



A fast multipole implementation of the simplified hybrid boundary element method: application to 2D potential problems

Ney Augusto Dumont  and Helvio de Farias Costa Peixoto 

Department of Civil and Environmental Engineering, PUC-Rio – Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil

ABSTRACT

The ultimate subject of this work is the implementation and testing of a novel numerical tool that can simulate on a personal computer and only in a few minutes a problem with many millions of degrees of freedom. The authors have already successfully developed and tested a technique that turned out to be a modified, reverse fast-multipole implementation for the conventional BEM. The variationally based hybrid BEM leads to a computationally less intensive formulation than in the conventional BEM for large-scale 2D and 3D problems of potential and elasticity. This formulation is especially advantageous for problems of complicated geometry and topology or requiring complicated fundamental solutions. The proposed implementation of the fast multipole method (FMM) for the simplified, hybrid BEM deals with the transpose of the double-layer potential matrix as well as with the nodal matrix expression of the potential fundamental solution. The basic aspects of the FMM are firstly introduced for the conventional BEM as well as for its expedite version. This takes most part of the present paper, which ends up with some validating numerical results. The FMM outline for the simplified hybrid BEM is shown in a separate section, as its numerical implementation is still in progress.

ARTICLE HISTORY

Received 4 November 2016
Accepted 21 February 2017

KEYWORDS

Boundary elements; hybrid boundary elements; fast multipole method; variational methods; potential problems

1. Introduction

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, Novelino, & Dumont, 2015a, 2015b). 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 (Liu, 2009; Liu & Nishimura, 2006; Nishimura, 2002).

In the proposed FMM implementation, a 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. This FMM – which differs from the formulations classically presented in the literature not only because it follows a reverse strategy – has been already assessed for a variety of patch and cut-out tests for 2D potential problems and is being presently implemented for elasticity and 3D problems. It has not been inserted into an iterative solver yet since our goal has consisted in first to validate and assess the isolated FMM algorithm for accuracy, computational effort and storage allocation. The code is written in C++ and can automatically deal with elements of any order – although only linear and quadratic elements have actually been tested. A separate code for constant elements is also implemented.

The following outline combines, extends and improves two papers recently presented at international conferences (Dumont & Peixoto, 2016a, 2016b).

2. Complex formulation of the two-dimensional potential problem

In the following equations $z_s = x_s + iy_s$ is the complex representation of the reference coordinates of a *source* point (as conceptualised in the frame of a boundary element development) and the difference of a field point $z = x + iy$ to the source point is $z - z_s$. Just for the sake of notation simplicity we temporarily assume $z_s = 0$ as the reference coordinates origin.

The fundamental solution for a potential problem is written in complex notation as

$$\theta_s^* = \Re\left(\frac{-1}{2\pi k} \ln(z)\right) \equiv \Re(\theta_s^C) \quad (1)$$

with $z = x + iy$ and including a material property k , such as a conductivity parameter in a stationary heat propagation analysis for a homogeneous and isotropic domain. We check that

$$\frac{-1}{2\pi k} \ln(z) = \frac{-1}{2\pi k} \ln(r) + i \arctan(y, x) \quad (2)$$

where $r = \sqrt{x^2 + y^2}$ is the distance between source and field points. Since

$$\frac{d \ln(z)}{dz} = \frac{1}{z} \equiv \frac{1}{x + iy} = \frac{x}{r^2} - i \frac{y}{r^2} \quad (3)$$

the potential flux in the Cartesian coordinates is defined as

$$q_x = -k \frac{\partial \theta_s^*}{\partial x} = -\frac{x}{r^2} \equiv -\Re \left(k \frac{d\theta_s^C}{dz} \right); \quad q_y = -k \frac{\partial \theta_s^*}{\partial y} = -\frac{y}{r^2} \equiv \Im \left(k \frac{d\theta_s^C}{dz} \right) \quad (4)$$

or, in complex notation,

$$q = q_x + iq_y \equiv -k \frac{d\theta_s^C}{dz} \quad (5)$$

The complex expression of the boundary outward unit vector is

$$n = n_x + in_y \equiv \frac{1}{|J|} \left(\frac{dy}{d\xi} - i \frac{dx}{d\xi} \right) \quad (6)$$

and the expression of the boundary normal flux q^n becomes

$$q_n = -q_x n_x - q_y n_y = k \frac{\partial \theta_s^*}{\partial x} n_x + k \frac{\partial \theta_s^*}{\partial y} n_y \equiv \Re \left(k \frac{d\theta_s^C}{dz} n \right) \quad (7)$$

(The superscript n stands for ‘normal’ to differentiate from the notation used in Equation (5) for flux in the Cartesian directions.). This comes from $\frac{n}{z} \equiv \frac{1}{x+iy} (n_x + in_y) \equiv \frac{1}{r^2} (xn_x + yn_y) + \frac{i}{r^2} (xn_y - yn_x)$.

3. Basic equations of the conventional, collocation boundary element method

In the collocation boundary element method, a generic potential problem is formulated, in the absence of domain sources just for the sake of simplicity, as the compatibility matrix equation

$$\mathbf{Hd} = \mathbf{Gq} \quad (8)$$

for the vectors of boundary nodal potentials $\mathbf{d} \equiv d_f$ and nodal normal flux attributes $\mathbf{q} \equiv q_\ell$ applied as mixed boundary conditions. The double-layer and single-layer potential matrices \mathbf{H} and \mathbf{G} are defined as

$$\mathbf{H} \equiv H_{sf} = \Re \left(H_{sf}^C \right) \quad \text{and} \quad \mathbf{G} \equiv G_{s\ell} = \Re \left(G_{s\ell}^C \right) \quad (9)$$

With

$$H_{sf}^C = k \int_{\Gamma} \frac{d\theta_s^C(z(\xi) - z_s)}{dz(\xi)} n(\xi) u_f(\xi) d\Gamma(\xi) \equiv \int_{\Gamma} q_s^C u_f d\Gamma \quad (10)$$

$$G_{s\ell}^C = \int_{\Gamma} \theta_s^C(z(\xi) - z_s)t_{\ell}(\xi)d\Gamma(\xi) = \frac{-1}{2\pi k} \int_{\Gamma} \ln(z - z_s)t_{\ell}d\Gamma \equiv \int_{\Gamma} \theta_s^C t_{\ell}d\Gamma \quad (11)$$

In the above definitions of $\mathbf{H} \equiv H_{sf}$ and $\mathbf{G} \equiv G_{s\ell}$, the subscript s denotes a node at which a point *source* is applied, f is a *field* node to which a nodal potential d_f is attached and ℓ is a point on the boundary corresponding to the nodal normal flux q_{ℓ} . Integration is carried out along a boundary segment in terms of a parametric variable ξ , as indicated in the latter two equations, in which it is also shown that, for the sake of notation simplicity, the argument ξ may be dropped.

While the fundamental solution defined in Equation (1) has global support, both potential and normal flux quantities θ and q^n are piecewise approximated along the boundary (thus with local support), as already indicated in Equations (10) and (11), by:

$$\theta = u_f(\xi)d_f \quad \text{and} \quad q^n = t_{\ell}(\xi)q_{\ell}, \quad \text{where} \quad t_{\ell} = u_{\ell} \frac{|J|_{(\text{at } \ell)}}{|J|} \quad (12)$$

where $u_f(\xi) \equiv u_f$ and $u_{\ell}(\xi) \equiv u_{\ell}$ (the argument ξ is usually dropped, for notation simplicity) stand for the same type of real polynomial interpolation functions of a given order (constant, linear and quadratic functions are implemented in our code). In the above and in the following equations repeated indices mean that a sum is being carried out. There is no sum indicated in the latter expression above as ‘(at ℓ)’ is not an index. Since $u_f(\xi)$ and $u_{\ell}(\xi)$ have local support, there is no need to make explicit that the integrations indicated in the evaluation of H_{sf} and $G_{s\ell}$ are carried out segment by segment along the boundary. The expression of $t_{\ell}(\xi) \equiv t_{\ell}$ for the interpolation of the normal flux along an element segment (as well as for boundary traction forces in an elasticity problem) stems from a consistent boundary element formulation proposed by Dumont (2010) for generally curved elements.

The subscripts used in Equation (9) and in the following equations play an important role in a consistent formulation. Let $o_e = 1, 2, 3, \dots$ be the order of a generic boundary element, that is, linear, quadratic, cubic and so on, with $o_e + 1 = 2, 3, 4, \dots$ nodes in an element. If a given problem is discretised with n_e boundary elements, then the number of source points s is $n_e \times o_e$, which is also the number of field points f : the double layer potential matrix \mathbf{H} is square and of order $n_e \times o_e$, which is also the size of the vector of nodal potentials \mathbf{d} . On the other hand since two adjacent boundary segments in general do not share a common normal at their connecting nodes (maybe unwillingly, as the result of the imprecise geometry representation of a curved, piecewise smooth boundary), the left and right normals \vec{n} at these points – see Equation (6) – should be explicitly considered in a generic formulation, so that the size of the vector of normal fluxes \mathbf{q} becomes $n_e \times o_e + n_e$, which is also the number of columns of the single-layer potential matrix \mathbf{G} and which is also why subscript ℓ is used (f and ℓ would be

used indistinctly only in the case of constant elements). By the way, it is worth remarking that the concepts of *continuous* or *discontinuous* nodes at corner points are considered by the first author as inconsistent and misleading and should not take place in a consistent formulation (Dumont, 2010).

4. Basic equations of the simplified hybrid boundary element method

The simplified hybrid boundary element method has been well explained by Dumont and Aguilar (2012), for example, and is derived from the fully variational formulation proposed by Dumont (1989), which is also shortly reviewed in Liu et al. (2011). For the simplest case of a potential problem, it relies on the assumption that the potential θ and its gradients θ_j inside a domain can be described in terms of a series of point source parameters p_s^* applied along the boundary plus some arbitrary particular solution θ^p ,

$$\theta = (\theta_s^* + C_s)p_s^* + \theta^p, \quad \theta_j = \theta_{s,j}^*p_s^* + \theta_j^p \quad (13)$$

where θ_s^* is a fundamental solution of the corresponding differential equation of the problem, as given in Equation (1) for the Laplace equation. In the present case, $\nabla^2\theta_s^* = 0$ except for the point of application of p_s^* , when θ_s^* becomes undetermined. Moreover, θ_s^* is obtained except for a constant C_s (Dumont, 2010; Dumont & Aguilar, 2012). This belongs to the basic theory of the conventional boundary element method only bearing in mind that in a variational formulation θ_s^* is used as a numerical approximation of the actual problem and not just as a weighting function [references given in Dumont (2010)].

Assigning subscripts D and N to subvectors of nodal potentials \mathbf{d} and equivalent nodal fluxes \mathbf{p} to characterise whether the boundary conditions are of Dirichlet or Neumann type, the final matrix equation system of the simplified hybrid boundary element method is expressed as

$$\begin{bmatrix} \mathbf{H}_N^T \\ \mathbf{H}_D^T \end{bmatrix} \mathbf{p}^* = \begin{Bmatrix} \mathbf{p}_N - \mathbf{p}_N^p \\ \mathbf{p}_D - \mathbf{p}_D^p \end{Bmatrix}, \quad \begin{bmatrix} \mathbf{U}_N^* \\ \mathbf{U}_D^* \end{bmatrix} \mathbf{p}^* = \begin{Bmatrix} \mathbf{d}_N - \mathbf{d}_N^p \\ \mathbf{d}_D - \mathbf{d}_D^p \end{Bmatrix} \quad (14)$$

where the quantities with superscript p stand for nodal potentials or fluxes related to an assumed, arbitrary particular solution of the problem. \mathbf{H} is the same double-layer potential matrix of the conventional, collocation boundary element method given in Equation (10), and \mathbf{U}^* represents the fundamental solutions θ_s^* evaluated at the boundary nodal points, that is, $\mathbf{U}^* \equiv U_{f_s}^* \equiv \theta_s^*(z_f - z_s)$. The vector $\mathbf{p} \equiv p_f$ of equivalent nodal normal flux is obtained from the distributed normal flux $\mathbf{q} \equiv q_\ell$ introduced in Equation (8):

$$\mathbf{p} = \mathbf{L}^T \mathbf{q}, \quad \text{with} \quad \mathbf{L} \equiv L_{\ell f} = \int_{\Gamma} u_f(\xi) t_\ell(\xi) d\Gamma(\xi) \quad (15)$$

The above transformation matrix \mathbf{L} can be expressed as

$$L_{\ell f} \equiv \tilde{L}_{\ell f} |J|_{(\text{at}\ell)} \tag{16}$$

where $|J|_{(\text{at}\ell)}$ is defined as in Equation (12) and $\tilde{L}_{\ell f}$ is a pre-evaluated result given as the block matrices below for linear, quadratic and cubic elements:

$$\tilde{L}_{\ell f} = \left[\begin{array}{c} \frac{1}{3} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \\ \frac{1}{15} \begin{bmatrix} 4 & 2 & -1 \\ 2 & 16 & 2 \\ -1 & 2 & 4 \end{bmatrix} \\ \frac{1}{840} \begin{bmatrix} 128 & 99 & -36 & 19 \\ 99 & 648 & -81 & -36 \\ -36 & -81 & 648 & 99 \\ 19 & -36 & 99 & 128 \end{bmatrix} \end{array} \right] \tag{17}$$

Equation (14) can be firstly solved for \mathbf{p}^* in terms of the known nodal quantities $\mathbf{p}_N - \mathbf{p}_N^p$ and $\mathbf{d}_D - \mathbf{d}_D^p$ provided that the problem is well posed, with the subsequent evaluation of the unknown boundary potentials and fluxes. Then, the equation to be implemented in the frame of the fast multipole method is

$$\begin{bmatrix} \mathbf{H}_N^T \\ \mathbf{U}_D^* \end{bmatrix} \mathbf{p}^* = \begin{Bmatrix} \mathbf{p}_N - \mathbf{p}_N^p \\ \mathbf{d}_D - \mathbf{d}_D^p \end{Bmatrix} \tag{18}$$

Results at internal points are obtained from \mathbf{p}^* directly using Equation (13). Results close to or at nodal points can also be obtained (see, for instance, Dumont & Aguilar, 2012). The implemented 2D boundary element code works with linear, quadratic or cubic elements.

5. Basic equations of the expedite boundary element method

The matrices \mathbf{H} and \mathbf{G} of the conventional, collocation boundary element method and, in a similar reasoning, the matrix \mathbf{H}^T of the simplified hybrid boundary element method, may be obtained in an expedite way that consists in approximating the fundamental solution along a boundary segment using the same interpolation functions introduced in Equation (12). Equation (8) can be developed – in its complex version as ready to be implemented in the frame of a fast multipole algorithm – according to the following equation, which uses in the second row the definitions of Equations (10) and (11) and then approximates the fundamental solutions themselves along each boundary segment according to Equation (12), thus resulting into a very simplified (thus the term *expedite*) form in terms of the transformation matrix \mathbf{L} of Equation (15):

$$\begin{aligned} H_{sf}^C d_f &= G_{s\ell}^C q_\ell \\ \left(\int_\Gamma q_s^C u_f d\Gamma \right) d_f &= \left(\int_\Gamma \theta_s^C t_\ell d\Gamma \right) q_\ell \\ T_{\ell s}^C \left(\int_\Gamma t_\ell u_f d\Gamma \right) d_f &\approx \theta_{sf}^C \left(\int_\Gamma u_f t_\ell d\Gamma \right) q_\ell \\ T_{\ell s}^C L_{\ell f} d_f &\approx U_{sf}^C L_{\ell f} q_\ell \\ &\text{or} \\ T_{\ell s}^C L_{\ell f} d_f &\approx U_{sf}^C p_f \end{aligned} \tag{19}$$

In this development, $T_{\ell s}^C$ expresses the complex fundamental solution q_s^C evaluated at node ℓ (and corresponding outward normal for the considered boundary segment) and $U_{fs}^C \equiv U_{sf}^C$ is the result of θ_s^C evaluated at node f . The expedite integration scheme represented as

$$H_{sf}^C \approx T_{\ell s}^C L_{\ell f} \quad \text{and} \quad G_{s\ell}^C \approx U_{sf}^C L_{\ell f} \quad (20)$$

only applies if the source point given by s is sufficiently far from the boundary segment where an integration should be carried out.

6. Proposed FM algorithm for a general, complex function

The following basic definitions are used in the present developments to represent a general function in the complex domain $f(z)$:

- $z - z_s$ = difference between the source point z_s and the field point z .
- z_{c^k} , $k = 1, 2, \dots, n_c$: hierarchical levels of poles about which the fundamental solution will be successively expanded for the field point z (then, by definition, $z_{c^0} \equiv z$).
- z_{L^l} , $l = 1, 2, \dots, n_L$: hierarchical levels of poles about which the fundamental solution will be successively expanded for the source point z_s (by definition, $z_{L^0} \equiv z_s$).

The above definitions of a pole z_{c^k} that is *close* (lower case c) to the field point z and of a pole z_{L^l} that is *local* (upper case L) to the source point z_s follow the notation introduced by Liu (2009). In the following developments, each *close* pole z_{c^k} and each *local* pole z_{L^l} are actually array representations of different hierarchical levels of poles, as illustrated in Figure 2, where the attached superscripts (here omitted, for simplicity) denote an individual pole in the array.

The expression of a generic fundamental solution for 2D problems is ultimately expanded for the field point z about the *close* pole $z_{c^{n_c}}$ (of highest level, as developed next) using n terms as well as for the source point z_s about the local point $z_{L^{n_L}}$ (also of highest level) using m terms:

$$f(z - z_s) = \sum_{i=1}^{n+1} \frac{1}{(i-1)!} P_i(z - z_{c^{n_c}}) \sum_{j=1}^{m+1} \frac{1}{(j-1)!} P_j(z_{L^{n_L}} - z_s) Q_{i+j-1}(z_{c^{n_c}} - z_{L^{n_L}}) \quad (21)$$

for truncation order given by $\max\left(|(z - z_{c^{n_c}})/(z - z_s)|^{n+1}, |(z_{L^{n_L}} - z_s)/(z - z_s)|^{m+1}\right)$.

Using Δz as a generic difference between poles, the expansion terms of the array $Q(\Delta z)$ in the above equation are

$$Q(\Delta z) = \left\langle f(\Delta z) \quad \frac{df(\Delta z)}{dz} \quad \frac{df^2(\Delta z)}{dz^2} \quad \frac{df^3(\Delta z)}{dz^3} \quad \dots \quad \frac{df^{m+1}(\Delta z)}{dz^{m+1}} \right\rangle \quad (22)$$

The terms $P_i(z - z_{c^{n_c}})$ and $P_j(z_{L^{n_L}} - z_s)$ in Equation (21) are polynomials evaluated recursively as shown in the following for $P_i(\Delta z) \equiv P_i(z - z_{c^i})$. If $\Delta z \equiv z - z_{c^i}$ characterises the difference between two consecutive poles, then

$$P(\Delta z) = \langle 1 \quad \Delta z \quad \Delta z^2 \quad \Delta z^3 \quad \dots \quad \Delta z^{n+1} \rangle \tag{23}$$

On the other hand, if $\Delta z \equiv z - z_{c^i}$ refers to differences between poles that are not consecutive, $P_i(z - z_{c^i})$ is expressed recursively in terms of a lower level polynomial $P_j(z - z_{c^{i-1}})$ and the polynomial $P_{i+1-j}(z_{c^{i-1}} - z_{c^i})$ of Equation (23) by

$$P_i(z - z_{c^i}) = \sum_{j=1}^i C_{j,i+1-j} P_j(z - z_{c^{i-1}}) P_{i+1-j}(z_{c^{i-1}} - z_{c^i}) \tag{24}$$

where $C_{ij} = 1$ if $i = 1$ or $j = 1$, otherwise, $C_{ij} = C_{i-1,j} + C_{i,j-1}$. With this recursive approach, the polynomials $P_i(z - z_{c^{n_c}})$ and $P_j(z_{L^{n_L}} - z_s)$ in Equation (21) always end up expressed in terms of arguments given as differences of poles in two consecutive levels, according to Equation (23).

In general, the higher derivatives in Equation (22) of the fundamental solution tend rapidly to zero when evaluated for large arguments. Equation (21) is the starting point for a procedure that leads to a computationally fast and economical evaluation of a given fundamental solution $f(z - z_s)$ for a very large number of source points z_s by means of a sufficiently approximate expression. As shown, $f(z - z_s)$ is expanded in terms of successive arrays of source poles z_{L^i} as well as of field poles z_{c^k} . The expansion ends up with a series of products of polynomials $P_i(z - z_{c^{n_k}})$ and $P_i(z_s - z_{L^{n_i}})$, which are independent from the complexity of the function $f(z - z_s)$, multiplied by functions $Q_i(z_{L^{n_i}} - z_{c^{n_k}})$ that are given as $f(z_{L^{n_i}} - z_{c^{n_k}})$ and its $2n$ first derivatives, for the expansion indicated in Equation (21). 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^i} and z_{c^k} . Then, the evaluation of $Q_i(z_{L^{n_i}} - z_{c^{n_k}})$ may end up orders of magnitude less intensive than the direct evaluation of $f(z - z_s)$ for all source and field points.

6.1. Adjacency search

The adjacency information to a boundary segment is based on a hierarchical boundary refinement. This scheme consists in splitting an element – be it constant, linear, quadratic or cubic, as implemented, – into two smaller ones and sequentially assigning a global numbering to the new nodes as they are created (Figure 1).

Figure 2 shows three cases of possible refinements, with 2, 4 or 8 child elements (n_c) per element. As each element is split into two new elements, n_c is always a power of 2 for a 2D problem.

This splitting scheme provides a direct way of assessing adjacency by node numbering (topological adjacency), which is adequate in the case of a convex domain

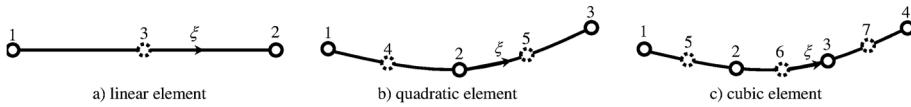


Figure 1. Scheme of three different elements that are split each into two sub-elements.

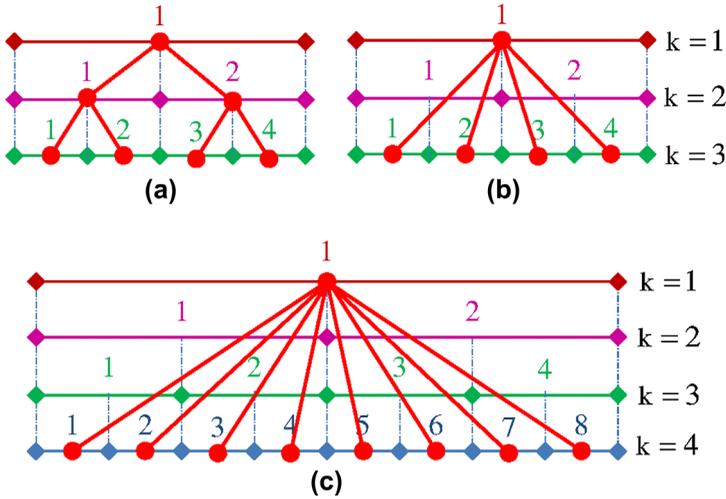


Figure 2. Schematic pole expansions using numbers of child poles $n_c = 2, 4$ or 8 (Novelino, 2015).

– or when its shape is not too irregular. For domains with holes, sharp corners or notches, for instance, a geometry-based adjacency assessment is required, which may become computationally expensive. The proposed adjacency search uses the hierarchical refinement shown in Figure 2 to reduce the number of possible adjacent elements, therefore reducing the need of evaluating distances geometrically.

Figure 3 shows a square domain with a hole to be assessed at two different refinement levels. If the topological adjacency were to be considered in such a domain for the refinement level $k = 0$, on the left, element 5 would not be detected as adjacent to element 1, as they are 4 elements apart. This illustrates a case that requires a geometry-based assessment. Using the hierarchical refinement, it is possible to assign to a given element at level k its child, split elements at level $k + 1$. This information is used to reduce the number of candidate adjacent nodes, and therefore the number of distance evaluations.

As illustrated on the left of Figure 3 for the level $k = 0$, a search is carried out for element 1 using the two circles centred on its nodes, and it comes out that the nodes inside the circles (corners of the square hole) are adjacent. Then, any element that contains at least one of these nodes is considered adjacent to element 1. The search radius is $AdjTOL \times L$, where L is the length of the reference element. Good numerical results have been obtained in the frame of the implemented fast multipole algorithm using $0.7 \leq AdjTOL \leq 2$.

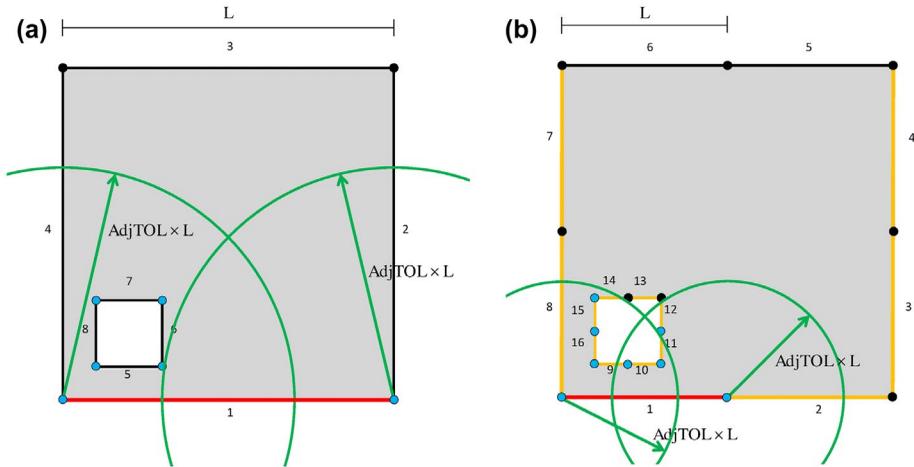


Figure 3. Schemes for the adjacency search at the coarser refinement level $k = 0$ and at the next refinement level $k = 1$ (Peixoto & Dumont, 2016b).

Element 1 on the right of Figure 3 corresponds to a level $k = 1$ and has been generated from element 1 at level $k = 0$. Since the adjacent elements of element 1 at level $k = 0$, on the left figure, are already known, they are the candidates to have adjacent child elements at level $k = 1$, that is, all elements in the figure except for elements 5 and 6. Once more, search circles with radius proportional to the element length are drawn and nodes inside them are marked as adjacent.

An element's adjacency list is built as the hierarchical mesh refinement proceeds up to the highest level. This list is generated and stored for just one element at a given refinement level.

6.2. Implemented FM algorithm

This section describes a compact version of the implemented fast-multipole algorithm, as applied to potential problems in the conventional BEM and described above. 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 (see also Dumont & Peixoto, 2016a).

The procedure Main (Figure 4) loads the input data, generates the hierarchical mesh according to the concepts briefly discussed in the Introduction and evaluates the kernel expansions according to Equation (22), which is the only part of the code that is kernel dependent. 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 .

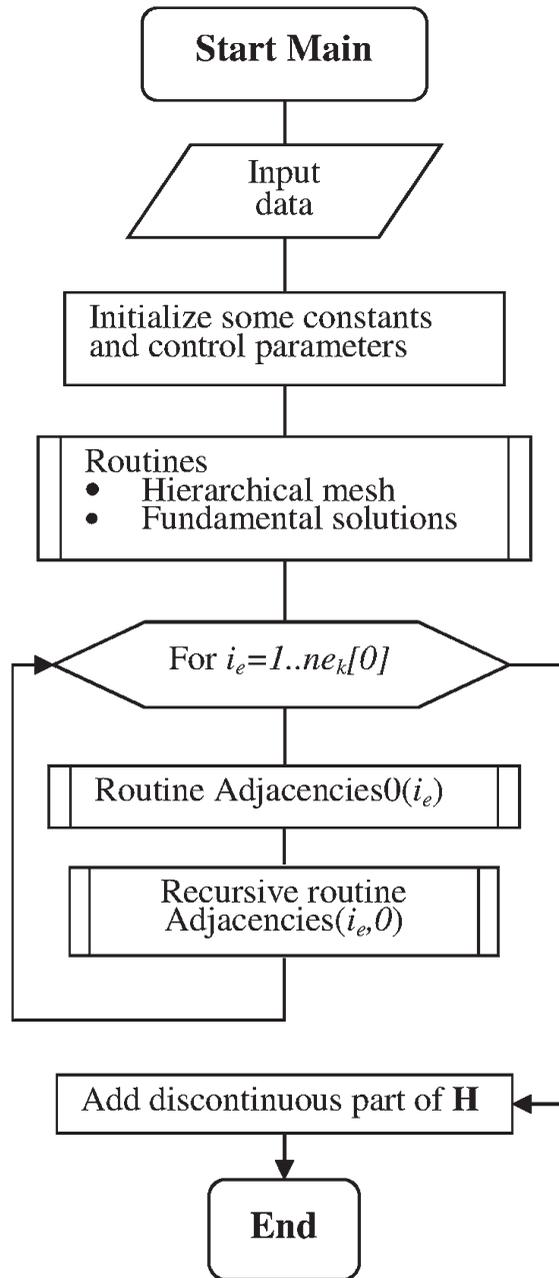


Figure 4. Procedure Main (Dumont & Peixoto, 2016a).

The routine Adjacencies (Figure 5), which is actually preceded by an initialising routine Adjacencies0 also called by the procedure Main, assembles the adjacency structure and when it reaches the most refined level ($k = n_v$), calls the routine Source. This routine (Figure 6) handles integrations in terms of the conventional BEM matrix-vector products (routine BEMAdj, not shown) for the

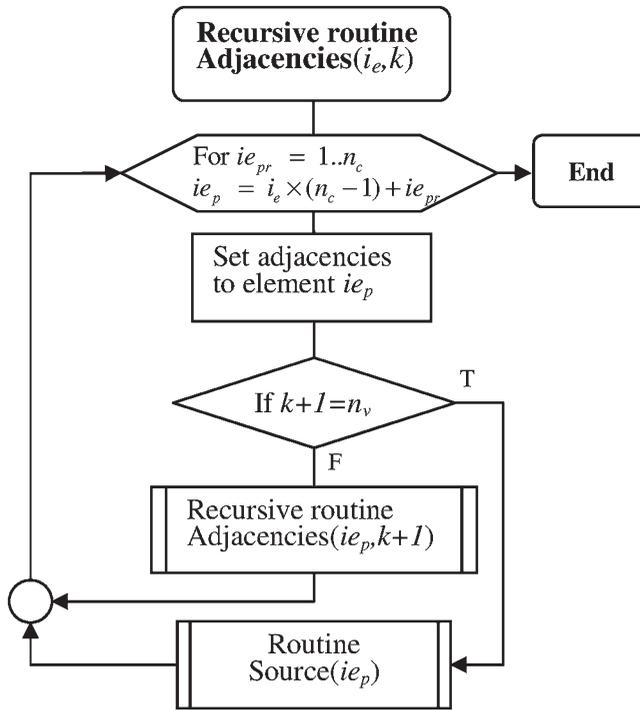


Figure 5. Recursive procedure Adjacencies (Dumont & Peixoto, 2016a).

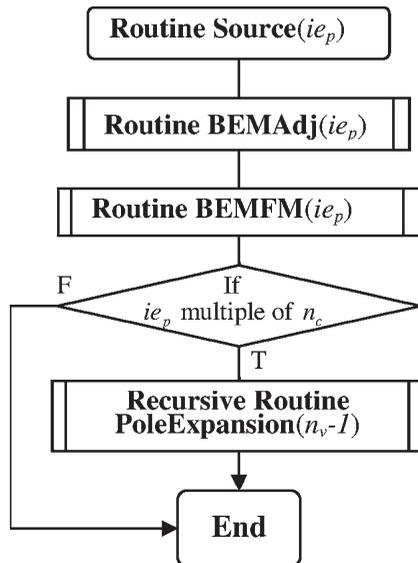


Figure 6. Procedure Source (Dumont & Peixoto, 2016a).

adjacent elements, as well as in terms of FM expansions (routine BEMFM, also not shown). 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

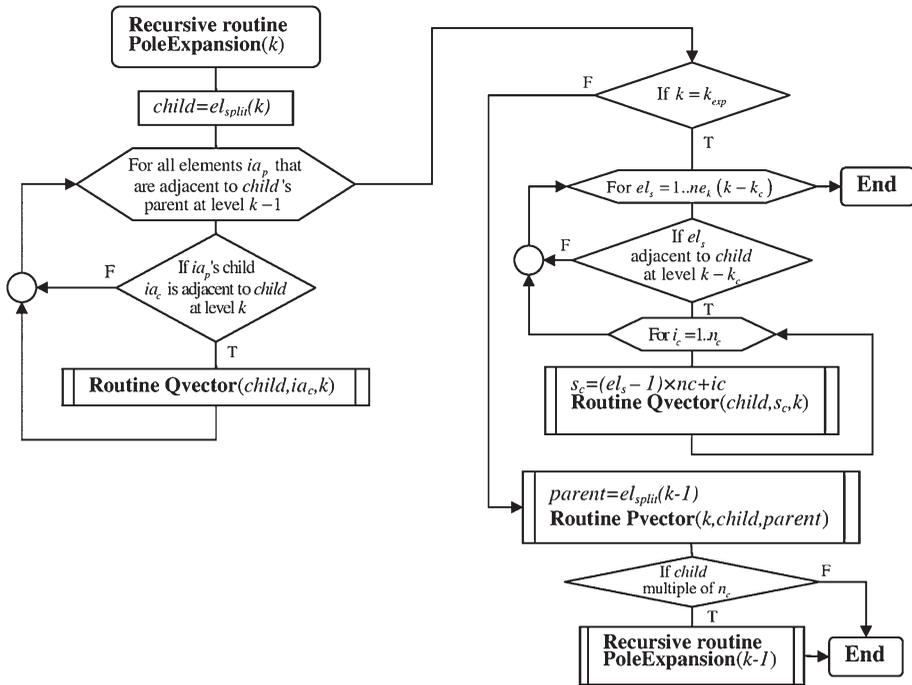


Figure 7. Recursive procedure PoleExpansion (Dumont & Peixoto, 2016a).

far-field elements in the routine PoleExpansion. The routine Source also leads to the successive expansion of the FM integration terms, thus delivering data information to the upper refinement levels, if this is the case, by calling the recursive routine PoleExpansion.

Finally, the recursive routine PoleExpansion (Figure 7) delivers the FM-integrated data to the source poles that are considered sufficiently far by calling the routine Qvector to evaluate the array of kernel expansions defined in Equation (22) and then evaluating the expansion series for the source point, Equation (21). 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 directly delivered to the remaining source points. If $k = k_{exp}$ has not been reached, the routine calls routine Pvector (not shown) to convey the obtained data, according to Equation (24), to the upper pole levels and, when all elements of level k have been processed, calls itself (thus recursively) to proceed to the immediately upper refinement level of the hierarchical structure.

7. Numerical results

In the following, some basic results obtained in the frame of the conventional boundary element method (CBEM), that is, using Equation (8), are displayed in order to set the highest accuracy one may expect to achieve in the analyses.

Some numerical experimentations with the expedite boundary element method (EBEM) are shown subsequently.

7.1. Some target results for the implemented fast multipole method

Figure 8 shows two irregularly shaped domains that will undergo an hierarchical mesh refinement and then will be submitted to a testing potential field (Peixoto & Dumont, 2016). Given an in principle arbitrary analytical solution of the Laplace equation, for the present potential problem, a vector \mathbf{d} of potential values and a vector \mathbf{q} of normal flux values are obtained along the nodal points of the boundary drawn in the open field, as illustrated in the figure. This is called a cut-out test, as the accuracy of the BEM Equation (8), that is, of the matrices \mathbf{H} and \mathbf{G} , can be assessed for arbitrarily distorted domains and different mesh refinements, as the vectors \mathbf{d} and \mathbf{q} are always known analytically. The code is implemented in the language C++ and runs on a desktop computer (i7™-4770 CPU 3.4 GHz, 16 GB RAM in Windows® 7). The errors presented on the right of Figures 9 and 10 represent the Euclidean error norm

$$\varepsilon = \frac{|\mathbf{H}\mathbf{d} - \mathbf{G}\mathbf{q}|}{|\mathbf{G}\mathbf{q}|} \quad (25)$$

For the domain on the left of Figure 8, a quadratic field $x^2 - y^2$ is applied as a target, analytical solution of the Laplace equation. The boundary is discretised with constant, linear and quadratic elements with up to $2^{24} = 16,777,216$ degrees of freedom as represented in the horizontal axis of both graphs in Figure 9.

Figure 9 shows on the left the time required for running simulations with different degrees of freedom (horizontal axes) and different numbers n of terms in the series, according to Equation (21), for each element type. It is worth noticing that, for a given element type, increasing the number of terms in the series does

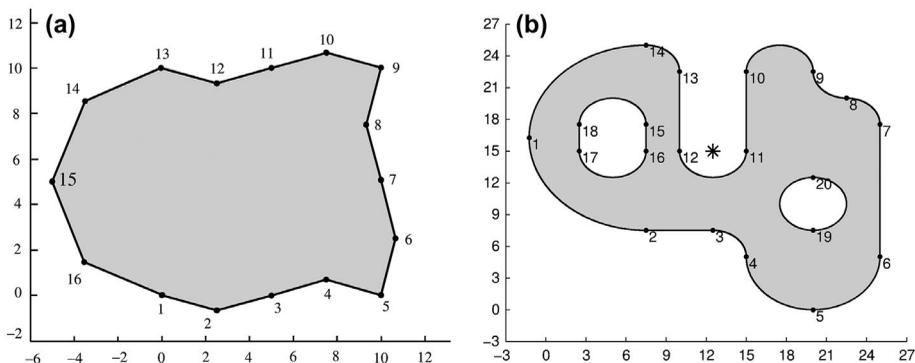


Figure 8. Domains used in the numerical assessments.

Notes: Left: stepwise linear boundary submitted to a quadratic field. Right: irregularly shaped domain submitted to a logarithmic field with source applied at the point marked * (Peixoto & Dumont, 2016a).

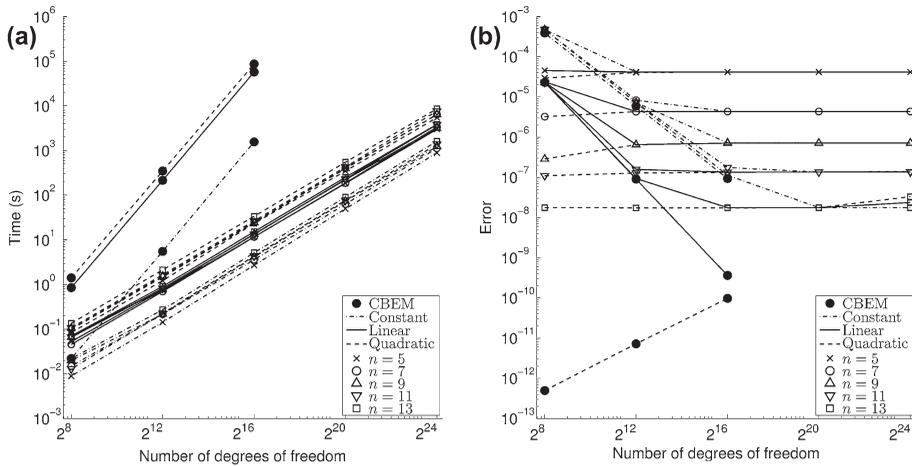


Figure 9. Execution times (left) for the evaluation of Equation (8) for the domain on the left of Figure 8 using constant, linear and quadratic elements, and accuracy results (right) for different numbers n of expansion terms (Peixoto & Dumont, 2016a).

not lead to a considerable increase in the execution time. On the other hand, the graph on the right of Figure 9 shows that the number n of expansion terms considerably affects the numerical accuracy. The full circles characterise in both graphs of Figure 9 results obtained by evaluating the matrix-vector products \mathbf{Hd} and \mathbf{Gq} as in a CBEM implementation. Since the applied analytical open-field is quadratic, the CBEM solution for quadratic elements (dashed lines) is as accurate as the numerical integration and round-off errors allow. However, when evaluated via the FMM, there is an intrinsic error due to the series expansions. This error poses an accuracy threshold to simulations with the lower order elements.

For the curved domain on the right of Figure 8, only quadratic elements are used. This structure is discretised with up to $5 \times 2^{18} = 1,310,720$ degrees of freedom and is submitted to a logarithmic field $\ln |z - z_s|$ where $z_s = 12.5 + 15i$ is the source point, represented by (*) in Figure 8, and $z = x + iy$ is a generic field point. The execution time and error results are given in Figure 10 exactly as outlined for the previous numerical example.

As already observed, one sees on the left of Figure 10 that the computational effort increases only slightly as the number n of expansion terms increases. This graph also displays the curves proportional to N (dotted line), $N \log N$ (dashed line) and N^2 (dash-dot line). One observes that, while the implementation of the matrix-vector product in terms of the CBEM requires a computational time proportional to N^2 , the present FMM implementation performs close to N , as already suggested by Liu (2009) as an achievable goal.

The error assessment on the right of Figure 10 goes up to 5×2^{18} degrees of freedom, although the error threshold for the FMM expansions is already arrived

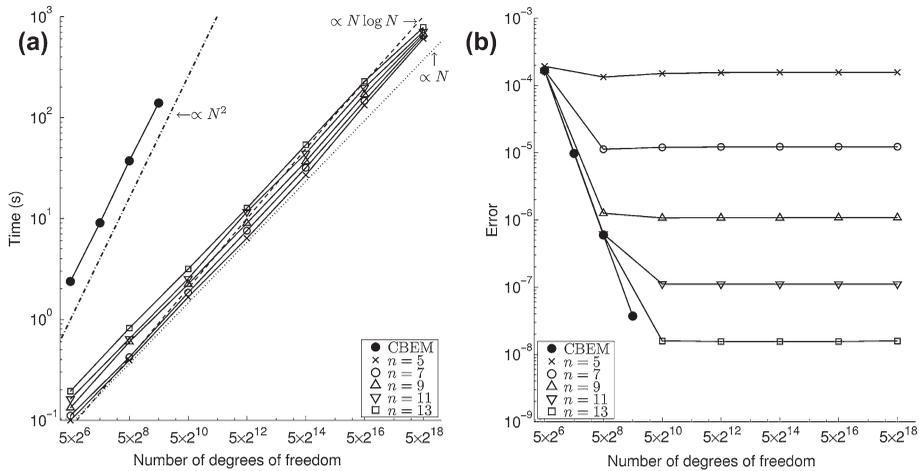


Figure 10. Execution times (left) for the evaluation of Equation (8) for the domain on the right of Figure 8 using quadratic elements, and accuracy results (right) for different numbers n of expansion terms (Peixoto & Dumont, 2016a).

at for 5×2^{10} degrees of freedom, with the same convergence behaviour observed in the previous example.

7.2. Assessment of the expedite approximation of the boundary element method

Computation cost and numeric accuracy of the expedite approximation of the CBEM are assessed in this section by means of a few simulations using either linear or quadratic (and curved) elements. The proposed approximations of the single- and double-layer potential matrices are given in Equation (20).

Two different domains, as depicted in Figure 11, are considered. The square domain on the left is discretised with either linear or quadratic elements with up to 1024 degrees of freedom. On the right of Figure 11 is shown an irregularly shaped domain defined by 16 initial nodes and discretised with quadratic elements with up to 2048 degrees of freedom. Both domains are submitted to a logarithmic field due to a point source applied at $z_s = 3 + 11i$ for the domain on the left, and $z_s = -3 - i$ for the one on the right of Figure 11. Corresponding nodal potentials \mathbf{d} as well as normal fluxes \mathbf{q} are evaluated along the boundary nodes to assess the accuracy of Equation (8) by applying the Euclidean error norm of Equation (25).

7.2.1. Results for the square domain of Figure 11

The times required to run the simulations for the square domain (left of Figure 11) with linear and quadratic discretisations are shown on the left and right graphics of Figure 12. Four $AdjTOL$ parameters, as presented in the section on adjacency search, are studied: 10^{-4} , 3, 6 and 10. The value $AdjTOL = 10^{-4}$ leads to the same results of a topological adjacency.

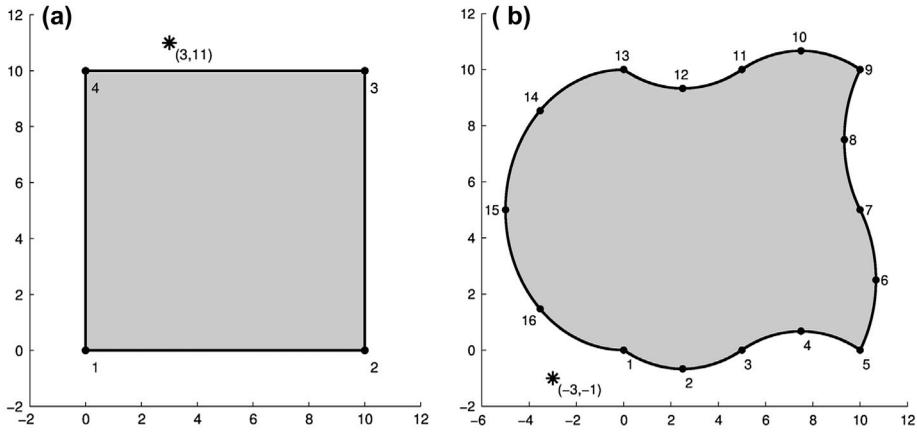


Figure 11. Regular square and deformed quadrilateral domains used for the numerical assessments (Peixoto & Dumont, 2016b).

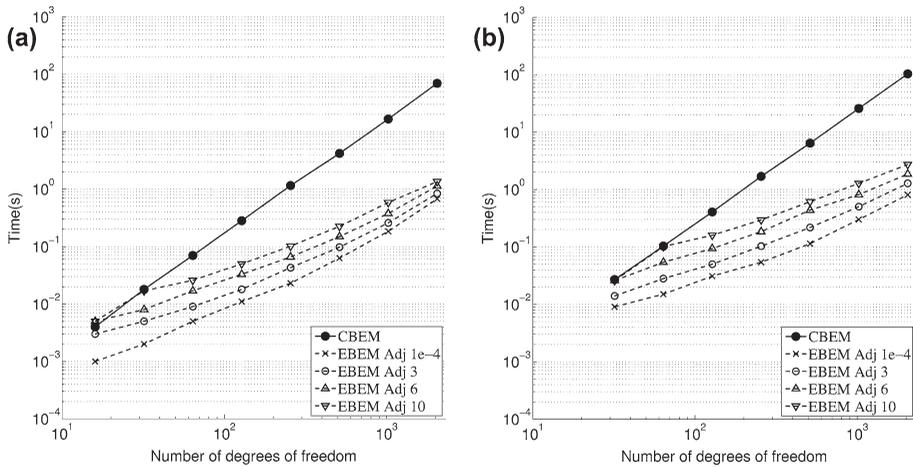


Figure 12. Execution times for the discretisation of the square domain of Figure 11 with linear (left) and quadratic (right) elements (Peixoto & Dumont, 2016b).

In both sets of simulations it may be seen that the CBEM is by far more time expensive than the EBEM, even for large *AdjTOL* values (when an element has a large number of adjacent elements and thus few expedite approximations take place). If the *AdjTOL* parameter is large enough all elements of a problem end up located inside the search circles and the EBEM simulation performs as a CBEM one. This threshold case happens for the first two quadratic discretisations with *AdjTOL* = 10. The results of Figure 12 show that the computational costs with the EBEM in general increase at a by far lower rate than in the case of the CBEM.

Euclidean error norms, evaluated as in Equation (25) for either case of linear or quadratic element, are shown in Figure 13 for several mesh refinements. As expected, the EBEM simulations lead to larger errors when compared to the

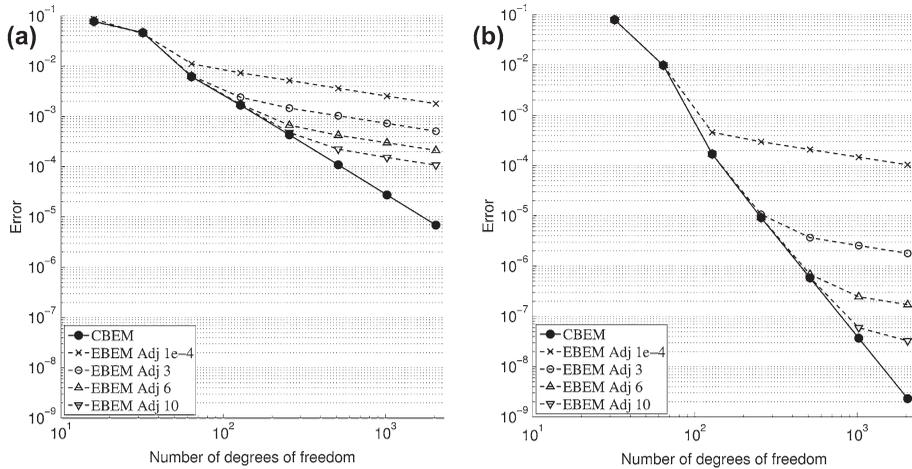


Figure 13. Euclidean error norms for the discretisation of the square domain of Figure 11 with linear (left) and quadratic (right) elements (Peixoto & Dumont, 2016b).

CBEM simulations, except for some trivial cases when they actually coincide computationally. For small values of $AdjTOL$ – that is, more expedite approximations –, it may be seen that the errors are not satisfactory. In fact, the errors for simulations with the EBEM using the topological adjacency (equivalent to $AdjTOL = 10^{-4}$) are at least one order of magnitude larger than with the CBEM whether using linear or quadratic elements.

If one weighs computational effort and accuracy convergence in Figures 12 and 13 it is reasonable to conclude that the CBEM and the EBEM are equivalent in terms of performance if a not too high accuracy is pursued.

7.2.2. Results for the curved domain of Figure 11

In a second assessment, the curved domain on the right of Figure 11 is discretised with quadratic elements, which keep the problem's original geometry throughout refinement. Figure 14 presents both the execution times (left) and the Euclidean error norms (right) for a numerical analysis carried out with up to 2048 nodes. The same four values of $AdjTOL$ of the preceding study are used.

As the adjacency search radius increases, the time needed to execute the simulations with the EBEM also increases, although with significant accuracy improvement, as shown on the right of Figure 14. In the results for $AdjTOL = 10$, that is, for a search radius ten times an element length, the simulation with 1024 nodes shows an error $\varepsilon = 1.6 \times 10^{-7}$ after about 1.2s execution time (result marks surrounded by circles). To achieve such a precision with the CBEM, it is needed to run a simulation with 700 DOFs, which takes slightly more computational time. This analysis shows that the EBEM is capable of delivering small errors with competitive computational time. This method may be recommended as a fast means of obtaining a fair approximation of a complex problem in a reduced amount of

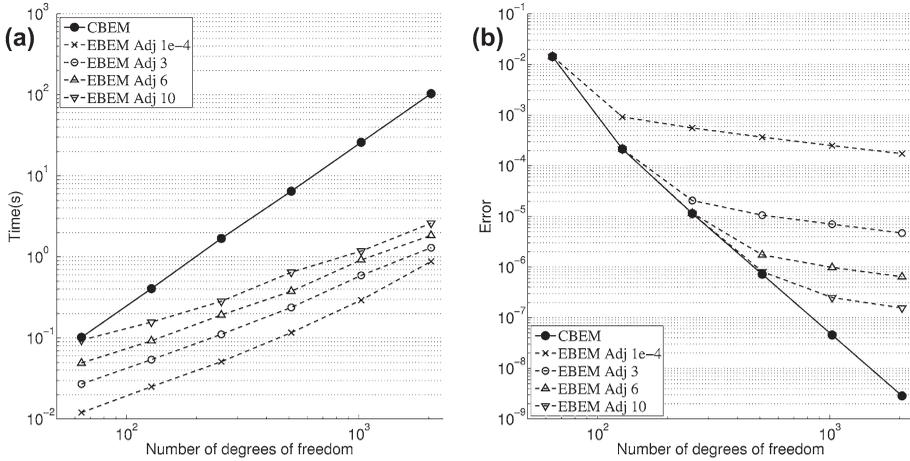


Figure 14. Execution time (left) and Euclidean error norms (right) for the curved domain on the right of Figure 11 discretised with quadratic elements.

time, such as in the evaluation of initial results for iterative methods, as well as for initial mesh approximations in highly convoluted domains.

7.3. Assessment of the expedite fast multipole boundary element method

The fast multipole technique briefly presented in this paper – and thoroughly discussed in Peixoto (2014), Novelino (2015) and Dumont & Peixoto (2016a) – may be applied together with the EBEM in order to push even further the gain in the algorithm speedup delivered solely by the EBEM. Since the FMM relies on polynomial expansions of the fundamental solution in the complex z direction and the EBEM consists in approximating the same fundamental solution along boundary segments, it is worth assessing the error when these two methods are combined.

The expansion of the fundamental solution of Equation (1) as in Equation (21), when applied to the matrices of Equation (20), leads to the fast multipole expansions for the boundary element method together with the expedite approximation:

$$\begin{aligned}
 H_{sf}^C &\approx T_{\ell_s}^C L_{\ell_f} \\
 &= q_s^C n \Big|_{(\text{at}\ell)} L_{\ell_f} = q_s^C \tilde{n} \Big|_{(\text{at}\ell)} \tilde{L}_{\ell_f} \\
 &\approx \tilde{n} \Big|_{(\text{at}\ell)} \tilde{L}_{\ell_f} \sum_{i=2}^{n+2} \frac{1}{(i-1)!} P_{i-1}(z_\ell - z_c) Q_i(z_c - z_s)
 \end{aligned} \tag{26}$$

$$G_{s\ell}^C \approx U_{sf}^C L_{\ell_f} = U_{sf}^C \tilde{L}_{\ell_f} |J|_{(\text{at}\ell)} \approx \tilde{L}_{\ell_f} |J|_{(\text{at}\ell)} \sum_{i=1}^{n+1} \frac{1}{(i-1)!} P_i(z_f - z_c) Q_i(z_c - z_s) \tag{27}$$

Equations (5) and (6) were also used in the above transformations. These expansions are only carried out when the source point s is sufficiently far from the field points f or ℓ , as both the expedite and the fast multipole methods depend on a sufficient distance in order to arrive at a reasonable accuracy. Otherwise the integrals indicated in Equations (10) and (11) are to be evaluated as usually, which includes the correct consideration of the cases when the integrals become singular or improper.

7.3.1. Some numeric results for the FMM applied to the EBEM

The square domain on the left of Figure 11 is used to assess both accuracy and computational cost for the FMM applied to the EBEM. The domain is discretised with linear and quadratic elements, and two adjacency parameters ($AdjTOL$) are studied for each discretisation: 0.1 and 5. The parameter $AdjTOL = 0.1$ corresponds to the topological distance, for which computational time is very low at the expense of accuracy. The same logarithmic field of the first numeric example – for a source at point ($z_s = 12.5 + i15$) – is applied. The discretisations with linear and quadratic elements go up to $2^{18} = 262,144$ and $2^{17} = 131,072$ degrees of freedom, respectively.

For the simulations using linear elements, time and error results, evaluated as in the previous examples, are presented in Figures 15 and 16 using the rather topological adjacency search with $AdjTOL = 0.1$ as well as $AdjTOL = 5$. In all cases, the execution times for the CBEM are shown to be proportional to N^2 , while the fast multipole simulations present an execution time proportional to N . Independently of the $AdjTOL$ value, all fast multipole simulations perform faster than the CBEM, even for a small number of degrees of freedom.

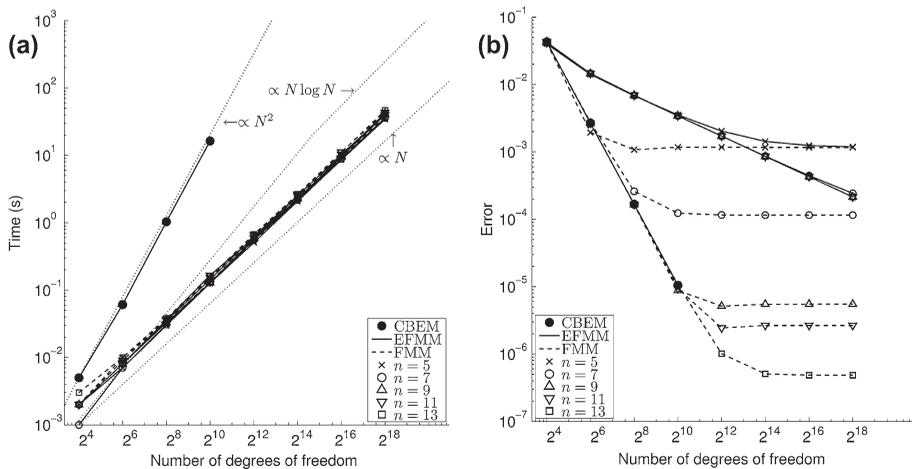


Figure 15. Execution time (left) and error measure (right) for the square domain of Figure 11 discretised with linear elements and with $AdjTOL = 0.1$: topological distance (Peixoto & Dumont, 2016b).

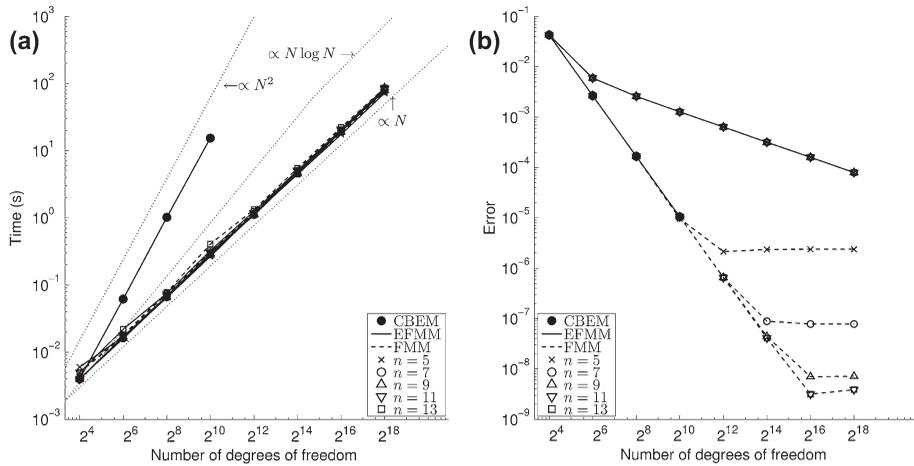


Figure 16. Execution time (left) and error measure (right) for the square domain of Figure 11 discretised with linear elements and with $AdjTOL = 5$ (Peixoto & Dumont, 2016b).

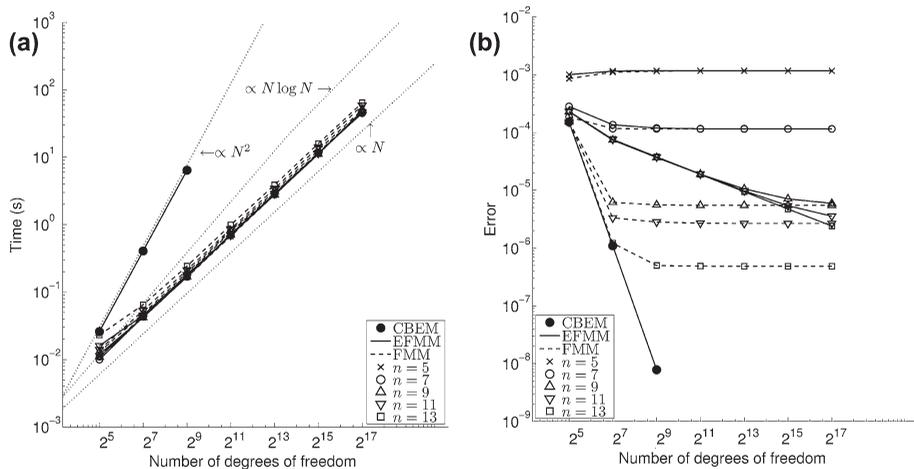


Figure 17. Execution time (left) and error measure (right) for the square domain of Figure 11 discretised with quadratic elements and with $AdjTOL = 0.1$: topological distance (Peixoto & Dumont, 2016b).

The execution times for the FM algorithm applied to the CBEM and to the EBEM are visually indistinguishable, but the EBEM runs always slightly faster. This is an expected result, as instead of evaluating polynomial integrations in a CBEM context, the EBEM pre-evaluates only polynomial interpolations, according to the array of results given in Equation (17).

Although the simulations with the EBEM run slightly faster, accuracy is definitely worse than with the CBEM, as shown on the right of Figures 15 and 16.

Results for a quadratic discretisation are shown in Figures 17 and 18 for $AdjTOL = 0.1$ and $AdjTOL = 5$. The same behaviour observed for the linear

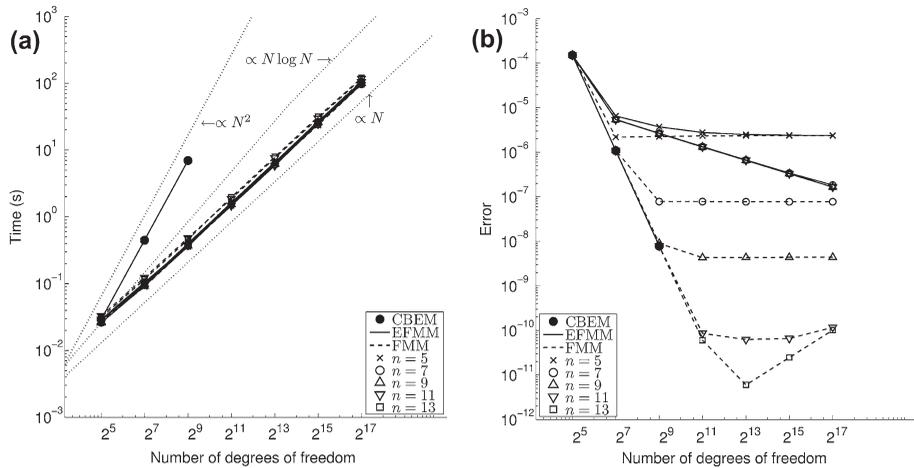


Figure 18. Execution time (left) and error measure (right) for the square domain of Figure 11 discretised with quadratic elements and with $AdjTOL = 5$ (Peixoto & Dumont, 2016b).

discretisation is seen in these results. The error norms are of a smaller order of magnitude, just as expected when comparing linear and quadratic elements.

8. Concluding remarks

This paper presents a novel, kernel-independent fast multiple formulation to be used with the BEM. 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 two irregularly shaped domains with up to $N = 16,777,216$ degrees of freedom.

The numerical assessments show that the proposed algorithm is seamlessly applicable to generally curved elements of any order. The simulation of extremely convoluted shapes including multiply connected domains seems to present no difficulties. The computational cost for all examples run so far has shown to be proportional to $O(N)$, 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 a very small number of degrees of freedom, as observed in the graphs on the left of Figures 9 and 10.

Although the proposed expedite implementation of the boundary element matrices seems to be competitive in a conventional formulation, as shown for some examples, its use in the frame of a fast multiple method does not seem to lead to a substantial gain in computational time and presents a bad convergence rate. As already mentioned, this method may be recommended as a fast means of obtaining a fair approximation of a complex problem in a reduced amount of

time, such as in the evaluation of initial results for iterative methods, as well as for initial mesh approximations in highly convoluted domains.

The simplified hybrid boundary element method, briefly outlined in this paper, is being presently implemented in the frame of the proposed fast multipole algorithm. As shown, this method makes use of the transpose of the double-layer potential matrix of the conventional method, which requires special care for a fast multipole implementation. The conclusions regarding an expedite evaluation of the matrices of the conventional method should apply to the simplified hybrid boundary element method as well.

The present fast-multipole algorithm is being implemented for elasticity problems, as well. Its generalisation to three-dimensional problems should not offer any conceptual difficulties, although the series expansions of the fundamental solution no longer relies on a complex-variable representation (see Y. Liu, 2009, for instance, and references in there). The authors are also working on the implementation of the proposed technique in the frame of an available iterative solver, which is of large practical interest but not at all in terms of research achievements.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the Brazilian agencies Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES); Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ); Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

ORCID

Ney Augusto Dumont  <http://orcid.org/0000-0003-4147-3130>

Hélvio de Farias Costa Peixoto  <http://orcid.org/0000-0003-4424-5272>

References

- Dumont, N. A. (1989). The hybrid boundary element method: An alliance between mechanical consistency and simplicity. *Applied Mechanics Reviews*, 42, S54–S63. doi:10.1115/1.3152408
- Dumont, N. A. (2010). The boundary element method revisited. In C. A. Brebbia (Ed.), *Boundary elements and other mesh reduction methods XXXII* (Vol. 50, pp. 227–238). Southampton: WIT Press. doi:10.2495/BE100201
- Dumont, N. A., & Aguilar, C. A. (2012). The best of two worlds: The expedite boundary element method. *Engineering Structures*, 43, 235–244. doi:10.1016/j.engstruct.2012.04.042
- Dumont, N. A., & Peixoto, H. C. F. (2016a). A fast-multipole unified technique for the analysis of potential problems with the boundary element methods. *Proceedings of the Indian National Science Academy*, 82, 289–299. doi:10.16943/ptinsa/2016/48420

- Dumont, N. A., & Peixoto, H. F. C. (2016b). A fast-multipole implementation of the simplified hybrid boundary element method. In M. Tezer-Sezgin, B. Karasözen, & M. H. Aliabadi (Eds.), *Advances in boundary element techniques XVII* (pp. 51–58). EC, Ltd., ISBN 978-0-9576731-3-7.
- Liu, Y. J. (2009). *Fast multipole boundary element method*. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511605345
- Liu, Y. J., Mukherjee, S., Nishimura, N., Schanz, M., Ye, W., Sutradhar, A., ... Saez, A. (2011). Recent advances and emerging applications of the boundary element method. *Applied Mechanics Reviews*, 64, 030802 (38 pp). doi:10.1115/1.4005491
- Liu, Y. J., & Nishimura, N. (2006). The fast multipole boundary element method for potential problems: A tutorial. *Engineering Analysis with Boundary Elements*, 30, 371–381. doi:10.1016/j.enganabound.2005.11.006
- Nishimura, N. (2002). Fast multipole accelerated boundary integral equation methods. *Applied Mechanics Reviews*, 55, 299–324. doi:10.1115/1.1482087
- Novelino, L. S. (2015). *A novel fast multipole technique in the boundary element methods* (M.Sc. thesis in Portuguese). PUC-Rio, Brazil.
- Peixoto, H. F. C. (2014). *A study of the fast multipole method applied to boundary element problems* (M.Sc. thesis in Portuguese). PUC-Rio, Brazil.
- Peixoto, H. F. C., & Dumont, N. A. (2016a). A kernel-independent fast multipole technique for the analysis of problems with the boundary element method. In T. P. Lima (Ed.), *Proceedings of the XII SIMMEC - Minas Gerais' Symposium on Computational Mechanics* (pp. 419–426). Brazil: Diamantina.
- Peixoto, H. C. P., & Dumont, N. A. (2016b). On the fast-multipole implementation of the simplified hybrid boundary element method. *Revista Interdisciplinar de Pesquisa em Engenharia - RIPE*, 2, 127–145. ISSN 2447-6102.
- Peixoto, H. F. C., Novelino, L. S., & Dumont, N. A. (2015a). A fast-multipole unified technique for the analysis of continuum mechanics problems with the boundary element methods. In N. A. Dumont (Ed.), *XXXVI Ibero-Latin American Congress on Computational Methods in Engineering* (16 pp). Brazil: Rio de Janeiro. doi:10.20906/CPS/CILAMCE2015-0688. Retrieved from <http://www.swge.inf.br/proceedings/CILAMCE2015/>
- Peixoto, H. F. C., Novelino, L. S., & Dumont, N. A. (2015b). Basics of a fast-multipole unified technique for the analysis of several classes of continuum mechanics problems with the boundary element method. In A. H.-D. Cheng & C. A. Brebbia (Eds.), *Boundary Elements and Other Mesh Reduction Methods XXXVIII* (pp. 47–59). Southampton: WITPress. doi:10.2495/BEM380041