Engineering Analysis with Boundary Elements 34 (2010) 549–556

Contents lists available at ScienceDirect

Engineering Analysis with Boundary Elements

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

An *hpr*-mesh refinement algorithm for BEM

Jaihind Reddy Maddi, Madhukar Vable *

Mechanical Engineering—Engineering Mechanics, Michigan Technological University, Houghton, MI 49931, USA

ARTICLE INFO

Article history:
Received 28 October 2009
Accepted 6 January 2010
Available online 18 February 2010

Keywords:
Boundary element method
Mesh refinement

ABSTRACT

This paper describes a mesh refinement scheme for boundary element method in which the number of elements, the size of elements, element end locations and the element polynomial order are determined to meet the user specified accuracy. The use of grading function in conjunction with the L_1 norm makes the mesh refinement scheme applicable to a variety of boundary element formulation and applications. The algorithm is stable for smooth, discontinuous, as well as singular density functions. Numerical results for mathematical test functions as well as for elastostatic problems demonstrate the viability and versatility of the algorithm for BEM applications.

1. Introduction

Computational tools like finite element method (FEM) and boundary element method (BEM) have become integral part of engineering design. Efforts to improve the accuracy of the analysis by refining the mesh have been going on since late seventies [1]. The mesh refinement schemes in BEM, like in FEM, can be classified [2] into the *h*, *p*, and *r* refinement and their combination. In the *h*-method, the accuracy is improved by increasing the total number of elements, but the order of the polynomial remains invariant. The high computational cost associated with the *h*-method due to the large number of unknowns can be reduced to some extent by the *h*-hierarchical refinement scheme [3]. In the *p*-method the polynomial order is increased uniformly or selectively to improve the accuracy while keeping the node location and element length unchanged during the iterative process [4]. The convergence rate of *p*-method is always better than the *h*-method for smooth functions [5] but may not converge near a singularity [6]. In the *r*-method [7], the total number of elements and the order of the polynomial are kept invariant, but the spacing of the elements is adjusted to minimize the error. If the initial mesh does not have sufficient degrees of freedom (DOF) then the desired accuracy cannot be obtained with the *r*-method [8].

The *hp*-method is a combination of the *h* and *p*-method that overcomes the convergence difficulty of the *p*-method near the singularity. Initial papers published on *p* and *hp* mesh refinement method were mostly by Babuska and his colleagues [9]. Most of Babuska work was done in relation to FEM and later extended to BEM [10]. In FEM he showed that mesh generated using grading function performs better than a graded mesh [9]. In FEM he has shown that optimal meshes are graded towards the singular point and starting with the second element away from singularity the polynomial order increases linearly. He has shown that for the Galerkin BEM, *hp*-method, with geometric mesh graded towards the point of singularity, has exponential rate of convergence [11]. In some cases the location [5] and the strength of the singularity [10] has to be specified. The *p*-method and the *hp*-method are usually restricted to the Galerkin method of satisfying the boundary conditions in BEM due to the difficulty of determining the collocation points with the increase in polynomial order.

Sun and Zamani [8] developed a *hr*-method for direct BEM. Finding residuals is computationally expensive in case of indirect BEM and thus the use of residuals as an error indicator makes the method suitable primarily for the direct BEM. Ammons and Vable [12] developed a *hr*-method which can be used for both direct and indirect BEM by using grading function [7] and L_1 norm. The *hr*-method of Ammons and Vable has very slow convergence rate near discontinuities and singularities in density function and can have very large DOF if a lower order polynomial is specified by the user.

This paper presents an *hpr*-method that has fast convergence for non-singular density functions and is stable for discontinuous and singular density functions. The use of grading function with L_1 norm as an error measure makes it possible to use the refinement scheme for approximation of any mathematical function of one variable and thus is independent of BEM methodology and its application. The three functions used for identification of critical ideas and for testing the algorithm are:

$$\text{Function 1}: \quad u = \sin(\pi s) \quad 0 \le s \le 1 \tag{1a}$$

$$\text{Function 2}: \quad u = \sqrt{s} \quad 0 \le s \le 1 \tag{1b}$$

$$\text{Function 3}: \quad u = 1/\sqrt{s} \quad 0 \le s \le 1 \tag{1c}$$

* Corresponding author. Tel.: +1 906 487 2747; fax: +1 906 487 2822.
E-mail address: mavable@mtu.edu (M. Vable).

0955-7997/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.enganabound.2010.01.004

The mathematical functions have counterparts in the problems considered for demonstrating the application of the algorithm in BEM. The three BEM problems considered are:

(i) A circular hole in an infinite plane under uniaxial tension modelled using direct BEM. This has a smooth density function analogous to function 1 in Eq. (1a).
(ii) A pressured crack in an infinite plane. Displacement discontinuity is used for modelling the crack. It has a singular slope at the crack tip analogous to function 2 in Eq. (1b).
(iii) A square under uniaxial tension is modelled by indirect BEM. The density function is a discontinuous and can be singular at each corner analogous to function 3 in Eq. (1c).

The various formulations of BEM, the errors that arise from the discretization, and the algorithms that reduce these errors are discussed elsewhere by Vable [13] and are not repeated here. A program called BEAMUP[1] has been built at Michigan Technological University that incorporates the various algorithms that reduce the errors of discretization. BEAMUP will be used for BEM analysis to get density functions used in this paper.

2. *hr*-mesh refinement

The *hr*-mesh refinement scheme of Ammons and Vable [12] that uses the grading function and L_1 norm will be modified for the proposed *hpr*-mesh refinement scheme in this paper. This section briefly describes the basic equations and ideas needed for this paper.

A grading function $G(s)$ is a monotonic function varying from zero to one that maps the density function $u(s)$ such that

$$G(s_i) = \frac{i}{N} \quad 0 \le i \le N \tag{2}$$

where s_i represent the arc coordinate of the ith element end node and N represents the number of elements. The objective of the *hr*-mesh refinement scheme is to determine s_i and N such that the user specified average error per unit length e_{user} is achieved.

In this paper the error of a mesh is measured using the L_1 norm given in Eq. (3).

$$e_R = \sum_{i=1}^{N} \int_{s_{i-1}}^{s_i} |u(s) - p(s)| ds \tag{3}$$

where $u(s)$ is the density function and $p(s)$ is the polynomial approximation of the density function on the ith element. It can be shown [12] that the grading function as determined in Eq. (4) will minimize the L_1 error.

$$G(s) = \frac{G^*(s)}{G^*(s_N)} \quad \text{where} \quad G^*(s) = \int_{s_0}^{s} |u^{(k+1)}(s))|^{\frac{1}{k+2}} ds \tag{4}$$

where $u^{(k+1)}(s)$ is the $k+1$ derivative of $u(s)$, k is the order of polynomial used in approximating the density function, and s_N is the value of s at the last node on the boundary.

From Eq. (2) we note that $G(s_j) - G(s_{j-1}) = 1/N$, thus in an optimal mesh Eq. (4) implies the change in $G^*(s)$ is equal in all elements. Hence, for the jth element

$$\Delta G_j^* = G^*(s_j) - G^*(s_j - 1) = \frac{G^*(s_N)}{N} \tag{5}$$

The error in each element (e_{R_j}) can be found and summed over all elements to obtain the error in the old mesh as e_R shown below.

$$e_{R_j} = D_k (\Delta G_j^*)^{k+2} \tag{6a}$$

$$e_R = \sum_{j=1}^{N} e_{R_j} \tag{6b}$$

where D_k is a constant that depends only on the polynomial order [12,14].

2.1. *hr*-mesh refinement algorithm

The description here is conceptual with details for implementation described in Ref. [12]. The number of elements N_{new} needed for the next iteration can be found [12] from the equations below

$$N_{new} = \left(\frac{D_k}{e_{user}}\right)^{\frac{1}{k+1}} (G^*(s_N))^{\frac{k+2}{k+1}} \tag{7}$$

A boundary is considered made up of sub-boundaries. Corners or a change in type of boundary condition defines the ends of a sub-boundary. In the *hr*-mesh refinement scheme, it is assumed that all elements on a sub-boundary have the same polynomial order and continuity except the end elements where a discontinuity [15] is permitted in the density function. In other words, D_k is constant on a sub-boundary. The steps of the *hr*-mesh refinement algorithm are described below.

1. From BEM analysis (or from the mathematical functions in Eq. (1a)–(1c)) the density function $u(s_i)$ is found at collocation points.
2. A cubic spline is constructed through the nodal values of the density function.
3. The integral in Eq. (4) is evaluated numerically to obtain $G^*(s_i)$ at the nodal points of the old mesh.
4. Knowing the value of $G^*(s_N)$ the grading function $G(s_i)$ is found from Eq. (4) and number of elements for the new mesh is found from Eq. (7).
5. A cubic spline is constructed for $G(s)$ from the nodal values in step 4.
6. The ends of the element s_i for the new mesh are found by determining the roots of Eq. (2).
7. Steps 1 through 6 are repeated till mesh error e_R in Eq. (7) becomes less than or equal to the user error e_{user} or the limit on maximum iterations is met.

The *hr*-mesh refinement algorithm is applicable to both the direct and the indirect BEM and is independent of BEM application in two dimension (elastostatic, Poisson's equation, and plate bending). Eqs. (2)–(6a) are applicable to any order of polynomials and the program BEAMUP can incorporate polynomials up to order 15. However, the use of cubic spline to model the boundary data limits the mesh refinement schemes to cubic polynomials.

3. New error measure

Eqs. (5) and (6a) imply that the error in each element is equal for an optimum mesh. Thus, non-uniform distribution of error is an indicator of a non-optimum mesh. Convergence in the *hr*-refinement scheme described in Section 2 is assumed to occur when the total average error per unit length on the boundary e_R is less than or equal to the user specified error e_{user}. Thus, it is possible that on part of boundary the average error is less than the user specified and on other parts it is more, producing a non-uniform distribution of error. Results in Ref. [12] showed that near

[1] Details of BEAMUP can be found on http://www.me.mtu.edu/~mavable/BEAMUP/index.html.