

*Research Paper***A Fast-Multipole Unified Technique for the Analysis of Potential Problems with the Boundary Element Methods**

NEY AUGUSTO DUMONT\* and HÉLVIO DE FARIAS COSTA PEIXOTO

*Civil Engineering Department, Pontifical Catholic University of Rio de Janeiro, Brazil*

(Received on 05 May 2016; Accepted on 12 May 2016)

The proposed developments are based on a consistent implementation of the conventional, collocation boundary element method (BEM). A scheme is used to expand a generic (not problem-dependent) fundamental solution about hierarchical levels of source and field poles, which is particularly advantageous to make the technique seamlessly applicable to 2D and 3D problems of elasticity or potential, in terms of different types of curved elements for generally complicated geometry and topology. The proposed compact algorithm is more straightforward to lay out and seems to be more efficient than the ones available in the technical literature – particularly because the outermost loop refers to field nodes and geometry, in what may be called a reverse implementation. Some numerical results are shown for the conventional BEM, with validation and assessment for a few simple, but very large-scale, 2D potential problems with complicated geometry and topology for constant, linear and quadratic elements. Since iterative solvers are not required in this first step of numerical simulations, an isolated assessment of accuracy, computational effort and storage allocation of the proposed fast multipole technique becomes possible.

**Keywords:** Boundary Elements, Fast Multipole Methods, Variational Methods**Introduction**

The fast multipole technique (Greengard and Rokhlin, 1987), 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  $O(N)$ , as suggested by (Liu, 2009) and corroborated in the present developments – while requiring computer storage that is only a small fraction of what would be allocated for a different numerical method.

The present research work is part of the studies carried out by the second author (Peixoto, 2014) together with Novelino (2015) to develop a robust and efficient fast multipole code applicable to problems with generally curved boundaries, in a framework that is almost completely independent from the underlying fundamental solution (Peixoto *et al.*, 2015, 2015a). The basic concept of the fast multipole method (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 (Nishimura, 2002; Liu and Nishimura, 2006; Liu, 2009).

This paper proposes a scheme for expansions of a generic fundamental solution about hierarchical levels of source and field poles, which makes the fast-multipole technique directly applicable to different kinds of potential and elasticity problems. The proposed hierarchical tree of poles is built upon a topological concept of superelements inside superelements, which in part circumvents the need of evaluating geometrical distances between nodes as well as the need of concepts such as quadtrees or octrees for 2D or 3D problems, as basically adopted since the inception of the FMM (Liu, 2009).

***Proposed FM Algorithm for a General, Complex Function***

The following basic definitions are used in the present developments:

---

\*Author for Correspondence: E-mail: dumont@puc-rio.br, hfcpeixoto@gmail.com

- $z - z_0$  = 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  $z$ .
- $z_L$  = point about which the fundamental solution will be expanded for the source point  $z_0$ .

Expansions about successively farther poles  $z_{c^k}$ ,  $k = 1, 2, \dots, n_c$  (where, by definition,  $z_{c^0} \equiv z_c$ ) and  $z_{L^l}$ ,  $l = 1, 2, \dots, n_L$  (where, by definition,  $z_{L^0} \equiv z_L$ ) are also undertaken.

A generic fundamental solution for 2D problems is expanded as

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

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  $P(z)$  and  $Q(z)$  defined as

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

$$Q(z) = \left[ f(z) \quad \frac{\partial f(z)}{\partial z} \quad \dots \quad \frac{\partial f^{n+1}(z)}{\partial z^{n+1}} \right] \quad (3)$$

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 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 the expansion indicated in Eq. (1). Although these latter functions may be computationally intensive to evaluate, they are only needed for the multiplication of the arrays of poles represented by  $z_{L^l}$  and  $z_{c^k}$ .

Then, the evaluation of  $Q_i(z_{L^{n_l}} - z_{c^{n_k}})$  may end up orders of magnitude less intensive than the direct evaluation of  $f(z - z_0)$  for all source and field points.

#### **Expansion of $P_i(z - z_{c^k})$ and $P_j(z_{L^l} - z_0)$ About Successive Poles**

In Eq. (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 Eq. (2) are expressed for a lower-level pole  $z_{c^{l-1}}$  exactly as

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

where  $C_{ij} = 1$  if  $i = 1$  or  $j = 1$ , otherwise,  $C_{ij} = C_{i-1,j} + C_{i,j-1}$ . In Eq. (4),  $P_{i+1-j}(z_{c^{l-1}} - z_{c^l})$  is defined as in Eq. (2) since the argument refers to two consecutive levels. On the other hand,  $P_j(z - z_{c^{l-1}})$  is recursively evaluated according to Eq. (4) until the lowest level  $P_j(z - z_{c^0})$  is obtained. With this recursive approach,

the binomials  $P_i(z - z_{c^{n_k}})$  and  $P_j(z_{L^{n_l}} - z_0)$  always end up expressed in terms of arguments given as differences of poles in two consecutive levels.

As a matter of illustration, three expansion schemes are shown in Fig. (1). They depict the cases where several field points are expanded about a single

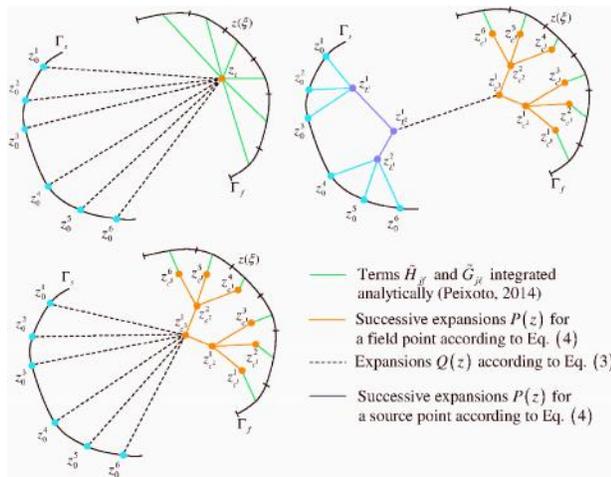


Fig. 1: Schematic representations of expansions about field and source poles (according to Peixoto *et al.*, 2015a)

pole  $z_c$  (upper left), about three successive layers of poles  $z_c$  (lower left) and, moreover, source points expanded about two layers of poles  $z_L$  (upper right). The continuous lines starting from the boundary  $\Gamma_f$  stand for analytical integrations of the BEM matrices for fast-multipole matrix-vector products corresponding to  $\tilde{H}_{jf}$  and  $\tilde{G}_{jt}$  in Eqs. (11) and (12), as proposed in Peixoto (2014).

**Hierarchical Refinement Algorithm**

Figure 2 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 (Dumont, 2012). The splitting algorithm for constant elements is the same one for linear elements. A corresponding algorithm has also been implemented for 3D problems (Dumont and Aguilar, 2011). The element numbering of the reference level leads immediately to the numbering of the splitted elements, as each element is splitted into two. The created nodes (indicated by dashed circles in Fig. 2) are sequentially numbered as they appear in the splitting process, so that their global numbering does not become sequential along the boundary. The element numbering leads to the straightforward identification of adjacent split elements at each refinement level.

Figure 3 shows three cases of possible refinements, with  $n_c = 2, 4$  or  $8$  children elements after an element splitting.

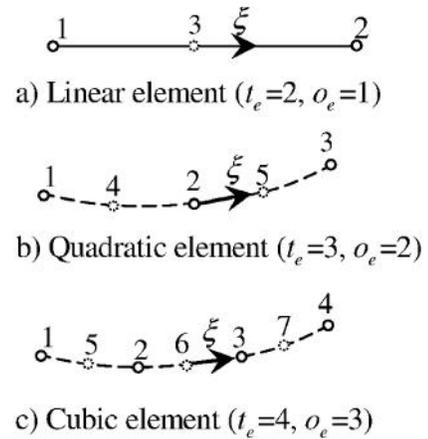


Fig. 2: Scheme of three different elements that are split each into two sub-elements (Dumont, 2012)

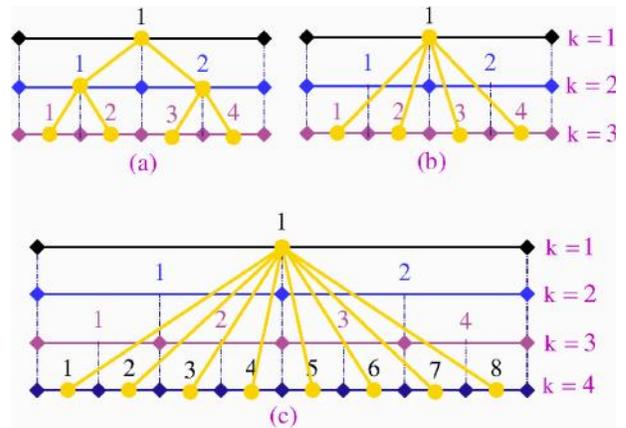


Fig. 3: Schematic pole expansions for constant element using numbers of children elements (and poles)  $n_c = 2, 4$  or  $8$  (Novelino, 2015)

**Implemented FM Algorithm**

This section describes a compact version of the implemented fast-multipole algorithm. The number of code lines is actually very small. However, as the algorithm calls a recursive routine (PoleExpansion) inside another recursive routine (Adjacencies), this makes it fairly convoluted and difficult to explain verbally, although the flowcharts can be easily translated into a code. The basic version presented below gives an overview of the algorithm’s four major routines: Main, Adjacencies, Source and PoleExpansion.

The procedure Main (Fig. 4) loads the input data, generates the hierarchical mesh according to the

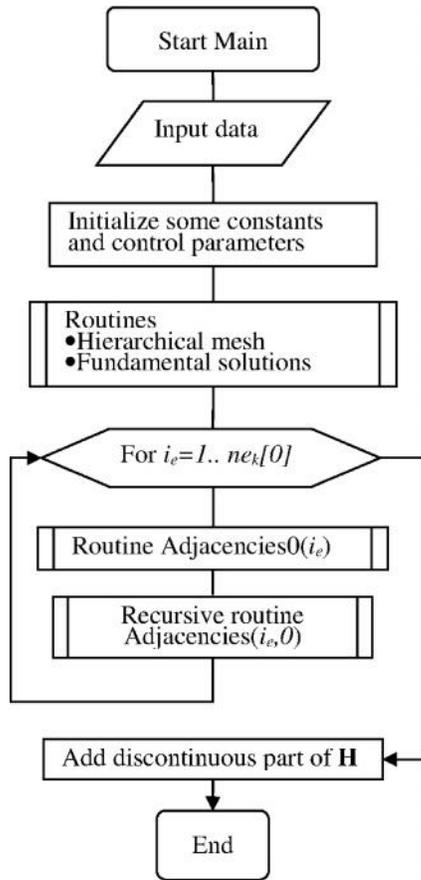


Fig. 4: Procedure main

concepts briefly discussed in the Introduction and evaluates the kernel expansions according to Eq. (3). Then it executes a small loop over all elements of the first level ( $k = 0$ ) of the hierarchical mesh in order to create the adjacency structure for each macro-element ( $i_e$ ), carrying out, at the same time, all the possible field evaluations for the child elements of element  $i_e$ .

The routine Adjacencies (Fig. 5) assembles the adjacency structure, and when it reaches the most refined level ( $k = n_v$ ), it calls the routine Source. This routine (Fig. 6) handles integrations in terms of the conventional BEM matrix-vector products (routine BEMAdj) for the adjacent elements, as well as in terms of FM expansions (routine BEMFM). The analytical integrations carried out in the frame of the routine BEMFM refer to the closest field pole and are successively stored for use with far-field elements in the routine PoleExpansion. The routine Source also leads to the successive expansion of the FM integration

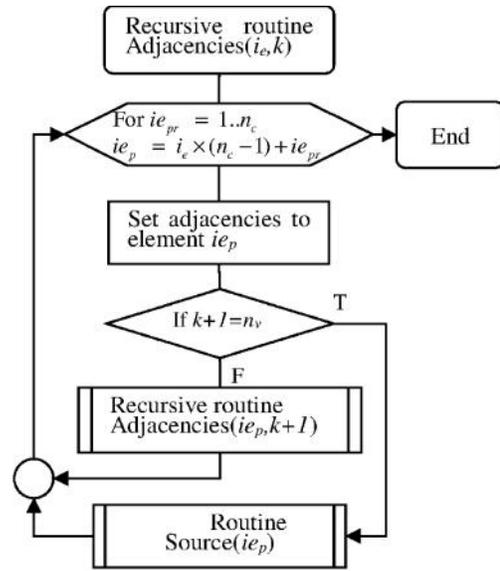


Fig. 5: Procedure adjacencies

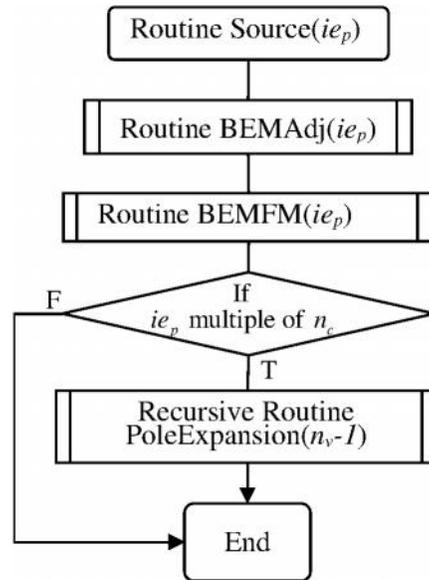


Fig. 6: Procedure source

terms, thus delivering data information to the upper refinement levels, if this is the case, by calling the routine PoleExpansion.

Finally, the recursive routine PoleExpansion (Fig. 7) delivers the FM-integrated data to the source poles that are considered sufficiently far by calling the routine Qvector to evaluate the  $Q$  vectors presented in Eq. and then evaluating the expansion series for the source point, Eq. (3). It also checks if the level  $k = k_{exp}$  has been reached, as expansions stop at this level, indicating that the results obtained so far are

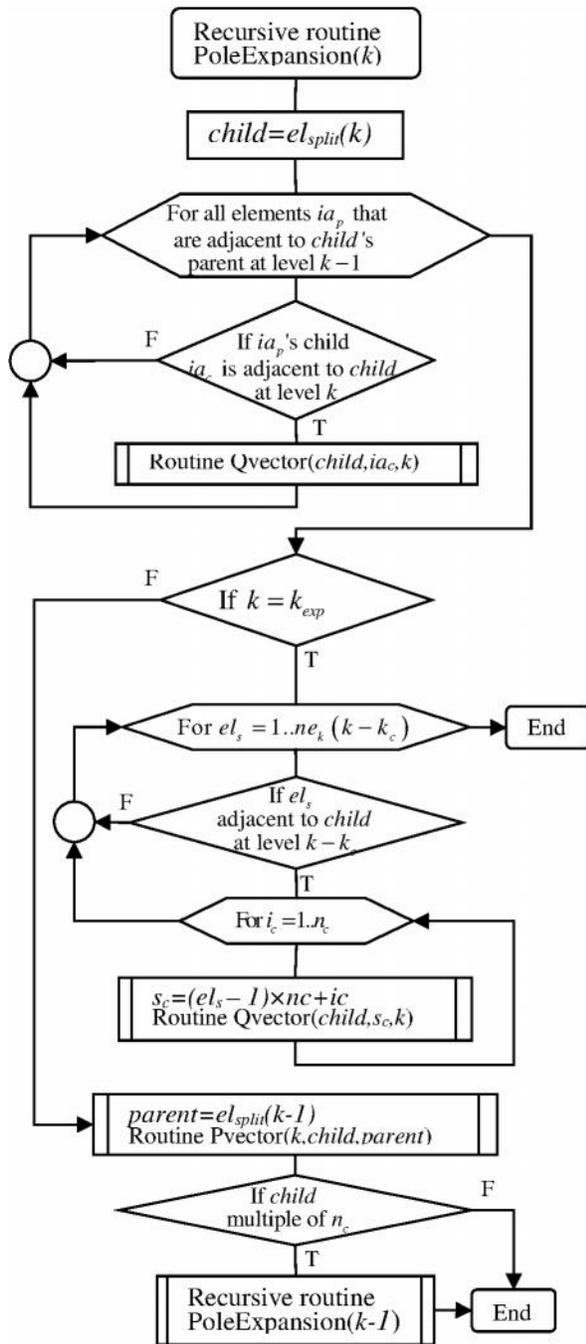


Fig. 7: Procedure PoleExpansion

directly delivered to the remaining source points. If  $k = k_{exp}$  has not been reached, the routine calls routine Pvector to convey the obtained data to the upper pole levels and, when all the elements of level  $k$  have been processed, calls itself (thus recursively) to proceed to the immediately upper refinement level of the hierarchical structure.

### Boundary Element Implementation

For a potential problem, 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 continuity equation

$$\mathbf{H}\mathbf{d} = \mathbf{G}\mathbf{t} \quad (5)$$

(body source not considered, for simplicity) of the nodal potentials  $\mathbf{d}$  and the boundary nodal flux attributes in terms of the boundary integrals (Brebbia *et al.*, 1984)

$$\begin{aligned} H_{sf} &= \int_{\Gamma} q_{ks}^*(z - z_0) \eta_k(z) \theta_f(z) d\Gamma(z), \\ G_{s\ell} &= \int_{\Gamma} \theta_s^*(z - z_0) q_{\ell}(z) d\Gamma(z), \end{aligned} \quad (6)$$

where  $q_{js}^*(z - z_0)$  and  $\theta_s^*(z - z_0)$  are the flux and potential fundamental solutions of the potential 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 source of the singular fundamental solution is applied) and the subscripts  $f$  (which stands for field) and  $\ell$  (also a field reference) indicate the nodes to which the potential-interpolation function  $\theta_f(z(\xi))$  and the flux-interpolation function  $q_{\ell}(z(\xi))$  are referred. In the above equation and in the following, repeated indices mean summation.  $\eta_k(z)$  are the Cartesian components of the outward unity vector to  $\Gamma(z)$ , and  $\theta_f(z)$  formally comes from the piecewise (with local support) interpolation of potentials  $\theta(z)$  along the boundary:  $\theta(z) = \theta_f(z) d_f$ , where  $d_f$  are the nodal potentials. In a usual isoparametric representation, potential and geometry are represented identically for each one of their Cartesian components, so that  $\theta_f(z) \equiv \theta_f(z(\xi))$  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  $z \rightarrow z_0$ , 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  $d\Gamma(z) = |J(z)|d\xi$ , in terms of the parametric variable  $\xi$ . Then, expanding  $q_{js}^*(z - z_0)$  according to Eq. (1) ends up with the evaluation of a polynomial integral corresponding to the first of Eq. (6). For the single-layer potential matrix  $\mathbf{G}$ , it is proposed that the usual interpolation polynomials  $q_\ell$  of normal fluxes in Eq. (6) be replaced with

$$q_\ell \leftarrow q_\ell |J|_{(\text{at } \ell)} / |J|, \quad (7)$$

where  $|J|_{(\text{at } \ell)}$  is the value of the Jacobian at the point characterized by the subscript (Dumont, 2010; Dumont and Aguilar, 2012). 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 (Dumont, 2010). In fact,  $|J|$  cancels out in the product  $q_\ell d\Gamma$  in Eq. (6) for  $q_\ell$  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  $\theta_s^*$ . In a practical implementation, the functions  $q_\ell(z(\xi))$  are replaced with the same shape functions  $N_k(\xi)$  used to represent potential, although the context differs conceptually, as  $\mathbf{G}$ , among other features, is in general a rectangular matrix ( $\ell$  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 potential as well as normal fluxes, as described above. Then, the only integrations that need to be carried out for a given boundary segment  $\Gamma_{seg}$  in the evaluation of the matrices of Eq.

(6) in the framework of the fast multipole developments represented by Eq. (1) are for terms of  $\mathbf{H}$  and  $\mathbf{G}$  defined similarly to Eq. (1) as

$$\begin{aligned} \tilde{H}_{jf} &= \int_{\Gamma_{seg}} P_j(z - z_c) z' N_f(\xi) d\Gamma \\ &= \int_0^1 P_j(N_k(\xi)z_k - z_c) N_f'(\xi) z_l N_i(\xi) d\xi \end{aligned} \quad (8)$$

$$\begin{aligned} \tilde{G}_{j\ell} &= \int_{\Gamma_{seg}} P_j(z - z_c) N_\ell(\xi) d\Gamma \\ &= \int_0^1 P_j(N_k(\xi)z_k - z_c) N_\ell(\xi) d\xi \end{aligned} \quad (9)$$

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  $\xi$ . Since the Jacobian inherent to  $d\Gamma$  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  $N_f'(\xi)z_l$  comes from the complex representation of  $\eta_k(z)$  along the boundary segment. The subscript  $j$  refers to one of the terms in the adopted expansion of Eq. (4), whereas  $i$  refers to each of the boundary element nodes. In the present implementation for 2D problems (Novelino, 2015), 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  $\tilde{H}_{jf}$  and  $\tilde{G}_{j\ell}$  given above become ultimately expressed as polynomials of the differences of the nodal coordinates of the boundary segment to the immediate pole,

$$\Delta_i = z_i - z_c, \quad \text{with } i = 1 \dots o_e + 1, \quad (10)$$

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

$$\tilde{H}_{jf} = \sum_{k=1}^{j+1} \sum_{l=1}^k A_{fjkl} \Delta_1^{j-k+1} \Delta_2^{k-l} \Delta_3^{l-1}, \quad (11)$$

$$\tilde{G}_{j\ell} = \sum_{k=1}^{j+1} \sum_{l=1}^k B_{\ell jkl} \Delta_1^{j-k+1} \Delta_2^{k-l} \Delta_3^{l-1}, \quad (12)$$

where the polynomial coefficients  $A_{fjkl}$  and  $B_{\ell jkl}$  can be previously evaluated for each node  $f$  or  $\ell$  and each series term  $j$  they refer to (Dumont and Peixoto, 2014; Peixoto, 2014; Novelino, 2015) and stored. However, a simpler procedure seems to be the direct storage of the polynomial expressions of Eqs. (11) and (12), as already implemented in a C++ code (Novelino, 2015). There is also the possibility of evaluating the indicated integrals in Eqs. (8) and (9) exactly by using Gauss-Legendre quadrature, which may become a faster procedure. However, the code would only be efficient by using different numbers of integration points for different terms of the series expansion. This is being currently assessed.

### Numerical Implementation

#### A Simple Numerical Example

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. The problem to be considered is the simplest case of linear potential in a square domain, Fig. 8, whose edges are progressively discretized with up to  $N = 2^{24} = 16,777,216$  constant elements, as indicated in the horizontal axes of Figs. 9 and 10. In Fig. 9 are plotted the execution times for the evaluation of Eq. (5) for the analytical values of potentials  $\mathbf{d}$  and normal gradients  $\mathbf{t}$ , as implemented

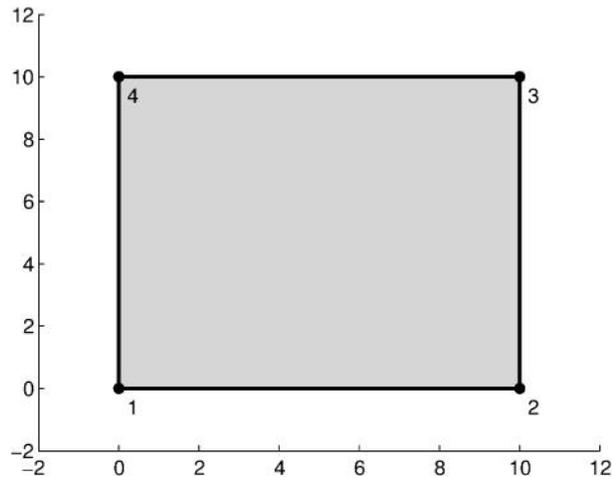


Fig. 8: Square domain for a first simple assessment

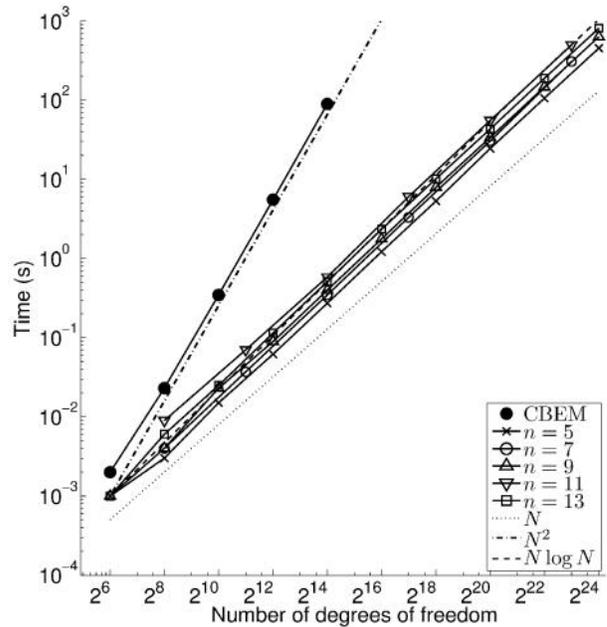


Fig. 9: Execution times for the evaluation of Eq. for a linear potential problem over the square of Fig. 8 using  $n_c = 2$  children poles and different numbers of expansion terms

in a C++ code and running on a desktop computer (i7-4770 CPU 3.4GHz, 16GB RAM in Windows 7) for this and all subsequent examples. For the sake of comparison, the execution time required for evaluating Eq. (5) in the frame of the conventional BEM is also plotted (represented by full circles in this and all successive graphs) and shown to be proportional to  $N^2$  (dash-dot line). The remaining plots in Fig. 9 give the execution time of Eq. (5) for the field poles successively expanded with number of children  $n_c = 2$  and the number  $n$  of terms in the series of Eq. (1) varying from 5 to 13. The results are in all cases quite the same and shown to be rather proportional to  $N$  (dotted line) than to  $N \log N$  (dashed line), according to the plots. The graphs in Fig. 10 show the Euclidean error norm

$$\varepsilon = \frac{|\mathbf{Hd} - \mathbf{Gt}|}{|\mathbf{Gt}|} \quad (13)$$

for evaluations carried out combining different children poles  $n_c = 2, 4, 8$  and different expansion terms  $n$ , 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

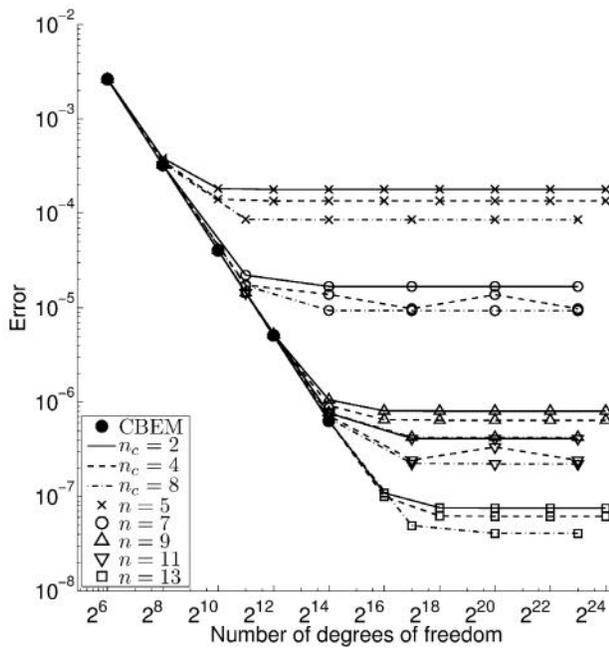


Fig. 10: Accuracy results for different numbers of children poles and expansion terms in the evaluation of Eq. for a linear potential problem over the square of Fig. 8

increases with  $n_c$ . Although the accuracy always increases with  $n$ , there is also always a threshold for the mesh refinement.

**Irregularly-Shaped Domain with Piece-Wise Straight Boundary**

The irregularly-shaped domain of Fig. 11 (Peixoto and Dumont, 2016) is built up with 16 straight segments, which then undergo successive splitting (always keeping the original geometry) in terms of either constant, linear or quadratic elements with up to  $N = 2^{24} = 16,777,216$  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 accuracy analysis of the conventional BEM Eq. (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 actually poses a good challenge to the numerical simulation.

Figure 12 shows the computational times required for constant, linear and quadratic elements

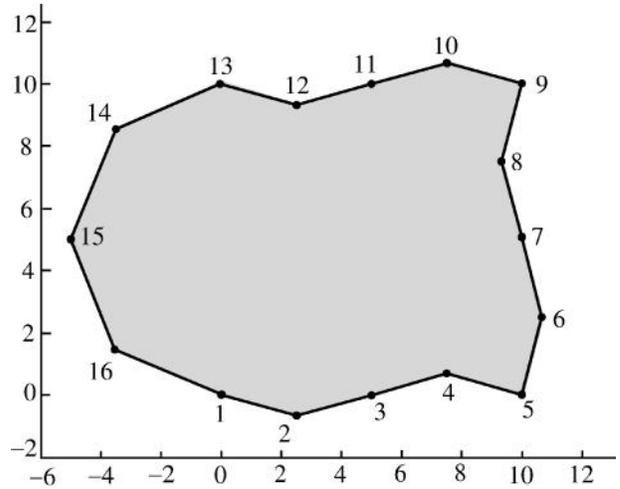


Fig. 11: Irregularly-shaped domain with the boundary given by piece-wise straight segments

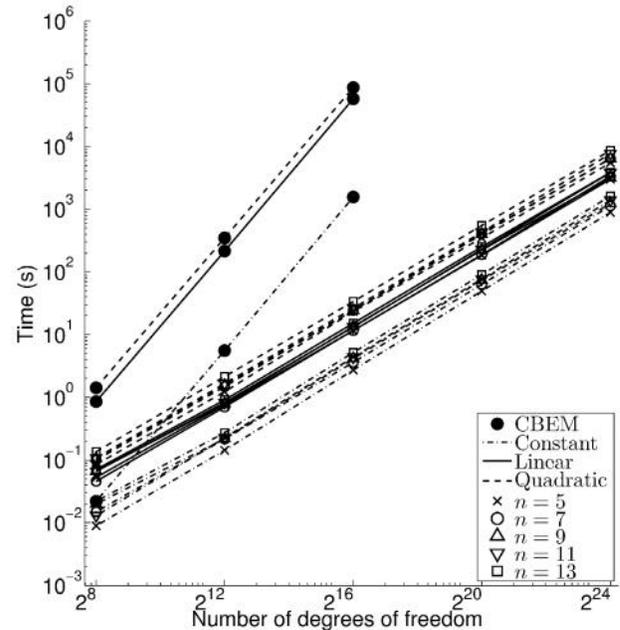


Fig. 12: Execution times for the irregularly-shaped domain of Fig. 11 with  $n_c = 2$  children pole expansions

using  $n = 5, 9, 11, 13$  expansion terms and  $n_c = 2$  children poles. The running times are all proportional to a function between  $N$  and  $N \log N$ , as in the previous example. The most time expensive case of all did not require more than 15 minutes to run, in the case of constant elements – and about ten times that for quadratic elements (the accuracy gain, according to Fig. 13, makes quadratic elements always incomparably more advantageous). For comparison, the evaluations in terms of the conventional BEM (solid

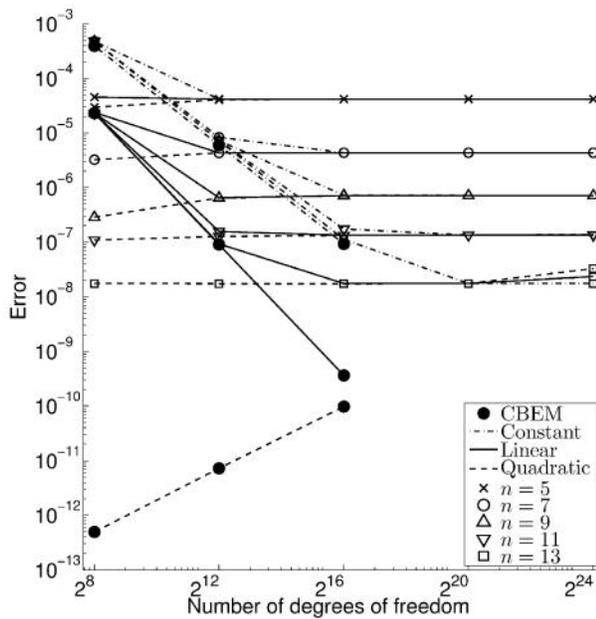


Fig. 13: Accuracy results for the irregularly-shaped domain of Fig. 11 with  $n_c = 2$  children pole expansions

circles), which are proportional to  $N^2$ , required over a day of computation for just  $2^{16} = 65,536$  degrees of freedom.

A convergence analysis using the Euclidean norm of Eq. (13) is shown in Fig. 13 for all cases discussed with regard to Fig. 11 and also using the results with the conventional BEM for comparison.

Owing to the applied quadratic potential field and the piece-wise straight boundary, the results for quadratic elements should be accurate up to the implemented numerical integration precision of the conventional BEM – and taking round-off errors into account. Then, the errors shown for the FMM using quadratic elements are exclusively related to the expansion approximations. In fact, except for some round-off errors, the error diagrams for quadratic elements are horizontal and serve as thresholds for the simulations with constant and linear elements.

### An Isolated Assessment for Quadratic Elements

For the domain in Fig. 14, only quadratic elements are used in order to have a cleaner display of results. This structure is discretized with up to  $N = 5 \times 2^{18} = 1,310,720$  degrees of freedom and is submitted to a logarithmic potential field  $\ln|z - z_s|$ , where  $z = x +$

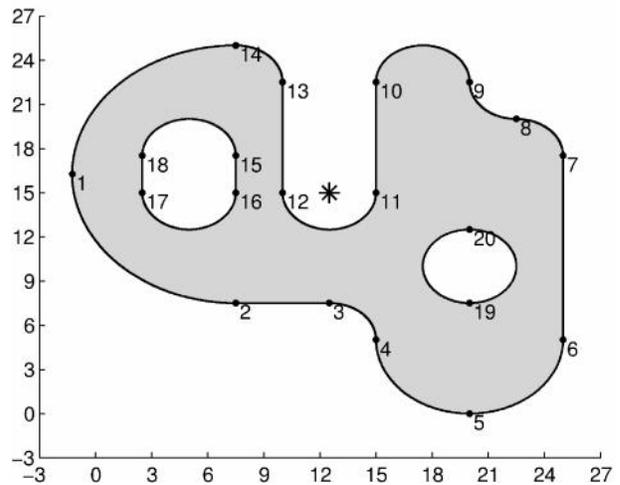


Fig. 14: Irregularly shaped domain for an isolated assessment with quadratic elements

$iy$  is the field point of the domain, and  $z_s = 12.5 + 15i$  is the source point, represented by (\*) in Fig. 14. The execution time and error results are given in Figs. 15 and 16 as outlined for the previous examples and in the case of  $n_c = 2$  children poles. A similar simulation with a larger number of degrees of freedom has already been explored by Peixoto and Dumont (2016).

As already observed, one sees in Fig. 15 that the computational effort increases only slightly with

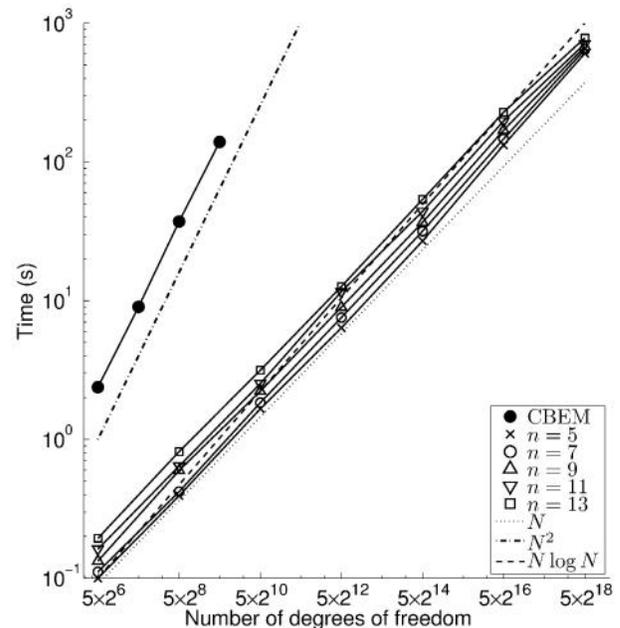


Fig. 15: Execution times for the domain of Fig. 17 using quadratic elements

the number  $n$  of expansion terms. This graph also displays the curves proportional to  $N$  (dotted line),  $N \log N$  (dashed line), and  $N^2$  (dash-dot line), which shows that the present FMM implementation performs close to  $N$ , as already suggested by (Liu, 2009) as an achievable goal.

According to Fig. 15, about three seconds were required for achieving an error  $\varepsilon \approx 10^{-8}$ , according to Eq. (13), in the frame of the FMM with  $N = 5 \times 2^{10} = 5,120$  degrees of freedom and using  $n = 13$  expansion terms, as given in Fig. 16. Over 15 minutes would be required to obtain the same results with the conventional BEM!

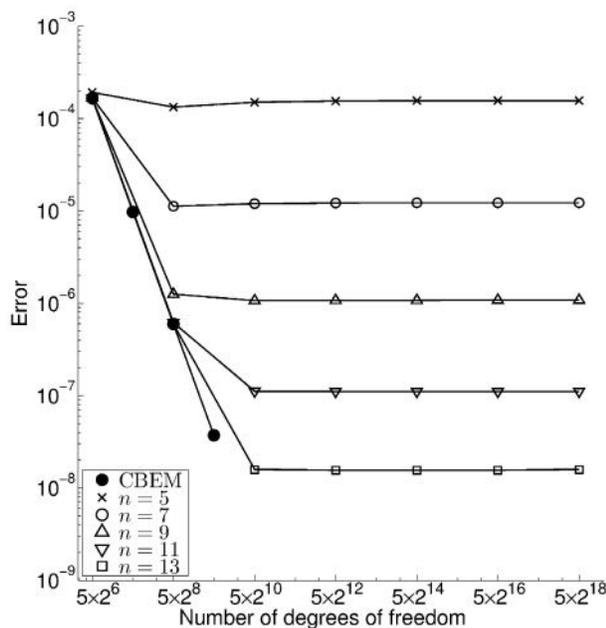


Fig. 16: Accuracy results for the domain of Fig. 17 using quadratic elements

## Conclusion

This paper presents a novel fast multipole implementation of the boundary element method that is almost independent from the type of potential or elasticity problem under investigation and then from the underlying fundamental solution. The formulation relies on a hierarchical mesh refinement strategy for generally curved boundary elements, which is also used in the evaluation of element adjacencies and is key to the proposed algorithm. A compact version of the implemented algorithm is presented, and its application is illustrated for 2D potential problems applied to some irregularly-shaped domains with up to several millions of degrees of freedom.

The core of the proposed fast-multipole methodology consists in implementing Eqs. (1)-(4), as applied to the matrix-vector products of the boundary element Eq. (5), in terms of a recursive procedure that scans the field elements hierarchically. Once the most refined level is achieved from a given superelement, the procedure scans back – also recursively and hierarchically – the whole tree of superelements inside superelements in search of all source poles and nodal points (Novelino, 2015; Peixoto *et al.*, 2015, 2015a) in order to build up fast and efficiently the required matrix-vector products.

The numerical assessments show that the proposed algorithm is seamlessly applicable to generally curved elements of any order, although numerical examples are displayed only for constant, linear and quadratic elements. The simulation of extremely convoluted shapes including multiply-connected domains presents no difficulties. The computational costs for all examples run so far have shown to be closely proportional to  $O(N)$  (order of the number of degrees of freedom), as opposed to a conventional BEM implementation, which requires operations of order  $O(N^2)$ . As a matter of fact, the proposed FMM implementation is superior to a conventional BEM implementation in terms of computational costs even for just a few tens of degrees of freedom, as observed in Figs. 9, 12 and 15.

The proposed FMM has not yet been implemented in an iterative solver, such as the GMRES, since this should be considered as the last step in the developments. In fact, issues such as preconditioning and numerical convergence of the iterative solver itself would only obscure the type of analyses presented in this paper.

Future developments concerning the proposed FMM aim at a three-dimensional formulation and implementation as well as the application to the variationally-based, hybrid boundary element method. Preliminary tests have shown that applications to problems with very complicated fundamental solutions, such as in fracture mechanics (Dumont and Mamani, 2011), may become advantageous regardless of problem size.

## Acknowledgements

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

## References

- Brebbia C A, Telles J C F and Wrobel L C (1984) *Boundary Element Techniques* Springer Berlin Heidelberg
- Dumont N A (2010) *The boundary element method revisited* Boundary elements and other mesh reduction methods XXXII 227-238 WITPress, Southampton.
- Dumont N A (2012) *Unified algorithm for the generation of refined 2D boundary meshes of linear, quadratic and cubic elements* Draft paper, PUC-Rio, Brazil
- Dumont N A and Aguilar C A (2011) *Three-dimensional implementation of the expedite boundary element method* In extended abstracts of the IABEM2011, symposium of the international association for boundary element methods 113-118, Brescia, Italy
- Dumont N A and Aguilar C A (2012) The best of two worlds: The expedite boundary element method *Engineering Structures* **43** 235-244
- Dumont N A and Mamani E Y (2011) Generalized Westergaard stress functions as fundamental solutions *Computer Modeling in Engineering and Sciences (CMES)* **78** 109-149
- Dumont N A and Peixoto H F C (2014) Application of the hybrid boundary element method to large-scale problems using a fast multipole technique. 6 pp on CD, PACAM XIV - 14th Pan-American Congress of Applied Mechanics
- Greengard L and Rokhlin V (1987) A fast algorithm for particle simulations *Journal of Computational Physics* **73** 325-348
- Liu Y (2009) *Fast Multipole Boundary Element Method: Theory and Applications in Engineering* Cambridge: Cambridge University Press
- Liu Y J and Nishimura N (2006) The fast multipole boundary element method for potential problems: A tutorial *Engineering Analysis with Boundary Elements* **30** 371-381
- Nishimura N (2002) Fast multipole accelerated boundary integral equation methods *Applied Mechanics Reviews* **55** 299
- Novelino L S (2015) *A novel fast multipole technique in the boundary element methods* Msc Thesis - PUC-Rio
- Peixoto H F C (2014) *Application of the hybrid boundary element method to large-scale problems using fast multipole techniques* Msc Thesis - PUC-Rio
- Peixoto H F C and Dumont N A (2016) *A kernel-independent fast multipole technique for the analysis of problems with the boundary element method* 8 pp, XII Simpósio de Mecânica Computacional, Diamantina, Brazil
- Peixoto H F C, Novelino L S and Dumont N A (2015) *Basics of a fast-multipole unified technique for the analysis of several classes of continuum mechanics problems with the boundary element method* 47-59 Boundary Elements and Other Mesh Reduction Methods XXXVIII, WITPress Southampton
- Peixoto H F C, Novelino L S and Dumont N A (2015a) *A fast-multipole unified technique for the analysis of continuum mechanics problems with the boundary element methods* 12 pp, CILAMCE – XXXVI Iberian Latin-American Congress on Computational Methods in Engineering, Rio de Janeiro, Brazil.