Basics of a fast-multipole unified technique for the analysis of several classes of continuum mechanics problems with the boundary element method

H. de Farias Costa Peixoto, L. Simões Novelino & N. A. Dumont
Civil Engineering Department,
Pontifical Catholic University of Rio de Janeiro, Brazil

Abstract

The proposed implementations are based on a consistent development of the conventional, collocation boundary element method (BEM) – with concepts taken from the variationally-based hybrid BEM – for large-scale 2D and 3D problems of potential and elasticity. The formulation is especially advantageous for problems of complicated topology or requiring complicated fundamental solutions. This paper, which is the sequel of a first paper presented at the PACAM 2014 Conference in Santiago, Chile, proposes a scheme for expansions of a generic fundamental solution about hierarchical levels of source and field poles. This makes the fast multipole technique directly applicable to different kinds of potential and elasticity problems with generally curved boundaries. The basic concept of the FMM, with the expansion of the fundamental solution about successive layers of source and field poles, is described in a compact algorithm that is more straightforward to lay out and seems to be more efficient than the ones available in the technical literature. The hierarchical tree of poles is built upon a topological concept of superelements inside superelements. The formulation is initially assessed and validated in terms of a simple 2D potential problem. Since iterative solvers are not required in this first step of numerical simulations, an isolated efficiency assessment of the implemented fast multipole technique is possible.
Keywords: hybrid boundary elements, fast multipole method, variational methods.

WIT Transactions on Modelling and Simulation, Vol 61, © 2015 WIT Press
www.witpress.com, ISSN 1743-355X (on-line)
doi:10.2495/BEM380041

1 Introduction

The fast multipole method (FMM), as developed by Greengard and Rokhlin in 1987 [1], was elected one of the top ten algorithms of the 20th century [2]. Although it was initially conceived for particle simulation problems involving coulombic and gravitational fields, it also turned out to be efficient in the solution of boundary integral equations [3]. The FMM, combined with an iterative solver such as the GMRES, can speed up the complete solution time of a problem with N unknowns from order $O(N^2)$ to $O(N \log N)$ – or even $O(N)$ [4] – while requiring computer storage that is only a small fraction of what would be allocated for a different numerical method N log N.

Liu [4] presents a thorough description of the FMM as applied to different types of BEM problems. A short review of the method is given by Liu *et al*. [5], a comprehensive review is given by Nishimura [3] and a tutorial has been prepared by Liu and Nishimura [6].

The present research work is part of the Ph.D and M.Sc. works of the two first authors [7, 8], and is mainly concerned with the application of the FMM to problems with generally curved boundaries, in a framework that is almost completely independent from the underlying fundamental solution. Moreover, some advantageous features of the hybrid BEM, which has a variational basis, are also explored. Not least, the basic concept of the FMM, with the expansion of the fundamental solution about successive layers of source and field poles, is described in a compact algorithm that is more straightforward to lay out and promises to be more efficient than the ones available in the technical literature [9]. Owing to space restrictions only the basic aspects of the proposed implementations [8] are described in the present paper, with a more extensive manuscript being prepared for publication in the near future.

2 Proposed FM algorithm for a general, complex function

The following basic definitions are used in the present developments:

- $z - z_0$ = difference between the source point z_0 and the field point z.
- z_c = point about which the fundamental solution will be expanded for the field point z. Expansions about successively farther poles z_{c^k}, $k = 1, 2, ...n_c$ (where, by definition $z_{c^0} \equiv z_c$) are also undertaken.
- z_L = point about which the fundamental solution will be expanded for the source point z_0. Expansions about successively farther poles z_{L^l}, $l = 1, 2, ...n_L$ (where, by definition $z_{L^0} \equiv z_L$) are also undertaken.

A generic fundamental solution for 2D problems can be expanded as [7–9]

$$f(z - z_0) = \sum_{i=1}^{n+1} \frac{1}{(i-1)!} P_i(z - z_{c^{n_k}}) \sum_{j=1}^{n+1} \frac{1}{(j-1)!} P_j(z_{L^{n_l}} - z_0) Q_{i+j-1}(z_{c^{n_k}} - z_{L^{n_l}}), \quad (1)$$

for truncation order given by $\max\left(\left|(z-z_{c^{n_k}})/(z-z_0)\right|^{n+1}, \left|(z_{L^{n_l}}-z_0)/(z-z_0)\right|^{n+1}\right)$ and with the arrays of functions $P(z)$ and $Q(z)$ defined as

$$P(z) = [1 \quad z \quad z^2 \quad z^3 \quad \cdots \quad z^{n+1}] \tag{2}$$

$$Q(z) = [f(z) \quad \partial f(z)/\partial z \quad \partial f^2(z)/\partial z^2 \quad \partial f^3(z)/\partial z^3 \quad \cdots \quad \partial f^{n+1}(z)/\partial z^{n+1}]. \tag{3}$$

In general, the higher derivatives of the fundamental solution tend rapidly to zero when evaluated for large arguments.

Equation (1) is the starting point for a procedure that leads to a computationally fast and economical evaluation of a given fundamental solution $f(z-z_0)$ for a very large number of source points z_0 and of field points z by means of a sufficiently approximate expression. As shown, $f(z-z_0)$ is expanded in terms of successive arrays of source poles z_{L^l} as well as field poles z_{c^k}. The indicated poles z_{L^l} and z_{c^k} are representative of arrays of poles corresponding to hierarchical expansion levels l and k. The expansion ends up with a series of products of binomials $P_i(z-z_{c^{n_k}})$ and $P_i(z_0-z_{L^{n_l}})$, which are independent from the complexity of the function $f(z-z_0)$, multiplied by functions $Q_i(z_{L^{n_l}}-z_{c^{n_k}})$ that are given as $f(z_{L^{n_l}}-z_{c^{n_k}})$ and its $2n$ first derivatives, for $n+1$ terms of the expansion indicated in eqn (1). The definition of the functions $P_i(z)$ and $Q_i(z)$ follows closely the definitions (3.12) of I and O by Liu [4], although without the factorials that are present there. Observe the mnemonic appeal of P (for polynomials, actually binomials) and Q (for quotients, as in the case of the expansion of the simplest fundamental solution conceivable [7–9]). Although these latter functions may be computationally intensive to evaluate, they are only needed for the array of poles represented by z_{L^l} multiplied by the array of poles represented by z_{c^k}. Then, the evaluation of $Q_i(z_{L^{n_l}}-z_{c^{n_k}})$ ends up, depending on the numerical implementation, orders of magnitude less intensive than the direct evaluation of $f(z-z_0)$ for all source and field points – the basis of the fast multipole method.

2.1 Expansion of $P_i(z-z_{c^k})$ and $P_j(z_{L^{n_l}}-z_0)$ about successive poles

In eqn (1), the superscripts n_k and n_l are the highest levels of field and source poles used in a given expansion. The binomials $P_i(z-z_{c^l})$ defined in eqn (2) can be expressed for a lower-level pole $z_{c^{l-1}}$ exactly as

$$P_i(z-z_{c^l}) = \sum_{j=1}^{i} C_{j,i+1-j} P_j(z-z_{c^{l-1}}) P_{i+1-j}(z_{c^{l-1}}-z_{c^l}), \tag{4}$$

where $C_{ij}=1$, if $i=1$ or $j=1$, and, otherwise, $C_{ij}=C_{i-1,j}+C_{i,j-1}$. In the above equation, $P_{i+1-j}(z_{c^{j-1}}-z_{c^j})$, for the argument referring to the difference of poles in two consecutive levels, is defined as in eqn (2). On the other hand, $P_j(z-z_{c^{j-1}})$ is recursively evaluated according to eqn (4) until the lowest level $P_j\left(z-z_{c^0}\right)$ is obtained. With this recursive approach, the binomials $P_i(z-z_{c^{n_k}})$ in eqn (1) – and, similarly, $P_j(z_{L^{n_l}}-z_0)$ – end up expressed in terms of arguments given as differences of poles in two consecutive levels.

3 Numerical implementation

In elastostatics, the double-layer potential matrix $\mathbf{H}$ and the single-layer potential matrix $\mathbf{G}$ of the conventional boundary element method are given for the compatibility equation

$$\mathbf{Hd}=\mathbf{Gt} \tag{5}$$

(body forces not considered, for simplicity) of the nodal displacements $\mathbf{d}$ and the boundary nodal traction attributes $\mathbf{t}$ in terms of the boundary integrals [10]

$$H_{sf}=\int_\Gamma \sigma^*_{kis}(z-z_0)\eta_k(z)\,u_{if}(z)\mathrm{d}\Gamma(z),\quad G_{s\ell}=\int_\Gamma u^*_{is}(z-z_0)t_{i\ell}(z)\mathrm{d}\Gamma(z), \tag{6}$$

where $\sigma^*_{jis}(z-z_0)$ and $u^*_{is}(z-z_0)$ are the stress and displacement (Kelvin's) fundamental solutions of the elastic problem – which have global support – and $\Gamma(z)$ is the integration boundary. The subscript *s* refers to a given *source* node (at which the unit point force of the singular fundamental solution is applied) and the subscripts *f* (which stands for *field*) and ℓ (also a field reference) indicate the nodes to which the displacement-interpolation function $u_{if}(z(\xi))$ and the traction-interpolation function $t_{i\ell}(z(\xi))$ are referred. In the above equation and in the following, repeated indices mean summation. $\eta_k(z)$ are the Cartesian components of the unity outward vector to $\Gamma(z)$ and $u_{if}(z)$ formally comes from the piecewise (with local support) interpolation of displacements $u_i(z)$ along the boundary: $u_i(z)=u_{if}(z)d_f$, where d_f are the nodal displacements. In an usual isoparametric representation, displacement and geometry are represented identically for each one of their Cartesian components, so that $u_{if}(z)\equiv u_{if}(z(\xi))$ are actually replaced with shape functions $N_k(\xi)$, for a given boundary segment, with $\xi\in(0,1)$ or $\xi\in(-1,1)$, depending on the preferred parametric representation. For $z\to z_0$, strong and weak singularities must be taken care of in the evaluation of $\mathbf{H}$ and $\mathbf{G}$. However, this issue can be disregarded in the present developments, which are actually only concerned with what occurs when z and z_0 are very far from each other. The Jacobian used in the definition of $\eta_j(z)$ cancels out with the Jacobian

of $\mathrm{d}\Gamma(z)=|J(z)|\mathrm{d}\xi$, in terms of the parametric variable ξ. Then, expanding $\sigma^*_{jis}(z-z_0)$ according to eqn (1) ends up with the evaluation of a polynomial integral corresponding to the first of eqns (6). For the single-layer potential matrix $\mathbf{G}$, it is proposed that the usual interpolation polynomials $t_{i\ell}$ of traction forces in eqn (6) be replaced with $t_{i\ell} \leftarrow t_{i\ell}\,|J|_{(\text{at }\ell)}/|J|$, where $|J|_{(\text{at }\ell)}$ is the value of the Jacobian at the point characterized by the subscript ℓ [11, 12]. Nothing changes formally in the developments of the BEM, except that the numerical integration of the matrix $\mathbf{G}$ becomes much easier and actually more consistent as compared to proposed implementations given in the technical literature [12]. In fact, $|J|$ cancels out in the product $t_{i\ell}\mathrm{d}\Gamma$ in eqn (6) for $t_{i\ell}$ defined as suggested, and the integrand of $\mathbf{G}$, in the frame of a fast multipole implementation, also becomes a polynomial, independently from the assumed kernel u^*_{is}. In a practical implementation, the functions $t_{i\ell}(z(\xi))$ are replaced with the same shape functions $N_k(\xi)$ used to represent displacements, although the context differs conceptually, as $\mathbf{G}$, among other features, is in general a rectangular matrix (ℓ in general spans a larger number of nodes than *f*).

As proposed, let $z=N_k(\xi)z_k$ be the complex geometric representation of a boundary segment, for a 2D implementation, expressed in terms of the nodal points z_k and the shape functions $N_k(\xi)$, which are also used to interpolate displacements as well as traction forces, as described above. Then, the only integrations that need to be carried out for a given boundary segment Γ_{seg} in the evaluation of the matrices of eqn (6) in the framework of the fast multipole developments represented by eqn (1) are for terms of $\mathbf{H}$ and $\mathbf{G}$ defined as

$$\begin{aligned}\tilde{H}_{ji} &= \int_{\Gamma_{seg}} P_j\left(z-z_c\right) z' N_i(\xi)\mathrm{d}\Gamma = \int_0^1 P_j\left(N_k(\xi)z_k - z_c\right) N_l'(\xi) z_l N_i(\xi)\mathrm{d}\xi \\ \tilde{G}_{ji} &= \int_{\Gamma_{seg}} P_j\left(z-z_c\right) N_i(\xi)\mathrm{d}\Gamma = \int_0^1 P_j\left(N_k(\xi)z_k - z_c\right) N_i(\xi)\mathrm{d}\xi,\end{aligned} \tag{7}$$

where, recalling, repeated indices mean summation, in this case covering all shape functions $N_k(\xi)$ of a boundary segment representation, and a prime $(')$ means derivative with respect to ξ. Since the Jacobian inherent to $\mathrm{d}\Gamma$ in the above integrals cancels out, the indicated interval $\xi\in(0,1)$ is actually irrelevant. In the first row of the above equation, the term $N_l'(\xi)z_l$ comes from the complex representation of $\eta_k(z)$ along the boundary segment. The subscript *j* refers to the number of terms in the adopted expansion of eqn (4), whereas *i* refers to each of the boundary element nodes. In the present implementation for 2D problems [8], either linear, quadratic or cubic boundary elements may be represented. Constant elements are considered in a separate implementation. The main feature of the proposed implementation of the FMM is that the arrays $\tilde{H}_{ji}$ and $\tilde{G}_{ji}$ given above

become ultimately expressed as polynomials of the differences of the nodal coordinates of the boundary segment to the immediate pole,

$$\Delta_i = z_i - z_c, \text{ with } i = 1 \ldots o_e + 1, \tag{8}$$

where $o_e = 1, 2, 3$ for linear, quadratic or cubic boundary elements. As an illustration for quadratic elements (3 nodes),

$$\tilde{H}_{ji} = \sum_{k=1}^{j+1} \sum_{l=1}^{k} A_{ijkl} \Delta_1^{j-k+1} \Delta_2^{k-l} \Delta_3^{l-1}, \quad \tilde{G}_{ji} = \sum_{k=1}^{j+1} \sum_{l=1}^{k} B_{ijkl} \Delta_1^{j-k+1} \Delta_2^{k-l} \Delta_3^{l-1}, \tag{9}$$

where the polynomial coefficients A_{ijkl} and B_{ijkl} can be previously evaluated for each node i and each series term j they refer to [7–9] and stored. However, a simpler procedure seems to be the direct storage of the polynomial expressions of eqn (9), as already implemented in a C^{++} code [8].

The expression of the array $Q(z)$ in eqn (3) is given in the technical literature for the most common fundamental solutions [4] and need not be reproduced here. However, as an illustration, the Westergaard stress function for a rotated, elliptic semicrack [13] may be developed as

$$Q(Z) = \frac{-1}{2\pi} \left[\frac{\pi Z}{2} + 1 - (1 - Z^2) F \quad FZ + \frac{1}{Z} + \frac{\pi}{2} \quad \frac{F - 1/Z^2}{1 - Z^2} \quad \frac{3ZF - (5Z^2 - 2)/Z^3}{(1 - Z^2)^2} \right], \tag{10}$$

where $F = \ln\left(-\left(1 + \sqrt{1 - Z^2}\right) / Z \right) / \sqrt{1 - Z^2}$ and $Z = z e^{-\theta\sqrt{-1}} / a$ for the semicrack length a and inclination θ. (Recall that $\tilde{H}_{ji}$ and $\tilde{G}_{ji}$ remain defined as above.)

4 The expedite boundary element method

The expedite BEM [11] makes use of some features borrowed from both the variationally based hybrid BEM and the conventional BEM in order to arrive at a simplified formulation. This method basically relies on the equations

$$\begin{aligned} \mathbf{H}^{\mathrm{T}} \mathbf{p}^* &= \mathbf{p} \\ \mathbf{U}^* \mathbf{p}^* &= \mathbf{d}, \end{aligned} \tag{11}$$

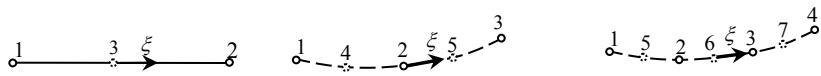
for generally mixed boundary conditions in terms of nodal displacements $\mathbf{d}$ and equivalent forces $\mathbf{p}$, where $\mathbf{H}$ is the same double-layer potential matrix of eqn (6) and $\mathbf{U}^*$ is the matrix of nodal-displacement fundamental solutions of the problem that is being analyzed. Once the vector of internal force parameters $\mathbf{p}^*$ is solved from eqn (11), displacements and stresses at internal points of the elastic body can be directly evaluated with no need of resorting to a boundary integral equation (Somigliana's identity), as in the conventional BEM.

In a FMM implementation, which is always embedded in the algorithm of the chosen iterative solver, such as the GMRES [4], the matrix products $\mathbf{H}^{\mathrm{T}} \mathbf{p}^*$ and $\mathbf{U}^* \mathbf{p}^*$ are evaluated according to the features briefly outlined in Section 3 for the

conventional BEM. The FMM evaluation of $\mathbf{U}^*\mathbf{p}^*$ is in part simplified, since no integration is required, but in part involves some additional computational effort, as the elements of $\mathbf{U}^*\mathbf{p}^*$ for coinciding source and field points are in principle undetermined and must be obtained separately [11]. Owing to space restrictions, this expedite formulation is no longer addressed in this paper.

5 A unified algorithm for hierarchical mesh refinement

Figure 1 shows the schemes of three different elements considered in the present algorithm, as taken out of a general 2D mesh corresponding to a given level of refinement. The algorithm is also implemented for constant elements. A corresponding algorithm has also been implemented for 3D problems [14].



a) Linear element (t_e=2, o_e=1) b) Quadratic element (t_e=3, o_e=2) c) Cubic element (t_e=4, o_e=3)

Figure 1: Schemes for splitting a general element into two sub-elements.

The input data for the hierarchical mesh refinement are:

- o_e = either 1, 2 or 3, which defines the element type (t_e = 2, 3 or 4).
- nee : Initial number of elements of the initial level.
- nne : Number of nodes of the initial level.
- nv : Number of additional levels of mesh refinement: ($nv = 1$, for example, indicates that the structure will be refined once).
- Table $Coord[\ ]$, with nee node coordinate entries in the format $[x, y]$, which are the initial nodes of the mesh structure to be refined. This table is successively expanded, as new nodes are added during the mesh refinement.
- Table $Inc[k][\ ]$, where $k = 1$ refers to the initial, first level local-to-global nodal incidence of the input mesh and the second entry are nee arrays, each one with t_e global node numbers of the elements. Arrays $Inc[k][\ ]$, for $k = 2...nv+1$, will be generated as a result of the mesh refinement.

The output data are a generalization of the input data, which now refer to $nv+1$ levels of refinement:

- $nel[k = 1...nv+1]$ Number of elements at each one of the $nv+1$ levels.
- $nnl[k = 1...nv+1]$ Number of nodes at each one of the $nv+1$ levels.
- Table $Coord[\ ]$ with $nnl[nv+1]$ node coordinate entries, which correspond to the input values plus the coordinates of the generated nodes.

- A table $Inc[k=1...nv+1][\ ]$ with $nv+1$ levels of arrays of local-to-global node incidences. The second entry are $nel[k=1...nv+1]$ arrays, each one with t_e global node numbers of the elements.

The algorithm, which is not shown in the present short paper, generates $nv+1$ levels of mesh refinement, including the initial one, which is referred to as level 1. The number of elements on any level is two times the number of elements of the preceding level. However, the number of nodes is not known in advance and is evaluated inside the procedure.

6 A unified algorithm for pole expansions

In the conventional BEM, as given in eqn (5), the fundamental solutions $\sigma^*_{jis}(z-z_0)$ and $u^*_{is}(z-z_0)$ shown in eqn (6), which have global support, can in principle be expanded according to eqns (1) and (4) about arbitrary (but feasible, depending on the mesh discretization) numbers of source poles z_{L^l} as well as of field poles z_{c^k}. In the proposed FMM algorithm, a pole expansion is combined with the mesh refinement introduced (but not detailed) in the previous section, in such a way that the element (be it linear, quadratic or cubic) dictates the expansion. The scheme is applicable to constant elements, although in a separate, simpler, code, and this is shown in Fig. 2 to illustrate that a pole expansion can be undertaken either at each mesh split (a), at every two mesh splits (b), or at every four mesh splits (c) and so on, generating in the illustration either two, four or eight children poles.

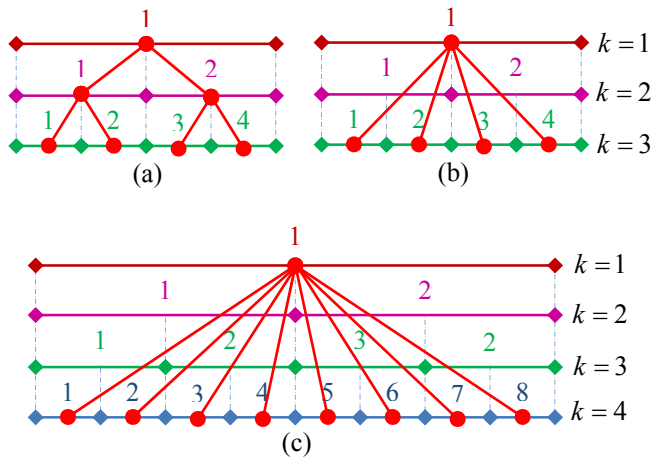


Figure 2: For constant elements, schematic pole expansions using numbers of children poles $n_c = 2$, 4 or 8.

The complete code combines the algorithm of Section 5 with the compact, self-recurring expansions of eqns (1) and (4) and is too convoluted to be shown here [8]. Given an initial, rough, mesh configuration, as illustrated in Fig. 3, field as well as source pole expansions can be undertaken along with the process of dyadic refinement of the mesh, as illustrated in Fig. 1, so that, for known values of $\mathbf{d}$ and $\mathbf{t}$ in eqn (5), line multiplications (for instance, the multiplication $\mathbf{H}_i\mathbf{d}$ of a given line i of matrix $\mathbf{H}$) are carried out in a faster way that requires usually only $\log(N)$ instead of N operations, where N is the vector dimension of $\mathbf{d}$. Basic entries of the developed FMM algorithm are, for a given mesh and a given boundary element type, the numbers of field and source poles as well as the numbers of children poles to be used at each expansion, according to Fig. 2.

The code developed so far by the authors is still a long way from being considered an accomplished work. Our goal is to rethink and carefully assess every step towards the development of a complete FMM procedure applicable to generally curved elements and for general underlying fundamental solutions.

The complete analysis of a problem in the frame of a FMM procedure presupposes the implementation of a direct solver (GMRES, for instance) and the use of a robust preconditioner. In order to efficiently assess the performance of the proposed FMM algorithm, the complete solution of a problem has been postponed to our last step of developments (when most possibly a well-proven solver – such as given by Liu [4] – will be adopted). Moreover, the geometric distance between source and field points is at the moment replaced by the topological distance obtained in the mesh refinement introduced in Section 5 and used as illustrated in Fig. 2. This is not correct for domains that are too irregular. There is already a work in progress to insert a problem's domain in a mesh of hierarchical squares, making use of developments done for 3D problems [14]. This concept is not far from the developments of the technical literature [4], only that we work with Boolean algebra rather than with geometric distances to evaluate how far two poles are from each other.

Our initial assessments have shown that, for the solution of eqn (5), the expansion of the source poles does not contribute to the efficiency of the proposed FMM algorithm [8]. As a result, the following numerical assessments only consider expansion of the field poles.

7 A basic assessment for constant elements

Although the proposed algorithms are for general curved elements, the simplicity of the constant element enables the assessment of the basic features of the algorithms introduced in Sections 3, 5 and 6. The problem to be considered is the simplest case of linear potential in a square domain, whose edges are progressively discretized with up to $2^{20} = 1,048,576$ constant elements, as indicated in the horizontal axes of both graphs of Fig. 3. On the left of this figure is plotted the execution time for the evaluation of eqn (5) for the analytical values of potentials $\mathbf{d}$ and normal gradients $\mathbf{t}$, as implemented in a C^{++} code and running on a desktop computer. For the sake of comparison, the execution time required for evaluating

eqn (5) in the frame of the conventional BEM is plotted and shown to be proportional to N^2. The remaining plots on the left give the execution time of eqn (5) for the field poles successively expanded with number of children $n_c = 2$ (first illustration of Fig. 2) and the number n of terms in the series of eqn (1) varying from 5 to 13. The results are in all cases quite the same and shown to be proportional to $N \log N$. The graph on the right of Fig. 3 shows the Euclidean error norm $\varepsilon = |\mathbf{Hd} - \mathbf{Gt}| / |\mathbf{Gt}|$ for evaluations carried out combining different $n_c = 2, 4, 8$ (Fig. 2) and different n, also comparing with results of the CBEM. The accuracy tends to increase with n_c mainly because the number of adjacent elements to a given element – for which no series expansion is carried out – also increases with n_c. Although the accuracy always increases with n, there is also always a threshold for the mesh refinement.

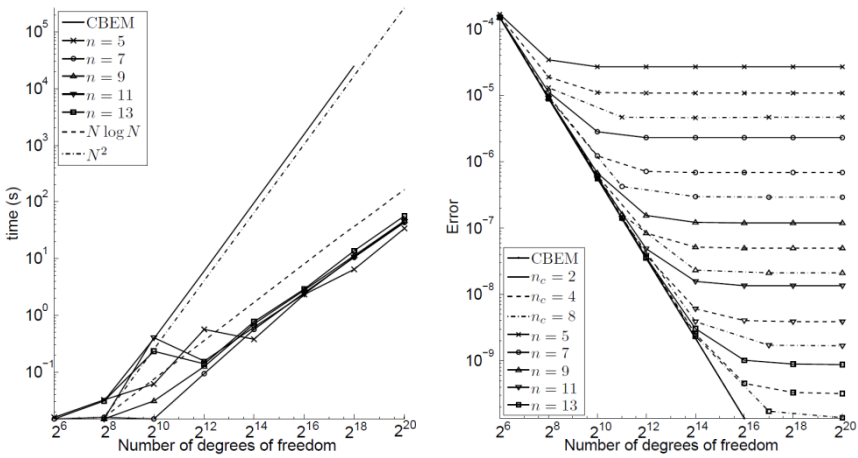


Figure 3: Execution time for the evaluation of eqn (5) for a linear potential problem over a square, and accuracy results for different numbers of children poles and expansion terms.

8 Some results for curved elements

Figure 4 shows some results for curved elements, which, although preliminary and of a rather academic nature [8], already give an idea of the capability of the algorithms that are being implemented. The deformed quadrilateral on the left is built up with eight quadratic macroelements, which will undergo successive splitting (always keeping the original geometry) up to 8,192 quadratic, curved elements corresponding to 16,384 degrees of freedom for a potential problem analysis. This irregularly-shaped domain is submitted to a potential field $x^2 - y^2$, with corresponding nodal potentials $\mathbf{d}$ and normal gradients $\mathbf{t}$ evaluated for the

numerical analysis using eqn (5) in terms of FMM expansions. Owing to the polynomial characteristics of the applied potential together with the large domain size and the irregular boundary, the proposed problem poses a good challenge for the numerical simulation. A convergence analysis using the same Euclidean norm of the previous example is shown on the right for three different numbers of children poles and five different numbers of expansion terms. Results with the conventional BEM are also shown, for comparison. The convergence pattern is the same of the previous example, although not as regular. Moreover, although both discretization and expansion errors have shown to be small from the very beginning, there is also an accuracy threshold indicating that it is not worth refining the mesh further. The execution time was much larger than in the case of constant elements, although the achieved accuracy is also incomparably larger.

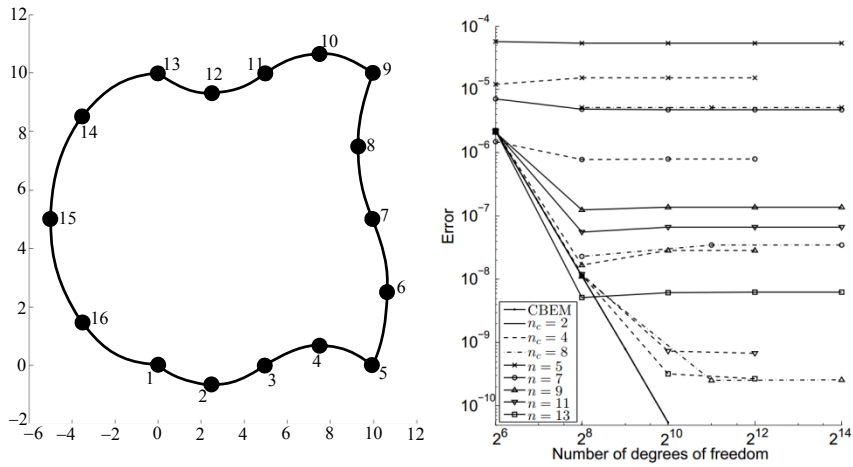


Figure 4: Deformed quadrilateral domain for a study with quadrilateral elements and accuracy results for different numbers of children poles and expansion terms.

9 Concluding remarks

Owing to space restrictions, only the basic features of the proposed developments could be presented. They are the compact expression of the expansion of a general fundamental solution about successive levels of source and field poles, as given in eqn (1), and the pre-integrations represented in eqn (7) and in part illustrated in eqn (9) for quadratic elements considered in the frame of a consistent BEM formulation [12]. The still preliminary examples of high-order, curved elements attest the possibilities of analysis that are being achieved. The combination with the expedite BEM in order to further accelerate the whole computational process and the use of a mesh with hierarchical squares [14] to evaluate pole distances in terms of Boolean algebra are some of the implementations in progress. There are some conceptual differences between the conventional and the expedite BEM, as

for instance the fact that the matrix **H** is used in eqn (5) whereas its transpose appears in eqn (11), which lead to implementation peculiarities still to be explored.

The proposed FMM must be ultimately inserted into an iterative solver. This is not the authors' concern at the moment, since the use of a GMRES solver, for instance, together with the identification of the best pre-conditioner, seems to be well established in the technical literature [4]. Thus, the proposed systematic assessment takes apart the issues related to a plain FMM procedure from the issues related to an iterative solver particularly with respect to execution time and may lead to a more efficient codification of the introduced algorithms.

Acknowledgements

This work was supported by the Brazilian agencies CAPES, CNPq and FAPERJ.

References

[1] Greengard, L. & Rokhlin, V., A fast algorithm for particle simulations, *Journal of Computational Physics*, **135**, 280–292, 1987.

[2] Dongarra, J. & Sullivan, F., The top ten algorithms of the century, *Computing in Science and Engineering*, **2**(1), 22–23, 2000.

[3] Nishimura, N., Fast multipole accelerated boundary integral equation methods, *Applied Mechanics Reviews*, **55**, 299–324, 2002.

[4] Liu, Y., *Fast Multipole Boundary Element Method: Theory and Applications in Engineering*, Cambridge University Press, New York, USA, 2009.

[5] Liu, Y.J., Mukherjee, S., Nishimura, N., Schanz, M., Ye, W., Sutradhar, A., Pan, E., Dumont, N.A., Frangi, A. & Saez, A., Recent advances and emerging applications of the boundary element method, *Applied Mechanics Reviews*, **64**(3), 030802 (38 pp), 2011.

[6] Liu, Y.J. & Nishimura, N., The fast multipole boundary element method for potential problems: a tutorial, *Eng. Anal. Boundary Elem*., **30**, 371–381, 2006.

[7] Peixoto, H.F.C., *Application of the Hybrid Boundary Element Method to Large-Scale Problems Using Fast Multipole Techniques*, M.Sc. thesis, PUC-Rio, Brazil, 2014.

[8] Novelino, L.S., *A Novel Fast Multipole Technique in the Boundary Element Methods*, M.Sc. thesis, PUC-Rio, Brazil, 2015.

[9] Dumont, N.A. & Peixoto, H.F.C., Application of the hybrid boundary element method to large-scale problems using a fast multipole technique, *Procs*. *PACAM XIV – 14th Pan-American Congress of Applied Mechanics*, ed. R. Bustamante, Santiago, Chile, 6 pp on CD, 2014.

[10] Brebbia, C.A., Telles, J.C.F. & Wrobel, L.C., *Boundary Element Techniques*, Springer-Verlag: Berlin and New York, 1984.

[11] Dumont, N.A. & Aguilar, C.A., The best of two worlds: the expedite boundary element method, *Engineering Structures*, **43**, 235-244, 2012.

[12] Dumont, N.A., The boundary element method revisited, *Boundary Elements and Other Mesh Reduction Methods XXXII*, ed C.A. Brebbia, 227–238, WIT Press, Southampton, 2010.
[13] Dumont, N.A. & Mamani, E.Y., Generalized Westergaard Stress Functions as Fundamental Solutions, *CMES – Computer Modeling in Engineering & Sciences*, **78**(2), 109–150, 2011.
[14] Dumont, N.A. & Aguilar, C.A., Three-dimensional implementation of the expedite boundary element method, *Procs. IABEM2011 – Symposium of the International Association for Boundary Element Methods*, 113–118, Brescia, Italy, 2011.