

The meshless standard and hypersingular boundary node methods—applications to error estimation and adaptivity in three-dimensional problems

Mandar K. Chati^{1,†} Glaucio H. Paulino² and Subrata Mukherjee^{1,*}

¹*Department of Theoretical and Applied Mechanics, Cornell University, Ithaca, NY 14853, U.S.A.*

²*Department of Civil and Environmental Engineering, University of Illinois, 2209 Newmark Laboratory, 205 N. Mathews Av., Urbana, IL 61801, U.S.A.*

SUMMARY

The standard (singular) boundary node method (BNM) and the novel hypersingular boundary node method (HBNM) are employed for the usual and adaptive solutions of three-dimensional potential and elasticity problems. These methods couple boundary integral equations with moving least-squares interpolants while retaining the dimensionality advantage of the former and the meshless attribute of the latter. The ‘hypersingular residuals’, developed for error estimation in the mesh-based collocation boundary element method (BEM) and symmetric Galerkin BEM by Paulino *et al.*, are extended to the meshless BNM setting. A simple ‘*a posteriori*’ error estimation and an effective adaptive refinement procedure are presented. The implementation of all the techniques involved in this work are discussed, which includes aspects regarding parallel implementation of the BNM and HBNM codes. Several numerical examples are given and discussed in detail. Conclusions are inferred and relevant extensions of the methodology introduced in this work are provided. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: mesh-free methods; boundary node method (BNM); hypersingular boundary node method (HBNM); singular residuals; hypersingular residuals; error estimates; adaptivity; parallel computing

1. INTRODUCTION

The combination of boundary integral equations (BIEs), both in their standard and hypersingular forms, together with moving least-squares (MLS) interpolants, leads to a novel and effective environment for reliable computations in applied mechanics. The key features of

*Correspondence to: Subrata Mukherjee, Department of Theoretical and Applied Mechanics, Cornell University, 212 Kimball Hall, Ithaca, NY 14853, U.S.A.

†Current address: General Electric, Corporate Research and Development, Schenectady, NY 12301, U.S.A.

Contract/grant sponsor: Ford Motor Company.

Contract/grant sponsor: National Science Foundation; contract/grant number: CMS-9713008.

Received 10 December 1999

Revised 17 July 2000

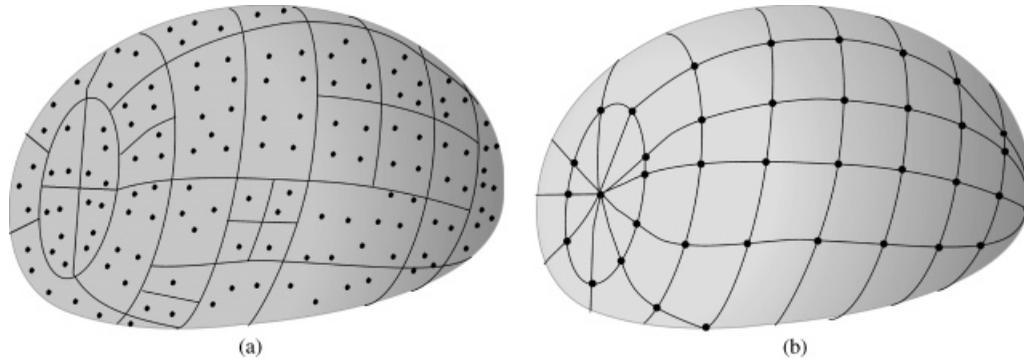


Figure 1. Comparison of BNM or HBNM and BEM input data structure: (a) BNM or HBNM—cells and collocation nodes; (b) BEM—mesh with elements and collocation nodes.

this environment are reduction of dimension, achieved by means of BIEs (as in the standard boundary element method—BEM), and the meshless attribute of MLS interpolants. The reliability of the simulations is achieved by means of self-adaptive techniques leading to refinement of

- cells,
- node density,
- regions of influence of the nodes.

It is worth mentioning that the cells are used just for integration, and pose no restriction on shape or compatibility. This feature makes meshless methods especially suited for self-adaptive techniques. The input data structure for solving a boundary value problem (BVP) involves nodes on the boundary and surface cells. The geometry/topology of the cells can be much simpler than the actual mesh required for conventional boundary elements in the sense that cells can be divided into smaller ones without affecting their neighbours—such is not the case with boundary (or finite) elements. To illustrate this point, Figure 1(a) shows the cell structure and collocation nodes in the meshless boundary node method (BNM) or hypersingular boundary node method (HBNM), and Figure 1(b) shows the surface mesh, i.e. elements and collocation nodes, necessary in the conventional boundary element method (BEM). The (conformal) BEM data structure requires that the collocation nodes be tightly coupled to the surface mesh. In the BEM, it is possible to collocate a BIE at an arbitrarily located collocation node—however, interpolants of primary variables are still related to the geometry of the elements. On the other hand, the BNM data structure is more flexible and allows input for stress analysis from a solid modeler in a natural fashion. This feature makes meshless methods especially suited for self-adaptive techniques. In particular, the h -version is explored and two self-adaptive strategies are presented: one is based on progressive refinement (iterative) and the other is based on a ONE-step refinement (non-iterative). Both adaptive procedures are guided by residual error estimates which are intrinsic to the nature of BIE-based meshless methods.

From a computational point of view, the present meshless method leads naturally to parallelism. Thus, a simple parallel implementation for the matrix assembly and residual computation is developed using the message passing interface (MPI) standard. It is shown that, even

with such simple parallel implementation, significant gains in wall-clock time, compared to serial implementation, are obtained. The development, and practical use of this computational environment, is addressed in this work.

This paper is organized as follows. A brief literature review is provided in Section 2. Next, Section 3 presents the MLS approximation, which includes a technique for evaluation of tangential derivatives on the boundary. Section 4 reviews the standard BIE employed in the traditional collocation-based BEM, and Section 5 its corresponding meshless version called the boundary node method (BNM). Section 6 presents the hypersingular BIE (HBIE) method and Section 7 the hypersingular BNM (HBNM) which is obtained from the corresponding HBIE. The concept of hypersingular and singular residuals in the meshless setting is explained in Section 8. Afterwards, Section 9 employs these residuals for *a posteriori* error estimation and for guiding the *h*-version of a self-adaptive refinement procedure (iterative). The parallel implementation of the BNM/HBNM and the residual computation is explained in Section 10. Numerical results for progressively adaptive solutions are given in Section 11. All the numerical results are for three-dimensional (3D) problems in potential theory and linear elasticity, and include some parallel computing solutions. A novel and alternative ONE-step adaptive technique based on the idea of multilevel cell refinement is developed in Section 12, and this heuristic idea is validated by means of numerical examples. Finally, some concluding remarks are made in Section 13.

2. RELATED WORK

The task of meshing a 3D object with complicated geometry can be arduous, time consuming and computationally expensive. Although significant progress has been made in 3D meshing algorithms (see Reference [1]), a considerable computational burden is associated with these algorithms. Conventional computational engines such as the finite difference method (FDM), finite element method (FEM), and BEM can be used, but often with difficulty, to solve problems involving changing domains such as large deformation or crack propagation. The main difficulty in these problems is the task of re-meshing a 3D object after large deformation or crack propagation. In recent years, novel computational algorithms have been proposed that circumvent some of the problems associated with 3D meshing. These methods have been collectively referred to as ‘Meshless’ methods.

Nayroles *et al.* [2] proposed a method which they call the diffuse element method (DEM). The main idea of their work is to replace the usual FEM interpolation by a ‘diffuse approximation’. Their strategy consists of using a least-squares approximation scheme to interpolate the field variables, which are called MLS interpolants. Nayroles *et al.* [2] have applied the DEM to two-dimensional (2D) problems in potential theory and linear elasticity.

Meshless methods proposed to date include the element-free Galerkin (EFG) method [3], the reproducing kernel particle method (RKPM) [4], *h-p* clouds [5, 6], the meshless local Petrov–Galerkin (MLPG) approach [7, 8], local boundary integral equation (LBIE) method [9], and the natural element method (NEM) [10]. The main idea in the EFG method is to use moving least-squares (MLS) interpolants to construct the trial functions used in the Galerkin weak form. A wide variety of problems have been solved using the EFG method. In the introductory paper by Belytschko *et al.* [3], the EFG method was applied to 2D problems in linear elasticity and heat conduction. Since then, the method has been applied

for example, to solve problems in elasto-plasticity [11] fracture mechanics [12], crack growth [13, 14], dynamic fracture [15–17], elasto-plastic fracture mechanics [18, 19], plate bending [20], thin shells [21] and sensitivity analysis and shape optimization [22]. A special issue of the journal *Computer Methods in Applied Mechanics and Engineering* contains review articles by Belytschko *et al.* [23] and Liu *et al.* [24] on meshless methods. Another source of information on the RKPM is an overview article by Liu *et al.* [25].

Recently, Mukherjee and Mukherjee [26] proposed the meshless method called BNM. As indicated above, this type of method involves a coupling between MLS interpolants and BIEs. The BNM has been used for solving 2D problems in potential theory [26] and linear elasticity [27], and for 3D problems in potential theory [28] and linear elasticity [29].

The present paper develops and employs the HBNM to solve boundary value problems and also for error estimation and adaptivity. Hypersingular boundary integral equations (HBIEs) have diverse important applications and are the subject of considerable current research (see References [30–33]). HBIEs, for example, have been employed for the evaluation of boundary stresses [34–36], in wave scattering (e.g. Reference [37]), in fracture mechanics (e.g. References [32, 38, 39]), in symmetric Galerkin boundary element formulations (e.g. References [40–42]), to obtain the hypersingular boundary contour method (e.g. References [43, 44]), and for adaptive analysis (e.g. References [45–48]).

Another area of major interest in this work is error estimation and adaptivity for meshless methods. Previous work on domain-based meshless methods include the articles by Chung and Belytschko [49] and Oden *et al.* [6], among others. *To the best of the authors knowledge, this is the first paper in the literature to address error estimation and adaptivity for boundary-based (e.g. BIE or HBIE-based) meshless methods.*

3. SURFACE APPROXIMANTS

A moving least-squares (MLS) approximation scheme, using curvilinear co-ordinates on the surface of a three-dimensional (3D) solid body, is suitable for the BNM. Such a scheme (see Reference [28] for problems in potential theory and Reference [29] for linear elasticity) is briefly described here and employed in the theoretical and numerical schemes.

3.1. Moving least-squares (MLS) approximants

It is assumed that, for 3D problems, the bounding surface ∂B of a solid body is the union of piecewise smooth segments called panels. On each panel, one defines surface curvilinear co-ordinates (s_1, s_2) . For problems in potential theory, let u be the unknown potential function and $\tau \equiv \partial u / \partial n$ (where \mathbf{n} is an unit outward normal to ∂B at a point on it). For 3D linear elasticity, let u denote a component of the displacement vector \mathbf{u} and τ be a component of the traction vector $\boldsymbol{\tau}$ on ∂B . One defines

$$u(\mathbf{s}) = \sum_{i=1}^m p_i(\mathbf{s} - \mathbf{s}^E) a_i = \mathbf{p}^T(\mathbf{s} - \mathbf{s}^E) \mathbf{a}, \quad \tau(\mathbf{s}) = \sum_{i=1}^m p_i(\mathbf{s} - \mathbf{s}^E) b_i = \mathbf{p}^T(\mathbf{s} - \mathbf{s}^E) \mathbf{b} \quad (1)$$

The monomials p_i (see below) are evaluated in local co-ordinates $(s_1 - s_1^E, s_2 - s_2^E)$ where (s_1^E, s_2^E) are the global co-ordinates of an evaluation point E . It is important to state here that a_i and b_i are not constants. Their functional dependencies are determined later. The name

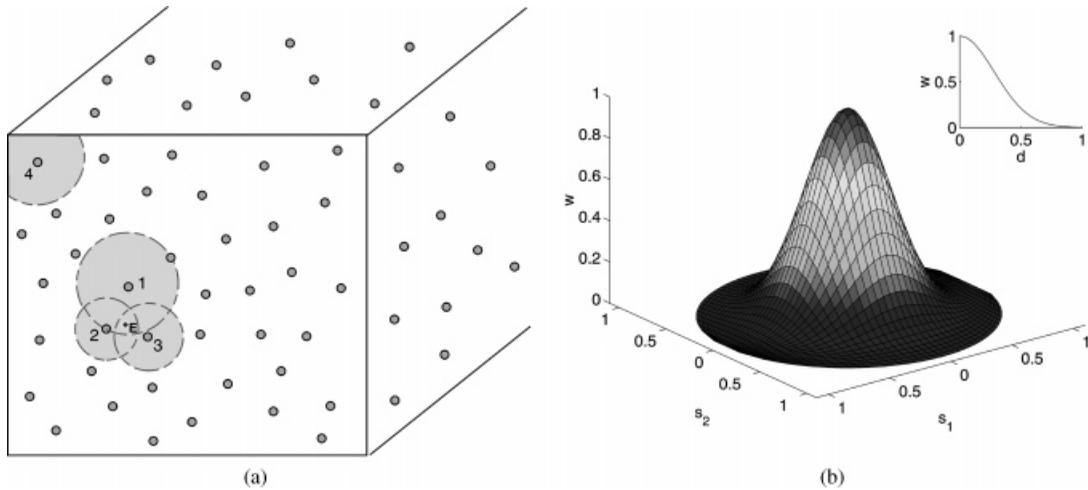


Figure 2. Domain of dependence and range of influence: (a) Nodes 1–3 lie within the domain of dependence of the evaluation point E . The ranges of influence of nodes 1–4 are shown as grey circles. The range of influence of node 4 is truncated at the edges of the body; (b) Gaussian weight function defined on the range of influence of a node

‘moving least squares’ arises from the fact that the quantities a_i and b_i are not constants. The integer m is the number of monomials in the basis used for u and τ . Quadratic interpolants, for example, are of the form

$$\mathbf{p}^T(\tilde{s}_1, \tilde{s}_2) = [1, \tilde{s}_1, \tilde{s}_2, \tilde{s}_1^2, \tilde{s}_2^2, \tilde{s}_1\tilde{s}_2], \quad m = 6, \quad \tilde{s}_i = s_i - s_i^E, \quad i = 1, 2 \quad (2)$$

The coefficients a_i and b_i are obtained by minimizing the weighted discrete L_2 norms

$$R_u = \sum_{l=1}^n w_l(d) [\mathbf{p}^T(\mathbf{s}^l - \mathbf{s}^E) \mathbf{a} - \hat{u}_l]^2, \quad R_\tau = \sum_{l=1}^n w_l(d) [\mathbf{p}^T(\mathbf{s}^l - \mathbf{s}^E) \mathbf{b} - \hat{\tau}_l]^2 \quad (3)$$

where the summation is carried out over the n boundary nodes for which the weight function $w_l(d) \neq 0$ (weight functions are defined in Section 3.3). The quantity $d = g(\mathbf{s}, \mathbf{s}^l)$ is the length of the geodesic on ∂B between \mathbf{s} and \mathbf{s}^l . These n nodes are said to be within the domain of dependence of a point \mathbf{s} (evaluation point E in Figure 2(a)). Also, $(s_1^l - s_1^E, s_2^l - s_2^E)$ are the local surface co-ordinates of the boundary nodes with respect to the evaluation point $\mathbf{s}^E = (s_1^E, s_2^E)$ and \hat{u}_l and $\hat{\tau}_l$ are the approximations to the nodal values u_l and τ_l . These equations above can be rewritten in compact form as

$$R_u = (\mathbf{P}(\mathbf{s}^l - \mathbf{s}^E) \mathbf{a} - \hat{\mathbf{u}})^T \mathbf{W}(\mathbf{s}, \mathbf{s}^l) (\mathbf{P}(\mathbf{s}^l - \mathbf{s}^E) \mathbf{a} - \hat{\mathbf{u}}) \quad (4)$$

$$R_\tau = (\mathbf{P}(\mathbf{s}^l - \mathbf{s}^E) \mathbf{b} - \hat{\boldsymbol{\tau}})^T \mathbf{W}(\mathbf{s}, \mathbf{s}^l) (\mathbf{P}(\mathbf{s}^l - \mathbf{s}^E) \mathbf{b} - \hat{\boldsymbol{\tau}}) \quad (5)$$

where $\hat{\mathbf{u}}^T = (\hat{u}_1, \hat{u}_2, \dots, \hat{u}_n)$, $\hat{\boldsymbol{\tau}}^T = (\hat{\tau}_1, \hat{\tau}_2, \dots, \hat{\tau}_n)$, $\mathbf{P}(\mathbf{s}^l)$ is an $n \times m$ matrix whose k th row is

$$[1, p_2(s_1^k, s_2^k), \dots, p_m(s_1^k, s_2^k)]$$

and $\mathbf{W}(\mathbf{s}, \mathbf{s}^l)$ is an $n \times n$ diagonal matrix with $w_{kk} = w_k(d)$ (no sum over k).

The stationarity of R_u and R_t , with respect to \mathbf{a} and \mathbf{b} , respectively, leads to the equations

$$\mathbf{a}(\mathbf{s}) = \mathbf{A}^{-1}(\mathbf{s})\mathbf{B}(\mathbf{s})\hat{\mathbf{u}}, \quad \mathbf{b}(\mathbf{s}) = \mathbf{A}^{-1}(\mathbf{s})\mathbf{B}(\mathbf{s})\hat{\boldsymbol{\tau}} \tag{6}$$

where

$$\mathbf{A}(\mathbf{s}) = \mathbf{P}^T(\mathbf{s}^I - \mathbf{s}^E)\mathbf{W}(\mathbf{s}, \mathbf{s}^I)\mathbf{P}(\mathbf{s}^I - \mathbf{s}^E), \quad \mathbf{B}(\mathbf{s}) = \mathbf{P}^T(\mathbf{s}^I - \mathbf{s}^E)\mathbf{W}(\mathbf{s}, \mathbf{s}^I) \tag{7}$$

It is noted from above that the coefficients a_i and b_i turn out to be functions of \mathbf{s} . Substituting Equations (6) into Equations (1), leads to

$$u(\mathbf{s}) = \sum_{I=1}^n \Phi_I(\mathbf{s})\hat{u}_I, \quad \tau(\mathbf{s}) = \sum_{I=1}^n \Phi_I(\mathbf{s})\hat{\tau}_I \tag{8}$$

where the approximating functions Φ_I are

$$\Phi_I(\mathbf{s}) = \sum_{j=1}^m p_j(\mathbf{s} - \mathbf{s}^E)(\mathbf{A}^{-1}\mathbf{B})_{jI}(\mathbf{s}) \tag{9}$$

As mentioned previously, $\hat{\mathbf{u}}$ and $\hat{\boldsymbol{\tau}}$ are approximations to the actual nodal values \mathbf{u} and $\boldsymbol{\tau}$. The two sets of values can be related by finding the number of nodes in the range of influence of each collocation node and then evaluating the shape function at each of these nodes. This procedure leads to

$$[\mathbf{H}]\{\hat{\mathbf{u}}_k\} = \{\mathbf{u}_k\}, \quad [\mathbf{H}]\{\hat{\boldsymbol{\tau}}_k\} = \{\boldsymbol{\tau}_k\}, \quad k = 1, 2, 3 \tag{10}$$

Equations (10) relate the nodal approximations of u and τ to their nodal values.

3.2. Surface derivatives

Surface derivatives of the potential (or displacement) field u are required for the HBIE. These are computed as follows. With

$$\mathbf{C} = \mathbf{A}^{-1}\mathbf{B}$$

Equations (8) and (9) give

$$u(\mathbf{s}) = \sum_{I=1}^n \sum_{j=1}^m p_j(\mathbf{s} - \mathbf{s}^E)\mathbf{C}_{jI}(\mathbf{s})\hat{u}_I \tag{11}$$

and the tangential derivatives of u can be written as

$$\frac{\partial u(\mathbf{s})}{\partial s_k} = \sum_{I=1}^n \sum_{j=1}^m \left[\frac{\partial p_j}{\partial s_k}(\mathbf{s} - \mathbf{s}^E)\mathbf{C}_{jI}(\mathbf{s}) + p_j(\mathbf{s} - \mathbf{s}^E)\frac{\partial \mathbf{C}_{jI}(\mathbf{s})}{\partial s_k} \right] \hat{u}_I, \quad k = 1, 2 \tag{12}$$

The derivatives of the monomials p_j can be easily computed. These are

$$\frac{\partial \mathbf{p}^T}{\partial s_1}(s_1 - s_1^E, s_2 - s_2^E) = [0, 1, 0, 2(s_1 - s_1^E), 0, (s_2 - s_2^E)] \tag{13}$$

$$\frac{\partial \mathbf{p}^T}{\partial s_2}(s_1 - s_1^E, s_2 - s_2^E) = [0, 0, 1, 0, 2(s_2 - s_2^E), (s_1 - s_1^E)] \tag{14}$$

After some simple algebra [50], the derivatives of the matrix \mathbf{C} with respect to s_k take the form

$$\frac{\partial \mathbf{C}(\mathbf{s})}{\partial s_k} = -\mathbf{A}^{-1}(\mathbf{s}) \frac{\partial \mathbf{B}(\mathbf{s})}{\partial s_k} \mathbf{P}(\mathbf{s}^I - \mathbf{s}^E) \mathbf{A}^{-1}(\mathbf{s}) \mathbf{B}(\mathbf{s}) + \mathbf{A}^{-1}(\mathbf{s}) \frac{\partial \mathbf{B}(\mathbf{s})}{\partial s_k}, \quad k = 1, 2 \quad (15)$$

with

$$\frac{\partial \mathbf{B}(\mathbf{s})}{\partial s_k} = \mathbf{P}^T(\mathbf{s}^I - \mathbf{s}^E) \frac{\partial \mathbf{W}(\mathbf{s}, \mathbf{s}^I)}{\partial s_k} \quad (16)$$

In deriving Equation (15), the following identity has been used:

$$\frac{\partial \mathbf{A}^{-1}(\mathbf{s})}{\partial s_k} = -\mathbf{A}^{-1}(\mathbf{s}) \frac{\partial \mathbf{A}(\mathbf{s})}{\partial s_k} \mathbf{A}^{-1}(\mathbf{s}), \quad k = 1, 2 \quad (17)$$

Tangential derivatives of the weight functions (described in Section 3.3) are easily computed [50]. The final form of the tangential derivatives of the potential (or displacement) u , at an evaluation point E , takes the form

$$\begin{aligned} \frac{\partial u}{\partial s_k}(\mathbf{s}^E) &= \sum_{I=1}^n \sum_{j=1}^m \left[\frac{\partial p_j}{\partial s_k}(0, 0) \mathbf{C}_{jI}(\mathbf{s}^E) \right] \hat{u}_I \\ &+ \sum_{I=1}^n \sum_{j=1}^m \left[p_j(0, 0) \left\{ \mathbf{A}^{-1}(\mathbf{s}^E) \frac{\partial \mathbf{B}}{\partial s_k}(\mathbf{s}^E) (\mathbf{I} - \mathbf{P}(\mathbf{s}^I - \mathbf{s}^E) \mathbf{A}^{-1}(\mathbf{s}^E) \mathbf{B}(\mathbf{s}^E)) \right\} \right] \hat{u}_I \quad (18) \end{aligned}$$

with $k = 1, 2$. In the above equation \mathbf{I} is the identity matrix.

One also needs the spatial gradient of the function u in order to solve the HBIE. For problems in potential theory, this is easily obtained from its tangential and normal derivatives, i.e. $\partial u / \partial s_k$ and $\partial u / \partial n$. For elasticity problems, however, one must also use Hooke's law at a point on the surface ∂B . Details of this procedure are given in Reference [51].

Equation (18) can be rewritten in compact form as:

$$\frac{\partial u}{\partial s_k}(\mathbf{s}^E) = \sum_{I=1}^n \Psi_I^{(k)}(\mathbf{s}^E) \hat{u}_I, \quad k = 1, 2 \quad (19)$$

where the approximating functions Ψ_I^k are:

$$\begin{aligned} \Psi_I^{(k)}(\mathbf{s}^E) &= \sum_{j=1}^m \left[\frac{\partial p_j}{\partial s_k}(0, 0) \mathbf{C}_{jI}(\mathbf{s}^E) \right] \\ &+ \sum_{j=1}^m \left[p_j(0, 0) \left\{ \mathbf{A}^{-1}(\mathbf{s}^E) \frac{\partial \mathbf{B}}{\partial s_k}(\mathbf{s}^E) (\mathbf{I} - \mathbf{P}(\mathbf{s}^I - \mathbf{s}^E) \mathbf{A}^{-1}(\mathbf{s}^E) \mathbf{B}(\mathbf{s}^E)) \right\} \right] \quad (20) \end{aligned}$$

3.3. Weight functions

The basic idea behind the choice of a weight function is that its value should decrease with distance from a node and that it should have compact support so that the region of influence of a node is of finite extent (Figure 2(b)). A possible choice is the Gaussian weight function

$$w_I(d) = \begin{cases} e^{-(d/d_I)^2} & \text{for } d \leq d_I \\ 0 & \text{for } d > d_I \end{cases} \quad (21)$$

Here $d = g(\mathbf{s}, \mathbf{s}^I)$ is the *minimum distance*, measured on the surface ∂B , (i.e. the geodesic) between a point \mathbf{s} and the collocation node I . In the research performed to date, the region of influence of a node has been truncated at the edge of a panel (Figure 2(a)) so that geodesics, and their derivatives (for use in Equation (21)), need only be computed on piecewise smooth surfaces. Finally, the quantities d_I determine the extent of the region of influence (the compact support) of node I . They can be made globally uniform, or can be adjusted such that approximately the same number of nodes get included in the region of influence of any given node I or in the domain of dependence of a given evaluation point E . Such ideas have been successfully implemented in References [28, 29].

4. BOUNDARY INTEGRAL EQUATIONS

Particular instances of the standard (singular) BIEs for potential theory and linear elasticity are given below.

4.1. Potential theory

The well-known regularized BIE for 3D problems in potential theory is (see, for example, Reference [52])

$$0 = \int_{\partial B} [G(P, Q)\tau(Q) - F(P, Q)(u(Q) - u(P))] dS_Q \quad (22)$$

where, as mentioned before, u is the potential, $\tau = \partial u / \partial n$ is the flux, and the well-known kernels for 3D problems are

$$G(P, Q) = \frac{1}{4\pi r(P, Q)}, \quad F(P, Q) = \frac{\partial G(P, Q)}{\partial n_Q} \quad (23)$$

Here, r is the Euclidean distance between the source point P and field point Q and n_Q is the unit normal to ∂B at a field point Q .

4.2. Linear elasticity

For 3D linear elasticity, the standard boundary integral equation, in regularized form, and in the absence of body forces, can be written as (see Reference [53])

$$0 = \int_{\partial B} [U_{ik}(P, Q)\tau_k(Q) - T_{ik}(P, Q)(u_k(Q) - u_k(P))] dS_Q \quad (24)$$

where u_k and τ_k are the components of the displacement and traction, respectively, and the well-known Kelvin kernels are

$$U_{ik} = \frac{1}{16\pi(1-\nu)Gr} [(3-4\nu)\delta_{ik} + r_{,i}r_{,k}] \tag{25}$$

$$T_{ik} = \frac{-1}{8\pi(1-\nu)r^2} \left[\{(1-2\nu)\delta_{ik} + 3r_{,i}r_{,k}\} \frac{\partial r}{\partial n} - (1-2\nu)(r_{,i}n_k - r_{,k}n_i) \right] \tag{26}$$

In the above, n_i are the components of the unit normal at the field point Q , G is the shear modulus, ν is the Poisson ratio and δ_{ij} denotes the Kronecker delta. A comma denotes a derivative with respect to a field point, i.e.

$$r_{,i} = \partial r / \partial y_i = (y_i(Q) - y_i(P)) / r \tag{27}$$

5. BOUNDARY NODE METHOD

The MLS interpolants derived in Section 3 are used to approximate u and τ on the boundary ∂B . In order to carry out the integrations, the bounding surface is discretized into cells. A variety of shape functions have been used in this work in order to interpolate the geometry. In particular, the bilinear (Q4) element and quadratic (T6) triangle have been used. These ‘geometric’ shape functions can be found in any standard text on the FEM (see References [54, 55]).

5.1. Potential theory

Substituting the expressions for u and τ from Equation (8) into Equation (22), and dividing ∂B into N_c cells, one gets the discretized form of the BIE for potential problems as follows:

$$0 = \sum_{k=1}^{N_c} \int_{\partial B_k} \left[G(P, Q) \sum_{I=1}^{N_Q} \Phi_I(Q) \hat{\tau}_I - F(P, Q) \left\{ \sum_{I=1}^{N_Q} \Phi_I(Q) \hat{u}_I - \sum_{I=1}^{N_P} \Phi_I(P) \hat{u}_I \right\} \right] dS_Q \tag{28}$$

where $\Phi_I(P)$ and $\Phi_I(Q)$ are the contributions from the I th node to the collocation point P and field point Q , respectively. Also, N_Q nodes are situated in the domain of dependence of the field point Q and N_P nodes are situated in the domain of dependence of the source point P .

5.2. Linear elasticity

The BNM equation for elasticity is obtained by substituting the expressions for u_k and τ_k from Equation (8) into Equation (24), leading to

$$0 = \sum_{m=1}^{N_c} \int_{\partial B_m} \left[U_{ik}(P, Q) \sum_{I=1}^{N_Q} \Phi_I(Q) \hat{\tau}_{kl} - T_{ik}(P, Q) \left\{ \sum_{I=1}^{N_Q} \Phi_I(Q) \hat{u}_{kl} - \sum_{I=1}^{N_P} \Phi_I(P) \hat{u}_{kl} \right\} \right] dS_Q \tag{29}$$

5.3. Discretization

In order to evaluate the non-singular integrals in Equations (28) and (29) over (possibly curved) triangular or rectangular surface cells, 7 point and 3×3 Gauss quadrature are used, respectively. However, as $Q \rightarrow P$ the kernels G and U_{ik} become weakly singular and the kernels F and T_{ik} become strongly singular. As shown in Equations (28) and (29), the strongly singular integrands are regularized by using rigid body modes and the regularized versions are weakly singular. Finally, special integration techniques are used to evaluate the resulting weakly singular integrals in Equations (28) and (29) [28, 56].

The final discretized version of either Equation (28) or Equation (29) has the form

$$[\mathbf{A}_{(\hat{u})}]\{\hat{\mathbf{u}}\} + [\mathbf{A}_{(\hat{\tau})}]\{\hat{\boldsymbol{\tau}}\} = \{\mathbf{0}\} \quad (30)$$

With respect to elasticity theory, the count for the number of equations and unknowns follows. For N_B nodes on the bounding surface, there are a total of $12N_B$ quantities on the boundary, i.e. $3N_B$ values for each of u_i and its nodal approximation \hat{u}_i , and similarly for τ_i . For a well-posed problem, values of either u_i or τ_i are known at each node on the boundary, so $3N_B$ nodal values are given. Therefore, $9N_B$ equations are needed to solve for the $9N_B$ remaining unknowns. Equation (30) consists of $3N_B$ equations and Equations (10) consist of $3N_B$ equations each. Thus, a well-posed boundary value problem can be solved using Equation (30), in combination with Equations (10). An analogous count of equations and unknowns applies to Equation (28) for potential theory.

6. HYPERSINGULAR BOUNDARY INTEGRAL EQUATIONS

Continuing the basic development of Section 4, we present here the HBIEs for potential theory and linear elasticity.

6.1. Potential theory

The HBIE is obtained upon differentiation of the primary BIE at an internal source point with respect to the co-ordinates of that source point. Due to differentiation, the kernels in the HBIE become strongly singular and hypersingular, respectively, and appropriate regularization procedures need to be employed in order to use the HBIEs for carrying out meaningful computations. The fully regularized HBIE for the Laplace's equation, at a regular point on ∂B (where it is locally smooth) can be written as (see Reference [30]),

$$\begin{aligned} 0 = & \int_{\partial B} \frac{\partial G(P, Q)}{\partial x_m(P)} [\tau(Q) - \tau(P)] dS_Q - u_{,k}(P) \int_{\partial B} \frac{\partial G(P, Q)}{\partial x_m(P)} (n_k(Q) - n_k(P)) dS_Q \\ & - \int_{\partial B} \frac{\partial F(P, Q)}{\partial x_m(P)} [u(Q) - u(P) - u_{,k}(P)(x_k(Q) - x_k(P))] dS_Q \end{aligned} \quad (31)$$

Carrying out the inner product of Equation (31) with the source point normal $\mathbf{n}(P)$, one obtains

$$0 = \int_{\partial B} \frac{\partial G(P, Q)}{\partial n(P)} [\tau(Q) - \tau(P)] dS_Q - u_{,k}(P) \int_{\partial B} \frac{\partial G(P, Q)}{\partial n(P)} (n_k(Q) - n_k(P)) dS_Q - \int_{\partial B} \frac{\partial F(P, Q)}{\partial n(P)} [u(Q) - u(P) - u_{,k}(P)(x_k(Q) - x_k(P))] dS_Q \tag{32}$$

The gradient of the potential function is required in the HBIEs (31) and (32). For potential problems, the gradient (at a regular boundary point) can be written as

$$\nabla u = \tau \hat{\mathbf{n}} + \frac{\partial u}{\partial s_1} \hat{\mathbf{t}}_1 + \frac{\partial u}{\partial s_2} \hat{\mathbf{t}}_2 \tag{33}$$

where $\tau = \partial u / \partial n$ is the flux, $\hat{\mathbf{n}}$ is the unit normal, $\hat{\mathbf{t}}_1, \hat{\mathbf{t}}_2$ are the appropriately chosen unit vectors in two tangential directions on the surface of the body, and $\partial u / \partial s_i, i = 1, 2$ are the tangential derivatives of u (along \mathbf{t}_1 and \mathbf{t}_2) on the surface of the body.

6.2. Linear elasticity

Similarly, the fully regularized HBIE for linear elasticity can be written as (see Reference [57])

$$0 = \int_{\partial B} D_{ijk}(P, Q)(\tau_k(Q) - \tau_k(P)) dS_Q - \sigma_{km}(P) \int_{\partial B} D_{ijk}(P, Q)(n_m(Q) - n_m(P)) dS_Q - \int_{\partial B} S_{ijk}(P, Q)[u(Q) - u(P) - u_{,m}(P)(y_m(Q) - y_m(P))] dS_Q \tag{34}$$

where the (strongly singular) kernel D_{ijk} and (hypersingular) kernel S_{ijk} are

$$D_{ijk} = \frac{1}{8\pi(1-\nu)r^2} [(1-2\nu)(\delta_{ki}r_{,j} + \delta_{kj}r_{,i} - \delta_{ij}r_{,k}) + 3r_{,i}r_{,j}r_{,k}] \tag{35}$$

$$S_{ijk} = \frac{G}{4\pi(1-\nu)r^3} \left[3 \frac{\partial r}{\partial n} [(1-2\nu)\delta_{ij}r_{,k} + \nu(\delta_{ik}r_{,j} + \delta_{jk}r_{,i}) - 5r_{,i}r_{,j}r_{,k}] \right] + \frac{G}{4\pi(1-\nu)r^3} [3\nu(n_i r_{,j} r_{,k} + n_j r_{,i} r_{,k}) + (1-2\nu) \times (3n_k r_{,i} r_{,j} + n_j \delta_{ik} + n_i \delta_{jk}) - (1-4\nu)n_k \delta_{ij}] \tag{36}$$

Again, taking the inner product of Equation (34) with the source normal, one gets the equation

$$0 = \int_{\partial B} D_{ijk}(P, Q)n_j(P)(\tau_k(Q) - \tau_k(P)) dS_Q - \sigma_{km}(P) \int_{\partial B} D_{ijk}(P, Q)n_j(P)(n_m(Q) - n_m(P)) dS_Q$$

$$-\int_{\partial B} S_{ijk}(P, Q)n_j(P)[u_k(Q) - u_k(P) - u_{k,m}(P)(y_m(Q) - y_m(P))]dS_Q \tag{37}$$

The procedure for obtaining the displacement gradients $u_{k,m}$, that are required in Equations (34) and (37), is described in Reference [51]. The stress components σ_{ij} can be easily obtained from the displacement gradients using Hooke’s law.

7. HYPERSINGULAR BOUNDARY NODE METHOD

Continuing the early development for the BNM in Section 5, we present here the derivation for the HBNM for potential theory and linear elasticity.

7.1. Potential theory

Using the interpolation functions for τ (Equation (8)₂) and for the tangential derivatives of u (Equation (19)), one can obtain the discretized forms of the potential gradient (see Equation (33)) and the HBIEs (31) and (32) as follows:

$$\nabla u = \sum_{I=1}^n \Phi_I \hat{\tau}_I \hat{\mathbf{n}} + \sum_{I=1}^n \Psi_I^{(1)} \hat{u}_I \hat{\mathbf{t}}_1 + \sum_{I=1}^n \Psi_I^{(2)} \hat{u}_I \hat{\mathbf{t}}_2 \tag{38}$$

$$\begin{aligned} 0 = & \sum_{i=1}^{N_c} \int_{\partial B_i} \frac{\partial G(P, Q)}{\partial x_m(P)} \left[\sum_{I=1}^{N_Q} \Phi_I(Q) \hat{\tau}_I - \sum_{I=1}^{N_P} \Phi_I(P) \hat{\tau}_I \right] \\ & - u_{,k}(P) \int_{\partial B_i} \frac{\partial G(P, Q)}{\partial x_m(P)} (n_k(Q) - n_k(P)) dS_Q \\ & - \int_{\partial B_i} \frac{\partial F(P, Q)}{\partial x_m(P)} \left[\sum_{I=1}^{N_Q} \Phi_I(Q) \hat{u}_I - \sum_{I=1}^{N_P} \Phi_I(P) \hat{u}_I - u_{,k}(P)(x_k(Q) - x_k(P)) \right] dS_Q \end{aligned} \tag{39}$$

$$\begin{aligned} 0 = & \sum_{i=1}^{N_c} \int_{\partial B_i} \frac{\partial G(P, Q)}{\partial n(P)} \left[\sum_{I=1}^{N_Q} \Phi_I(Q) \hat{\tau}_I - \sum_{I=1}^{N_P} \Phi_I(P) \hat{\tau}_I \right] \\ & - u_{,k}(P) \int_{\partial B_i} \frac{\partial G(P, Q)}{\partial n(P)} (n_k(Q) - n_k(P)) dS_Q \\ & - \int_{\partial B_i} \frac{\partial F(P, Q)}{\partial n(P)} \left[\sum_{I=1}^{N_Q} \Phi_I(Q) \hat{u}_I - \sum_{I=1}^{N_P} \Phi_I(P) \hat{u}_I - u_{,k}(P)(x_k(Q) - x_k(P)) \right] dS_Q \end{aligned} \tag{40}$$

respectively, where $\Phi_I(P)$ and $\Phi_I(Q)$ are the contributions from the I th node to the collocation point P and field point Q , respectively, with N_P and N_Q nodes in their respective domains of dependence. Of course, the gradient of u from Equation (38) is used in Equations (39) and (40).

7.2. Linear elasticity

As mentioned before, the procedure for obtaining $u_{k,m}$ in the elasticity equations (34) and (37), from the tangential derivatives and tractions, is described in Reference [51]. Once this is done, discretized versions of Equations (34) and (37) are readily obtained as:

$$\begin{aligned}
 0 = & \sum_{l=1}^{N_c} \int_{\partial B_l} D_{ijk}(P, Q) \left[\sum_{I=1}^{N_Q} \Phi_I(Q) \hat{\tau}_{kl} - \sum_{I=1}^{N_P} \Phi_I(P) \hat{\tau}_{kl} \right] dS_Q \\
 & - \sigma_{km}(P) \int_{\partial B_l} D_{ijk}(P, Q) (n_m(Q) - n_m(P)) dS_Q \\
 & - \int_{\partial B_l} S_{ijk}(P, Q) \left[\sum_{I=1}^{N_Q} \Phi_I(Q) \hat{u}_{kl} - \sum_{I=1}^{N_P} \Phi_I(P) \hat{u}_{kl} - u_{k,m}(P) (y_m(Q) - y_m(P)) \right] dS_Q \quad (41)
 \end{aligned}$$

and

$$\begin{aligned}
 0 = & \sum_{l=1}^{N_c} \int_{\partial B_l} D_{ijk}(P, Q) n_j(P) \left[\sum_{I=1}^{N_Q} \Phi_I(Q) \hat{\tau}_{kl} - \sum_{I=1}^{N_P} \Phi_I(P) \hat{\tau}_{kl} \right] dS_Q \\
 & - \sigma_{km}(P) \int_{\partial B_l} D_{ijk}(P, Q) n_j(P) (n_m(Q) - n_m(P)) dS_Q - \int_{\partial B_l} S_{ijk}(P, Q) n_j(P) \\
 & \times \left[\sum_{I=1}^{N_Q} \Phi_I(Q) \hat{u}_{kl} - \sum_{I=1}^{N_P} \Phi_I(P) \hat{u}_{kl} - u_{k,m}(P) (y_m(Q) - y_m(P)) \right] dS_Q \quad (42)
 \end{aligned}$$

Remark 1. Please note that Equations (39) and (40) are the HBNM equations for potential theory, and Equations (41) and (42) are the HBNM equations for linear elasticity. Equation (39) is used to obtain the hypersingular residual in the gradient of the potential, and Equation (28) or (40) is used to solve boundary value problems in potential theory. Similarly, Equation (41) is used to obtain the hypersingular residual in the stress and Equation (29) or (42) is used to solve boundary value problems in linear elasticity.

7.3. Discretization

The procedure followed for discretization of Equations (40) and (42) is quite analogous to the BNM case described before in Section 5.3. These equations are fully regularized and contain either non-singular or weakly singular integrands. Non-singular integrals are evaluated using the usual Gauss quadrature over surface cells, while the weakly singular integrals are evaluated using the procedure outlined in References [28, 56]. The discretized version of either Equation (40) or (42) has the generic form shown in Equation (30).

Numerical results from the BNM, for 3D potential theory and linear elasticity, are available in References [28, 29], respectively, while corresponding numerical results from the HBNM are available in Reference [51].

8. HYPERSINGULAR AND SINGULAR RESIDUALS

The idea of using hypersingular or singular residuals, to obtain local error estimates in the BEM, was first proposed by Paulino [33] and Paulino *et al.* [45]. This concept has been applied to the collocation BEM (see References [33, 45, 47, 48]), to the boundary contour method (BCM) (Mukherjee and Mukherjee) [58], and to the symmetric-Galerkin BEM (see Reference [46]). In this work, it is extended to the BNM setting. The main idea is as follows.

8.1. The hypersingular residual

8.1.1. *Potential theory.* Let the BIE (Equation (28)) for potential theory be written in operator form as

$$\mathcal{L}_{\text{BNM}}(u, \tau) = 0 \quad (43)$$

and its *numerical solution* be (u^*, τ^*) . Also, the HBIE (Equation (39)) is written in operator form as:

$$\mathcal{L}_{\text{HBNM}}(u, \tau) = 0 \quad (44)$$

The hypersingular residual in the potential gradient $u_{,j}$ is defined as,

$$r_j \equiv \text{residual}(u_{,j}) = \mathcal{L}_{\text{HBNM}}(u^*, \tau^*) \quad (45)$$

and is calculated from Equation (39).

8.1.2. *Linear elasticity.* Similarly, for elasticity problems, the BIE (Equation (29)) can be written in operator form as

$$\mathcal{L}_{\text{BNM}}(u_k, \tau_k) = 0, \quad k = 1, 2, 3 \quad (46)$$

with the *numerical solution* (u_k^*, τ_k^*) . Also, the HBIE (Equation (41)) is written in operator form as

$$\mathcal{L}_{\text{HBNM}}(u_k, t_k) = 0, \quad k = 1, 2, 3 \quad (47)$$

This time, the stress residual is defined from the stress HBIE (Equation (41)) as

$$r_{ij} \equiv \text{residual}(\sigma_{ij}) = \mathcal{L}_{\text{HBNM}}(u_k^*, t_k^*), \quad k = 1, 2, 3 \quad (48)$$

This idea is illustrated in Figure 3(a).

Remark 2. It has been proved by Menon *et al.* [48] that, under certain conditions, real positive constants c_1 and c_2 exist such that

$$c_1 r \leq \varepsilon \leq c_2 r \quad (49)$$

where r is a scalar measure of a residual (see Section 9 on ‘error estimation and adaptive strategy’) and ε is a scalar measure of the exact local error. Thus, the residual is expected to provide an estimate of the local error on a boundary element. It should be mentioned here that the definitions of the residuals used in Reference [48] are analogous to, but different in detail from, the ones proposed in this paper.

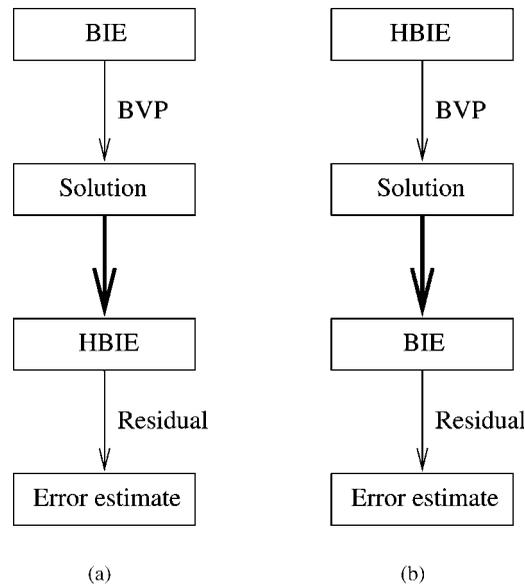


Figure 3. Interchange of BIE and HBIE: (a) hypersingular residual; (b) singular residual.

8.2. The singular residual

The argument for using the residuals as error estimates is symmetric (see References [33, 45]). Therefore, one can reverse the above procedure to define singular residuals by first solving the HBIE and then iterating with the BIE.

8.2.1. *Potential theory.* In this case, for potential theory, one gets from Equation (40):

$$\mathcal{L}_{\text{HBNM}}(u^o, \tau^o) = 0 \quad (50)$$

and from Equation (28)

$$r \equiv \text{residual}(u) = |\mathcal{L}_{\text{BNM}}(u^o, \tau^o)| \quad (51)$$

8.2.2. *Linear elasticity.* Similarly, for elasticity, one has from Equation (42)

$$\mathcal{L}_{\text{HBNM}}(u_k^o, \tau_k^o) = 0, \quad k = 1, 2, 3 \quad (52)$$

and from Equation (29)

$$r_i \equiv \text{residual}(u_i) = \mathcal{L}_{\text{BNM}}(u_k^o, \tau_k^o), \quad k = 1, 2, 3 \quad (53)$$

This idea is illustrated in Figure 3(b).

Remark 3. The above formulation for singular and hypersingular residuals is a generalization of the earlier work by Menon *et al.* [48] in the sense that Dirichlet, Neumann and mixed problems require separate prescriptions in Reference [48], while the current work presents a unified residual formulation.

9. ERROR ESTIMATION AND ADAPTIVE STRATEGY

There are similarities between adaptive techniques (e.g. *h*-version) for mesh-based methods (see References [59, 60]) and meshless methods, however, the latter set of methods provides substantially more flexibility in the (re-)discretization process than the former ones.

The *h*-version iterative self-adaptive procedure employed in this work is presented in the flowchart of Figure 4. The goal is to efficiently develop a final cell configuration which leads to a reliable numerical solution, in as simple a manner as possible.

9.1. Local residuals and errors—hypersingular residual approach

Potential theory: For potential theory problems (see Equation (45)),

$$r_j = \text{residual}(u_{,j}) \tag{54}$$

A scalar residual measure is defined as

$$r = r_j r_j \tag{55}$$

The exact local error in the gradient, $u_{,j}$, is defined as:

$$\varepsilon_j = u_{,j}^{(\text{exact})} - u_{,j}^{(\text{numerical})} \tag{56}$$

and the corresponding scalar measure is defined as

$$\varepsilon = \varepsilon_j \varepsilon_j \tag{57}$$

Equations (55) and (57) are used to calculate the hypersingular residual and exact error, respectively, in the gradient $u_{,j}$, at each node, for problems in potential theory.

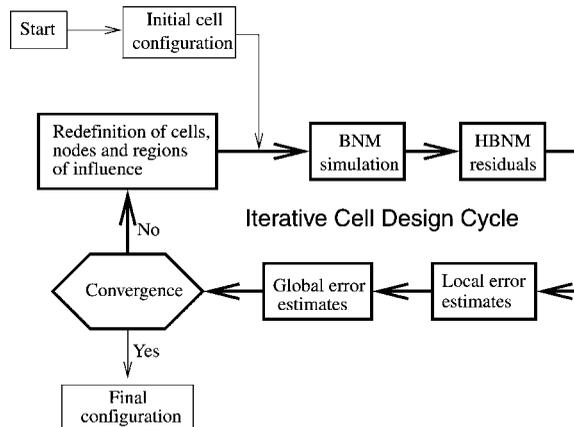


Figure 4. Typical self-adaptive iterative BNM algorithm (*h*-version) according to the scheme of Figure 3(a). The BNM equations used for solving the BVP are (28) and (29), and the HBNM equations used for residual computation are (39) and (41) for potential theory and linear elasticity, respectively.

Linear elasticity: Similarly, for elasticity problems (see Equation (48)):

$$r_{ij} = \text{residual}(\sigma_{ij}) \quad (58)$$

A scalar residual measure is defined as

$$r = r_{ij}r_{ij} \quad (59)$$

The exact local error in stress is defined as

$$\varepsilon_{ij} = \sigma_{ij}^{(\text{exact})} - \sigma_{ij}^{(\text{numerical})} \quad (60)$$

and the corresponding scalar measure is defined as

$$\varepsilon = \varepsilon_{ij}\varepsilon_{ij} \quad (61)$$

Equations (59) and (61) are used to compute the hypersingular residual and exact error, respectively, in the stress σ_{ij} , at each node, for problems in linear elasticity. These equations are presented here for the sake of completeness.

9.2. Local residuals and errors—singular residual approach

Potential Theory: The singular residual is defined in an analogous fashion. For potential problems (see Equation (51)),

$$r = \text{residual}(u) \quad (62)$$

and the exact local error in u is defined as

$$\varepsilon = |u^{(\text{exact})} - u^{(\text{numerical})}| \quad (63)$$

Here, r and ε are themselves scalar measures of the residual and exact error, respectively. *Equations (51) and (63) are used to obtain the singular residual and exact error, respectively, in the potential u , at each node, for problems in potential theory.* These equations are presented here for the sake of completeness.

Linear elasticity: For elasticity problems (see Equation (53)):

$$r_i = \text{residual}(u_i) \quad (64)$$

so that a scalar residual measure is

$$r = r_i r_i \quad (65)$$

The exact local error in u_i is defined as

$$\varepsilon_i = u_i^{(\text{exact})} - u_i^{(\text{numerical})} \quad (66)$$

with a corresponding scalar measure

$$\varepsilon = \varepsilon_i \varepsilon_i \quad (67)$$

Equations (65) and (67) are used to obtain the singular residual and exact error, respectively, in the displacement u_i , at each node, for elasticity problems.

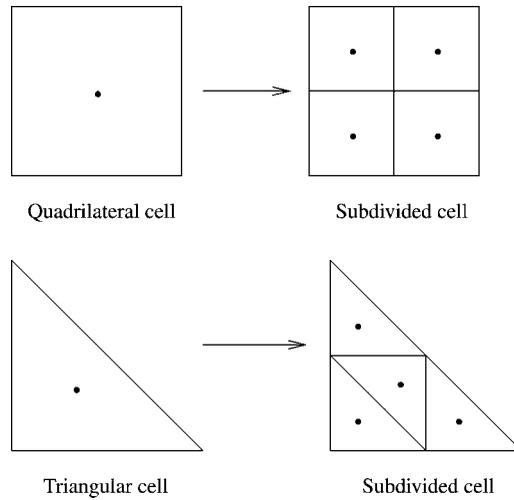


Figure 5. Cell refinement for quadrilateral and triangular cells with one node per cell.

Remark 4. The local error measure (Equation (63)) is also used for $\partial u/\partial n$ at points on the surface of a cube (see examples of Section 11). This quantity is defined as

$$e\left(\frac{\partial u}{\partial n}\right) = \left| \frac{\partial u^{(\text{exact})}}{\partial n} - \frac{\partial u^{(\text{numerical})}}{\partial n} \right| \tag{68}$$

This error measure is used only for the colour plots presented in Plate 1 of this paper. The scalar residual measures, defined above, evaluated at nodes, are used as error estimators. In all the adaptivity examples presented in this paper, one node is used for each cell and is placed at its centroid. The scalar residual measure at this centroidal node is used as an error estimator for that cell. A comparison of the residual (r) and exact error (ε) demonstrates the effectiveness of residuals as error estimates.

9.3. Cell refinement criterion

A simple criterion for cell refinement consists of subdividing the cells for which the error indicator is larger than a certain reference value. In this work, the reference quantity is taken as the average value of the error indicator (here the average residual) given by

$$\bar{r} = \frac{1}{N_n} \sum_{i=1}^{N_n} r^{(i)} \tag{69}$$

where N_n is the total number of nodes. If the inequality,

$$r > \gamma \bar{r} \tag{70}$$

is satisfied, then the cell is subdivided into four pieces (see Figure 5). The parameter γ in Equation (70) is a weighting coefficient that controls the ‘cell refinement velocity’. The standard procedure consists of using $\gamma = 1.0$. If $\gamma > 1.0$, then the number of cells to be refined

is less than with $\gamma = 1.0$. According to Figure 4, the numerical solution of the next iterative step is expected to be more accurate than that of the current step; however, the increase on the total number of cells is comparatively small when $\gamma > 1.0$.

If $\gamma < 1.0$, then the number of cells to be refined is larger than that with $\gamma = 1.0$. The advantage in this case is that the refinement rate increases, however, the computational efficiency may decrease owing to likely generation of an excessive number of cells. *An alternative procedure, for a ONE-step refinement, is presented in Section 12 of this paper.*

9.4. Global error estimation and stopping criterion

Global L_2 error: A global L_2 error, on a panel, or over the whole boundary ∂B , is defined as

$$\bar{\varepsilon}(\phi) = \frac{\int_A (\phi^{(\text{exact})} - \phi^{(\text{numerical})})^2 dA}{\int_A (\phi^{(\text{exact})})^2 dA} 100\% \quad (71)$$

where ϕ is a variable of interest and A is the area of a panel or of the whole surface ∂B . These global errors are used in many of the tables that are presented later in this paper.

An indication of overall convergence may be obtained by evaluating either \bar{r} (Equation (69)) or $\bar{\varepsilon}$ from Equation (71). Of course, Equation (71) is only useful for test examples in which the exact solution is known.

Stopping criterion: For generic problems where the exact solution is not available (e.g. most engineering problems), cell refinement (see Figure 5) can be stopped when

$$\bar{r} \leq r_{\text{global}} \quad (72)$$

where r_{global} has a preset value, which depends on the overall level of accuracy desired. The goal of the adaptive procedure is to obtain well-distributed (i.e. near optimal) cell configurations. Ideally, as the iterative cell refinement progresses, the error estimates should decrease both locally and globally.

10. PARALLEL COMPUTING

An important aspect of any new numerical method is its computational cost. It has been observed that although meshless methods have NO connectivity requirements on the underlying cell structure, they are nonetheless computer intensive. This aspect is addressed in this work by means of a simple parallel implementation of the BNM/HBNM and the adaptive procedure. Here, only the assembly of the system matrix and the residual computation are parallelized, as illustrated by the flow-charts in Figures 6 and 7, respectively. Parallelization of the solution phase is a separate and challenging problem that has not been implemented in this work.

The parallel code uses the message passing interface (MPI) standard and accesses multiple processors of the IBM SP2 (R6000 architecture, 120 MHz P2SC Processor). However, similar concepts are also applicable to the parallel virtual machine (PVM) standard operating on a cluster of engineering workstations (see for example, References [61–63]).

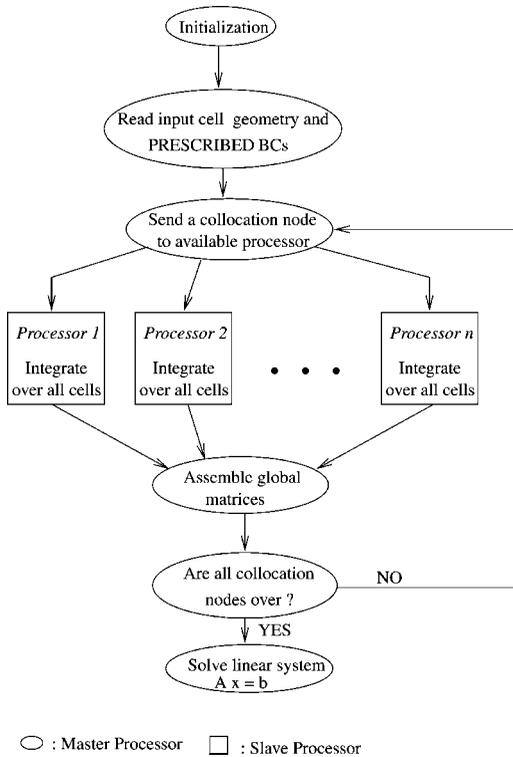


Figure 6. Flow-chart for parallel BNM or HBNM code.

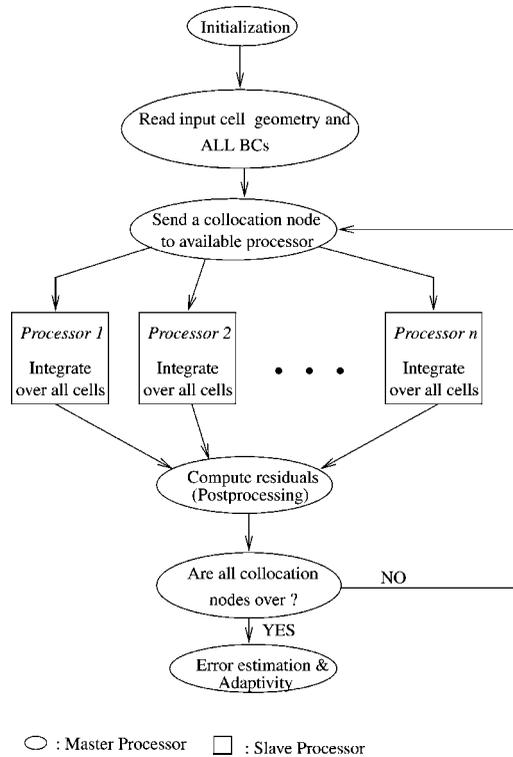


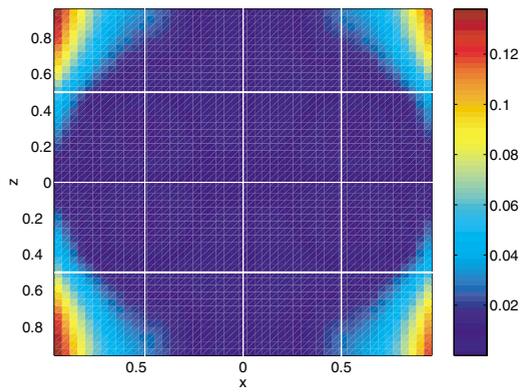
Figure 7. Flow-chart for parallel meshless adaptive code.

11. PROGRESSIVELY ADAPTIVE SOLUTIONS

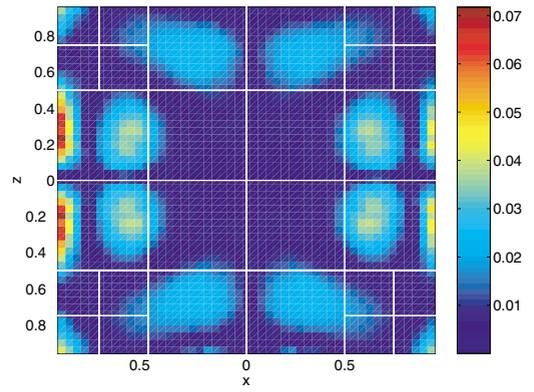
The adaptive process illustrated by Figure 4 is applied to two representative examples:

- Dirichlet problem on a cube. Laplace’s equation is solved using the BNM, and the (hypersingular) residuals are obtained using the HBNM, according to Figure 3(a).
- Stretching of an elastic cylindrical rod clamped at one end. This time, the role of the BIE and HBIE is reversed, i.e. the HBNM is employed for solving the boundary value problem, and the (singular) residuals are calculated from the BNM, according to Figure 3(b).

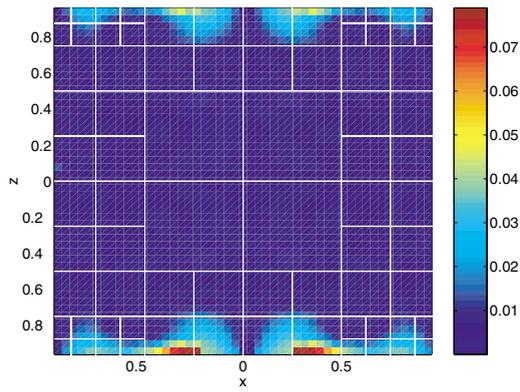
This set of problems permit assessment of various parameters of the adaptive strategy for meshless methods based on BIE techniques. Several aspects are investigated such as the quality of the adaptive solution obtained for scalar (potential theory) and vector field (elasticity theory) problems, performance of the method on problems with either pure or mixed boundary conditions, evaluation of the quality of error estimates obtained by means of hypersingular or singular residuals, sensitivity of the ‘final’ solution with respect to the starting cell configuration (initial condition of the self-adaptive problem), and convergence properties.



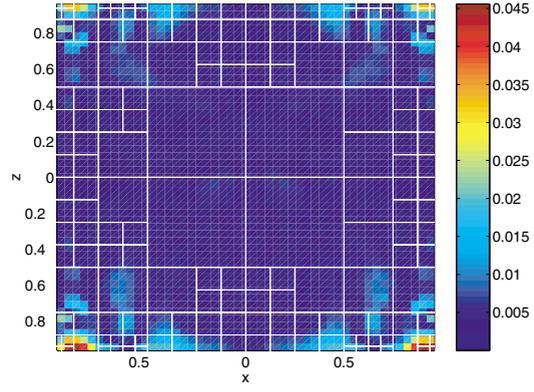
(a)



(b)



(c)



(d)

Plate 1. Error in $\partial u/\partial n$ ($e(\partial u/\partial n)$) on the face $y=-1$ of the cube: (a) initial configuration #2 (96 surface cells); (b) first adapted step (168 cells); (c) second step (456 cells); and (d) third step (1164 cells).

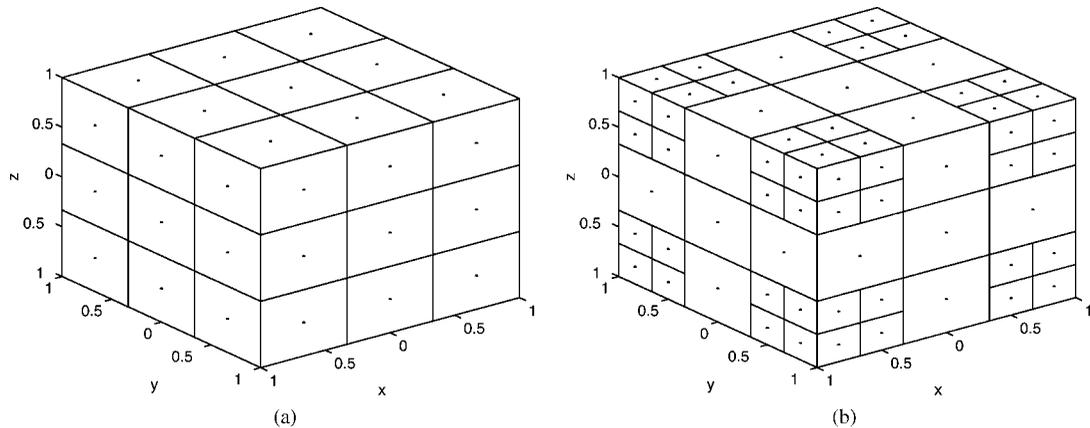


Figure 8. Cell configurations on the surface of a cube: (a) initial configuration # 1: 54 surface cells. (b) first adapted step: 126 cells; obtained with $\gamma = 1$.

11.1. Dirichlet problem on a cube

The following exact solution, which satisfies the 3D Laplace's equation, is used in this example:

$$u = \sinh\left(\frac{\pi x}{2}\right) \sin\left(\frac{\pi y}{2\sqrt{2}}\right) \sin\left(\frac{\pi z}{2\sqrt{2}}\right) \quad (73)$$

Note that the solution is symmetric with respect to y and z but that its dependence on x is different from its dependence on y or z . The appropriate value of u is prescribed on ∂B (Dirichlet problem) and $\partial u/\partial n$ is computed on ∂B . Because the exact solution cannot be represented in terms of polynomials, this is a proper test of the meshless method and the adaptivity procedure. A quadratic basis is used for the construction of the MLS interpolating functions, i.e. $m = 6$ (see Equation (2)). The idea behind the adaptive procedure is to start with a rather crude cell configuration and carry out cell refinement in the region where the residual is large according to a certain criterion. Hence, the adaptivity results in this section have been obtained starting with two relatively coarse initial cell configurations. This comparative procedure tests the sensitivity of the adaptive scheme with respect to the initial conditions.

11.1.1. Initial cell configuration # 1. Figure 8(a) shows a discretization consisting of 54 rectangular cells with one (centroidal) node per cell. The boundary value problem is solved using the BNM (Equation (28)). Then the results are used in the HBNM (Equation (39)) to obtain the hypersingular residual. Figure 9 shows a comparison between the hypersingular residual (from Equations (45) and (55)) and the exact local error ε in u_j (from Equations (56) and (57)) computed for the initial configuration # 1 (Figure 8(a)) at each node on the surface. It can be clearly seen that the hypersingular residual tracks the exact error perfectly.

Cell refinement is carried out using $\gamma = 1.0$ in Equation (70), and the resulting refined cell configuration consisting of 126 cells is shown in Figure 8(b). It can be seen from Figure 8(b) that the cell refinement occurs *only* at the corners where the exact error is the largest. This is an indication that the procedure for error estimation and adaptivity is moving in the

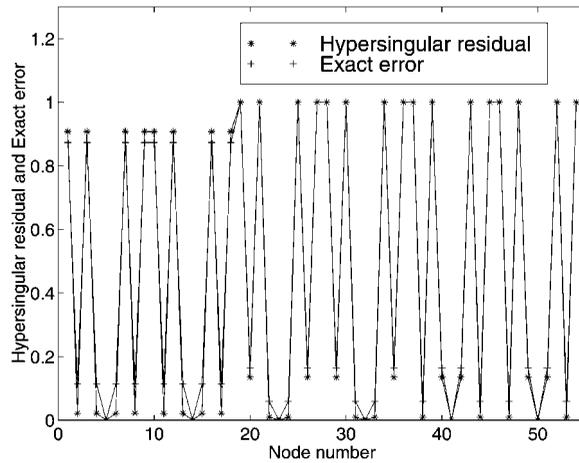


Figure 9. Comparison of hypersingular residual and exact local error ε in u_j for the initial configuration # 1 (54 cells, one node per cell). These quantities have been normalized by their respective maximum values, where $r_{\max} = 0.5197 \times 10^{-1}$ and $\varepsilon_{\max} = 0.2051$.

Table I. $\bar{\varepsilon}(\partial u/\partial n)$ and residuals \bar{r} , r_{\max} for the initial cell configuration (Figure 8(a)) and the configuration obtained at the end of the first step of the adaptivity process using $\gamma = 1.0$ (Figure 8(b)).

Output parameters	Initial	Final
Number of cells	54	126
$x = \pm 1$	1.4209%	0.0238%
$y = \pm 1$	7.6911%	0.2773%
$z = \pm 1$	7.6911%	0.2578%
All faces	2.1450%	0.0519%
Average residual (\bar{r})	0.2366E-01	0.7605E-02
Maximum residual r_{\max}	0.5197E-01	0.3068E-01

right direction. Now, the boundary value problem is solved again using the BNM. Table I summarizes the various output parameters of the adaptivity procedure. It can be seen from Table I that excellent numerical results are obtained in a single step of the adaptivity process and hence the adaptive procedure is not continued further.

11.1.2. Initial cell configuration # 2. The initial configuration # 2 is for the same physical cube with 16 uniform cells on each face with, as always, one node at the centroid of each cell. As before, the boundary value problem is solved using the BNM (Equation (28)), and the results obtained are used in the HBNM (Equation (39)). Figure 10 shows a comparison between the hypersingular residual (from Equations (45) and (55)) and the exact local error ε in u_j (from Equations (56) and (57)), computed for the initial configuration # 2 (Figure 11(a)). It can be clearly seen that the hypersingular residual tracks the exact error very accurately. In fact, the results for the finer cell configuration # 2 are very similar to those shown in Figure 9 for the coarser initial cell configuration # 1.

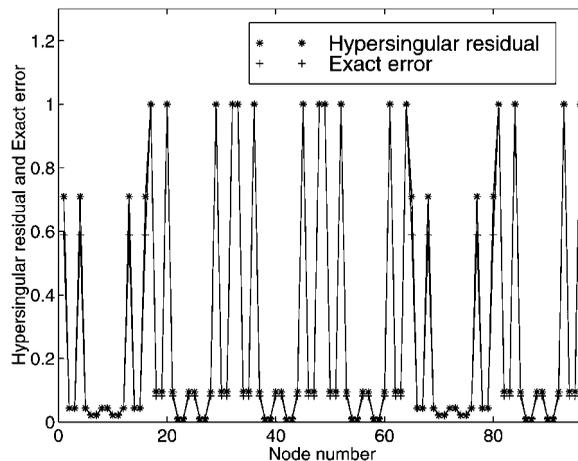


Figure 10. Comparison between hypersingular residual and exact local error ε in u_j for the initial configuration # 2 (96 cells, one node per cell). These quantities have been normalized by their respective maximum values, where $r_{\max} = 0.1829 \times 10^{-1}$ and $\varepsilon_{\max} = 0.6223 \times 10^{-1}$.

Adaptivity Results: In order to obtain a better understanding of the adaptivity procedure, the local error e in $\partial u/\partial n$ (from Equation (68)) is calculated on each of the faces of the cube. The iterative cell design cycle of Figure 4 is repeated three times using $\gamma = 0.5$ in Equation (70) and starting from the initial configuration # 2 given in Figure 11(a). The resulting refined cell configurations are shown in Figures 11(b), 11(c) and 11(d), respectively. It is noted that the cell refinement should begin at the corners of the cube where the error in $\partial u/\partial n$ is the largest.

Plate 1 shows colour contour plots of the exact local error e in $\partial u/\partial n$ on the $y = -1$ face of the cube. The underlying cell structure on the face is also shown in the colour plots. The resolution of these and subsequent contour plots is much finer than the corresponding cell discretization because the error is actually evaluated at a large number of points on the boundary (panels) of the body. These results confirm the observation made at the end of the previous paragraph regarding regions of large errors which demand a finer discretization. Thus, refinement occurs close to the edges and corners where the error in $\partial u/\partial n$ is largest.

Other relevant comments are in order. For the first step of the adaptive procedure (see Figure 11(b)), selected results are shown in Figure 12 which provides a comparison between the hypersingular residual (from Equations (45) and (55)) and the exact local error ε (from Equations (56) and (57)). The results are shown on the $x = -1$ and $z = 1$ faces as a representative sample of the results over the 168 nodes. It can be seen from Figure 12 that the hypersingular residual tracks the exact error reasonably well.

Plate 1(b) shows a contour plot for the exact local error e in $\partial u/\partial n$ on the $y = -1$ face of the cube for adapted cell configuration of Figure 11(b). Note that, due to the refinement procedure, the error in $\partial u/\partial n$ has reduced substantially, especially at the corners (cf. Plate 1(a) and 1(b)).

Plate 1(c) and 1(d) show the exact local error e in $\partial u/\partial n$ on the $y = -1$ face of the cube for the adapted cell configurations consisting of 456 cells and 1164 cells, respectively (see Figures

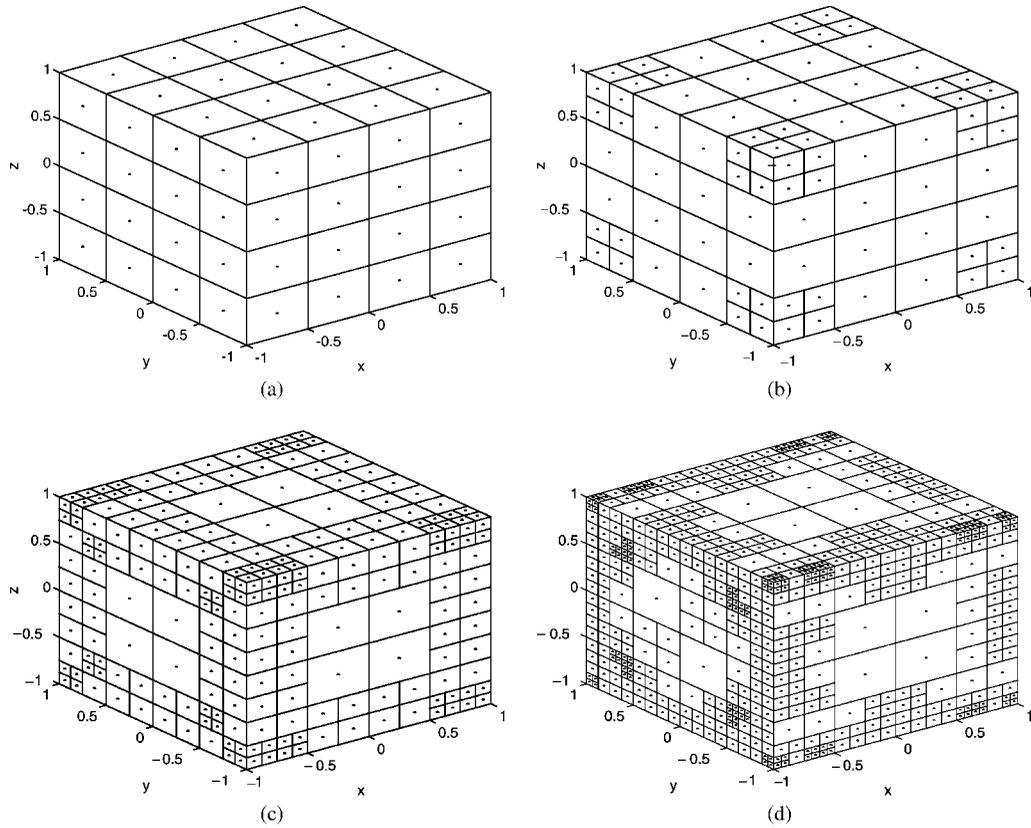


Figure 11. Cell configurations on the surface of a cube: (a) initial configuration # 2 (96 surface cells); (b) first adapted step (168 cells); (c) second step (456 cells); (d) third step (1164 cells).

11(c) and 11(d)). Comparing the contour plots of Plate 1(a)–(d), one can readily verify that the error in $\partial u/\partial n$ decreases substantially during the adaptive process. It is interesting to note that the absolute value of the exact solution (Equation (73)) has the same functional dependence on the $y = -1$ and $z = 1$ faces and different on the $x = -1$ face of the cube. Step 1 (Figure 11(b)) is not sensitive to this fact, however, Steps 2 (Figure 11(c)) and 3 (Figure 11(d)) of the adaptive procedure are. This is a tribute to the quality of residuals as error estimates.

Table II summarizes the results of the adaptive process for the cube problem starting with the initial cell configuration consisting of 96 cells (Figure 11(a)). Note that, on the faces $x = \pm 1$, $\bar{\epsilon}(\partial u/\partial n)$ increases from the initial configuration to Step 1, and from Step 1 to Step 2. However, $\bar{\epsilon}(\partial u/\partial n)$ finally decreases from Steps 2 to 3 and reaches its lowest value at this step, which has the sophisticated cell pattern of Figure 11(d). On the faces $y = \pm 1$ and $z = \pm 1$, $\bar{\epsilon}(\partial u/\partial n)$ monotonically decreases as the number of adaptive cycles increases. Moreover, as expected, the global $\bar{\epsilon}(\partial u/\partial n)$ for ‘all faces’, as well as the average and maximum residuals, decrease as the adaptive process progresses.

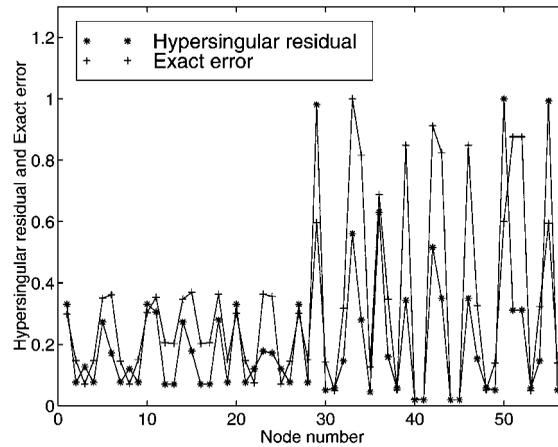


Figure 12. Comparison of hypersingular residual and exact local error ε in $u_{,j}$ on the faces $x = -1$ and $z = 1$ of the cube of Figure 11(b) (*first step of the adaptive procedure*). The quantities have been normalized by their respective maximum values, where $r_{\max} = 0.1567 \times 10^{-1}$ and $\varepsilon_{\max} = 0.3645 \times 10^{-1}$.

Table II. L_2 error in $\partial u/\partial n$ ($\bar{\varepsilon}(\partial u/\partial n)$) and residuals \bar{r} , r_{\max} for the various steps of the adaptivity process starting with the initial cell configuration consisting of 96 cells with one node per cell (Figure 11(a)). Here $\gamma = 0.5$ is used for the cell refinement of the cube.

Output parameters	Initial	Step 1	Step 2	Step 3
Number of cells	96	168	456	1164
$x = \pm 1$	0.0759%	0.1062%	0.1135%	0.0438%
$y = \pm 1$	1.0654%	0.2785%	0.2089%	0.0551%
$z = \pm 1$	1.0696%	0.2781%	0.2091%	0.0551%
All faces	0.1899%	0.1269%	0.1247%	0.0451%
Average residual \bar{r}	0.4963E-02	0.3661E-02	0.5643E-03	0.1811E-03
Maximum residual r_{\max}	0.1829E-01	0.1567E-01	0.3579E-02	0.2537E-02

Parallel computation. An important aspect of any new numerical method is the computational burden associated with it. Thus the parallel implementation described in Section 10, and illustrated in Figures 6 and 7, is employed here and compared with the serial implementation. The running times for different cell configurations are shown in Table III. It can be seen that order of magnitude gains in wall-clock times are possible by using a parallel version of the serial program. The last line in Table III, where the ‘Total time’ is listed, is the total time taken for the entire adaptive procedure, starting from the initial configuration (Figure 11(a)) to the final adapted configuration in Step 3 (Figure 11(d)).

11.2. Adaptivity on a cylindrical elastic rod

The adaptive procedure is described in this section for a problem in linear elasticity. A schematic of the physical problem under consideration is shown in Figure 13(a). In this case (for variety) the boundary value problem is solved using the HBNM (Equation (42)) and the singular residual is obtained using the standard BNM (Equation (29)).

Table III. Comparison of wall-clock times on the IBM SP2 for the various steps of the adaptive process ($\gamma=0.5$) starting with the initial cell configuration of 96 cells with one node per cell on the cube.

Configuration	Serial	Parallel BNM and adaptivity (MPI)		
		4 Procs	16 Procs	64 Procs
96 cells	3 min 6.3 s	1 min 3.3 s	14.7 s	5.6 s
168 cells	10 min 15.1 s	3 min 27.4 s	46.4 s	13.7 s
456 cells	1 h 38 min	32 min 18.4 s	6 min 43.7 s	1 min 52.4 s
1168 cells	16 h	5 h 16 min	1 h 4 min	16 min 51.2 s
Total time	17 h 51 min	5 h 53 min	1 h 12 min	19 min 2.9 s

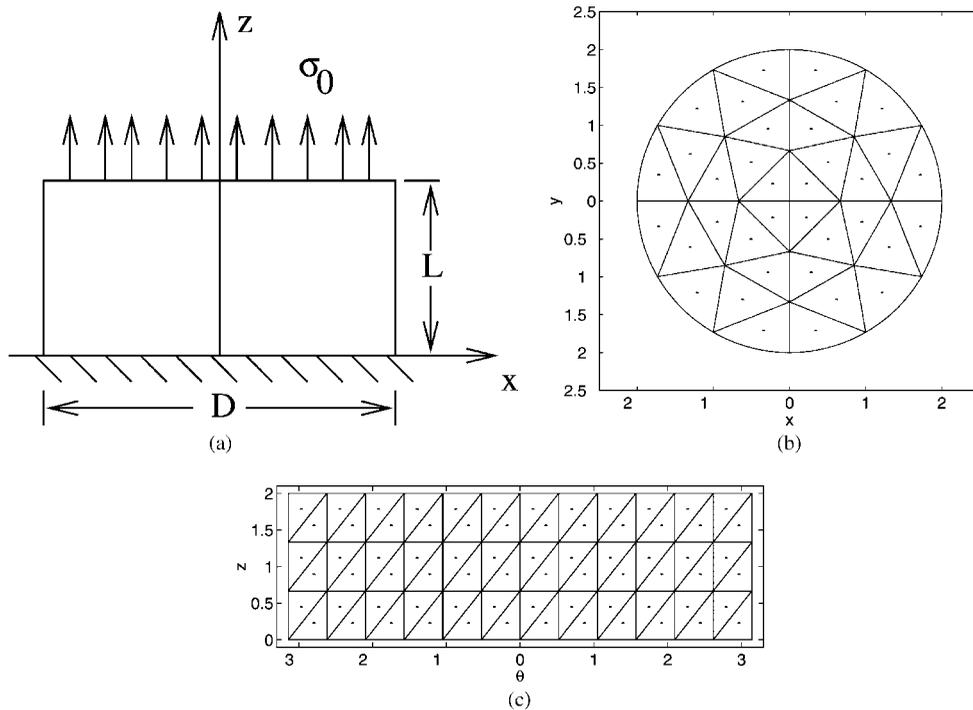


Figure 13. Stretching of a short clamped cylindrical rod by an uniform tensile load: (a) Physical situation: $L=2.0$, $D=4.0$, $\nu=0.25$, $E=1.0$, $\sigma_0=1.0$; (b) and (c) Initial cell configuration with 144 cells (one node per cell)—(b) clamped and loaded faces; (c) curved surface of the cylindrical rod mapped onto the (z, θ) plane.

Initial cell configuration: The geometric and material parameters chosen are: $E=1.0$, $\nu=0.25$, $\sigma_0=1.0$, $L=2.0$, and $D=4.0$. Figures 13(b) and 13(c) show the initial cell configuration on the clamped and loaded faces and on the curved surface of the rod. The boundary value problem is solved by prescribing tractions on the top face of the cylinder, with the bottom surface completely clamped, and the curved surface traction free. Upon obtaining the solution to the

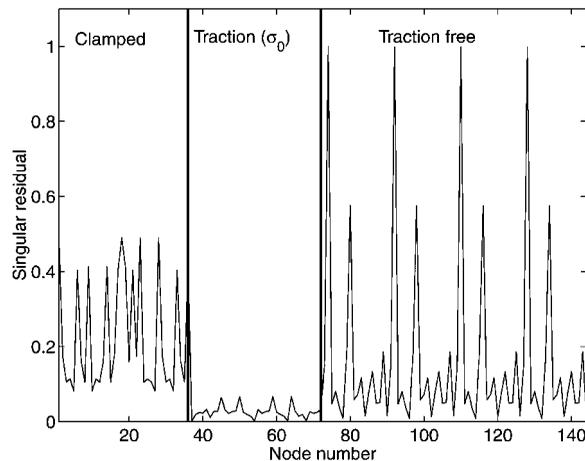


Figure 14. Singular residual for the initial configuration of 144 cells on the cylindrical rod of Figure 13(b) and (c). The residual has been normalized with respect to its maximum value, $r_{\max} = 0.2419 \times 10^{-2}$.

boundary value problem, the singular residual is obtained at each node. Figure 14 shows the singular residual (from Equations (42), (53), and (65)) obtained for the initial cell configuration (144 cells). It can be seen that the residual is considerably higher on the clamped face and on the curved surface near the clamped face, than on the loaded face. This is to be expected considering the physical nature of the problem at hand which has a singularity on the bounding circle of the clamped face.

Adaptivity results: The adaptive strategy is carried out according to the flow chart of Figure 4. However, the boxes for the ‘BNM simulation’ and ‘HBNM residuals’ are replaced by ‘HBNM simulation’ and ‘BNM residuals’, according to the scheme of Figure 3(b). Since the singular residual is higher on the clamped and curved faces, most of the subdivision of cells occur on those faces. The curved surface of the cylinder near the clamped surface is refined due to the singularity at the edge of the clamped face. However, the top face (the loaded face) is NOT refined at all and so the cell structure on that face remains as shown in Figure 13(b).

Three steps of adaptivity are pursued using $\gamma = 1.25$ in Equation (70) and starting from the initial configuration of Figures 13(b) and 13(c). The resulting refined cell configurations are shown in Figures 15–17. As expected, Figures 15(a), 16(a) and 17(a) show that the loaded face is not refined at all and remains as in the initial configuration (Figure 13(b)). On the clamped face, a comparison of Figures 13(b), 15(b), 16(b), and 17(b) indicates that cell refinement only takes place near the edge of the face, which is the region where gradients in stresses are largest.

Figures 13(c), 15(c), 16(c), and 17(c) show the progressive refinement on the curved surface of the cylinder. One can observe that refinement primarily occurs along the curved surface near the clamped edge of the cylindrical rod. Note that, for the initial configuration (Figures 13(b) and 13(c)), the number of subdivisions along the edge of the top and bottom faces is the same as the number of subdivisions along the edge of the curved surface

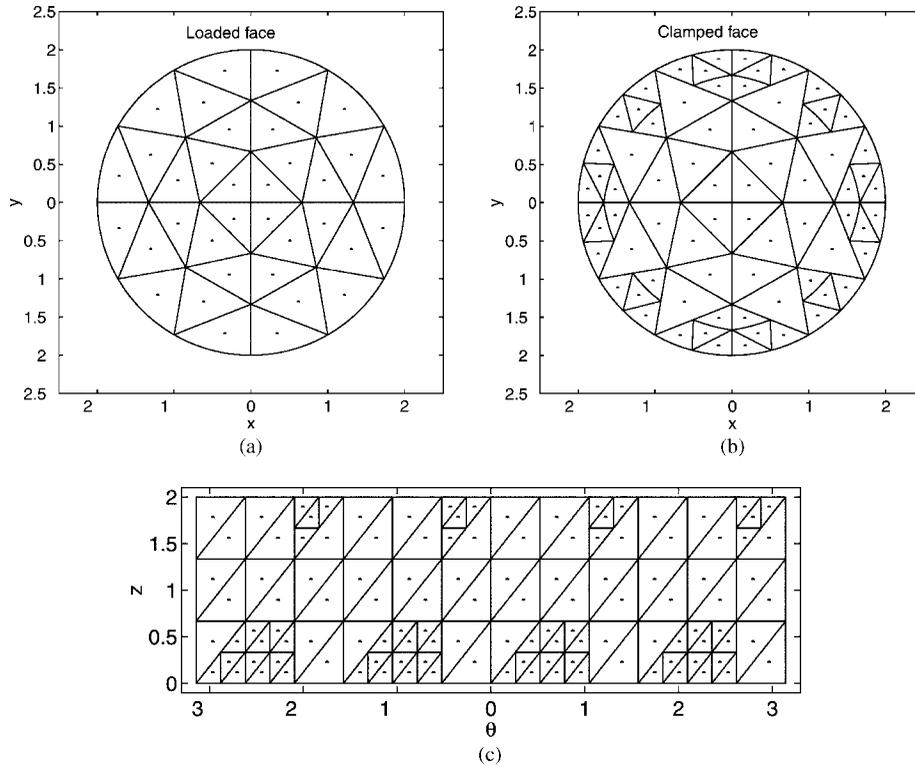


Figure 15. Short clamped cylindrical rod Step 1: Adapted configuration consisting of 228 cells obtained with $\gamma = 1.25$: (a) loaded face; (b) clamped face; (c) curved surface of the rod.

(12 subdivisions). However, when adaptivity is carried out, a significant mismatch in the number of subdivisions is created at every adaptive step. This does not present any problem for the meshless method, and such freedom in modeling is expected to be especially advantageous in analysing problems with complicated geometry.

12. MULTILEVEL (ONE-STEP) ADAPTIVE CELL REFINEMENT

The previous section has dealt with an iterative adaptive technique for cell refinement (*h*-version). Here the interest is on developing a simple ONE-step algorithm for cell refinement in the meshless BNM setting. The flowchart of Figure 18 illustrates this idea which is based on the concept of refinement level (RL) employed by Krishnamoorthy and Umesh [64] and Mosalam and Paulino [65].

Refinement strategy. Figures 19(a) and 19(b) show that different degrees of refinement are carried out for different values of the refinement level. From these figures, the expression relating the final cell size h_f to the refinement level RL is

$$h_f = \frac{h_i}{2^{RL}} \tag{74}$$

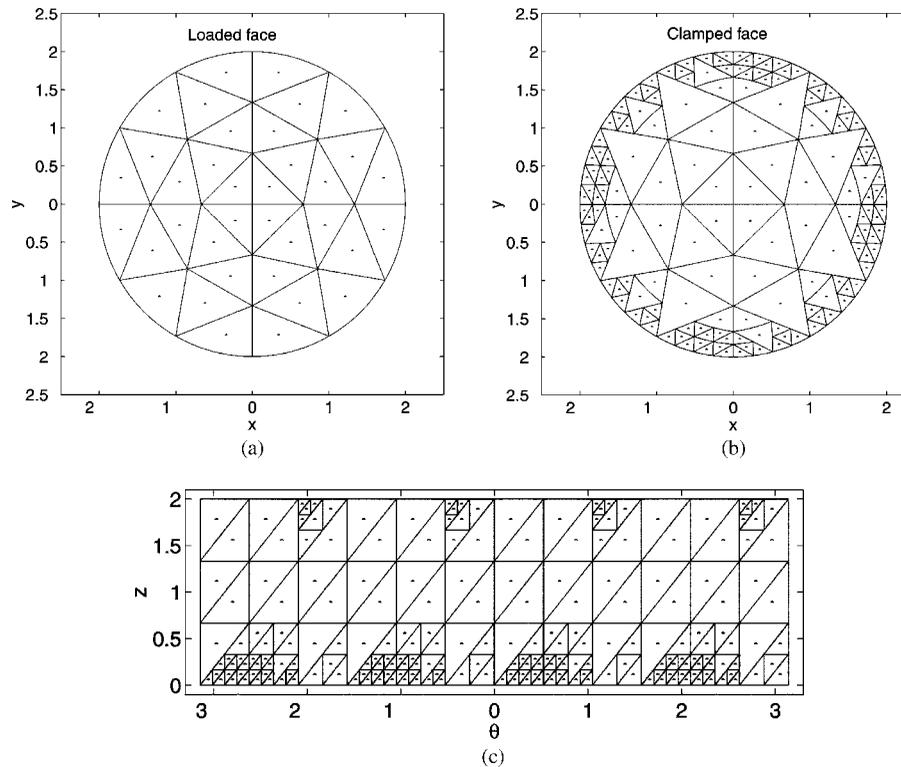


Figure 16. Short clamped cylindrical rod Step 2: Adapted configuration consisting of 324 cells obtained with $\gamma = 1.25$: (a) loaded face; (b) clamped face; (c) curved surface of the rod.

where h_i denotes the initial cell size. Assuming that the rate of convergence of the error is $\mathcal{O}(h^p)$, where h is a characteristic cell size in the area covered by the cells, which are of order p , and setting the error estimate equal to $\eta = r/(\gamma\bar{r})$ (see Equations (69) and (70)), one obtains

$$h_f = \frac{h_i}{\eta^{1/p}} \quad (75)$$

From Equations (74) and (75), the RL is given by

$$\text{RL} = \begin{cases} \frac{\log \eta}{p \log 2} & \text{for } \eta \geq 1 \\ 0 & \text{for } \eta < 1 \end{cases} \quad (76)$$

where p is order of the interpolating function. For the interpolation procedure used in this work, $p = m$. The second condition in Equation (76) is enforced because cell structure coarsening is not considered in this work. This idea of ONE-step refinement is applied to the cube problem of Section 12.1 and the elastic cylindrical rod problem of Section 12.2. On the cube problem, the errors are estimated by means of hypersingular residuals, and on the cylinder problem the errors are estimated by means of singular residuals.

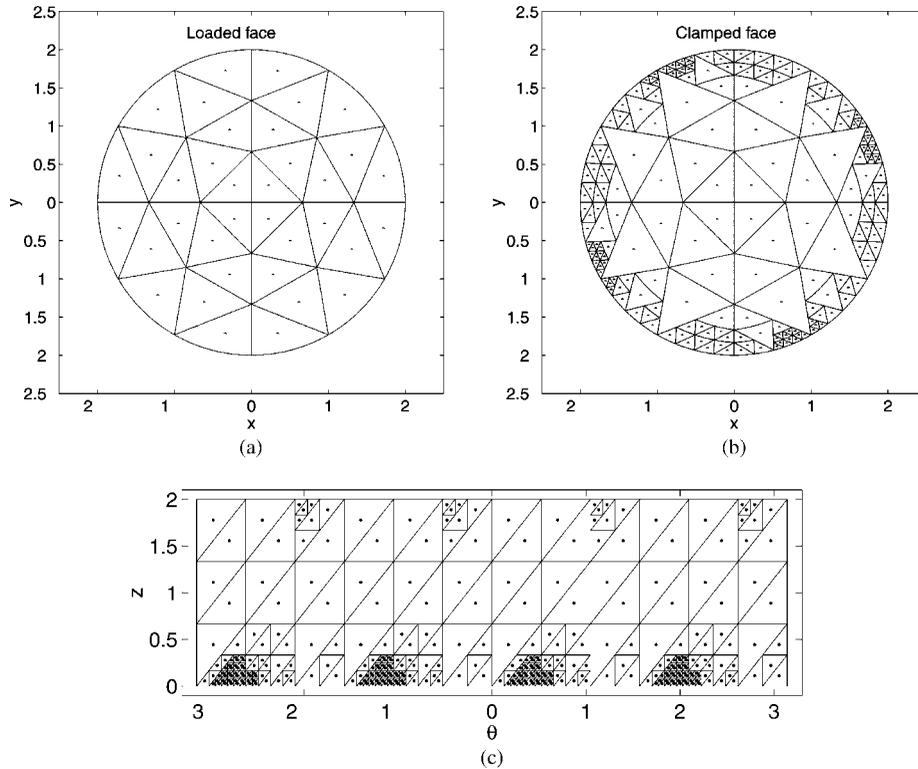


Figure 17. Short clamped cylindrical rod Step 3: Adapted configuration consisting of 576 cells obtained with $\gamma = 1.25$: (a) loaded face; (b) clamped face; (c) curved surface of rod.

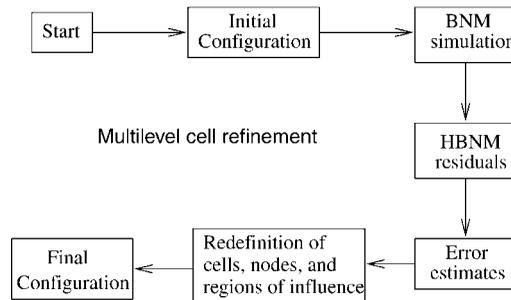


Figure 18. ONE-step adaptive BEM algorithm based on multilevel cell refinement.

12.1. Multilevel refinement on a cube with Dirichlet BCs

The Dirichlet problem for the cube solved in Section 11.1 by means of an iterative adaptive solution is reconsidered here with the ONE-step multilevel refinement scheme. The two initial cell configurations of Figures 8(a) and 11(a) are investigated again because this study permits evaluation of the sensitivity of the multilevel refinement scheme with respect to the initial conditions.

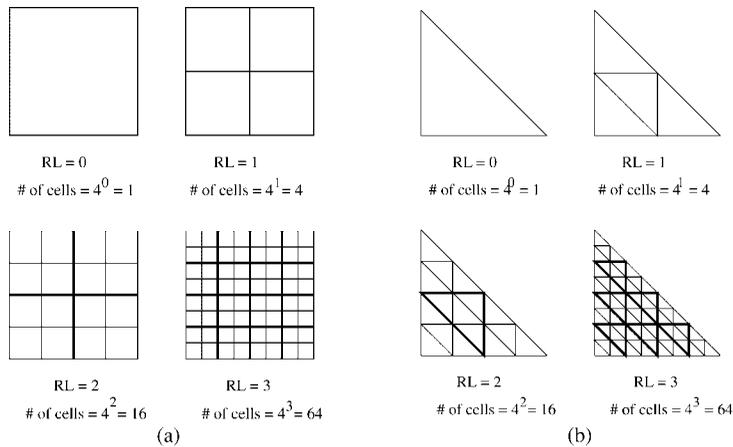


Figure 19. Refinement level RL using: (a) rectangular; and (b) triangular cells. The bold lines illustrate the idea of cell structure embedding.

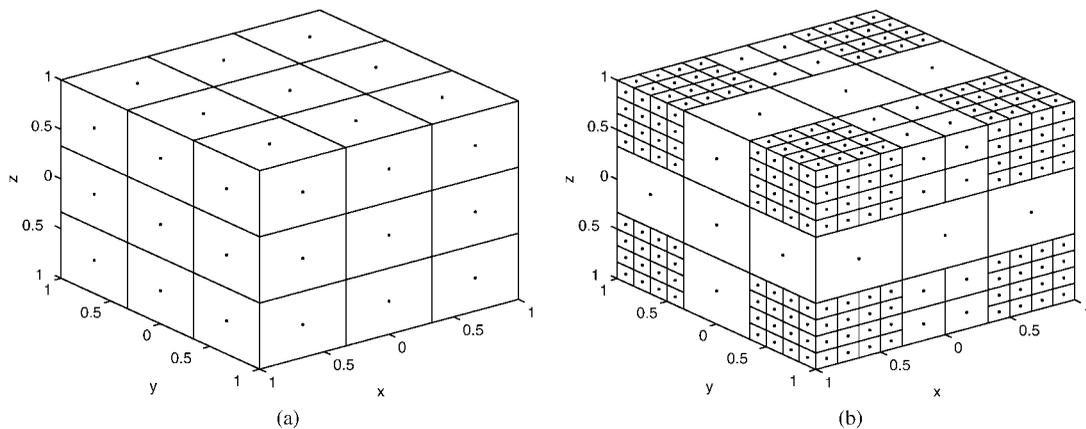


Figure 20. ONE-step multilevel cell refinement for the cube problem: (a) initial configuration # 1 with 54 cells; (b) adapted configuration with 438 cells using $\gamma = 0.15$

12.1.1. Initial cell configuration # 1.

Adaptivity Results: The multilevel strategy is implemented on the cube of Figure 20(a), which consists of 54 cells with one node per cell. The boundary value problem is solved using the BNM (Equation (28)) by imposing the exact solution in Equation (73) as Dirichlet boundary conditions. The hypersingular residual (from Equations (39 and (55))) is obtained and then the multilevel refinement procedure is carried out using $\gamma = 0.15$. The cell structure obtained in ONE-step is shown in Figure 20(b), which consists of 438 cells with one node per cell. Table IV shows a comparison of the results from the ONE-step multilevel refinement scheme starting with the configuration of Figure 20(a) and ending with the configuration of Figure 20(b). This table shows that $\bar{\epsilon}(\partial u/\partial n)$ and the residual consistently decrease with refinement.

Table IV. $\bar{\epsilon}(\partial u/\partial n)$ and residuals \bar{r} , r_{\max} for the initial configuration (Figure 20(a)), and the final configuration (Figure 20(b)) obtained by the multilevel refinement strategy with $\gamma = 0.15$.

Output parameters	Initial	Final
Number of cells	54	438
$x = \pm 1$	1.4209%	0.0411%
$y = \pm 1$	7.6911%	0.0339%
$z = \pm 1$	7.6911%	0.0343%
All faces	2.1450%	0.0403%
Average residual \bar{r}	0.2366E-01	0.4615E-03
Maximum residual r_{\max}	0.5197E-01	0.2618E-01

Table V. Comparison of wall-clock times on the IBM SP2 considering the multilevel refinement strategy for the cube problem of Figure 20 with $\gamma = 0.15$.

Configuration	Parallel BNM and adaptivity (MPI)			
	Serial	4 Procs	8 Procs	16 Procs
54 cells	33.1 s	12.9 s	6.9 s	4.6 s
438 cells	1 h 30 min	29 min 44.5 s	12 min 57.8 s	6 min 17.6 s
Total time	1 h 30 min	29 min 57.4 s	13 min 4.7 s	6 min 22.2 s

Parallel computation: As mentioned earlier, an important aspect of meshless methods is the computational cost. To address this problem, Table V shows a comparison of computing times for serial and parallel simulations using 4, 8, and 16 processors, respectively. The timing results show the superiority of parallel over serial computations. Thus, the parallel computing technique is also advantageous in the multilevel refinement scheme. Qualitatively, the conclusions regarding Table III (cube problem with iterative adaptive scheme) also hold for Table V.

12.1.2. Initial cell configuration # 2. The BNM (Equation (28)) is used to solve the boundary value problem using Equation (73) as the exact solution, and the hypersingular residual is obtained by means of Equations (39) and (55). Multilevel refinement is carried out using $\gamma = 0.15$. The cell structure obtained in ONE-step is shown in Figure 21(b), which consists of 1764 cells with one node per cell.

Table VI summarizes the results for $\bar{\epsilon}(\partial u/\partial n)$ and the residual for the multilevel refinement strategy starting from the initial configuration # 2 of 96 cells (see Figure 21(a)). Qualitatively, these results are analogous to those of Table II for the progressive adaptive refinement, which includes the peculiarity observed on the $x = \pm 1$ faces of the cube. Moreover, the remarks concerning Table II also hold for explaining the results of Table VI. Therefore, for further explanations, the reader is referred to Section 11.1.

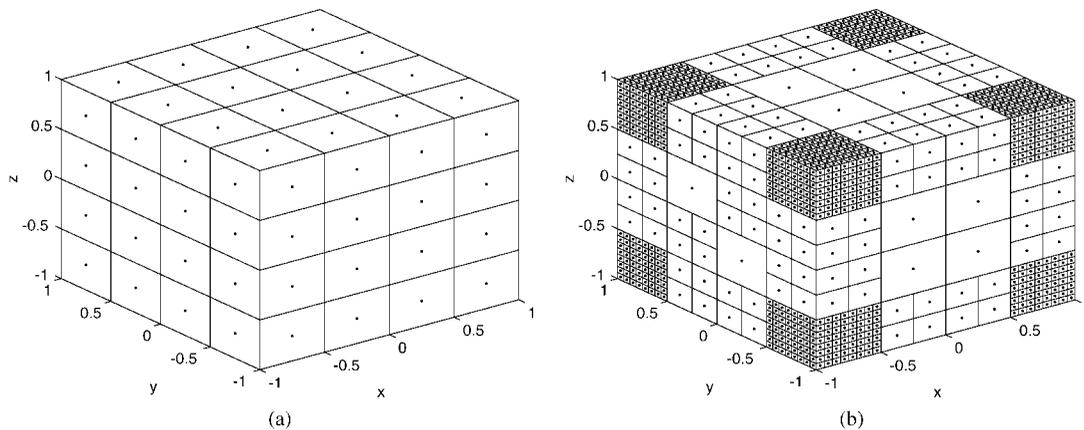


Figure 21. ONE-step multilevel cell refinement for the cube problem: (a) initial configuration # 2 with 96 cells; (b) adapted configuration with 1764 cells using $\gamma = 0.15$.

Table VI. $\bar{\epsilon}(\partial u/\partial n)$ and residual for the initial configuration (Figure 21(a)), and the final configuration (Figure 21(b)) obtained by the multilevel refinement strategy with $\gamma = 0.15$.

Output parameters	Initial	Final
Number of cells	96	1764
$x = \pm 1$	0.0759%	0.1034%
$y = \pm 1$	1.0654%	0.2400%
$z = \pm 1$	1.0696%	0.2400%
All faces	0.1899%	0.1169%
Average residual (\bar{r})	0.4963E-02	0.3370E-03
Maximum residual (r_{\max})	0.1829E-01	0.1535E-01

12.2. Multilevel refinement on a cylindrical elastic rod

The multilevel refinement procedure is demonstrated here for the physical problem of stretching of a cylinder with one end clamped. The initial configuration is the same as that of Figures 13(b) and 13(c). The present study also employs singular residuals (Equation (29)) for error estimation (see Section 11.2) rather than hypersingular residuals (Equation (41)). This is similar to the study carried out for the same problem by means of iterative adaptive cell refinement. The results for the multilevel refinement for the original problem of Figures 13(a) and 13(b) are given in Figure 22. Comparing these results with the ones of Section 11.2 (Figures 15–17), one verifies that the overall trends are quite similar in both situations. However, two main differences are noticeable. First, the loaded face is refined (a little) here (cf. Figures 13(b) and 22(a)), while it is not refined at all in the adapted configurations shown in Figures 15(a), 16(a) and 17(a). Second, as expected, the multilevel cell refinement does not allow the smooth cell gradation which occurs in progressively adapted cell configurations (cf. Figures 17 and 22). Nevertheless, such gradation, which is essential for mesh-based methods, is not required at all in the present meshless methods (BNM and HBNM).

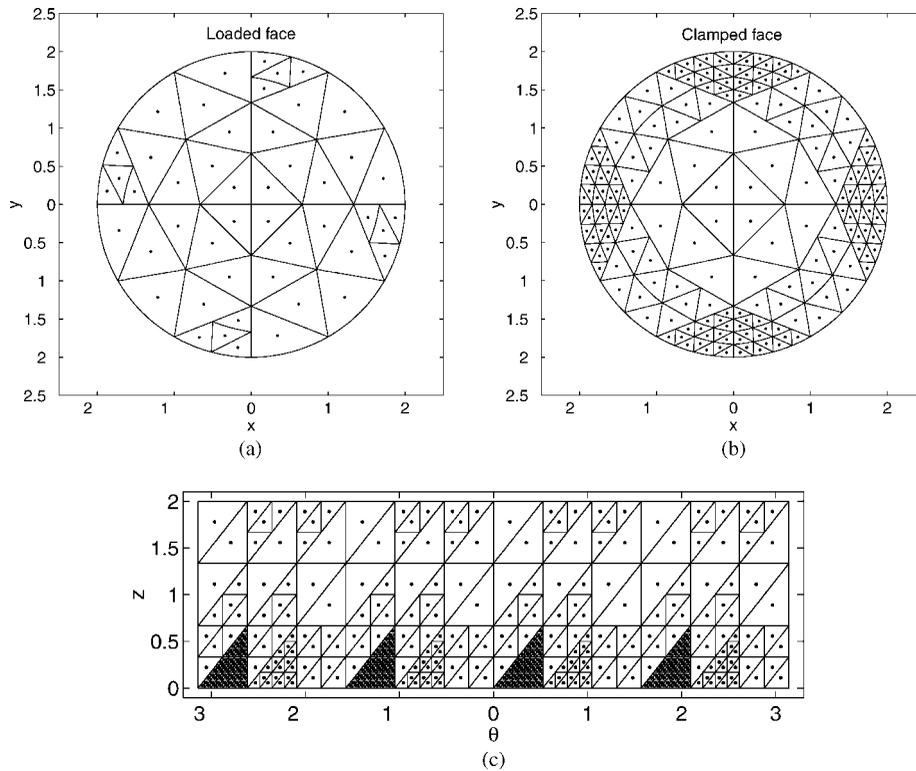


Figure 22. ONE-step multilevel cell refinement for short clamped cylinder: (a) loaded face; (b) clamped face; and (c) curved surface of cylinder. The initial cell configuration is shown in Figures 13(b) and 13(c).

13. CONCLUDING REMARKS

This paper presents several new research results related to meshless methods:

- A hypersingular boundary node method (HBNM) is developed for 3D problems in potential theory and linear elasticity.
- A unified residual formulation is developed for potential theory and linear elasticity.
- The concept of hypersingular and singular residuals is developed for *a posteriori* error estimation for the BNM and the HBNM.
- An h -adaptive cell refinement strategy is developed based on residual error estimates. It is remarkable that, in this approach, only surface cells need to be subdivided for 3D problems (a property inherited from the BEM) and a given cell can be subdivided without affecting its neighbours in any way (a consequence of decoupling interpolation from integration).
- Two adaptive criteria are presented: iterative and multilevel. In general, the iterative criterion leads to a cell structure with selective local refinement embedded within the primary cell configuration. However, being an iterative method, it is computer intensive. The multilevel criterion is a simple ONE-step procedure which circumvents the computational cost

of the iterative criterion; however, it does not allow the local selective cell control typically found in the iterative scheme.

- A new algorithm for obtaining accurate displacements and stresses at internal points *close* to the boundary, has been developed, and successfully implemented.
- A preliminary parallel implementation of the BNM/HBNM and the adaptive procedure has been presented. In this implementation, only the assembly of the system matrix and the residual computation are parallelized. Future investigation will involve development of a parallel version of the solution phase, which can be accomplished by means of software packages specifically designed to solve linear algebra problems on distributed memory computers, e.g. ScaLAPACK (parallel version of LAPACK), parallel basic linear algebra subprograms (PBLAS), and basic linear algebra communications subprograms (BLACS) (see Reference [66]). This matter is currently under investigation by the authors. The BNM is slower than the conventional BEM (collocation-based) by about an order of magnitude. Thus further research in parallel algorithms and other numerical linear algebra techniques are needed to make the method more efficient.
- A study of the various aspects of the above methods/technique are carried out by means of 3D numerical examples, e.g. a cube (Dirichlet) problem in potential theory, and a clamped-stretched cylinder in linear elasticity.

Overall, the frontiers of the boundary node method (BNM) and the hypersingular boundary node method (HBNM) have been extended to solve problems of engineering interest. Due to its considerable flexibility and relative ease of use, this method has potential for efficiently solving a wide range of industrial problems. The initial study presented in this paper holds significant possibilities for the future of adaptive numerical analysis.

ACKNOWLEDGEMENTS

The Cornell group (Chati and Mukherjee) acknowledges support by the University Research Programs (URP) grant from Ford Motor Company to Cornell University. Professor Paulino acknowledges support from the National Science Foundation under Grant CMS-9713008. The computing for this research was carried out using the resources of the Cornell Theory Center, which receives funding from Cornell University, New York State, the National Center for Research Resources at the National Institutes of Health, the National Science Foundation, the Defense Department Modernization program, the United States Department of Agriculture, and corporate partners. The efforts of Prachi Chati for carefully proof reading the manuscript are acknowledged.

REFERENCES

1. Mackerle J. Mesh generation and refinement for FEM and BEM—a bibliography (1990–1993). *Finite Elements in Analysis and Design* 1993; **15**:177–188.
2. Nayroles B, Touzot G, Villon P. Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational Mechanics* 1992; **10**:307–318.
3. Belytschko T, Lu YY, Gu L. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering* 1994; **37**:229–256.
4. Liu WK, Jun S, Zhang Y. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids* 1995; **20**:1081–1106.
5. Duarte CAM, Oden JT. H-p clouds—an *h-p* meshless method. *Numerical Methods for Partial Differential Equations* 1996; **12**:673–705.
6. Oden JT, Duarte CAM, Zienkiewicz OC. A new cloud-based *hp* finite element method. *Computational Methods in Applied Mechanics and Engineering* 1998; **153**:117–126.

7. Atluri SN, Zhu T. New meshless local Petrov–Galerkin (MPLG) approach in computational mechanics. *Computational Mechanics* 1998; **22**:117–127.
8. Atluri SN, Zhu T. A new meshless local Petrov–Galerkin (MPLG) approach to nonlinear problems in computer modeling and simulation. *Computer Modelling and Simulation in Engineering* 1998; **3**:187–196.
9. Zhu T, Zhang J-D, Atluri SN. A local boundary integral equation (LBIE) method in computational mechanics, and a meshless discretization approach. *Computational Mechanics* 1998; **21**:223–235.
10. Sukumar N, Moran B, Belytschko T. The natural element method. *International Journal for Numerical Methods in Engineering* 1998; **43**:839–887.
11. Barry W, Saigal S. A three-dimensional element-free Galerkin elastic and elastoplastic formulation. *International Journal for Numerical Methods in Engineering* 1999; **46**:671–693.
12. Sukumar N, Moran B, Black T, Belytschko T. An element-free Galerkin method for three-dimensional fracture mechanics. *Computational Mechanics* 1997; **20**:170–175.
13. Belytschko T, Lu YY, Gu L. Crack propagation by element-free Galerkin methods. *Engineering Fracture Mechanics* 1995; **51**:295–315.
14. Xu Y, Saigal S. An element-free Galerkin formulation for stable crack growth in elastic solids. *Computer Methods in Applied Mechanics and Engineering* 1998; **154**:331–343.
15. Krysl P, Belytschko T. The element free Galerkin method for dynamic propagation of arbitrary 3-D cracks. *International Journal for Numerical Methods in Engineering* 1999; **44**:767–800.
16. Belytschko T, Lu YY, Gu L, Tabbara M. Element-free Galerkin methods for static and dynamic fracture. *International Journal of Solids and Structures* 1995; **32**:2547–2570.
17. Belytschko T, Tabbara M. Dynamic fracture using element-free Galerkin methods. *International Journal for Numerical Methods in Engineering* 1996; **39**:923–938.
18. Xu Y, Saigal S. Element-free Galerkin study of steady quasi-static crack growth in plane strain tension in elastic–plastic materials. *Computational Mechanics* 1998; **21**:276–282.
19. Xu Y, Saigal S. An element-free Galerkin analysis of steady dynamic growth of a mode I crack in elastic–plastic materials. *International Journal of Solids and Structures* 1999; **36**:1045–1079.
20. Krysl P, Belytschko T. Analysis of thin plates by the element-free Galerkin method. *Computational Mechanics* 1995; **17**:26–35.
21. Krysl P, Belytschko T. Analysis of thin shells by the element-free Galerkin method. *International Journal of Solids and Structures* 1996; **33**:3057–3080.
22. Bobaru F, Mukherjee S. Shape sensitivity analysis and shape optimization using the element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, in press.
23. Belytschko T, Krongauz Y, Organ D, Fleming M, Krysl P. Meshless methods: an overview and recent developments. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:3–47.
24. Liu WK, Chen Y, Uras RA, Chang CT. Generalized multiple scale reproducing kernel particle methods. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:91–157.
25. Liu WK, Chen Y, Jun S, Belytschko T, Pan C, Uras RA, Chang CT. Overview and applications of the reproducing kernel particle methods. *Archives of Computer Methods in Engineering* 1996; **3**:3–80.
26. Mukherjee YX, Mukherjee S. The boundary node method for potential problems. *International Journal for Numerical Methods in Engineering* 1997; **40**:797–815.
27. Kothnur VS, Mukherjee S, Mukherjee YX. Two dimensional linear elasticity by the boundary node method. *International Journal of Solids and Structures* 1999; **36**:1129–1147.
28. Chati MK, Mukherjee S. The boundary node method for three-dimensional problems in potential theory. *International Journal for Numerical Methods in Engineering* 2000; **47**:1523–1547.
29. Chati MK, Mukherjee S, Mukherjee YX. The boundary node method for three-dimensional linear elasticity. *International Journal for Numerical Methods in Engineering* 1999; **46**:1163–1184.
30. Krishnaswamy G, Rizzo FJ, Rudolph TJ. Hypersingular boundary integral equations: their occurrence, interpretation, regularization and computation. In Banerjee PK, Kobayashi S (eds). *Developments in Boundary Element Methods*, vol. 7, Elsevier Applied Science: London, 1992; 207–252.
31. Tanaka M, Sladek V, Sladek J. Regularization techniques applied to boundary element methods. *ASME Applied Mechanics Reviews* 1994; **47**:457–499.
32. Chan Y-S, Fannjiang AC, Paulino GH. Integral equations with hypersingular kernels—theory and applications to fracture mechanics. Submitted.
33. Paulino GH. Novel formulations of the boundary element method for fracture mechanics and error estimation. *Ph.D. Dissertation*, Cornell University, Ithaca, NY, 1995.
34. Wilde AJ, Aliabadi MH. Direct evaluation of boundary stresses in the 3D BEM of elastostatics. *Communications in Numerical Methods in Engineering* 1998; **14**:505–517.
35. Zhao ZY, Lan SR. Boundary stress calculation—a comparison study. *Computers and Structures* 1999; **71**: 77–85.
36. Chati MK, Mukherjee S. Evaluation of gradients on the boundary using fully regularized hypersingular boundary integral equations. *Acta Mechanica* 1999; **135**:41–45.

37. Krishnaswamy G, Schmerr LW, Rudolphi TJ, Rizzo FJ. Hypersingular boundary integral equations: some applications in acoustics and elastic wave scattering. *ASME, Journal of Applied Mechanics* 1990; **57**: 404–414.
38. Lutz ED, Ingrassia AR, Gray LJ. Use of 'simple solutions' for boundary integral methods in elasticity and fracture analysis. *International Journal for Numerical Methods in Engineering* 1992; **35**:1737–1751.
39. Gray LJ, Paulino GH. Crack tip interpolation, revisited. *SIAM Journal of Applied Mathematics* 1998; **58**: 428–455.
40. Bonnet M. Regularized direct and indirect symmetric variational BIE formulations for three-dimensional elasticity. *Engineering Analysis with Boundary Elements* 1995; **15**:93–102.
41. Gray LJ, Paulino GH. Symmetric Galerkin boundary integral fracture analysis for plane orthotropic elasticity. *Computational Mechanics* 1997; **20**:26–33.
42. Gray LJ, Paulino GH. Symmetric Galerkin boundary integral formulation for interface and multizone problems. *International Journal for Numerical Methods in Engineering* 1997; **40**:3085–3101.
43. Phan A-V, Mukherjee S, Mayer JRR. The hypersingular boundary contour method for two-dimensional linear elasticity. *Acta Mechanica* 1998; **130**:209–225.
44. Mukherjee S, Mukherjee YX. The hypersingular boundary contour method for three-dimensional linear elasticity. *ASME Journal of Applied Mechanics* 1998; **65**:300–309.
45. Paulino GH, Gray LJ, Zirikian V. Hypersingular residuals—A new approach for error estimation in the boundary element method. *International Journal for Numerical Methods in Engineering* 1996; **39**:2005–2029.
46. Paulino GH, Gray LJ. Galerkin residuals for adaptive symmetric-Galerkin boundary element methods. *ASCE Journal of Engineering Mechanics* 1999; **125**:575–585.
47. Menon G. Hypersingular error estimates in boundary element methods. *M.S. Thesis*, Cornell University, Ithaca, NY, 1996.
48. Menon G, Paulino GH, Mukherjee S. Analysis of hypersingular residual error estimates in boundary element methods for potential problems. *Computer Methods in Applied Mechanics and Engineering* 1999; **173**: 449–473.
49. Chung HJ, Belytschko T. An error estimate in the EFG method. *Computational Mechanics* 1998; **21**:91–100.
50. Chati MK. Meshless standard and hypersingular boundary node method—applications in three-dimensional potential theory and linear elasticity. *Ph.D. Dissertation*, Cornell University, Ithaca, NY, 1999.
51. Chati MK, Mukherjee S, Paulino GH. The hypersingular boundary node method for three-dimensional potential theory and linear elasticity problems. Submitted.
52. Banerjee PK. *The Boundary Element Methods in Engineering* (2nd edn). McGraw Hill: New York, 1994.
53. Rizzo FJ. An integral equation approach to boundary value problems of classical elastostatics. *Quarterly of Applied Mathematics* 1967; **25**:83–95.
54. Zienkiewicz OC. *The Finite Element Method* (3rd edn). McGraw-Hill: New York, 1977.
55. Bathe KJ. *Finite Element Procedures in Engineering Analysis*. Prentice-Hall: Englewood Cliffs, NJ, 1976.
56. Nagarajan A, Mukherjee S. A mapping method for numerical evaluation of two-dimensional integrals with $1/r$ singularity. *Computational Mechanics* 1993; **12**:19–26.
57. Cruse TA, Richardson JD. Non-singular Somigliana stress identities in elasticity. *International Journal for Numerical Methods in Engineering* 1996; **39**:3273–3304.
58. Mukherjee YX, Mukherjee S. Error analysis and adaptivity in three-dimensional linear elasticity by the usual and hypersingular boundary contour methods. *International Journal of Solids and Structures* 2001; **38**:161–178.
59. Paulino GH, Menezes IFM, Cavalcante Neto JB, Martha LF. A methodology for adaptive finite element analysis: towards an integrated computational environment. *Computational Mechanics* 1999; **23**:361–388.
60. Paulino GH, Shi F, Mukherjee S, Ramesh P. Nodal sensitivities as error estimates in computational mechanics. *Acta Mechanica* 1997; **121**:191–213.
61. Mai-Duy N, Nguyen-Hong P, Tran-Cong T. A fast convergent iterative boundary element method on PVM cluster. *Engineering Analysis with Boundary Elements* 1998; **22**:307–316.
62. Davies AJ. Fine-grained parallel boundary elements. *Engineering Analysis with Boundary Elements* 1997; **19**:13–16.
63. Hsieh S-H, Paulino GH, Abel JF. Evaluation of automatic domain partitioning algorithms for parallel finite element analysis. *International Journal for Numerical Methods in Engineering* 1997; **40**:1025–1051.
64. Krishnamoorthy CS, Umesh KR. Adaptive mesh refinement for two-dimensional finite element stress analysis. *Computers and Structures* 1993; **48**:121–133.
65. Mosalam KM, Paulino GH. Towards adaptive nonlinear finite element analysis of reinforced concrete structures considering smeared cracking effects. Submitted.
66. Semeraro BD, Gray LJ. PVM implementation of the symmetric-Galerkin method. *Engineering Analysis with Boundary Elements* 1997; **19**:67–72.