements and circumferential stresses at the mean radius, showing the effectiveness of the BPM.

5.5. A 3D thick cylinder under inner pressure

The last example is a 3D thick cylinder under inner pressure as shown in Fig. 17. In the analysis, the boundary of the cylinder is discretized by 640 boundary nodes over a quarter of the model owing to the geometrical symmetry. Fig. 18 gives the results of the surface stresses of the cylinder under inner pressure $p = 1$, showing that the computed results are in good agreement with the exact solution, verifying the effectiveness of the BPM for the 3D analysis of elasticity.

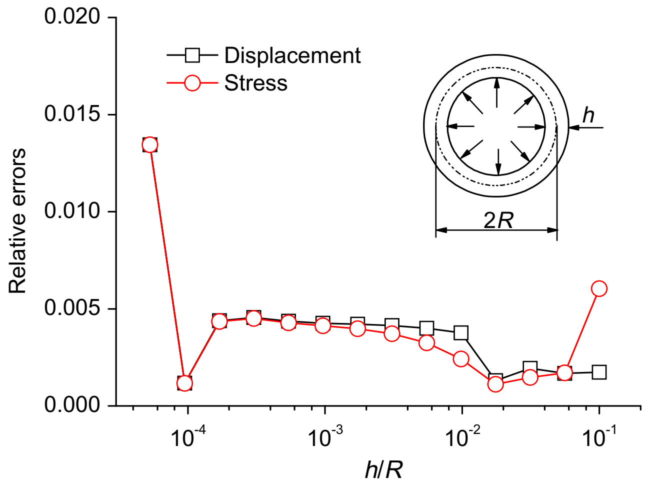

Fig. 16. Relative errors of radial displacements and circumferential stresses of the thin-wall cylinder.

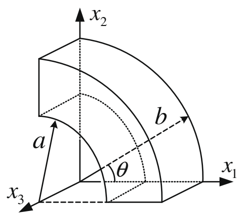

Fig. 17. The quarter of the thick cylinder for 3D elasticity.

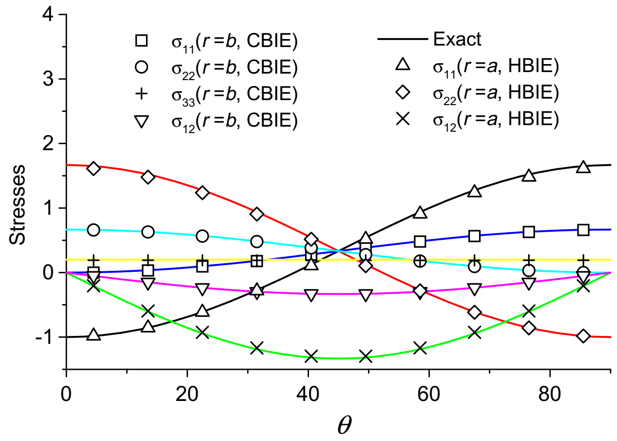

Fig. 18. The surface stresses of the thick cylinder under inner pressure $p = 1$.

6. Conclusions

In this study, based on the direct formulations of the BIE, the novel boundary-type meshless method, the BPM, is extended to the solution of linear elasticity problems. As the source points are located on the real boundary in the BPM, the main difficulty of coincidence of the source and observation points as well as the inconvenience of using fictitious boundary encountered in the MFS are removed. Comparing with the BEM, the use of the BPM can improve efficiency by both reducing the burden of data preparation and taking advantage of the *one-point* computing for most of the integrals of kernel functions while keeping a reasonable accuracy. With the introduction of the quadratic moving elements, the accuracy of the stresses near the boundaries is greatly improved in the post processing. The thin-wall structure can be analyzed with no difficulty. Numerical tests of the BPM are carried out by benchmark examples in the two- and three-dimensional elasticity. Good agreement is observed between the numerical and the exact solutions. It is considered also that the BPM would be more feasible to combine with the fast multipole technique [33] to solve large-scale problems to be carried out in the near future.

Acknowledgement

The work was supported by the National Natural Science Foundation of China (10972131/10772106).

References

[1] Brebbia CA, Telles JCF, Wrobel LC. Boundary element techniques—theory and applications in engineering. Berlin: Springer; 1984.

[2] Mukherjee S. Boundary element methods in creep and fracture. London: Applied Science Publishers; 1982.

[3] Nagarajan A, Lutz E, Mukherjee S. A novel boundary element method for linear elasticity with no numerical integration for two-dimensional and line integrals for three-dimensional problems. ASME J Appl Mech 1994;61:264–9.

[4] Nagarajan A, Mukherjee S, Lutz E. The boundary contour method for three-dimensional elasticity. ASME J Appl Mech 1996;63:273–86.

[5] Phan AV, Mukherjee S, Mayer JRR. The boundary contour method for two-dimensional linear elasticity with quadratic boundary elements. Computational Mechanics 1997;20:310–9.

[6] Mukherjee S, Mukherjee YX. The hypersingular boundary contour method for three-dimensional linear elasticity. ASME J Appl Mech 1998;65:300–9.

[7] Mukherjee YX, Mukherjee S. The boundary node method for potential problems. Int J Numer Methods Eng 1997;40:797–815.

[8] Chati MK, Mukherjee S, Paulino GH. The meshless hypersingular boundary node method for three-dimensional potential theory and linear elasticity problems. Eng Anal Boundary Elem 2001;25:639–53.

[9] Zhang JM, Tanaka M, Matsumoto T. Meshless analysis of potential problems in three dimensions with the hybrid boundary node method. Int J Numer Methods Eng 2004;59:1147–60.

[10] Liu GR, Gu YT. A point interpolation method for two-dimensional solid. Int J Numer Methods Eng 2001;50:937–51.

[11] Liu GR, Gu YT, Wang YY. A meshfree radial point interpolation method (RPIM) for three-dimensional solids. Computational Mechanics 2005;36:421–30.

[12] Liu GR, Gu YT. Boundary meshfree methods based on the boundary point interpolation methods. Eng Anal Boundary Elem 2004;28:475–87.

[13] Kita E, Kamiya N. Trefftz method: an overview. Adv Eng Softw 1995;24:3–12.

[14] Herrera I, Gourgeon H. Boundary methods, C-complete systems for Stokes problems. Comput Methods Appl Mech Eng 1982;30:225–41.

[15] Chen W, Tanaka M. A meshfree, integration-free and boundary-only RBF technique. Comput Math Appl 2002;43:379–91.

[16] Chen W, Hon YC. Numerical investigation on convergence of boundary knot method in the analysis of homogeneous Helmholtz, modified Helmholtz and convection–diffusion problems. Comput Methods Appl Mech Eng 2003;192:1859–75.

[17] Kita E, Kamiya N, Ikeda Y. An application of Trefftz method to the sensitivity analysis of two-dimensional potential problem. Int J Numer Methods Eng 1995;38:2209–24.

[18] Fairweather G, Karageorghis A. The method of fundamental solutions for elliptic boundary value problems. Adv Comput Math 1998;9:69–95.

[19] Golberg MA. The method of fundamental solution for Poisson's equation. Eng Anal Boundary Elem 1995;16:205–13.

[20] Fairweather G, Karageorghis A. The method of fundamental solution for elliptic boundary value problems. Adv Comput Math 1998;9:69–95.

[21] Berger JR, Karageorghis A. The method of fundamental solutions for heat conduction in layered materials. Int J Numer Methods Eng 1999;45:1681–94.

[22] Chen JT, Chen IL, Lee YT. Eigensolutions of multiply connected membranes using the method of fundamental solutions. Eng Anal Boundary Elem 2005;29:166–74.

[23] Young DL, Jane SJ, Fan CM, Murugesan K, Tsai CC. The method of fundamental solutions for 2D and 3D Stokes problems. J Comput Phys 2006;211:1–8.

[24] Young DL, Chen KH, Lee CW. Novel meshless method for solving the potential problems with arbitrary domain. J Comput Phys 2005;209:290–321.

[25] Ma H, Qin QH. Solving potential problems by a boundary-type meshless method—the boundary point method based on BIE. Eng Anal Boundary Elem 2007;31(9):749–61.

[26] Sladek N, Sladek J, Tanaka M. Regularization of hypersingular and nearly singular integrals in the potential theory and elasticity. Int J Numer Methods Eng 1993;36:1609–28.

[27] Guiggiani M, Casalini P. Direct computation of Cauchy principal value integrals in advanced boundary elements. Int J Numer Methods Eng 1987;24:1711–20.

[28] Guiggiani M, Gigante A. A general algorithm for multi-dimensional Cauchy principal value integrals in the boundary element method. ASME J Appl Mech 1990;57:906–15.

[29] Ma H, Kamiya N. Distance transformation for the numerical evaluation of near singular boundary integrals with various kernels in boundary element method. Eng Anal Boundary Elem 2002;26:329–39.

[30] Ma H, Kamiya N. A general algorithm for the numerical evaluation of nearly singular boundary integrals of various orders for two- and three-dimensional elasticity. Computational Mechanics 2002;29:277–88.

[31] Ma H, Kamiya N. Nearly singular approximations of CPV integrals with end- and corner-singularities for the numerical solution of hypersingular boundary integral equations. Eng Anal Boundary Elem 2003;27:625–37.

[32] Timoshenko SP, Goodier JN. Theory of elasticity. 3rd ed. Beijing: Tsinghua University Press; 2004.

[33] Greengard LF, Rokhlin V. A fast algorithm for particle simulations. J Comput Phys 1987;73(2):325–48.