REVISTA INTERDISCIPLINAR DE PESQUISA EM ENGENHARIA

CILAMCE 2016

XXXVII IBERIAN LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGINEERING
BRASÍLIA - DF - BRAZIL

On the fast-multipole implementation of the simplified hybrid boundary element method

Hélvio de Farias Costa Peixoto

Ney Augusto Dumont

hfcpeixoto@gmail.com

dumont@puc-rio.com.br

Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio

Rua Marquês de São Vicente, 225, Gávea, 22451-900, Rio de Janeiro, RJ, Brazil

Abstract. *The present paper is part of a research line to implement, test and apply a novel numerical tool that can simulate on a personal computer and in just a few minutes a problem of potential or elasticity with up to tens of millions of degrees of freedom. We have already developed our own version of the fast-multipole method (FMM), which relies on a consistent construction of the collocation boundary element method (BEM), so that ultimately only polynomial terms are required to be integrated – and in fact can be given as a table of pre-integrated values – for generally curved segments related to a given field expansion pole and no matter how complicated the problem topology and the underlying fundamental solution. The simplified hybrid BEM has a variational basis and in principle leads to a computationally less intensive analysis of large-scale 2D and 3D problems of potential and elasticity – particularly if implemented in an expedite version. One of the matrix-vector products of this formulation deals with an equilibrium transformation matrix that comes out to be the transpose of the double-layer potential matrix of the conventional BEM. This in principle requires a reverse strategy as compared to our first developed (and reverse) FMM. The effective application of these strategies to any high-order boundary element and any curved geometry needs to be adequately assessed for both numerical accuracy and computational effort. This is the subject of the present investigations, which are far from a closure. A few numerical examples are shown and some initial conclusions can already be drawn.*

Keywords: *Boundary elements, Hybrid boundary elements, Fast multipole method, Variational methods*

1 INTRODUCTION

The present research work is part of the studies carried out by 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 (Dumont & Peixoto, 2016; Peixoto & Dumont, 2016). 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 (Dongarra & Sullivan, 2000; Liu, 2009; Liu et al., 2012).

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 simplified hybrid BEM has a variational basis and leads to a computationally less intensive implementation, which may become relevant for large-scale 2D and 3D problems of potential and elasticity. One of the matrix-vector products of this formulation deals with an equilibrium transformation matrix that comes out to be the transpose of the double-layer potential matrix of the conventional BEM. This in principle requires a reverse strategy as compared to our first implemented FMM, which has already turned out to be a reverse strategy as compared to the implementations given in the literature. The FMM for the simplified hybrid BEM can be implemented in a further simplified version, called *expedite*, in such a way that only matrix-vector products directly related to nodal values of the fundamental solution needs to be obtained and is seamlessly applicable to any high-order boundary element and curved geometry, which may come out as of further computational advantage. On the other hand, our initially proposed implementation of the fast multipole, collocation boundary element method seems to be already extremely fast and accurate, which makes it questionable whether further improvements are feasible. This is the core of the present paper and, although the required implementations are far from definitive, the conclusions that can be drawn so far are of conceptual and practical relevance. A few numerical results are shown to support the conceptual developments.

The complex formulation of the collocation boundary element method is shown in the following Section as applicable to a potential problem. (The formulation for an elasticity problem is almost as straight forward and is the subject of an implementation in progress in the frame of the fast multipole method.) The equations of the hybrid simplified boundary element method are shown next and the further simplification related to an expedite evaluation of the integrals required in the boundary element methods is presented in a separate subsection. After this introduction of the BEMs we present the steps and algorithms related to the fast multipole method, so that a general discussion is made possible.

2 Complex formulation of the two-dimensional potential problem

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

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

valid in the open domain (that is, with global support) for a unit point source applied at z_s, where $z = x + iy$ stands for the complex representation of the Cartesian coordinate system (x, y) and k is a material property, such as the conductivity for a steady state heat propagation problem in a homogeneous and isotropic medium.

One checks that, in fact,

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

where $r = \sqrt{x^2 + y^2}$.

For a general potential θ, the flux components in the Cartesian coordinates are defined as

$$\begin{aligned} q_x &= -k\frac{\partial\theta}{\partial x} \equiv -\Re\left(k\frac{d\theta^C}{dz}\right) \\ q_y &= -k\frac{\partial\theta}{\partial y} \equiv -\Im\left(k\frac{d\theta^C}{dz}\right), \end{aligned} \tag{3}$$

where the superscript C stands for complex notation. Then,

$$q_x + iq_y \equiv -k\frac{d\theta^C}{dz}. \tag{4}$$

Using the complex expression of the boundary outward unit vector

$$\eta = \eta_x + i\,\eta_y \equiv \frac{1}{|J|}\left(\frac{dy}{d\xi} - i\frac{dx}{d\xi}\right) \equiv \frac{\tilde{\eta}}{|J|}, \tag{5}$$

the general expression of a boundary normal flux $q = -q_x\eta_x - q_y\eta_y$ leads to the complex representation of the boundary normal flux q_s^* due to the potential defined in Eq. (1) for a unit point source applied at z_s:

$$q_s^* = k\frac{\partial\theta_s^*}{\partial x}\eta_x + k\frac{\partial\theta_s^*}{\partial y}\eta_y \equiv \Re\left(k\frac{d\theta_s^C}{dz}\eta\right) \equiv \Re\left(q_s^C\right). \tag{6}$$

2.1 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

$$\mathbf{Hd} = \mathbf{Gq}, \tag{7}$$

for the vectors of boundary nodal potentials $\mathbf{d}$ and normal fluxes $\mathbf{q}$ applied as mixed boundary conditions. The double layer and single layer potential matrices $\mathbf{H}$ and $\mathbf{G}$ are defined as

$$\mathbf{H} \equiv H_{sf} = \Re\left(H_{sf}^{C}\right) \quad \text{and} \quad \mathbf{G} \equiv G_{s\ell} = \Re\left(G_{s\ell}^{C}\right), \tag{8}$$

with

$$H_{sf}^{C} = k\int_{\Gamma} \frac{d\theta_s^C(z(\xi)-z_s)}{dz(\xi)}\eta(\xi)\, u_f(\xi)\mathrm{d}\Gamma(\xi) = \frac{-1}{2\pi}\int_{\Gamma}\frac{1}{z-z_s}\eta\, u_f\mathrm{d}\Gamma \equiv \int_{\Gamma} q_s^C\, u_f\mathrm{d}\Gamma \tag{9}$$

$$G_{s\ell}^{C} = \int_{\Gamma}\theta_s^C(z(\xi)-z_s)t_\ell(\xi)\mathrm{d}\Gamma(\xi) = \frac{-1}{2\pi k}\int_{\Gamma}\ln(z-z_s)t_\ell\mathrm{d}\Gamma \equiv \int_{\Gamma}\theta_s^C t_\ell\mathrm{d}\Gamma. \tag{10}$$

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 Eq. (1) has global support, both potential and normal flux quantities θ and q are piecewise approximated along the boundary (with local support):

$$\theta = u_f(\xi)d_f \quad \text{and} \quad q = t_\ell(\xi)q_\ell, \quad \text{where} \quad t_\ell = u_\ell\frac{|J|_{\text{at }\ell}}{|J|}, \tag{11}$$

where $u_f(\xi)$ and $u_\ell(\xi)$ stand for the same type of real polynomial interpolation functions of a given order (constant, linear and quadratic quadratic functions are implemented in our code). 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_ℓ 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 Eq. (8) and subsequently play an important role in a consistent formulation. Let $o_e = 1, 2, 3, ...$ be the order of a generic boundary element, that is, linear, quadratic, cubic and so on, with $o_e + 1 = 2, 3, 4, ...$ nodes in an element. If a given problem is discretized 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 elements are in general discontinuous at their connecting nodes (maybe unwillingly, as the result of the imprecise geometry representation of a curved boundary), the left and right normals at these points should be considered explicitly 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 the differentiating subscript ℓ is used. 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 should not take place in a consistent formulation (Dumont, 2010).

2.2 Basic equations of the simplified hybrid boundary element method

The simplified hybrid boundary element method has been well explained by Oliveira (2009), for example. For the simplest case of a potential problem, it relies on the assumption that the potential θ and its gradients $\theta_{,j}$ 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,

$$\theta = (\theta_s^* + C_s)\, p_s^* + \theta^p, \quad \theta_{,j} = \theta_{s,j}^*\, p_s^* + \theta_{,j}^p, \tag{12}$$

where θ_s^* is a fundamental solution of the corresponding differential equation of the problem, as given in Eq. (1). 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 (Dumont, 2010). Assigning subscripts D and N to subvectors of nodal potentials $\mathbf{d}$ and equivalent nodal gradients $\mathbf{p}$ to characterize 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

$$\begin{bmatrix} \mathbf{H}_N^{\mathrm{T}} \\ \mathbf{H}_D^{\mathrm{T}} \end{bmatrix} \mathbf{p}^* = \begin{Bmatrix} \mathbf{p}_N - \mathbf{p}_N^p \\ \mathbf{p}_D - \mathbf{p}_D^p \end{Bmatrix}, \qquad \begin{bmatrix} \mathbf{U}_N^* \\ \mathbf{U}_D^* \end{bmatrix} \mathbf{p}^* = \begin{Bmatrix} \mathbf{d}_N - \mathbf{d}_N^p \\ \mathbf{d}_D - \mathbf{d}_D^p \end{Bmatrix}, \tag{13}$$

where the quantities with superscript p stand for nodal potentials or gradients belonging 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 and $\mathbf{U}^*$ represents the fundamental solutions θ_s^* evaluated at the boundary nodal points, that is, $\mathbf{U}^* \equiv \theta_s^*(z_f - z_s)$. 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 Eq. (7):

$$\mathbf{p} = \mathbf{L}^{\mathrm{T}}\mathbf{q}, \quad \text{with} \quad \mathbf{L} \equiv L_{\ell f} = \int_\Gamma u_f(\xi) t_\ell(\xi) \mathrm{d}\Gamma(\xi). \tag{14}$$

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

$$L_{\ell f} \equiv \tilde{L}_{\ell f} |J|_{(\mathrm{at}\,\ell)}, \tag{15}$$

where $|J|_{(\mathrm{at}\,\ell)}$ is defined as in Eq. (11) and $\tilde{L}_{\ell f}$ is a pre-evaluated result given as the block matrices below for linear, quadratic and cubic elements:

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

Equation (13) 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

remaining boundary potentials and gradients. Then, the equation to be implemented in the frame of the fast multipole method is

$$\begin{bmatrix} \mathbf{H}_N^{\mathrm{T}} \\ \mathbf{U}_D^* \end{bmatrix} \mathbf{p}^* = \begin{Bmatrix} \mathbf{p}_N - \mathbf{p}_N^p \\ \mathbf{d}_D - \mathbf{d}_D^p \end{Bmatrix}. \tag{17}$$

Results at internal points are obtained directly by using Eq. (12). Results close to or at nodal points can also be obtained (Dumont, 2010). The implemented 2D boundary element code works with linear, quadratic or cubic elements.

2.3 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 $\mathbf{H}^{\mathrm{T}}$ 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 by using the same interpolation functions introduced in Eq. (11). Equation (7) 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 Eqs. (9) and (10) and then approximates the fundamental solutions themselves along each boundary segment according to Eq. (11), thus resulting in a very simplified (thus the term $expedite$) form in terms of the transformation matrix $\mathbf{L}$ of Eq. (14):

$$\begin{aligned} H_{sf}^C d_f &= G_{s\ell}^C q_\ell \\ \left(\int_\Gamma q_s^C u_f \mathrm{d}\Gamma\right) d_f &= \left(\int_\Gamma \theta_s^C t_\ell \mathrm{d}\Gamma\right) q_\ell \\ T_{\ell s}^C \left(\int_\Gamma t_\ell u_f \mathrm{d}\Gamma\right) d_f &\approx \theta_{sf}^C \left(\int_\Gamma u_f t_\ell \mathrm{d}\Gamma\right) q_\ell \\ T_{\ell s}^C L_{\ell f} d_f &\approx U_{sf}^C L_{\ell f} q_\ell \\ &\text{or} \\ T_{\ell s}^C L_{\ell f} d_f &\approx U_{sf}^C p_f \end{aligned} \tag{18}$$

In this development, $T_{\ell s}^C$ expresses the evaluation of the complex fundamental solution q_s^C 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

$$H_{sf}^C \approx T_{\ell s}^C L_{\ell f} \quad \text{and} \quad G_{s\ell}^C \approx U_{sf}^C L_{\ell f} \tag{19}$$

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

3 A kernel-independent fast multipole method

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}, $k = 1, 2, ...n_c$: 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}, $l = 1, 2, ...n_L$: 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_0 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 Fig. 1, 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 initially expanded about the *close* pole $z_{c^{n_c}}$ (of highest level, as developed next) using n terms:

$$f(z - z_s) = \sum_{i=0}^{n} \frac{1}{i!}(z - z_{c^{n_c}})^i D^{(i)} f(z_{c^{n_c}} - z_s) + O(z - z_{c^{n_c}})^{n+1} \tag{20}$$

where $D^{(0)} f(z) = f(z)$ and $D^{(i)} f(z) = d^i f(z)/dz^i$.

The truncated form of Eq. (20) is conveniently written as

$$f(z - z_s) = \sum_{i=1}^{n+1} \frac{1}{(i-1)!} P_i(z - z_{c^{n_c}}) Q_i(z_{c^{n_c}} - z_s) \tag{21}$$

for truncation order $O(z - z_{c^{n_c}})^{n+1}$ and with the arrays of functions $P(z)$ and $Q(z)$ defined for a generic argument z as

$$P(z) = \left\{ \begin{array}{ccccccc} 1 & z & z^2 & z^3 & \cdots & z^{n+1} \end{array} \right\} \tag{22}$$

$$Q(z) = \left\{ \begin{array}{ccccccc} f(z) & \frac{df(z)}{dz} & \frac{df^2(z)}{dz^2} & \frac{df^3(z)}{dz^3} & \cdots & \frac{df^{n+1}(z)}{dz^{n+1}} \end{array} \right\}. \tag{23}$$

Expansions about the source point are also possible. Let the derivatives $D^{(i)} f(z_{c^{n_c}} - z_s)$ be also expanded for the source point z_s about the *local* point $z_{L^{n_L}}$ (of highest level, as to be also shown subsequently) using m terms:

$$D^{(i)} f(z_{c^{n_c}} - z_s) = \sum_{j=0}^{m} \frac{1}{j!}(z_{L^{n_L}} - z_s)^j D^{(i+j)} f(z_{c^{n_c}} - z_{L^{n_L}}) + O(z_{L^{n_L}} - z_s)^{m+1} \tag{24}$$

Substituting for $D^{(i)} f(z_{c^{n_c}} - z_s)$ in Eq. (20) according to above, it results

$$\begin{aligned} f(z - z_s) = & \sum_{i=0}^{n} \tfrac{1}{i!}(z - z_{c^{n_c}})^i \sum_{j=0}^{m} \tfrac{1}{j!}(z_{L^{n_L}} - z_s)^j D^{(i+j)} f(z_{c^{n_c}} - z_{L^{n_L}}) \\ + & \; O(z - z_{c^{n_c}})^{n+1} + O(z_{L^{n_L}} - z_s)^{m+1} \end{aligned} \tag{25}$$

The truncated form of Eq. (25) is conveniently written as

$$f(z - z_s) = \sum_{i=1}^{n+1} \frac{1}{(i-1)!} P_i(z - z_{c^{n_c}}) \sum_{j=1}^{m+1} \frac{1}{(j-1)!} P_j(z_{L^{n_L}} - z_s) Q_{i+j-1}(z_{c^{n_c}} - z_{L^{n_L}}) \tag{26}$$

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

3.1 Adjacency search

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

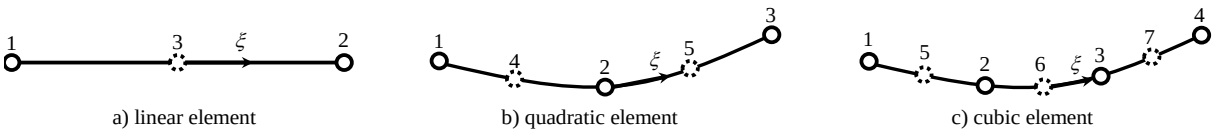

Figure 1: Schemes for splitting a general element into two sub-elements.

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.

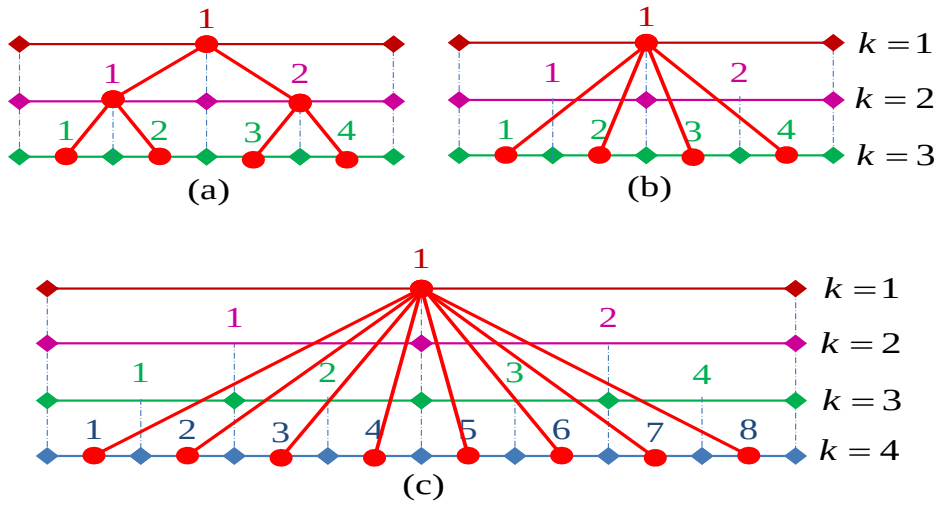

Figure 2: Schematic pole expansions using numbers of child poles $n_c = 2, 4$ or 8 (constant elements).

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 – 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 Fig. 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

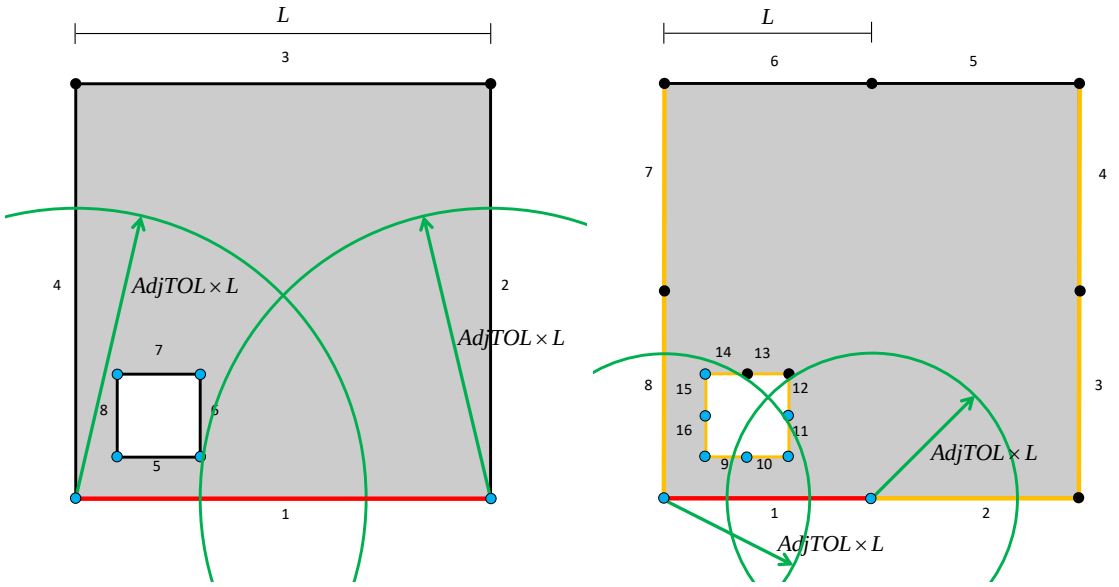

Figure 3: Schemes for the adjacency search at the coarser refinement level $k = 0$ (left) and at the next refinement level $k = 1$ (right). $AdjTOL \times L$ is the search radius. The red elements are the ones being assessed and the green circles represent the search radius. The yellow elements are the adjacent elements at the previous level (only these are considered for the search). The blue nodes are adjacent, and thus the elements they belong are adjacent as well.

elements at level $k + 1$. This information is used to reduce the number of possible adjacent nodes, and therefore, the number of distance evaluations.

As illustrated on the left of Fig. 3 for the level $k = 0$, a search is carried out for element 1 using the two green circles centred on its nodes, and it comes out that the nodes marked as blue (corners of the square hole) are adjacent. Then, any element that contains at least one of theses 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 $0.7 \leq AjdTOL \leq 2$.

The red element on the right of Fig. 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 $k = 1$, as shown in yellow. 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 build 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.

4 Some numerical results

4.1 Assessment of the expedite approximation of the conventional 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 Eq. 19.

Two different domains, as depicted in Fig. 4, are considered. The square domain on the left is discretized with either linear or quadratic elements with up to 1024 degrees of freedom. On the right of Fig. 4 is shown an irregularly shaped domain defined by 16 initial nodes and discretized with quadratic elements with up to 1024 degrees of freedom. Both domains are submitted to a logarithmic field due to a point source applied at $z_s = 12.5 + i15$ and nodal potentials $\mathbf{d}$ as well as normal fluxes $\mathbf{q}$ are evaluated along the boundary nodes to assess accuracy the accuracy of Eq. (12) by applying the Euclidean error norm

$$\epsilon = \frac{|\mathbf{Hd} - \mathbf{Gt}|}{|\mathbf{Gt}|}. \tag{27}$$

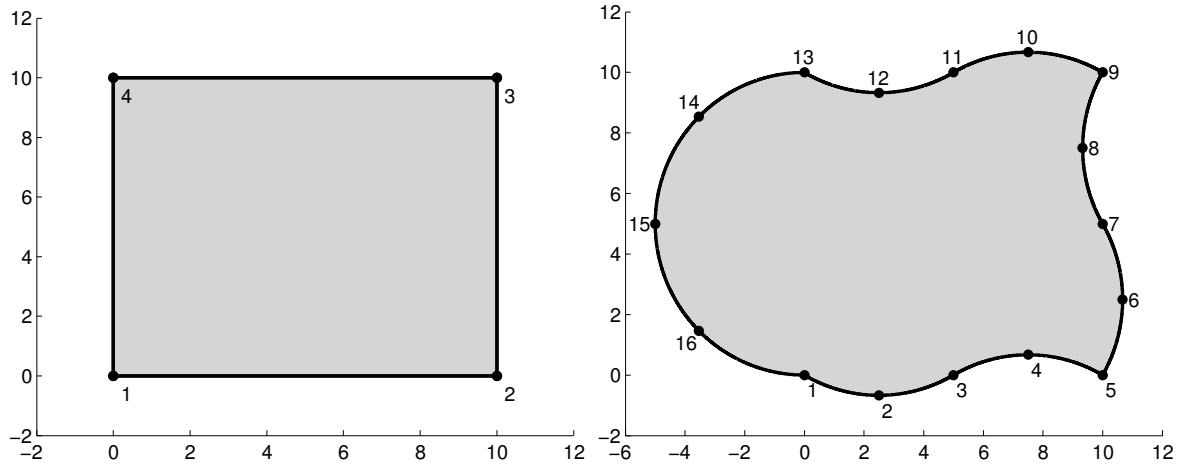

Figure 4: Regular square and deformed quadrilateral domains used for the numerical assessments of Sections 4.1 and 4.1.

Results for the square domain

The times required to run the simulations for the square domain (left of Fig. 4) with linear and quadratic discretizations are shown on the left and right graphics of Fig. 5. Four $AdjTOL$ parameters, as presented in Section 3.1, are studied: 10^{-4}, 3, 6, and 10. The value $AdjTOL = 10^{-4}$ leads to the same results of a topological adjacency.

In both sets of simulations it may be seen that the CBEM takes by far more time to run than the ones carried out with 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 discretizations with $AdjTOL = 10$. The results of Fig. 5 show that the computational costs with the EBEM in general increase at a by far lower rate then in the case of the CBEM.

Euclidean error norms, evaluated as in Eq. 27 for either case of linear or quadratic element, are shown in Fig. 6 for several mesh refinements. As expected, the EBEM simulations lead

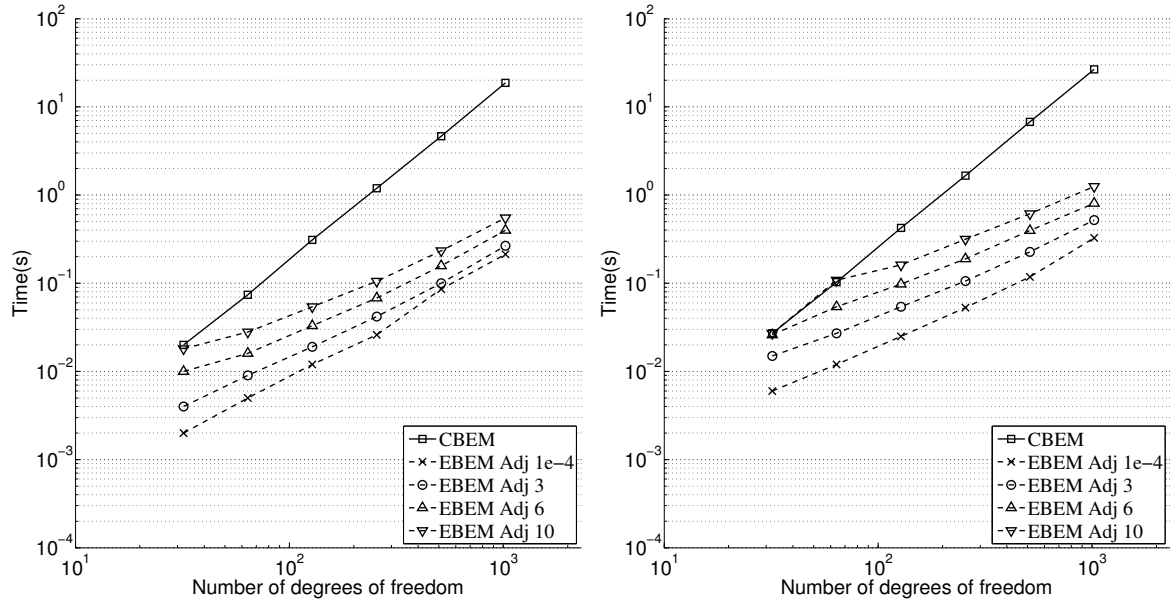

Figure 5: Execution times for the linear (left) and quadratic (right) discretization of the square domain in the left of Fig. 4.

to larger errors when compared to the 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 larger then with the CBEM whether using linear or quadratic elements.

If one weighs computational effort and accuracy convergence in Figs. 5 and 6 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.

Results for the curved domain

In a second assessment, the curved domain on the right of Fig. 4 is discretized with quadratic elements in order to keep its original shape. Figure 7 presents both the execution times (left) and the Euclidean error norms (right) for a numerical analysis carried out with up to 1024 nodes. The same four vales 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 Fig. 7. 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 $\epsilon = 6 \times 10^{-7}$ after about 1.2s execution time. To achieve such a precision with the CBEM, it is needed to run a simulation with 230 DOFs, which takes about the same 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 time, which may prove to be a useful means of evaluating initial results for iterative methods, as well as for initial mesh approximations in highly convoluted domains.

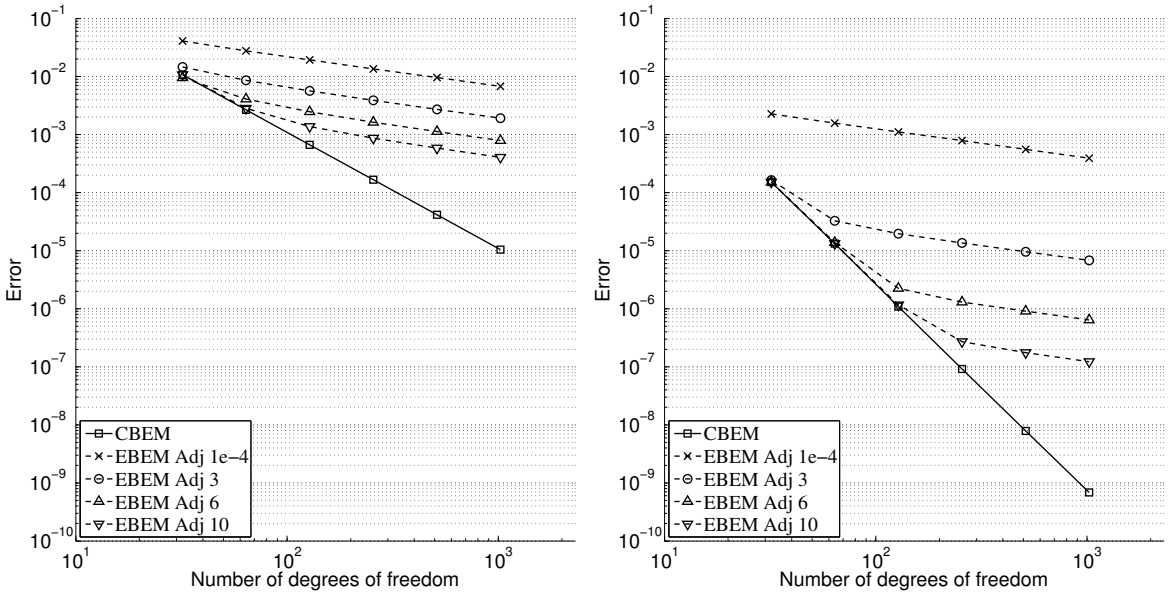

Figure 6: Euclidean error norms for the linear (left) and quadratic (right) discretizations of the square domain on the left of Fig. 4.

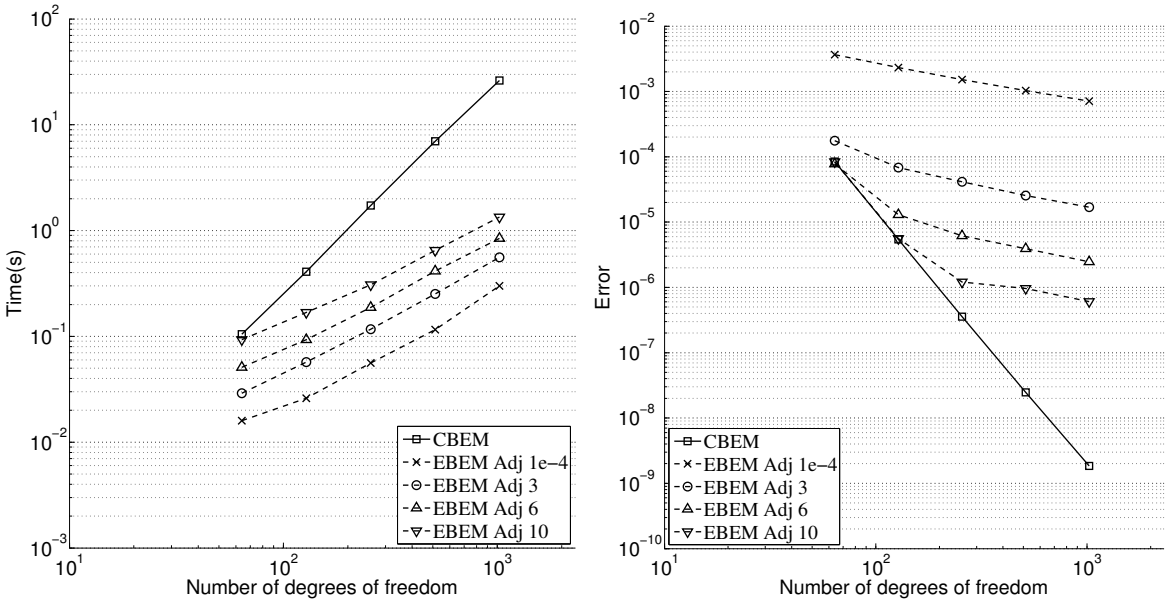

Figure 7: Assessment results for the curved domain in the right of Fig. 4. Execution time (left) and Euclidian error norms for the quadratic discretization (right)

4.2 Basic, target results for the implemented fast multipole method

A multiply connected domain with a very irregular boundary is shown in Fig. 8 to illustrate the excellent results we can achieve with the proposed fast multipole algorithm implemented in the frame of the conventional, collocation boundary element method (Dumont, 2010; Dumont & Peixoto, 2016). The boundary is discretized with quadratic elements up to $5 \times 2^{18} = 1,310,720$ degrees of freedom as a cut-out test for the open domain submitted to a logarithmic field given by $\ln|z - z_{s_1}| + \ln|z - z_{s_2}|$, where $z = x + iy$ is a domain point and $z_{s_1} = 12.5 + 14i$ and

$z_{s_2} = 20 + 10i$ are source points, represented by $(*)$ in the figure. The execution time and error results are given in Fig. 9 for several levels of mesh refinement. The solid circles refer to the evaluation of the matrix-vector products of Eq. (7) using the conventional, collocation boundary element method without the fast multipole algorithm. We see on the left that the computational time is proportional to N^2, where N is the number of degrees of freedom. The graphics on the right present the Euclidean error norm of Eq. (27) to assess the accuracy of results. Blue squares mark the time and accuracy results for the simulation when $N = 5 \times 2^{10}$. On the other hand, several simulations are shown for the same problem run in the frame of the implemented fast multipole method with several numbers n of the series expansion given in Eq. (20). The results for $n = 17$ terms are marked with red circles to show that, for the same level of accuracy of the conventional boundary element method, as given on the right, only approximately $1/1000$ of computational time was required. As a matter of fact, a dash line proportional to $NlogN$ and a dot line proportional to N are also drawn on the left graphic to show that the computational costs – in this example as well as in several other numerical tests – grow rather proportional to N.

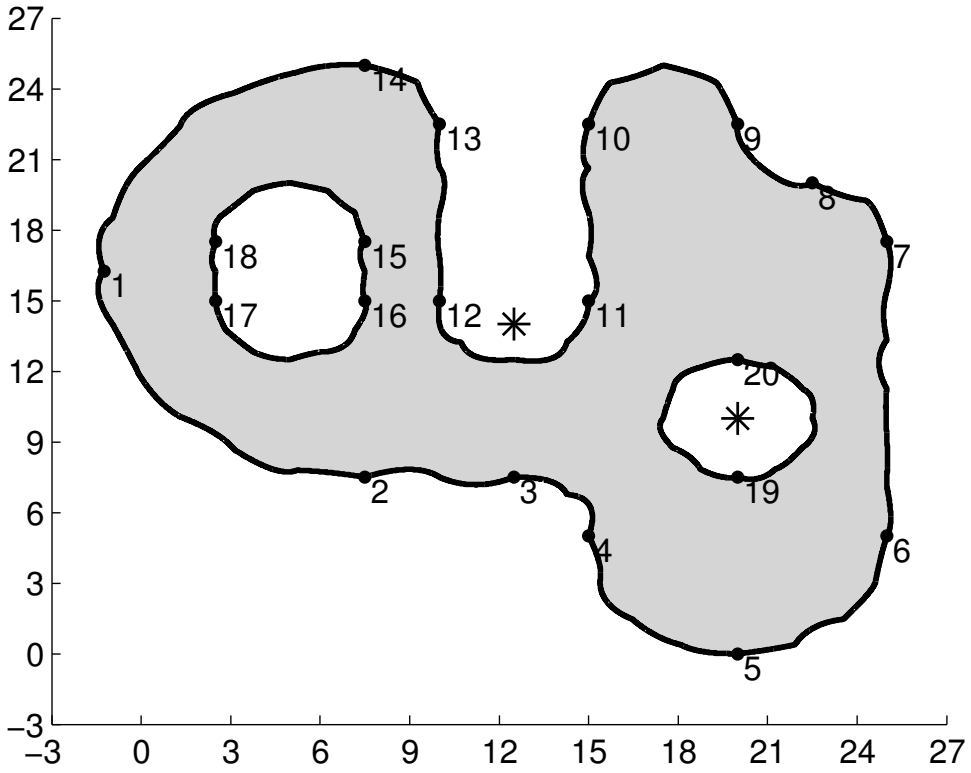

Figure 8: Curved domain, with a random noise on its boundary generation, discretized with quadratic elements and submitted to a logarithmic field with two sources (asterisks).

4.3 Assessment of the expedite fast multipole boundary element method

The Fast Multipole technique briefly presented in Section 3 – but thoroughly discussed in Peixoto (2014), Novelino (2015), and Dumont & Peixoto (2016) – 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

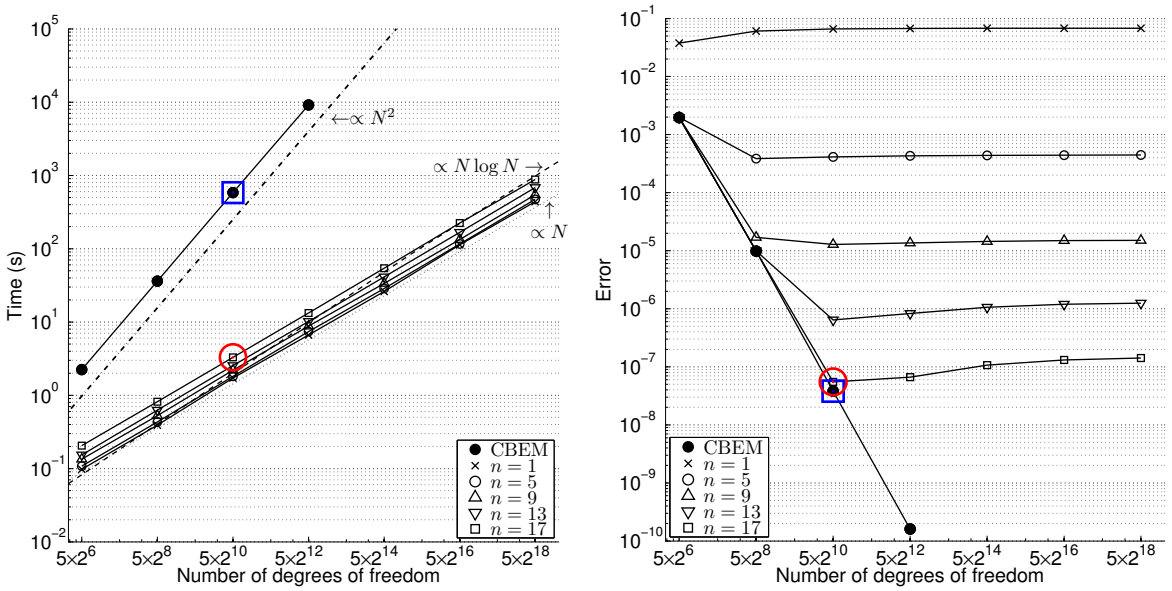

Figure 9: Execution times for the linear (left) and quadratic (right) discretization of the square domain in the left of Fig. 4.

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 Eq. (1) as in Eq. (26), when applied to the matrices of Eq. (19), leads to the fast multipole expansions for the boundary element method when the expedite approximation is also used:

$$\begin{aligned} H^C_{sf} &\approx T^C_{\ell s} L_{\ell f} = q^C_s \eta\big|_{(\text{at}\ell)} L_{\ell f} = q^C_s \tilde{\eta}\big|_{(\text{at}\ell)} \tilde{L}_{\ell f} \\ &\approx \tilde{\eta}|_{(\text{at}\ell)} \tilde{L}_{\ell f} \sum_{i=2}^{n+2} \frac{1}{(i-1)!} P_{i-1}(z_\ell - z_c) Q_i(z_c - z_s) \end{aligned} \tag{28}$$

$$\begin{aligned} G^C_{s\ell} &\approx U^C_{sf} L_{\ell f} = U^C_{sf} \tilde{L}_{\ell f} |J|_{(\text{at}\,\ell)} \\ &\approx \tilde{L}_{\ell f} |J|_{(\text{at}\,\ell)} \sum_{i=1}^{n+1} \frac{1}{(i-1)!} P_i(z_f - z_c) Q_i(z_c - z_s). \end{aligned} \tag{29}$$

Equations (4) and (5) have also been 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 Eqs. (9) and (10) are to be evaluated as usually, which includes the correct consideration of the cases when the integrals become singular or improper.

Some numeric results for the FMM applied to the EBEM

The square domain on the left of Fig. 4 is used to assess both accuracy and computational cost for the FMM applied to the EBEM (FMEBEM). The domain is discretized with linear and quadratic elements, and two adjacency parameters ($AdjTOL$) are studied for each discretization: 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 Section 4.1 for a source at point $(z_s = 12.5 + i15)$ is applied. The discretizations with linear and quadratic elements goes up to 262144 and 137072 degrees of freedom, respectively.

For the simulation using linear elements, time and error results, evaluated as in the previous examples, are presented in Figs. 10 and 11 for the rather topological adjacency search with $(AdjTOL = 0.1)$ as well as for $(AdjTOL = 5)$. In both 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 then the CBEM, even for a small number of DOFs.

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 table of results given in Eq. (16).

Although the simulations with the EBEM run slightly faster, accuracy is definitely worse then the with the CBEM, as it is shown on the right of Figs. 10 and 11.

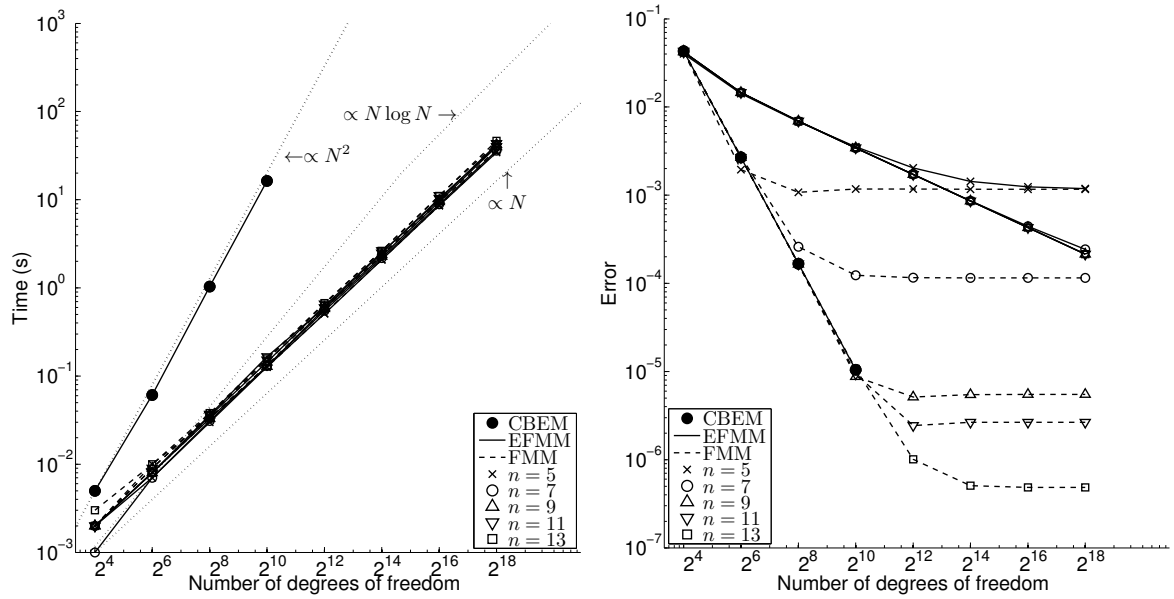

Figure 10: Execution time (left) and error measure (right) for the square domain in the left of Fig. 4 discretized with linear elements and with $AdjTOL = 0.1$ (topological distance).

Results for a quadratic discretization are shown in Figs. 12 and 13 for $(AdjTOL = 0.1)$ and $(AdjTOL = 5)$. The same behaviour observed for the linear discretization is seen in these results. The error norms are of a smaller order of magnitude, just as expected when comparing linear and quadratic elements.

5 CONCLUDING REMARKS

This paper assesses the main issues of an attempt to further improve the fast multipole method already proposed by the authors by implementing an expedite version of the way the matrices of the boundary element method are evaluated. Several basic concepts of a consistent

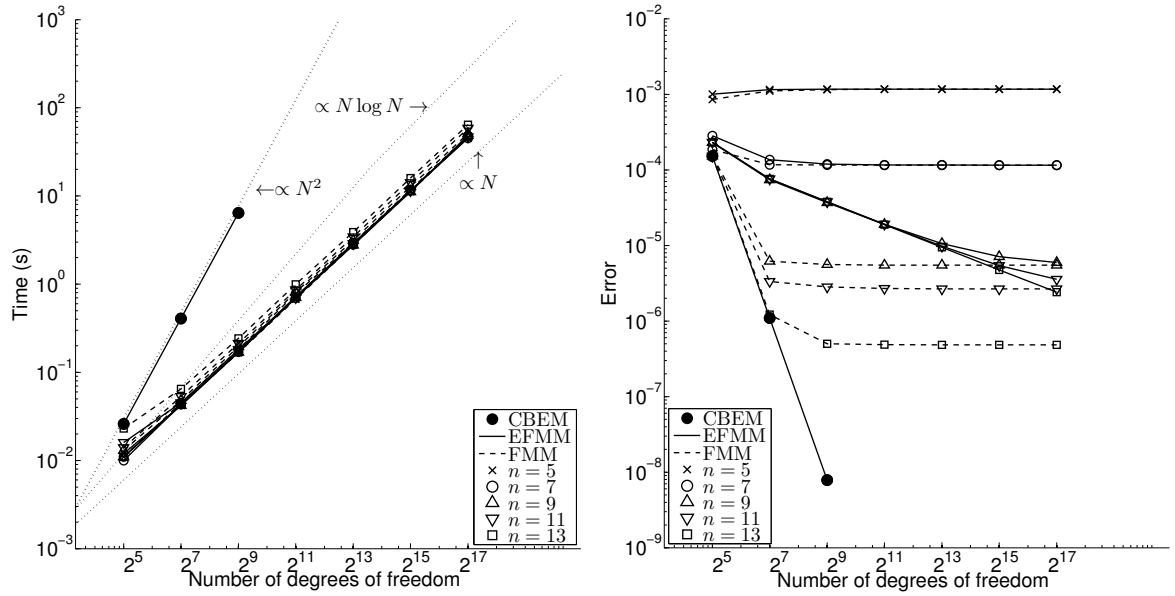

Figure 11: Execution time (left) and error measure (right) for the square domain in the left of Fig. 4 discretized with linear elements and with $AdjTOL = 5$.

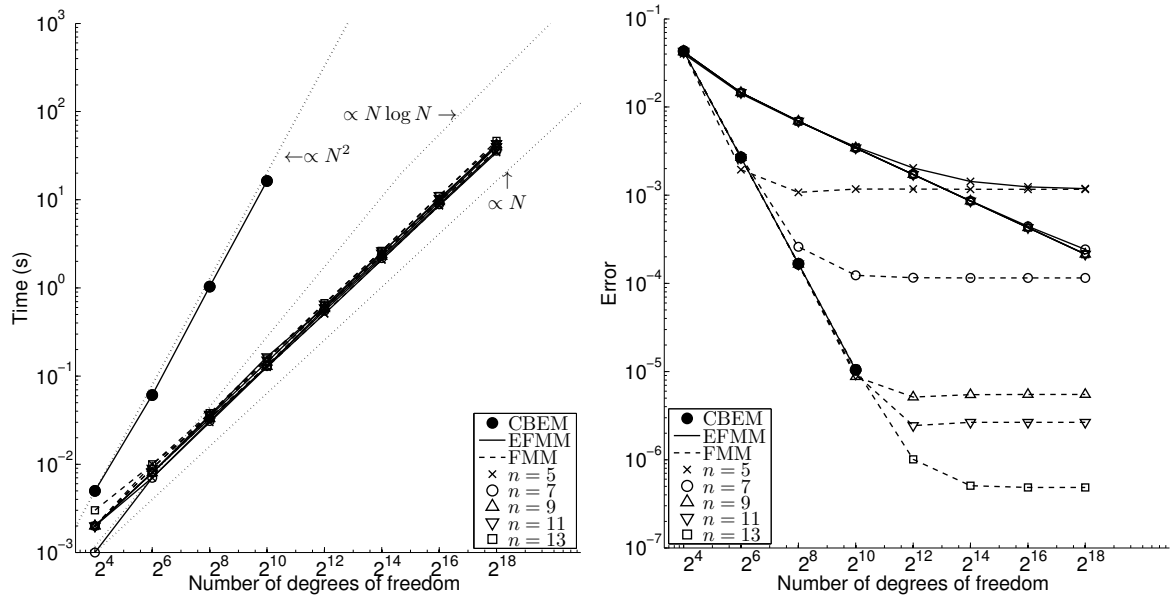

Figure 12: Execution time (left) and error measure (right) for the square domain in the left of Fig. 4 discretized with quadratic elements and with $AdjTOL = 0.1$ (topological distance).

implementation of the boundary element method are reviewed and the basic features of the proposed fast multipole algorithm are discussed particularly as concerning an efficient element adjacency search.

The fast multipole implementation in the frame of a consistent version of the conventional, collocation boundary element method leads to high accuracy of results while keeping the computational cost extremely low, as presented in a first group of results. Although the proposed expedite implementation of the boundary element matrices seems to be competitive in a con-

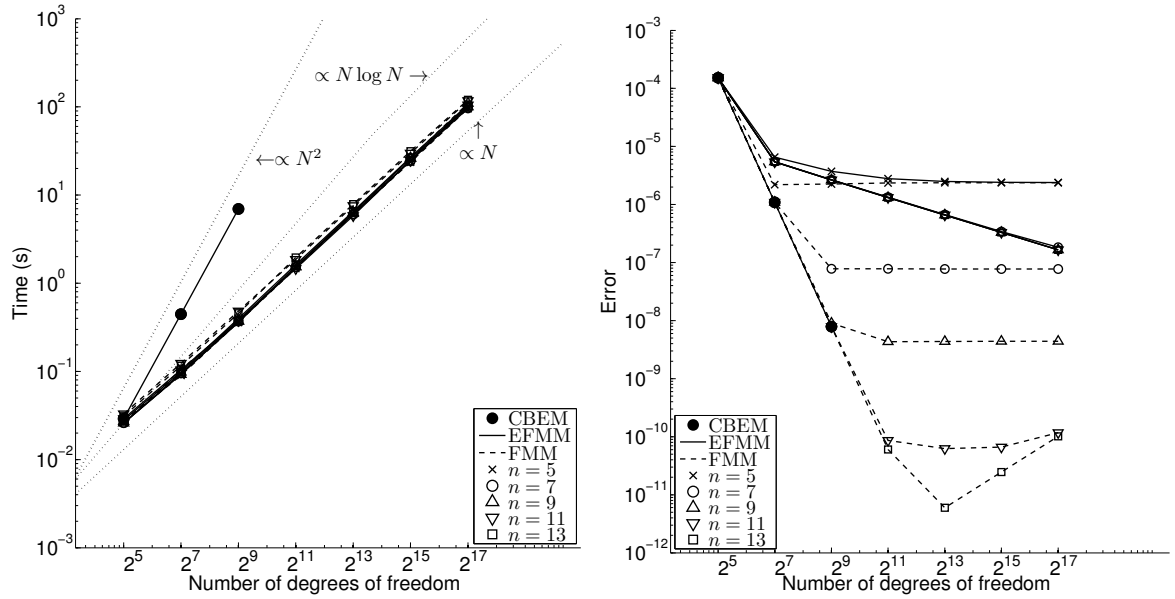

Figure 13: Execution time (left) and error measure (right) for the square domain in the left of Fig. 4 discretized with quadratic elements and with $AdjTOL = 5$.

ventional formulation, as shown for some examples, its use in the frame of a fast multipole 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 Section 2.2, 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. This is being presently developed. The conclusions regarding an expedite evaluation of the matrices of the conventional method should apply to the simplified hybrid boundary element method as well.

ACKNOWLEDGEMENTS

This section should be positioned between the end of the text and the reference list. Type **Acknowledgements** in boldface italics, skip one line of space and type the text in regular type.

REFERENCES

Dongarra, J. & Sullivan, F., 2000. Guest editors introduction to the top 10 algorithms. *Computing in Science & Engineering*, 2(1):22–23.

Dumont, N. A., 2010. The boundary element method revisited. In *Boundary Elements and Other Mesh Reduction Methods XXXII*, volume 50, pages 227–238. WIT Press.

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

Dumont, N. A. & Peixoto, H. F. C., 2016. A fast-multipole unified technique for the analysis of potential problems with the boundary element methods. *Proceedings of the Indian National Science Academy*, 82(2):289–299.

Liu, Y., 2009. *Fast Multipole Boundary Element Method*. Cambridge University Press, Cambridge.

Liu, Y. J., Mukherjee, S., Nishimura, N., Schanz, M., Ye, W., Sutradhar, A., Pan, E., Dumont, N. A., Frangi, A., & Saez, A., 2012. Recent advances and emerging applications of the boundary element method. *Applied Mechanics Reviews*, 64(3):030802.

Novelino, L. S., 2015. *A novel fast multipole technique in the boundary element methods*. Master's thesis (in portuguese), PUC-Rio.

Oliveira, M. F. F., 2009. *Conventional and simplified-hybrid boundary element methods applied to axisymmetric elasticity problems in fullspace and halfspace*. Phd thesis (in portuguese), PUC-Rio.

Peixoto, H. D. F. C. & Dumont, N. A., 2016. A kernel-independent fast multipole technique for the analysis of problems with the boundary element method. In Lima, T. P., editor, *Anais do XII Simpósio de Mecânica Computacional*, volume 1, pages 419–426, Diamantina.

Peixoto, H. F. C., 2014. *A Study of the Fast Multipole Method Applied to Boundary Element Problems*. Master's thesis (in portuguese), PUC-Rio.