Transactions on Modelling and Simulation vol 22, © 1999 WIT Press, www.witpress.com, ISSN 1743-355X

Minimizing some of the mesh errors in boundary element method

Madhukar Vable[(1)] and Bruce A. Ammons[(2)]
[(1)]*Mechanical-Engineering and Engineering Mechanics, Michigan Technological University, Houghton, MI 49931, USA*
E-mail: mvable@mtu.edu
[(2)]*Dateppli Inc., 3333 Patrick Road, Midland, MI 48642, USA*
E-mail: bammons@dateppli.com

Abstract

The discretization of the boundary in Boundary Element Method (BEM) requires several decisions that affect the accuracy of the BEM solution. These decisions include: the order of polynomial in each element, the continuity requirement at the element end, the location of the nodes inside the element, the size of the element, the location of the element end nodes, and the location of collocation points where the boundary conditions are imposed. The errors that are generated from these decisions are referred to as the mesh errors in this paper.

In Ammons and Vable[1] the errors from continuity and collocation were discussed in detail and will not be considered here. But how to numerically determine an interpolation functions of a given order to satisfy a given continuity requirement will be discussed. The presentation will briefly describe three algorithms that minimize the L_1 norm of the mesh error from the remaining sources. These algorithms are applicable two-dimensional problems of Poisson's equation, Plate-Bending, Elastostatic, and Fracture mechanics formulated using direct or indirect BEM. Numerical examples will be presented showing the validity of these algorithms.

1 Introduction

In Boundary Element Method (BEM) a quantity of interest (temperature, displacement, stresses, etc.) is represented by an integral over the boundary. The integrands in the integral are products of influence functions and density functions. The boundary is sub-divided into elements over which the density functions are approximated by polynomials. The continuity requirement on the density function depends upon the order of singularity in the influence function. If the order of singularity is n, then the density function should be $C^{(n-1)}$ continuous at a regular boundary points to ensure that the quantities of interest are bounded at all points including the element ends. In a previous paper by the authors[1] the effect of satis-

fying or violating the continuity conditions at the element ends were discussed. In this paper we address the issue of generating optimum interpolation functions of different orders to meet different continuity requirements at the element end and hr-method of mesh refinement[2] which address the mesh errors that are generated by the user's decisions in creating the mesh.

Computer codes for BEM usually have an element library where each element type has a program segment in which analytical expressions for the interpolation functions are written. An element is classified by the order of the approximating polynomial and the degree of continuity that must be ensured at each element end. In addition, the base points in the interpolation functions are either uniform[5,6] or located to facilitate hierarchical elements[7,8]. In this paper an alternative is presented.

The interpolation functions for one-dimension will be numerically generated to satisfy user specified polynomial order and continuity requirement at each element end. The continuity requirement at each element end can be different. To develop the algorithm to generate the interpolation functions, each interpolation function is written as a power series and the coefficients in the power series are determined to satisfy the interpolation function property that the interpolation function value is one at the associated nodal value and zero at the other nodes. The algorithm should help produce compact computer programs providing greater flexibility in the choice of elements to the user. The one dimensional interpolation functions can be used to generate two and three dimensional interpolation functions by tensor products[6]. But this extension is limited to rectangular master elements in two dimension and cubical master elements in three dimension.

The base point location can be uniform, or correspond to hierarchical elements, or chosen as per an algorithm described in this paper to minimize the error in an element as measured by the L_1 norm. The reduction in error per element should result in faster convergence in mesh refinement and non-linear applications that solve problems iteratively.

An hr-mesh refinement[2] scheme minimizes the L_1 norm to obtain the optimum number, location and size of elements. Numerical examples are reported that validate the algorithms.

2 Numerical Generation of Interpolation Functions

We develop the interpolation function for an element of unit length in natural coordinate and then scale it to obtain interpolation functions for elements of different lengths. The approximating polynomial p(x) can be written as

$$p(x) = \sum_{i=0}^{k} F_i \; f_i(x) \qquad 0 \le x \le 1 \tag{1}$$

where, k is the order of polynomial, F_i are the nodal values of p(x) and its derivatives, and f_i the associated interpolation functions (shape functions).
Figure 1shows the relationship between F_i and the nodal value of the ith derivative of p(x) represented by $p^{(i)}$. N_0 and N_1 are the number of degrees of freedom which are shared with adjacent elements at x=0 and x=1, respectively. These correspond to continuity of the function, slope, and higher order derivatives. For example, $N_0 = 0$ indicates the function is discontinuous at x=0, and $N_1 = 2$ indicates the function and slope are continuous at x=1. N_M represent the number of additional

degrees of freedom within the element. This represents additional points within the element at which the value of the function is defined. x_i represent the location of the N_M base points within the element. The location of these points can either be uniformly spaced, located to facilitate hierarchical elements, or selected using an algorithm to minimize the L_1 norm as described in the next section. The polynomial order k can be written as $k = N_0 + N_M + N_1 - 1$.

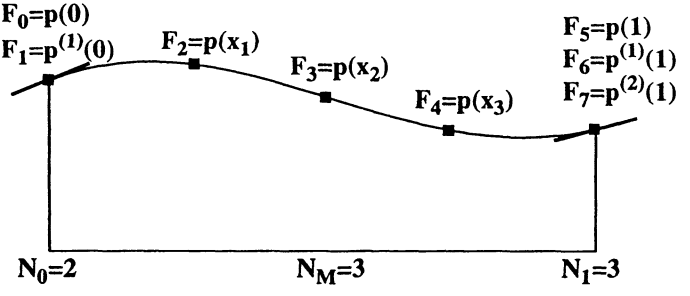

Figure 1: Example of Nodal Values for Interpolating Polynomial.

The objective is to generate the interpolation functions $f_i(x)$ that meet the user specified order of polynomial k and the specified continuity at the end (N_0 and N_1).

The key ideas in the derivation are briefly outlined below.

$$f_n(x) = \sum_{j=0}^{k} C_{nj}\ x^j \tag{2}$$

where, C_{nj} is a matrix corresponding to the jth coefficient of the nth interpolation polynomial. To find the coefficients we use the properties of the interpolation function: it is equal to one for its own nodal value, and equal to zero for all other nodal values. This leeds to the following equation:

$$[C][X] = [I] \tag{3}$$

The matrix [C] contains the unknown coefficients of the interpolation functions that are to be determined. The matrix [I] is the identity matrix. The coefficient of matrix [X] are zero except for the elements shown below:

$$\begin{aligned} X_{ii} &= i! && 0 \le i < N_0 - 1 \\ X_{ji} &= x_i^j && 0 \le i < N_M - 1 \quad 0 \le j \le k \\ X_{ji} &= j!/(j-i)! && 0 \le i < N_1 - 1 \quad i \le j \le k \end{aligned} \tag{4}$$

where, i! represents the factorial of i. Equation (3) can be solved to obtain the coefficient of matrix [C]. The matrix [X] depends upon N_0, N_1, N_M and x_i. The numbers N_0, N_1, N_M define a type of element for a given continuity at element end and a given order of polynomial. The base points are so far arbitrary and could be uniformly located, or be at positions to define hierarchical elements, or chosen to minimize the error of approximation as described in next section. Thus, the matrix [C]

and the associated interpolation functions are that of a master element. In other words, if only a single type of element is used in the mesh then the coefficients of matrix [C] need to be determined only once in the program. The derivatives of the density function are tangential derivatives and hence by scaling the column of matrix [C] by h^i corresponding to the ith derivative the interpolation functions for each element can be obtained.

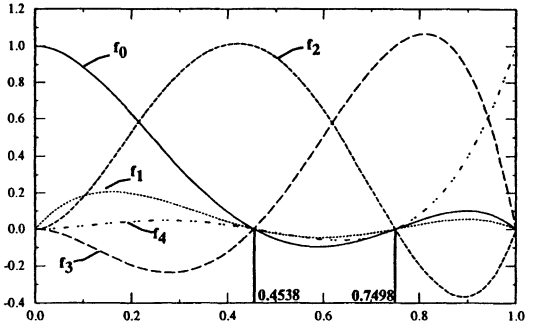

Figure 2: Interpolation Functions for N_0=2, N_M=2, and N_1=1.

Consider an element for which a fourth order polynomial is needed with function and slope continuity required at x=0 and function continuity required at x=1. Thus N_0=2, N_1=1 and k=4 is specified by the user. Using the algorithm described in next section, the base points were selected at x_0=0.4538 and x_1=0.7498. Figure 2 shows the interpolation functions generated using the algorithm described as above, with f_1 scaled for an element of length of 3 for visibility.

We see that at x=0, f_0 equals one, and all other interpolation functions are zero. Likewise, at x=0 the slope of f_1 equals three (due to scaling), and the slopes of all other functions are zero. At the other base points, the corresponding interpolation function equals one and all others equal zero.

3 Minimizing error inside an element

Let u(x) represent the density function being approximated by the polynomial p(x). The error function e(x)=u(x)-p(x) can be written[9] as

$$e(x) = \frac{u^{(k+1)}(\xi_x)}{(k+1)!} \prod_{i=0}^{k} (x - c_i) \tag{5}$$

where, $u^{(k+1)}(\xi_x)$ is the k+1 derivative of u(x) evaluated at some unknown point ξ_x within the element. Point c_i are collocation points where $e(c_i) = 0$. We shall assume that the collocation points and base points are the same i.e., $c_i = x_i$. The derivation is presented for the case in which the approximating function can be discontinuous at the ends i.e., $N_0 = 0$ and $N_1 = 0$. The derivation can be extended to include the case in which continuity of the function and/or its derivative must be met at the element end.

The L_1 norm on the element $\|e\| = \int_0^1 |e(x)|dx$ can be written as:

$$\|e\| \le \left(\max_{0 \le x \le 1} \left| \frac{u^{(k+1)}(\xi_x)}{(k+1)!} \right| \right) \left(\int_0^1 \left| \prod_{i=0}^{k} (x - x_i) \right| dx \right) \tag{6}$$

Within an element the first term in Equation (6) is independent of the location of the base points x_i. Minimizing the contribution of the first term towards the error is briefly discussed in Section 4. Letting P(x) represent the product in the second term we seek to minimize $E = \int_0^1 |P(x)|dx$ with respect to x_n, i.e., $\frac{\partial E}{\partial x_n} = 0$. Noting that the function P(x) changes signs at each base point, we obtain:

$$\left[\int_0^{x_1} R_n(x)dx + \sum_{j=1}^{N_M - 1} \int_{x_j}^{x_{j+1}} (-1)^j R_n(x)dx + \int_{x_{N_M}}^{1} (-1)^{N_M} R_n(x)dx \right] = 0 \tag{7}$$

where, $R_n(x) = \frac{\partial}{\partial x_n} P(x) = -\prod_{\substack{i=1 \\ i \ne n}}^{N_M} (x - x_i)$. The following notation is introduced for ease of explanation. The function $U_n(x)$ is the integral of the function $R_n(x)$, and A_j is the absolute value of the area under the curve $R_n(x)$ for the jth interval, as shown in Figure 3.

$$U_n(x) = \int_0^x R_n(x)dx \qquad A_j = \left| U_n(x_{j+1}) - U_n(x_j) \right| \quad 1 \le j < N_M$$
$$A_0 = \left| U_n(x_1) - U_n(0) \right| \qquad A_{N_M} = \left| U_n(1) - U_n(x_{N_M}) \right| \tag{8}$$

It should be noted that $R_n(x)$ changes signs at every base point except for x_n, which has been removed from the product term. Thus the area under the curve of R_n changes signs for each interval except for the intervals adjacent to x_n, as shown in Figure 3. The term $(-1)^j$ in Equation (7) changes sign in every interval. Thus, all of the terms to the left of x_n in Equation (7) have one sign, and all the terms to the right of x_n have the opposite sign. Based on this observation, we can interpret Equation (7) as: *The optimum location of x_n is such that the sum of the areas to the left of x_n is equal to the sum of the areas to the right of x_n.* In the example shown in Figure 3, this implies that the optimum location of x_3 will result in $A_0 + A_1 + A_2 = A_3 + A_4$.

We intent to find the N_M base points iteratively. For a given iteration, the difference between the sum of the areas on left of x_n and the sum of areas on right of x_n will not be zero. Let this difference be equal to Δ_n. Equation may thus be written in the following form.

$$\sum_{j=0}^{n-1} A_j - \sum_{j=n}^{k} A_j = \Delta_n \tag{9}$$

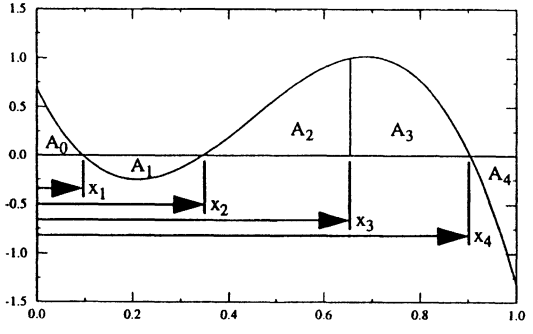

Figure 3: $R_3(x)$ for k=3 and N_0=N_1=0

The objective of the algorithm is to find the set of x_n points such that all Δ_n=0 for $1 \le n \le N_M$. It should be noted that A_2+A_3 in Figure 3 is independent of the location of x_3. In other words, the following equation does not change with variation of location of $x_n$when the other base points are fixed

$$A_{n-1} + A_n = constant \tag{10}$$

We could add (Δ_n/2) to A_{n-1} and subtract from A_n and still satisfy Equations (9) and (10). We want to change the location of x_n such that the new area between x_n and x_{n-1} is (A_{n-1}+Δ_n/2). This may be stated as:

$$U_n(\bar{x}_n) - U(x_n) + \Delta_n/2 = 0 \tag{11}$$

where, $\bar{x}_n$ is the new location of x_n. Thus $\bar{x}_n$ represents the root of the function given by the left hand side of Equation (11) and it can be found by number of methods. Newton Raphson method is used by the authors, since both $U_n(x)$ and its derivative $R_n(x)$ are known.

Other cases for which $N_o \ge 1$ and /or $N_1 \ge 1$ result in a different expressions for P(x) and will be presented at the talk. But in all cases the optimum location of base points is determined by finding the roots of Equation (11).

4 Minimizing Error On the Boundary

Briefly described here are the basic ideas of the authors paper[2] for minimizing the error on the boundary for completeness. The L_1 norm of the error on the boundary can be obtained by adding the error given by Equation (6) for all elements on the boundary. To minimize this total error on the boundary, the density function was mapped using a grading function. A grading function[11] is a monotonic function with $G(s_0) = 0$ and $G(s_N) = 1$ that maps the density function $u(s_i)$ such that

$G(s_i) = \frac{i}{N}$, where, N are the number of element on the boundary and s_i is the arc length coordinate of the starting point of the ith element. The algorithm found the location s_i and N in the grading function such that the user specified value of the error on the boundary was minimized. The hr-algorithm of the paper is independent of the boundary element formulation and can be used for Direct and Indirect BEM applied to elastostatics, fracture mechanics, Poisson's equation and plate bending problems.

5 Numerical Results

Example 1

This example is a test for validation of algorithm described in Section 3. The optimum location for collocation points for discontinuous elements (N_0=0 and N_1=0) correspond to the roots of the Chebyshev polynomial of the second kind[10], which can be calculated using Equation 12 given below.

$$x_i = \left[1 - \cos\left(\frac{i}{N_M + 1}\pi\right)\right]/2 \qquad 1 \le i \le N_M \qquad (12)$$

The base points were initialized with uniformly spaced points. Results for optimum location as obtained from the algorithm described in Section 3 were compared to the roots of the Chebyshev polynomial given in Equation 12 for different orders of polynomials. Table 1 list the number of mid points N_M, the number of iterations needed for convergence, and the difference between location obtained from Equation 12 to that obtained from the algorithm in Section 3. Table 1 show that even for 16 degrees of freedom the algorithm converges very rapidly to the analytical solution of Equation 12. Similar rapid convergence was obtained for 32 degrees of freedom when the coordinate range was changed to $-1 \le x \le 1$. It should be emphasized that the optimum location algorithm is invoked once per master element which may be only once in the program if a single type of element is used in the mesh.

Table 1: Comparison to Chebyshev Roots for Discontinuous Elements.

N_M	# Iterations	Condition Number	Max. Difference
2	1	7.00	111×10^{-18}
4	4	$320 \times 10^{+3}$	333×10^{-18}
8	5	$410 \times 10^{+6}$	11×10^{-12}
16	7	$586 \times 10^{+9}$	65×10^{-9}

Example 2

As a first step a 100 x 100 square subjected to a unit value of uniform uniaxial tension was taken. Each side of the square was modeled by 12 linear elements and the problem solved by the indirect BEM with force discontinuity. Stresses in the central region of 2x2 square were computed and no error was seen up to six places

of decimals. Thus the square represented an infinite plate subjected to a unit value of uniform uniaxial tension. The hole boundary was represented by 6 elements and integration was performed using the semi-analytical integration scheme in which the integration path is dynamically created[3]. Lagrange polynomials of different order were considered as shown in Table 2 . Then Hermite polynomials with different order of continuity ($C^{(n)}$) at each element end were considered. For each case the stress concentration factor was computed and compared with the analytical value of three and the percentage error is shown in column four of Table 2 . Also reported is the matrix conditioning number.

Table 2: Results of Example 2

Polynomial			Number of Unknowns on Hole Boundary	%error in Stress Concentration Factor	Matrix Condition Number
Type	Order	$C^{(n)}$			
Lagrange	1	0	6	-10.367	4553
	2	0	12	0.933	5026
	3	0	18	0.193	5353
	4	0	24	0.150	5480
	6	0	36	0.153	5701
	8	0	48	-1.1	1.0×10^{10}
Hermite	3	1	12	0.057	3143
	5	2	18	0.068	6548
	7	3	24	0.077	2.4×10^{6}

Though the primary purpose of this example is to demonstrate the validity of the algorithms in Sections 2 and 3, results show some interesting observations. Initially the accuracy improves as the order of Lagrange polynomial increases, but after the 4th order it starts decreasing even though there is no significant change in condition number from 4th order to 6th order. Another observation is that for the same number of unknowns the Hermite Polynomials give an order of magnitude better accuracy than the Lagrange polynomials. It should be emphasized that Lagrange polynomials meet the continuity requirements on the density functions as the highest order of singularity in the influence function associated with force discontinuity is one. The high condition numbers for 8th order Lagrange polynomial is surprising, as outside BEM the algorithm of Sections 2 and 3 showed these type of condition number around the 16th order of polynomials (see Table 1). The most likely cause is the size of circle boundary in comparison to the square boundary. To check this a circular hole in infinite body (no 100x100 square) under uniform pressure was solved and with six elements and eighth order Lagrange polynomial. The matrix condition number was found to be only 3.03. Similarly when a seventh

order Hermite polynomial with $C^{(3)}$ continuity was used the matrix condition number was found to be 15,769. Thus, the large matrix conditioning number that arises when using the algorithms of Sections 2 and 3 with high order polynomial is dependent upon the dimensions of the boundary relative to dimensions of the outer most boundary. Though the above observations merit further study it needs to be emphasized that these observations may be uniquely tied to this particular problem.

Example 3

Once more a 100x100 square under uniaxial tension in the y-direction was modeled using 12 linear elements per side. Using the Direct BEM stresses in the central region of 2x2 square was checked and no error was found up to six decimal places. A crack of 2 unit length was introduced and modeled using the displacement discontinuity. Boundary conditions on integrated tractions were imposed[4]. The displacement discontinuity density function was modeled using linear, quadratic, and cubic Lagrange polynomials and the hr-method described in Section 4 was used for refining the mesh for three iterations for each polynomial. In authors previous papers[2,4], high accuracies were obtained for stress intensity factors using few unknowns—a consequence that stresses away from the crack tip can be used in the calculation of the factor. In this paper we report the stresses in front of the crack, and the percentage difference in σ_{yy} with the analytical series expression[12] of σ_{yy} given below was computed and plotted as shown in Figure 4.

$$\frac{\sigma_{yy}}{\sigma_\infty} = \sqrt{\frac{2a}{r}}\left[1 + \left(\frac{3}{4}\right)\left(\frac{r}{a}\right) - \left(\frac{5}{32}\right)\left(\frac{r}{a}\right)^2 + \left(\frac{7}{128}\right)\left(\frac{r}{a}\right)^3 + \cdots\right] \qquad (13)$$

where, r is the radial distance from the crack tip and a is the half-crack length.

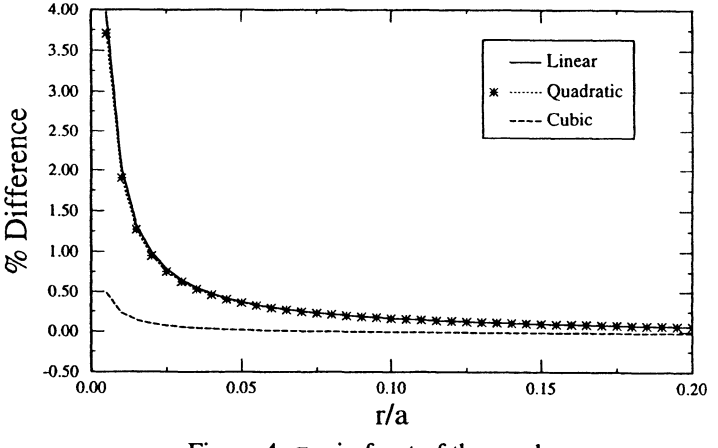

Figure 4: σ_{yy} in front of the crack

Figure 4 shows that stress values for all polynomial approximation rapidly converge towards each other away from the crack tip. But at a distance of less than 0.025a, only the cubic interpolation gives good results. This accuracy issue can be critical if one was trying to determine the plastic zone in front of a crack. The number of unknowns used in modeling the crack for linear, quadratic and cubic

Lagrange interpolation were 442, 136, and 112, respectively. The results highlight that a faster convergence will be achieved by using a higher order of polynomial for modeling the displacement discontinuity density function, which is not surprising as the density function on the crack is a smooth function. But it is possible that further increases in the order of polynomial order may not give better accuracy as in previous example. The only way to resolve the appropriate order of polynomial approximation is to design a full hrp-mesh refinement algorithm.

6 Conclusions

The paper briefly described three algorithms for minimizing the mesh error. The first algorithm is used for generating interpolation functions to satisfy user specified polynomial order and continuity requirements. The second algorithm found optimum location of collocation points. The third algorithm is an hr-method of mesh refinement that finds the optimum number, location, and size of elements. Numerical results validate these algorithms but also point the need to develop a full hrp-mesh refinement scheme.

References

1. Ammons, B. A. & Vable, M. Continuity and Collocation effects in the boundary element method. *IJNME,* **40**, pp 1877-1891, 1997.
2. Ammons B.A. & Vable M. An hr-method of Mesh Refinement for Boundary Element Method *IJNME*, **43**, pp 979-996, 1998.
3. Ammons B.A. & Vable M. A Dynamic Algorithm for Integration in the Boundary Element Method. *IJNME*,**41**, pp 639-650,1998.
4. Ammons B.A. & Vable M. Boundary Element Analysis of Cracks. *Int. J. for Solids and Structures*, **33**, No. 13, pp. 1853-1865, 1996.
5. Brebbia, C.A. and Walker, S. *Boundary Element Techniques in Engineering*, Newnes-Butterworths, Boston, 1980.
6. Reddy, J.N. *An Introduction to the Finite Element Method*, McGraw Hill, New York, 1993.
7. Guiggiani, M. Hypersingular integral equations and super accurate stress evaluation. *Boundary Element XV*, eds. C.A. Brebbia & J.J. Rencis, Computational Mechanics Publications, Southampton, 1993.
8. Bathe, K.J., *Finite Element Procedures*, Prentice Hall, New Jersey, 1996.
9. Davis, P. (1963). *Interpolation and Approximation*, Blaisdell Publishing Company, New York.
10. Kincaid, D. and Cheney, W. *Numerical Analysis*, Brooks / Cole, 1991.
11. Carey, G. and Dinh, H. Grading functions and mesh redistribution, *SIAM J. Num. Anal.* **22**, 1028-1040, 1985.
12. Unger, D.J. *Analytical Fracture Mechanics*, Academic Press, San Diego, 1995.