ORIGINAL ARTICLE

lajss.org
Latin American Journal of Solids and Structures

www.lajss.org

Two accelerated isogeometric boundary element method formulations: fast multipole method and hierarchical matrices method

Emerson Bastos[a]*, Éder Lima de Albuquerque[b], Lucas Silveira Campos[c], Luiz Carlos Wrobel[d]

[a] Universidade Federal dos Vales do Jequitinhonha e Mucuri, Unaí, Minas Gerais, Brasil. Email: emersonbastos@ufvjm.edu.br
[b] Universidade de Brasília, Faculdade de Tecnologia, Departamento de Engenharia Mecânica, Brasília, Distrito Federal, Brasil. Email: eder@unb.br
[c] Universidade Federal do Espírito Santo, Departamento de Engenharia Mecância, Vitória, Espírito Santo, Brasil. Email: lucas.s.campos@ufes.br
[d] Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil. Email: luiz.wrobel@puc-rio.br

* Corresponding author

https://doi.org/10.1590/1679-78257244

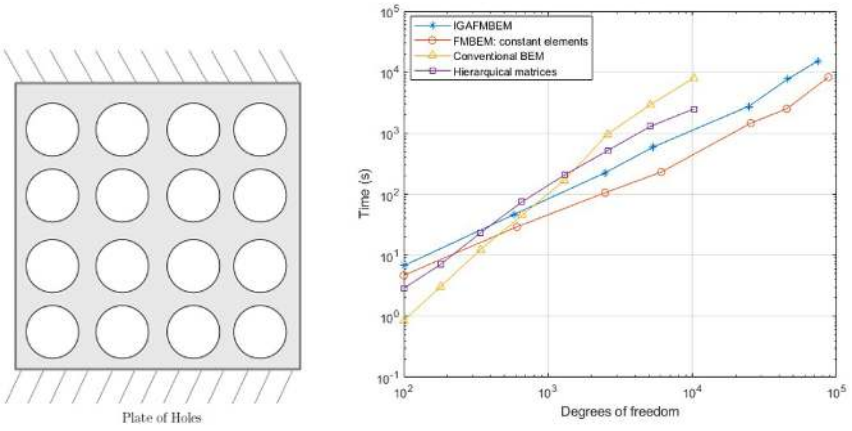
Abstract

This work presents two fast isogeometric formulations of the Boundary Element Method (BEM) applied to heat conduction problems, one accelerated by Fast Multipole Method (FMM) and other by Hierarchical Matrices. The Fast Multipole Method uses complex variables and expansion of fundamental solutions into Laurant series, while the Hierarchical Matrices are created by low rank CUR approximations from the k−Means clustering technique for geometric sampling. Both use Non-Uniform Rational B-Splines (NURBS) as shape functions. To reduce computational cost and facilitate implementation, NURBS are decomposed into Bézier curves, making the isogeometric formulation very similar to the conventional BEM. A description of the hierarchical structure of the data and the implemented algorithms are presented. Validation is performed by comparing the results of the proposed formulations with those of the conventional BEM formulation. The computational cost of both formulations is analyzed showing the advantages of the proposed formulations for large scale problems.

Keywords

Boundary element method, Isogeometric analysis, Fast multipole method, Hierarquical Matrices

Graphical Abstract



Received: August 22, 2022. In revised form: September 23, 2022. Accepted: September 26, 2022. Available online: September 27, 2022.
https://doi.org/10.1590/1679-78257244

Latin American Journal of Solids and Structures. ISSN 1679-7825. Copyright © 2022. This is an Open Access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 INTRODUCTION

In general, numerical methods, such as Finite Element Method (FEM) and BEM, are based on the transformation of partial differential equations into integral equations and on the discretization of these integral equations through the creation of elements. There are some characteristics that are common to the numerical methods, among which we can mention the loss of precision of the results with the use of less refined meshes and the increase in processing time with the use of more refined meshes. These mentioned characteristics are partly due to the approximation of the geometry and field variables by the Lagrange polynomials. Lagrange polynomials are not able to accurately represent the geometry of most of the boundary of continuum mechanics problem domains such as circles, ellipses and hyperboles. Furthermore, there is no continuity of the derivatives of functions approximated by Lagrange polynomials between an element and its neighbors. With a view to solving problems like these, the idea was to use the same basis functions used in the Computer Aided Design (CAD) packages, called NURBS, to describe the geometry and to approximate the field variables. Thus arises the isogeometric analysis, which is widely discussed in the literature (Beer et al., 2019; Peigl and Tiller, 1996; Rogers, 2000; Hughes et al., 2005; Cottrell et al., 2009; Kagan and Fisher, 2000). With this new form of analysis, a segment in the parametric space corresponds to an isogeometric element located in the geometric space. Therefore, there is an elimination of the mesh generation step. In turn, the refinement is achieved without much additional effort (Li and Qian, 2011; Shene, 2011; Abramowitz and Stegun, 1972).

It is noteworthy that isogeometric analysis is more adaptable to BEM than to FEM specially for linear problems, as both the CAD system and BEM share surface definitions and do not need the domain discretization for linear problems. Even for some nonlinear problems, as contact mechanics, for example, the BEM doesn't require domain discretization. In the work of Loyola et al. (2022), an isogeometric boundary element formulation was successfully applied to contact mechanics problems. Another difficulty with the joining of CAD and FEM is the decrease in the sparsity of their matrices due to the high continuity of the NURBS basis functions (Collier et al., 2012). Note that the matrices of BEM are already full and this is not an extra problem. Thus, the isogeometric formulation of BEM brought encouraging results in terms of accuracy and efficiency. Some benefits can be observed in the literature (Beer et al., 2019), such as smoother geometries easily obtained, non-linear problems are solved without additional effort, among others.

Despite the known dimension reduction due to the fact that only the boundary is discretized, BEM has disadvantages due to the fact that its matrices are full and not symmetrical, imposing a high cost in the assembly and resolution of the linear system of equations. Several techniques were developed in order to face these problems, such as the Adaptive Cross Approximation (ACA) technique, which approximates the dense matrices of the BEM by a representation through hierarchical matrices. This representation is based on a binary tree structure that partitions the full matrix into smaller blocks, where each block will either be approximated by a low-rank matrix or the original block will be exactly used in the resolution of the linear system (Hackbusch, 2016). Other techniques were also developed, such as wavelets (Bucher et al., 2002), block methods (Rigby and Aliabadi, 1995; Crotty, 1982; Kane et al., 1990), agglutination processes and iterative techniques (Mansur et al., 1992; Barra et al., 1992).

Another well-known technique is the Fast Multipole Method (FMM) which has its roots in gravitational calculus of particle simulation models (Rokhlin, 1985; Greengard and Rokhlin, 1987). FMM improves the performance of the BEM due to the fact that the kernel of the fundamental solution can be expanded in series, which allows the separation of the relationship between the source point and the field point by inserting an intermediate point. There is also a decrease in the number of interactions due to the clustering of boundary elements in cells of a tree structure (Barnes and Hut, 1986). Due to its performance wherever it is applied, FMM is considered one of the top ten algorithms of the 20th century (Dongarra and Sullivan, 2000; Liu, 2009). There are other FMM methods developed for purely numerical kernels or for kernel-independent methods where there is no expansion of analytical kernels. Instead, they use an interpolation method like, for example, Chebyshev. These methods also make use of a matrix compression methods like Singular Value Decomposition (SVD) or Fast Fourier Transform (FFT) (Ying et al., 2004; Fong and Darve, 2009).

Techniques such as hierarchical matrices and FMM make use of iterative methods to solve the linear system of equations because in these techniques matrices of BEM cannot be calculated explicitly in order to save memory and allow large-scale simulations (Greenbaum, 1997). Some well-known iterative methods are the Gauss-Jacobi, Gauss-Seidel, minimum residue (MINRES), conjugate gradient, bi-stabilized conjugate gradient (BICGSTAB) and Generalized Minimal Residual Method (GMRES). These methods adapt very well to low rank approximation techniques such as those mentioned at the beginning of this paragraph.

Regarding the accelerated isogeometric formulation by the FMM, there are some related works. For example, in Matsumoto and Takahashi (2012), the FMM is coupled to an isogeometric formulation of the BEM, where B-splines of degree 2 and 3 are used explicitly as basis functions, applied to an infinite domain problem with Neumann boundary conditions and smooth boundary. In Wang et al. (2019), an element-free method (meshless method) coupled with isogeometric analysis and accelerated by the FMM is applied to potential problems. In Simpson and Liu (2016), a FMM coupled to Isogeometric Boundary Element Method (IGABEM) is presented for 3D problems, which uses Chebyshev

interpolation and the M2L operators are approximated by SVD. In the literature, no other article was found where the FMM, with an analytical kernel, was used together with the Bézier extraction operator for the isogeometric analysis in BEM.

For the accelerated formulation by the hierarchical matrices method, some related works are: Ozdemir and Lee (2004) presents an algorithm IE-QR that builds an approximation QR of low rank using the modified Gram-Schmidt algorithm with cost $O(N^{3/2})$, where $N = \max(m,n)$ with m and n being the admissible matrix block dimensions. In Kapur and Long (1998), an algorithm IES3 is presented, which consists of a kernel-independent method for electromagnetic simulations and costs $O(N \cdot \log N)$. An interpolative decomposition method based on QR rank-revealing factorization and costs $O(m \cdot n \cdot k)$ is known, where k is the rank of the matrix block (see (Boutsidis et al., 2009; Gu and Eisenstat, 1996; Voronin and Martinsson, 2017)). In Campos et al. (2017), a fast isogeometric formulation of the BEM is adapted to the hierarchical matrices method using ACA for low rank approximation. In Ayala et al. (2020), an approximation method CUR is presented for matrices corresponding to the admissible blocks, through a geometric sampling technique. To the best of the author's knowledge, there is no other article in the literature where the hierarchical matrix method, coupled with IGABEM with Bézier extraction, was subjected to a low-rank matrix compression method using the k-Means clustering technique for geometric sampling.

This paper presents two fast isogeometric formulations of the BEM, one using the FMM (Fast Multipole Accelerated Isogeometric BEM (IGAFMBEM)) and the other using hierarchical matrices with specific matrix compression. Both formulations use NURBS as shape functions and the Bézier decomposition method which, in turn, brings speed in the generation of NURBS and simplicity in the computational implementation. The boundary conditions are generic and are treated as Cabral et al. (1990 and 1991) and Campos et al. (2017).

2 ISOGEOMETRIC FORMULATION

The parametric curves of the isogeometric analysis are related to each other as can be seen in Figure 1. All of them have a common core, which are the Bézier curves, and from then on they specialize according to the properties of each one.

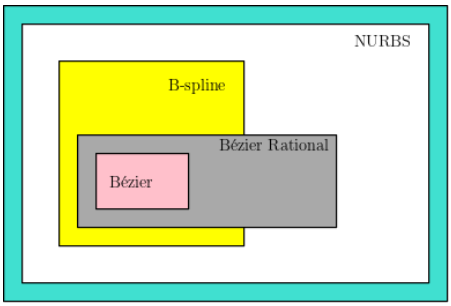


Figure 1: Curves.

2.1 Bézier curves

The i–th Bernstein polynomial base function, $B_{i,p}$, of degree p is defined by the expression:

$$B_{i,p}(\xi) = C(p,i)\xi^i(1-\xi)^{p-i}, \tag{1}$$

with $C(p,i) = \dfrac{i!}{p!(p-i)!}$.

A Bézier curve of degree p can be written as the linear combination of $p+1$ Bernstein polynomial basis functions, dependent on the real parameter ξ, with $0 \le \xi \le 1$, like:

$$\beta(\xi) = \sum_{i=1}^{p+1} \mathrm{P}_i B_{i,p}(\xi), \tag{2}$$

where P_i are control points that form its control polygon.

2.2 NURBS

B-spline curves are a generalized form of Bézier curves. It is composed of one or more Bézier curves or segments with a commitment to continuity between the curves. Each control point influences only some segments of the B-spline curve and thus has local control of the curve. A base of univariate B-spline functions is defined from a set of knots or parametric domain U, which is a set of non-descending parametric coordinates written as $\mathrm{U}=\left\{\xi_0,\xi_1,\xi_2,\ldots,\xi_{n+p}\right\}$ where $\xi_i\in\mathbb{R}$ is the i−th parametric knot, p is the polynomial degree of the B-splines basis functions and n is the number of basis functions. The B-splines base functions of degree p, $N_{i,p}$, are defined recursively on U through the Cox-de Boor recursive formula of significant computational cost (Peigl and Tiller, 1996).

A B-spline curve of degree p in $\mathbb{R}^2$ is defined by

$$\gamma\left(\xi\right)=\sum_{i=0}^{n}\mathrm{P}_i N_{i,p}\left(\xi\right), \tag{3}$$

where P_i are the control points.

A NURBS curve is also defined from the set of knot U and a set of control points P_i, of the form

$$\delta\left(\xi\right)=\frac{\sum_{i=0}^{n}\mathrm{P}_i N_{i,p}\left(\xi\right)w_i}{\sum_{i=0}^{n}N_{i,p}\left(\xi\right)w_i}=\sum_{i=0}^{n}\mathrm{P}_i R_{i,p}\left(\xi\right), \tag{4}$$

where $R_{i,p}$ are the NURBS basis functions and are defined as

$$R_{i,p}\left(\xi\right)=\frac{w_i N_{i,p}\left(\xi\right)}{W\left(\xi\right)}, \tag{5}$$

and

$$W\left(\xi\right)=\sum_{i=0}^{n}N_{i,p}\left(\xi\right)w_i \tag{6}$$

is the weight function with w_i being the weight corresponding to the i-th B-spline base function, $N_{i,p}$, or associated with the i-th control point.

2.3 Bézier decomposition

New knots can be inserted in the set of knots, U, without changing the parametric or geometric properties of the curve, provided that for each knot inserted a new control point is added. Thus, inserting a new knot $\overline{\xi}\in\left[\xi_k,\xi_{k+1}\right]$, with $k>p$, in the knot set, requires that $n+2$ new B-spline basis functions are defined with the new set of knot $\mathrm{U}=\left\{\xi_0,\ldots,\xi_k,\overline{\xi},\xi_{k+1},\ldots,\xi_{n+p}\right\}$. So that there is no change in the continuity of the curve, these $n+2$ new control points, $\left\{\overline{\mathrm{P}}_i\right\}_{i=1}^{n+2}$, are formed from the original control points and are given by:

$$\overline{\mathrm{P}}_i=\begin{cases}\mathrm{P}_1, & \text{if} \quad i=1\\ \alpha_i\mathrm{P}_i+(1-\alpha_i)\mathrm{P}_{i-1}, & \text{if} \quad 1<i<n+2,\\ \mathrm{P}_n, & \text{if} \quad i=n+2\end{cases} \tag{7}$$

where

$$\alpha_i = \begin{cases} 1, & \text{if} \quad 1 \le i \le k-p \\ \dfrac{\overline{\xi} - \xi_i}{\xi_{i+p} - \xi_i}, & \text{if} \quad k-p+1 \le i \le k \\ 0, & \text{if} \quad i \ge k+1 \end{cases}. \tag{8}$$

(see (Peigl and Tiller, 1996; Borden et al., 2011)).

The Bézier decomposition is obtained by inserting repeated knot together with all knots inside the knot set, U, until they have a multiplicity equal to the degree of the curve. After the insertions, the NURBS curve is decomposed into a set of Bézier curves, where each curve corresponds to a knot span of U. Considering a curve with n control points and calling $\overline{\xi}_j$ the r we need to perform a decomposition, we can define α_j according to Eq. (8). One can then write a matrix that relates the new control points to the old ones:

$$\mathrm{C}^j = \begin{bmatrix} \alpha_1 & 1-\alpha_2 & 0 & 0 & \cdots & 0 \\ 0 & \alpha_2 & 1-\alpha_3 & 0 & \cdots & 0 \\ 0 & 0 & \alpha_3 & 1-\alpha_4 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & 0 & \alpha_{(n+j-1)} & 1-\alpha_{(n+j)} \end{bmatrix}. \tag{9}$$

Writing $\overline{\mathrm{P}}^1 = \mathrm{P}$, we can rewrite Eq. (7) in matrix form in order to represent the sequence of control points created by the h −refinement,

$$\overline{\mathrm{P}}^{j+1} = \left(\mathrm{C}^j\right)^{\mathrm{T}} \overline{\mathrm{P}}^j. \tag{10}$$

By repeating this operation r times, the final form of the decomposition is obtained:

$$\mathrm{P}^b = \mathrm{C}^{\mathrm{T}}\mathrm{P}, \tag{11}$$

where $\mathrm{C}^{\mathrm{T}} = \left(\mathrm{C}^r\right)^{\mathrm{T}} \left(\mathrm{C}^{r-1}\right)^{\mathrm{T}} \left(\mathrm{C}^{r-2}\right)^{\mathrm{T}} \cdots \left(\mathrm{C}^1\right)^{\mathrm{T}}$, P is the original set of control points and P^b is the final set, which can be called the Bézier control points. Remember that P has dimension $n \times d$, C is $n \times (n+r)$ and P^b is $(n+r) \times d$, where $d = 2$ in the applications of this article.

Let $\mathrm{B}(\xi) = \left\{B_i(\xi)\right\}_{i=1}^{n+r}$ be the set of Bernstein basis functions defined by the final set of knots. As the insertion of extra knots does not cause any geometric or parametric changes, the Bézier curve described by the new control points must coincide with the initial B-spline curve. So, from Eq. we have that:

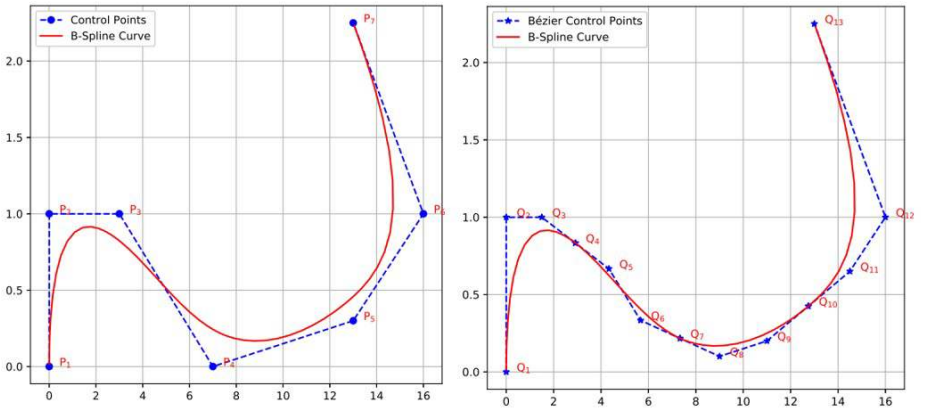
$$\gamma(\xi) = \mathrm{P}^{\mathrm{T}}\mathrm{N}(\xi) = \left(\mathrm{P}^{\mathrm{b}}\right)^{\mathrm{T}} \mathrm{B}(\xi) = \left(\mathrm{C}^{\mathrm{T}}\mathrm{P}\right)^{\mathrm{T}} \mathrm{B}(\xi) = \mathrm{P}^{\mathrm{T}}\mathrm{C}\mathrm{B}(\xi). \tag{12}$$

Since P is arbitrary:

$$\mathrm{N}(\xi) = \mathrm{C}\,\mathrm{B}(\xi), \tag{13}$$

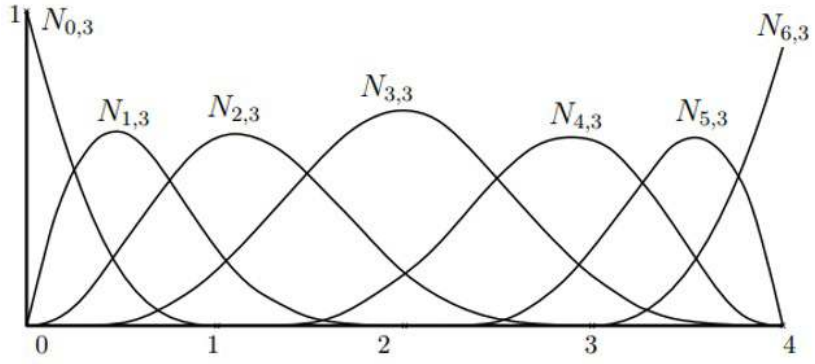
where C, called the Bézier extraction operator, relates the B-spline basis functions with the Bézier basis functions and, in this way, the NURBS curve can be generated directly from the Bézier base.

Figure 2a shows a cubic B-spline curve with the set of knots $\mathrm{U}=\{0,0,0,0,1,2,3,4,4,4,4\}$. Its decomposition is given in Figure 2b with the new set of parametric knots $\mathrm{U}=\{0,0,0,0,1,1,1,2,2,2,3,3,3,4,4,4,4\}$, and the Bezier control points $\mathrm{Q}_1, \mathrm{Q}_2, \ldots, \mathrm{Q}_{13}$. In Figure 3a we have the B-spline basis functions of the curve in Figure 2a, which are transformed into Bézier basis functions by parts given in Figure 3b. The procedure described in Eq. (7) and Eq. (8) ensures continuity for the curve of at least C^2 at breakpoints 1, 2 and 3. The numbers in Figure 3b denote the numbering scheme of the Bézier basis functions.

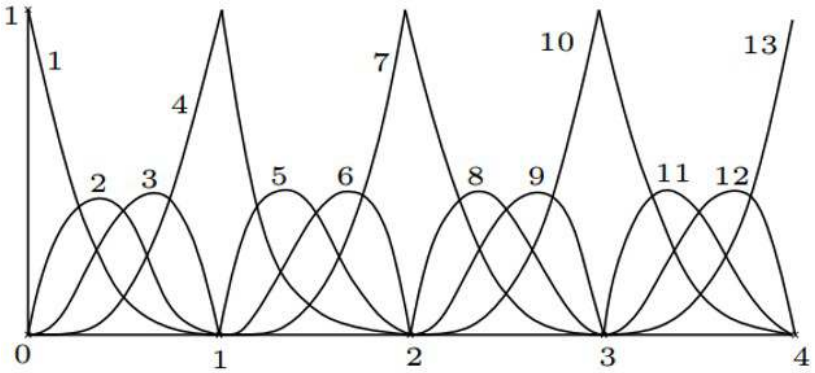


(a) Cubic B-spline curve and its control points. (b) Curve given in (a) decomposed into Bézier Curves.

Figure 2: Bézier decomposition.



(a) B-spline basis functions referring to the curve in Figure 2a.



(b) Bézier basis functions referring to the curve in Figure 2b.

Figure 3: Basis functions.

3 BOUNDARY INTEGRAL EQUATIONS

The proposed formulations are in line with the concept of the collocational boundary element method, but uses NURBS as shape functions for both geometry and numerical analysis. Let $\Omega \in \mathbb{R}^2$ be a bounded domain with the boundary $\Gamma := \partial\Omega$, which may contain non-smoothness points. We propose to solve the following mixed boundary value problem for the Laplace equation:

$$\begin{aligned} \nabla^2 u(x,y) &= 0, \quad \text{for } (x,y) \in \Omega \\ u &= \overline{u} \quad \text{in } \Gamma_1 \\ \frac{\partial u}{\partial n} &= \overline{u}_n \quad \text{in } \Gamma_2 \end{aligned}, \tag{14}$$

where n represents the outer normal to Γ, $\Gamma = \Gamma_1 \bigcup \Gamma_2$ and u is a function that satisfies Laplace's equation in the domain Ω. After an algebraic reduction process, this problem is solved using the following Boundary Integral Equation (BIE):

$$c(x)u(x) = \int_\Gamma \frac{\partial u(y)}{\partial n} u^*(x,y) d\Gamma_y - \int_\Gamma u(y) \frac{\partial u^*(x,y)}{\partial n} d\Gamma_y , \tag{15}$$

where $c(x)$ is a jump term that arises from the integration process of the integral equation and depends on the geometry at the source point $x \in \Gamma$. The second integral on the right side of Eq. (15) has its value given according to Cauchy's definition of principal value. Furthermore, $u(y)$ and $\frac{\partial u(y)}{\partial n}$ are the nodal values of the potential and the derivative of the potential, while u^* and $\frac{\partial u^*}{\partial n}$ refer to the fundamental solutions of the potential and the potential derivative, respectively, calculated on the boundary Γ.

For discretization purposes, the continuous fields u and $\frac{\partial u}{\partial n}$ are written for each Bézier element, whose boundary conditions are interpolated by the corresponding collocation points. In the isogeometric method, the variables are approximated using the same geometry functions, in this case, the NURBS and from Eq. (4) we can rewrite Eq. (15) in the form:

$$c(x)u(x) = \sum_{e_k} \left(\sum_j \int_{\Gamma_j} u^* \sum_{i=1}^{n_{\Gamma_j}} \left(\frac{\partial u_i^c}{\partial n} R_{i,p}(t) \right) d\Gamma - \sum_j \int_{\Gamma_j} \sum_{i=1}^{n_{\Gamma_j}} \left(u_i^c R_{i,p}(t) \right) \frac{\partial u^*}{\partial n} d\Gamma \right), \tag{16}$$

where u_i^c and $\frac{\partial u_i^c}{\partial n}$ are nodal values of potential and its corresponding derivative at the i–th control point, $\bigcup_k e_k = \Gamma$ and $\bigcup_j \Gamma_j = e_k$. It is worth noting that in Eq. (16) there is no approximation in geometry, but only in the variables u and $\frac{\partial u}{\partial n}$. In turn, with the boundary being parameterized by t, we can rewrite Eq. (16) as:

$$c(x)u(x) = \sum_{e_k}\left(\sum_{j}\left(\sum_{i=1}^{n_{\Gamma_j}-1}\left(\frac{\partial u_i^c}{\partial n}\int_{\xi_j}^{\xi_{j+1}} u^* R_{i,p}(t)\frac{d\Gamma}{dt}dt\right)\right)\right) - \sum_{e_k}\left(\sum_{j}\left(\sum_{i=1}^{n_{\Gamma_j}-1}\left(u_i^c\int_{\xi_j}^{\xi_{j+1}} R_{i,p}(t)\frac{\partial u^*}{\partial n}\frac{d\Gamma}{dt}dt\right)\right)\right), \tag{17}$$

where the transformation to a standard domain was performed, hence $\frac{d\Gamma}{dt}$ is the Jacobian of the transformation from physical coordinate space to parametric space.

It is still necessary one more change of variable in Eq. (17), in order to standardize the integration intervals for using the Gauss-Legendre quadrature, obtaining:

$$\sum_{e_k}\left(\sum_{j}\left(\sum_{i=1}^{n_{\Gamma_j}}\left(u_i^c\left(\int_{-1}^{1} R_{i,p}(\xi)\frac{\partial u^*}{\partial n}\frac{d\Gamma_y}{dt}\frac{dt}{d\xi}d\xi + c(x)R_{i,p}(t)\right)\right)\right)\right) = \sum_{e_k}\left(\sum_{j}\left(\sum_{i=1}^{n_{\Gamma_j}}\left(\frac{\partial u_i^c}{\partial n}\int_{-1}^{1} u^* R_{i,p}(\xi)\frac{d\Gamma_y}{dt}\frac{dt}{d\xi}d\xi\right)\right)\right) \tag{18}$$

where $\frac{dt}{d\xi}$ is the Jacobian of the transformation from parametric space to local coordinate. The Bézier decomposition operation together with the Bézier extraction operator promotes a structure of boundary elements very similar to the elements in the conventional BEM. Each knot span in the parametric space corresponds to an independent isogeometric element of the adjacent isogeometric elements in the physical coordinate space, which correspond to the adjacent parametric knot spans to the initial one.

3.1 Numerical calculation of integrals

A very important part of the proposed formulations here, which is responsible for the convergence of the iterative process, refers to the accurate calculation of the integrals in Eq. (15) when the source point is located in the element over the which integration takes place. The first integral on the right-hand side has a weak singularity kernel, because of the fundamental solution of the potential, while the second integral has a strong singularity kernel, because of the fundamental solution of the potential derivative, making both integrals singular. It is also important to consider the occurrence of quasi-singular integrals, where the source point is very close to the element where the integration.

To treat singular, quasi-singular, and regular integrals, a transformation based on the hyperbolic sine function was used, which takes into account the position of the orthogonal projection of the source point on the element in which the integration occurs, as well as the distance from the source point to the element (Johnston and Elliott, 2004). The hyperbolic sine transformation has the ability to distinguish and satisfactorily handle each case as will be seen in the numerical results. For regular integrals, the transformation coincides with the Gauss-Legendre quadrature.

3.2 Definition of collocation points

The most appropriate way to distribute the collocation points is through the definition of Greville abscissas, because it is on which the control points have maximum influence (Simpson et al., 2012; Farin, 1996; Scott et al., 2013). They are defined as the average of p parametric coordinates:

$$\lambda_i = \frac{\xi_{i+1} + \xi_{i+2} + \cdots + \xi_{i+p}}{p}, \tag{19}$$

where p is the degree of NURBS and they are in quantity equal to the control points.

In the case of problems with boundary presenting corners, the Greville abscissas as defined in Eq. (19) have the inconvenience of locating collocation points at these corners. One way to avoid this inappropriate occurrence is to change the positions of the first and last abscissa as follows:

$$\begin{aligned} \lambda_1 &= \lambda_1 + s \cdot \left(\lambda_2 - \lambda_1 \right) \\ \lambda_n &= \lambda_n - s \cdot \left(\lambda_n - \lambda_{n-1} \right) \end{aligned}, \tag{20}$$

where s is a coefficient that defines the displacement of the two extreme abscissas and, consequently, of the two corresponding collocation points. Here the value used was $s = 0.5$, as recommended by the results presented by Wang and Benson (2015).

Isogeometric formulations have an important particularity, which is the fact that the boundary integral equation given in Eq. (15) is written in terms of the control points. In Figure 2a it can be seen, for an example, that the control points may not be on the boundary of the problem and, therefore, they do not constitute a suitable place for the installation of the collocation points.

3.3 Treatment of boundary conditions

The variation of the potential and its derivative can also be represented by NURBS, thus gaining smoothness between adjacent isogeometric elements. For this, a concept analogous to that of control points for the representation of geometry is used for the representation of the field variable (Cabral et al., 1990; Campos et al., 1997).

Since the values of the boundary conditions are given at the collocation points that are on the boundary of the problem, then this conditions cannot be directly enforced on Eq. (15) still. To solve this impasse, we apply a transformation matrix, $\mathbf{E}$, constituted by the NURBS shape functions to relate the values of the boundary conditions at the collocation points with the values at the control points:

$$\begin{aligned} u &= \mathrm{E}\, u_c \\ \frac{\partial u}{\partial n} &= \mathrm{E}\, \frac{\partial u_c}{\partial n} \end{aligned}, \tag{21}$$

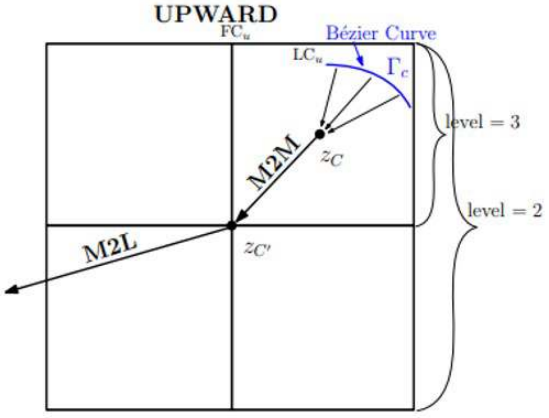
where the subindex indicates the values at the control points. By inverting the systems of equations given in Eq. (21), we can now solve Eq. (15). The solution is then obtained at control points and later through the system given in Eq. (21), the solution is obtained at the boundary. This procedure is carried out in each patch NURBS that composes the geometric boundary of the problem, each one with its type of boundary condition.

4 APPLICATION OF FMM TO ISOGEOMETRIC BEM

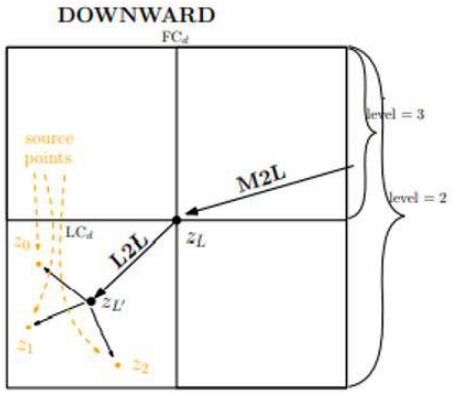
The Bézier decomposition operation and the Bézier extraction operator provided a simplification such that the FMM coupled to the isogeometric BEM practically does not differ from the FMM coupled to the conventional BEM using traditional elements (Liu, 2009).

The method starts with the construction of the quaternary tree from a square covering the entire boundary of the problem and from then on, each square is subdivided into four subsquares until each square has a number of Bézier elements, Γ_{C}, limited by a prefixed value and each subdivision corresponds to a level change, starting from zero where the square covers the entire original boundary. Each square is known as a cell and a leaf cell is a square that will not be further subdivided. For an illustrative example, Figure 4 shows the transition of operations from level 3 to level 2 and vice versa. A Γ_{C} element will belong to a cell if its center is inside that cell. In Figure 4a we have the *upward* process, step where the expansions of the multipole moments of the element Γ_{C} around the center of the cell of the lowest level are calculated, which is transferred by translation M2M to the center of the cell level 2. The *downward* process starts with the M2L translation transferring the moments accumulated in the center of the cell from the level 2, $z_{\mathrm{C}'}$, to the local expansion point z_{L}, from there to $z_{\mathrm{L}'}$ by the L2L translation and finally the contributions of the element Γ_{C}, from

a distant cell, will constitute the actions of the source points z_0, z_1, and z_2, see Figure 4b. The acronyms $\mathrm{FC_u}$ and $\mathrm{LC_u}$ (u meaning *upward*) stand for parent cell and child cell, respectively.



(a) *upward* process.



(b) *downward* process.

Figure 4: Validity condition: $\left|z_i - z_{\mathrm{L}}\right| << \left|z_{\mathrm{L}} - z_{\mathrm{C}}\right|$ and $\left|z - z_{\mathrm{C}}\right| << \left|z_{\mathrm{C}} - z_i\right|$ for $i = 0,1,2,\ldots,r$.

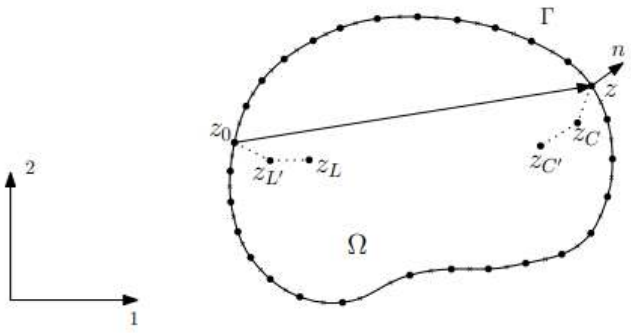


Figure 5: Expansion points.

In order to perform the matrix-vector product from Eq. (15), basically two expansions are performed, each one referring to the integrals on its right side. To define the algebraic scenario, the following notations and associations must be established: source point $x = \left(x_1, x_2\right) \in \mathbb{R}^2$ with $z_0 = x_1 + ix_2 \in \mathbb{C}$ and field point $y = \left(y_1, y_2\right) \in \mathbb{R}^2$ with $z = y_1 + iy_2 \in \mathbb{C}$, where i is the imaginary unit, see Figure 5. With the properties of complex variable differential and integral calculus in mind one can write:

$$u^*(x,y) = \mathrm{Re}\left\{u^*(z_0,z)\right\}, \tag{22}$$

where

$$u^*(z_0,z) = -\frac{1}{2\pi}\ln(z_0 - z) \tag{23}$$

is the fundamental solution of the potential in complex notation and $\mathrm{Re}\{+\}$ represents the real part of the argument $+$. As well as:

$$\frac{\partial u^*(x,y)}{\partial n} = \mathrm{Re}\left\{\frac{\partial u^*(z_0,z)}{\partial n}\right\}, \tag{24}$$

where

$$\frac{\partial u^*(z_0,z)}{\partial n} = (n_1 + in_2)\frac{\partial u^*(z_0,z)}{\partial z} \tag{25}$$

is the fundamental solution of the potential derivative also in complex notation and $n = n_1 + in_2$ is the outer normal vector at point z of Γ.

Therefore, the following equivalences are obtained:

$$\int_{\Gamma_C} u^*(x,y)\frac{\partial u(y)}{\partial n}d\Gamma_y \equiv \mathrm{Re}\left\{\int_{\Gamma_C} u^*(z_0,z)\frac{\partial u(z)}{\partial n}dS(z)\right\} \tag{26}$$

and

$$\int_{\Gamma_C} u(y)\frac{\partial u^*(x,y)}{\partial n}d\Gamma_y \equiv \mathrm{Re}\left\{\int_{\Gamma_C} u(z)\frac{\partial u^*(z_0,z)}{\partial n}dS(z)\right\} \tag{27}$$

where $\frac{\partial u(z)}{\partial n}$ and $u(z)$ come from the interpolation of the values of the boundary conditions corresponding to the element Γ_C with the NURBS basis functions from Eq. (4), as well as z_0 and z are also given by the same equation.

One of the fundamental ideas that speed up the matrix-vector product is the separation of the relationship between the source point z_0 and the field point z. For this purpose, the kernel function $u^*(z_0,z)$ given in Eq. (24) is expanded. Introducing an expansion point z_C close to the point z such that the validity condition, $|z - z_C| << |z_0 - z_C|$, is obeyed, see Figure 5, the following equation is obtained after an algebraic effort:

$$u^*(z_0,z) = -\frac{1}{2\pi}\ln(z_0 - z) = -\frac{1}{2\pi}\left[\ln(z_0 - z) + \ln\left(1 - \frac{z - z_C}{z_0 - z_C}\right)\right]. \tag{28}$$

Applying Taylor series to the second logarithm on the right side of Eq. (28), we arrive at:

$$u^*\left(z_0,z\right)=\frac{1}{2\pi}\sum_{k=0}^{\infty}O_k\left(z_0-z_{\mathrm{C}}\right)I_k\left(z-z_{\mathrm{C}}\right), \tag{29}$$

where the auxiliary functions $I_k\left(z\right)$ and $O_k\left(z\right)$ are defined as:

$$\begin{aligned} I_k\left(z\right) &= \frac{z^k}{k!}, \quad \text{for } k\geq 0 \\ O_k\left(z\right) &= \frac{\left(k-1\right)!}{z^k}, \quad \text{for } k\geq 1 \quad \text{and} \quad O_0\left(z\right)=-\ln\left(z\right) \end{aligned}. \tag{30}$$

4.1 Multipole moments

The integral in Eq. (26) can now be calculated by the following expansion in *multipole moments*:

$$\int_{\Gamma_{\mathrm{C}}}u^*\left(z_0,z\right)\frac{\partial u\left(z\right)}{\partial n}dS\left(z\right)=\frac{1}{2\pi}\sum_{k=0}^{\infty}O_k\left(z_0-z_{\mathrm{C}}\right)M_k\left(z_{\mathrm{C}}\right), \tag{31}$$

on what

$$M_k\left(z_{\mathrm{C}}\right)=\int_{\Gamma_{\mathrm{C}}}I_k\left(z-z_{\mathrm{C}}\right)\frac{\partial u\left(z\right)}{\partial n}dS\left(z\right) \tag{32}$$

are called moments of the Γ_{C} element around the z_{C} pole and independent of z_0.

Analogously in Eq. (27), the calculation of the expansion at multipole moments is given as follows:

$$\int_{\Gamma_{\mathrm{C}}}u\left(z\right)\frac{\partial u^*\left(z_0,z\right)}{\partial n}dS\left(z\right)=\frac{1}{2\pi}\sum_{k=1}^{\infty}O_k\left(z_0-z_{\mathrm{C}}\right)\tilde{M}_k\left(z_{\mathrm{C}}\right), \tag{33}$$

where

$$\tilde{M}_k\left(z_{\mathrm{C}}\right)=\int_{\Gamma_{\mathrm{C}}}n\left(z\right)I_{k-1}\left(z-z_{\mathrm{C}}\right)u\left(z\right)dS\left(z\right),\ k=1,2,\ldots \tag{34}$$

are the moments of Γ_{C} around the z_{C} pole, now referring to the kernel integral $\frac{\partial u^*}{\partial n}$.

4.2 Moment-to-Moment (M2M) translation

This operation is responsible for transporting the moment of Γ_{C} around the center of the lowest level cell until it reaches the center of the cell at the level 2 of the tree. Figure 4a illustrates the translation of the calculated moment in z_{C}, center of the LC_{u} cell at the level 3, to $z_{\mathrm{C}'}$, center of the FC_{u} parent cell at level 2, and the calculation is done as follows:

$$M_k\left(z_{\mathrm{C}'}\right)=\sum_{l=0}^{k}I_{k-l}\left(z_{\mathrm{C}}-z_{\mathrm{C}'}\right)M_l\left(z_{\mathrm{C}}\right). \tag{35}$$

4.3 Moment-to-Local (M2L) translation

The multipole moment of Γ_{C} is carried from the center, $z_{\mathrm{C}'}$, from FC_{u} cell, at level 2, to the center z_{L} of FC_{d} cell, also at level 2, by moment-to-local (M2L) translation, where FC_{u} is found in the interaction list of FC_{d}, through the local expansion coefficient given below:

$$L_l\left(z_{\mathrm{L}}\right)=\left(-1\right)^l\sum_{k=0}^{\infty}O_{l+k}\left(z_{\mathrm{L}}-z_{\mathrm{C}'}\right)M_k\left(z_{\mathrm{C}'}\right). \tag{36}$$

4.4 Local-to-Local (L2L) translation

The calculated local expansion coefficient at the center z_{L} of the FC_{d} cell, at the level 2, needs to be translated to the center $z_{\mathrm{L}'}$ of the LC_{d} child cell, at the level 3, considering the example in Figure 4b. In the general case, the translation must happen to the lowest level, so that the contribution can be attributed to the actions of the source points of the leaf cell. The translation responsible for this step is the L2L given below:

$$L_m\left(z_{\mathrm{L}'}\right)=\sum_{l=m}^{\infty}L_l\left(z_{\mathrm{L}}\right)I_{l-m}\left(z_{\mathrm{L}'}-z_{\mathrm{L}}\right). \tag{37}$$

4.5 Contribution of distant elements

Finally, the integrals referring to Eq. (26) and Eq. (27) when Γ_{C} is far from the source points are calculated by FMM as follows:

$$I=\sum_{m=0}^{\infty}I_m\left(z_i-z_{\mathrm{L}'}\right)L_m\left(z_{\mathrm{L}'}\right),\ \text{with}\ i\in\left\{0,1,2,\ldots\right\}. \tag{38}$$

All M2M, M2L and L2L translations can be written using $\tilde{M}_k$ in the same way as done with M_k.

FMM is used to calculate the contributions of elements from cells far from the cell holding the source point. The calculation of the contributions of the elements of neighboring cells is performed with the conventional integration of BEM. The number of levels in the quaternary tree favors the validity conditions of the upward and downward processes, expressed in the legend of Figure 4, and contributes to the precision and speed of FMM as can be inferred from the results.

5 HIERARQUICAL MATRICES METHOD WITH CUR APPROXIMATION

BEM, as is known, produces full and non-symmetric matrices, but due to the characteristic of asymptotic smoothness of the fundamental solutions of the potential and the potential derivative, these matrices have the property that part of their submatrices can be represented by low-rank matrices. These submatrices are the final product of the hierarchical matrices method applied to BEM. The hierarchical form divides the matrices into blocks of sub-matrices and it all starts with structuring the nodes and boundary elements of the problem in the form of two binary trees, one tree for nodes and another for the boundary elements represented by their central points (Hackbusch, 2016; Bebendorf, 2000 and 2008; Börm, 2010). From then on, each submatrix block will correspond to a group, which in turn is made up of two subgroups, one of source points $X=\left\{x_1,x_2,\ldots,x_m\right\}$ and another of boundary elements $Y=\left\{y_1,y_2,\ldots,y_n\right\}$, these being central points of the elements. The construction of these groups is done by the hierarchical clustering process from the two binary trees and is governed by the criterion called geometric admissibility condition, which takes into account the size and distance between the subgroups mentioned above and results in the largest possible blocks of low-rank submatrices, as follows:

$$\max\left\{\mathrm{diam}\left(X\right),\mathrm{diam}\left(Y\right)\right\}\leq\eta\cdot\mathrm{dist}\left(X,Y\right). \tag{39}$$

If a group is considered admissible, then its submatrix or block will be represented by a low-rank matrix providing the memory saving, otherwise, the submatrix will not be approximated and in this case there will be no savings because the representation will be full. The hierarchical matrices method applied to the matrices of BEM and coupled with a low-rank representation method seeks to obtain the largest possible number of admissible blocks or the largest possible blocks and this makes large-scale problems manageable both in terms of storage costs and in time spent on floating point arithmetic operations, as it can present linear or logarithmic-linear complexity depending on the implementation (Bebendorf and Rjasanow, 2003).

Let $\mathrm{A} \in \mathbb{C}^{m \times n}$ be a matrix block of a BEM matrix corresponding to an admissible group, with m and n being the amounts of source points and boundary elements, respectively. A k −rank CUR approximation (or skeleton) is defined as

$$\mathrm{A} \approx \xi_k = \mathrm{CUR}, \tag{40}$$

where, in Julia language notation, $\mathrm{C}=\mathrm{A}[:,\mathrm{J}]$, $\mathrm{R}=\mathrm{A}[\mathrm{I},:]$ and $\mathrm{U}=\mathrm{A}^{-1}[\mathrm{I},\mathrm{J}] \in \mathbb{C}^{k \times k}$, where $\mathrm{I}=\{i_1, i_2, \ldots, i_k\}$ and $\mathrm{J}=\{j_1, j_2, \ldots, j_k\}$ sets of row and column indexes, respectively, with cardinality k, adaptively selected to ensure that $\mathrm{A}[\mathrm{I},\mathrm{J}]$ is invertible and has the largest absolute value of possible determinant of all submatrices $k \times k$ of A (Mahoney and Drineas, 2009; Kumar and Schneider, 2017). There are different methods of selecting J such as the Nearest-Neighbors criterion (NN) and which has been evaluated in high-dimensional problems showing good accuracy (March and Biros, 2017). Another method is Gravity Centers Sampling (GCS) with accuracy comparable to the ACA method (Ayala et al., 2020). The approach of this work is to select J through the k −Means clustering technique (MacQueen, 1967), which presents good performance as will be seen in the results. A brief outline of the k-Means clustering technique is presented in appendix A.

To select I specifically, the k −Means clustering partitions the subgroup of boundary elements of the admissible group into k clusters and then selects the closest k's boundary elements to the centroids of the k clusters, where $k << n$. The indices of the selected k's boundary elements will make up the set J. Then the pivoted partial QR decomposition is used to find I and, in short, proceeds as follows: calculate the factors p_1, Q_1 and R_1 through the QR factorization of the matrix $\mathrm{A}[:,\mathrm{J}]$ and through the permutation vector p_2 of the factorization QR of the factor Q_1^{T}, one finds the indices of the set I referring to the source points of the subgroup of the source points of the admissible group, which in turn correspond to the largest singular values of the A matrix, that is, the main source points (Golub and Loan, 2013). From now on, BEM builds a submatrix with the most significant rows and columns of the A matrix, that is, the $\mathrm{A}[\mathrm{I},\mathrm{J}]$ submatrix. There are also other ways to obtain I in linear time, such as the *Strong* RRQR routines (Gu and Eisenstat, 1996) or *maxvol* (Goreinov et al., 2008). So getting a good set of column indexes J is critical.

The computational cost of CUR approximation with k −Means is given as follows: $O(n \cdot k \cdot t)$ floating point operations to obtain J, with t being the number of iterations of the k −Means algorithm (Han et al., 2012); $O(m \cdot k^2)$ complexity of operations to perform the factorization QR truncated over C; $n \cdot k^2$ complexity of operations to perform the factorization QR truncated over Q^{T}. So the total cost is $O((m+n) \cdot k^2)$ (Ayala et al., 2020). A summary of the definition of the QR decomposition can be seen in appendix B.

The fact that the kernel of the integrals of the matrix entries H is the fundamental solution of the potential derivative, which in turn is asymptotically smoother than the fundamental solution of the potential, makes its submatrices have the tendency to have rank lower than the submatrices of the G matrix. So it makes sense to base the partial factorization QR pivoted from the submatrices of the H matrix. The qualitative information about the factorizations made on the submatrices of the H matrix, are used to obtain the submatrices of low rank of the G matrix. Consequently, there is an important reduction in the number of rows and columns in relation to the original submatrix.

The implementation of the isogeometric formulation of BEM coupled with the hierarchical matrices method, proposed in this work, also experienced a simplification due to the use of the Bézier extraction operator derived from the Bézier decomposition operation. The integration did not undergo any changes and was carried out as is done in conventional BEM. Singular, quasi-singular and regular integrals were satisfactorily treated with the hyperbolic sine

transformation (Johnston and Elliott, 2004). The linear system is also solved with the GMRES iterative method restarted and unpreconditioned (Saad and Schultz, 1986).

5.1 Discussion of the effectiveness of k −Means clustering in the CUR approximation

The accuracy of the CUR approximation, of rank k, depends on the good conditioning of the matrix $\mathrm{A}\left[\mathrm{I},\mathrm{J}\right]$ which, in turn, depends on adequate choices for the indices J of columns and I of rows, both with cardinality k. Such conditioning is directly related to the maximal volume of the simplex formed by the j −th column vectors, $c_j \in \mathbb{C}^m$, of the A matrix (Goreinov and Tyrtyshnikov, 2001). The volume of this simplex is given by the Cayley-Menger determinant as follows:

$$v_k = \frac{\left(-1\right)^k}{2^{k-1}\left(k-1\right)!^2}\begin{Vmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & 0 & d_{12}^2 & \cdots & d_{1k}^2 \\ 1 & d_{21}^2 & 0 & \cdots & d_{2k}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & d_{k1}^2 & d_{k2}^2 & \cdots & 0 \end{Vmatrix}, \tag{41}$$

where $d_{jl} = \left\| c_l - c_j \right\|_2$ for $j,l = 1,\ldots,k$ (Sommerville, 1958).

For a simple argument, suppose $u : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$, that is, the fundamental solution of the potential relating real domains and remembering that it is used to calculate the matrix A entries. Without much algebraic effort, it can be deduced from the mean value theorem that

$$d_{jl}^2 = \left| y_j - y_l \right|^2 \sum_{i=1}^{m} \left| \partial_y u\left(x_i, \psi_{lj}\right) \right|^2, \tag{42}$$

where x_i's and y_j's are source points and field points corresponding to subgroups of an admissible group,

respectively, and ψ_{lj} is a real number between y_l and y_j. It can be noted that the values of d_{jl}^2 are related to the distance between the field points y's selected by the set J. Therefore, if the field points are very close to each other, then a small value of v_k will result, which harms the conditioning of the $\mathrm{A}\left[\mathrm{I},\mathrm{J}\right]$ matrix and decreases the performance of the CUR approximation. The k −Means clustering, due to its concept of intracluster similarities and intercluster differences, in terms of the distance metric, acts to keep these field points y's as far apart as possible and, consequently, guarantying d_{jl}^2 that are different from zero preserving the linear independence of the lines as much as possible, which favors the maximal volume of the simplex and the conditioning of the $\mathrm{A}\left[\mathrm{I},\mathrm{J}\right]$ matrix.

6 NUMERICAL RESULTS

In the results that follow, the two proposed formulations are evaluated and compared with the conventional BEM using constant elements. All are applied to problems with known analytical solution, with the exception of the plate with many hole problem where only the processing time was analyzed. The simulation was performed on an Acer notebook with an Intel i5 processor at 2.3 GHz with Turbo Boost up to 2.8 GHz, 8 Gigabytes of RAM. In this work, the Julia programming language in version 1.5.3 was used.

6.1 Numerical setup

The number of degrees of freedom is given as a function of the h −refinement, which in turn inserts parametric knots into the U knot set. At each insertion of knots, an additional control point is created respecting Eq. (7). So, there is no change in geometry and smoothness between the Bézier curves. The degree p, or order $p+1$, of the NURBS curves will be explained in each case. The npg is the number of parametric Gauss-Legendre coordinates and these will be readapted by the hyperbolic sine transformation for the calculation of integrals. In this work, for the two proposed formulations, it was

sufficient to set $npg = 8$. The tolerance $\varepsilon = 10^{-6}$ was used as a weight for the stopping criterion for the preconditioned and restarted GMRES method (Saad and Schultz, 1986), which criterion is the defined relative residual

$$\mathrm{R}^{(i)} = \frac{\left\| \mathrm{A}x^{(i)} - \mathrm{b} \right\|_2}{\left\| \mathrm{b} \right\|_2}, \tag{43}$$

where $x^{(i)}$ represents a candidate solution of the linear system in the i−th iteration and $\left\| * \right\|_2$ denotes the L^2 norm for a finite dimensional space.

The instrument that will be used to evaluate the performance of the methods is the calculation of the approximation error by the L^2 norm, for the continuous case, between the numerical solution u^h and the exact solution u on the boundary $\Gamma = \partial\Omega$ of the problem:

$$\text{error} = \sqrt{\int_\Gamma \left(u - u^h \right)^2 d\Gamma}\,. \tag{44}$$

6.1.1 FMM parameters:

1. $nexp$ is the number of terms of the expansion of moments in multipoles and $ntylr$ is the number of terms in the local expansion. $nexp$ and $ntylr$, for simplicity, they were considered equal to 20, which value can be estimated according to the formula $-\log_2(\varepsilon)$, deducted from the calculation of the error estimate of the expansion of moments (Liu, 2009).
2. $maxl$ is the maximum number of elements in a leaf cell: $maxl = 30$.
3. $levmx$ is the maximum number of levels in the quaternary tree: $levmx = 15$.
4. To calculate the multipole moments, $npg = 3$.

6.1.2 Parameters of the hierarchical matrices method:

1. $nnucleos$ is the number of clusters produced by k -Means technique: $nnucleos = 5$.
2. max_elem is the maximum number of elements in a leaf cell: $max_elem = 30$.
3. η is the parameter of the geometric admissibility condition: $\eta = 1$.

6.2 Heat transfer in a hollow cylinder

This is a heat transfer problem, by conduction, in a hollow cylinder modeled as a 2D problem as shown in Figure 6. This problem has an analytical solution that will be used to verify the results obtained by the proposed formulations in this work using NURBS of order 3. The results of conventional BEM using quadratic elements will also be compared. The known boundary conditions are temperature at the inner boundary, S_i, and flux at the outer boundary, S_e. The analytical solution for the temperature is given by:

$$T(r) = T_i - q_e \cdot r_e \cdot \log\left(\frac{r}{r_i} \right), \tag{45}$$

and for the flux:

$$q(r) = -q_e \frac{r_e}{r}, \tag{46}$$

where T_i and q_e are the temperature and flux in the inner and outer boundary, respectively, and r, r_i and r_e are given in Figure 6. The data for the simulation are from Table 1.

In Figure 7, it is possible to observe the approximation error, by the norm L^2, for the temperature, calculated along the boundaries S_i and S_e. It is observed that the three methods converge to the exact solution, however, the isogeometric formulations with fewer degrees of freedom reach greater accuracy in relation to the conventional BEM. Figure 8 presents the result for calculation of the approximation error for the heat flux and it is also possible to observe the convergence of the three methods for the exact solution.

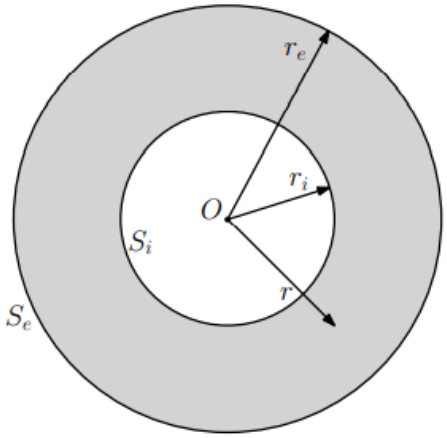


Figure 6: Potential problem in an annular region.

Table 1 T Parameters for the hollow cylinder simulation.

Input variables	r_i	$1\,\mathrm{m}$
	r_e	$2\,\mathrm{m}$
	T_i	$100\ \mathrm{K}$
	q_e	$-200\ \mathrm{W/m^2}$
	Thermal Conductivity	$1\ \mathrm{W/(mK)}$
Analytical results	T_e	$377.2588722\ \mathrm{K}$
	q_i	$400\ \mathrm{W/m^2}$

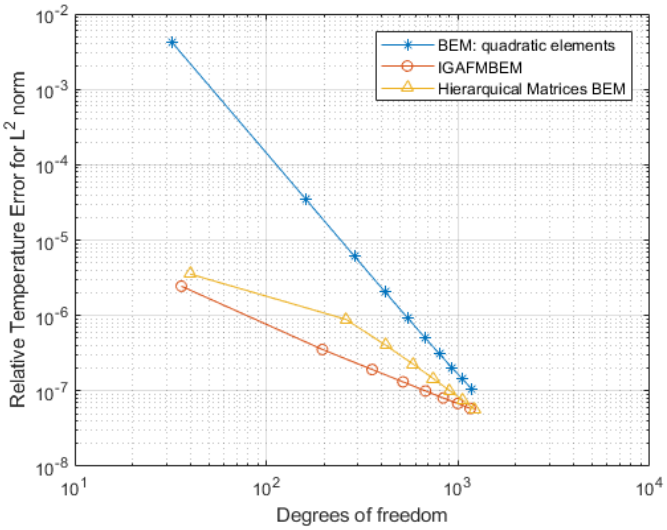


Figure 7: Hollow Cylinder: temperature results.

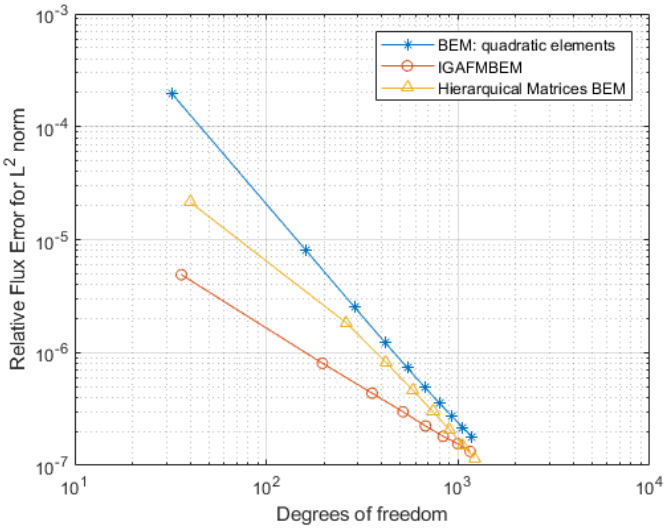


Figure 8: Hollow Cylinder: Flux results.

6.3 Moulton problem

Consider the problem of heat transfer, by conduction, now in a rectangular plate AOBCD as shown in Figure 9, with constant thermal conductivity of value $1\ \mathrm{W}\cdot\mathrm{m}^{-1}\cdot\mathrm{K}^{-1}$.

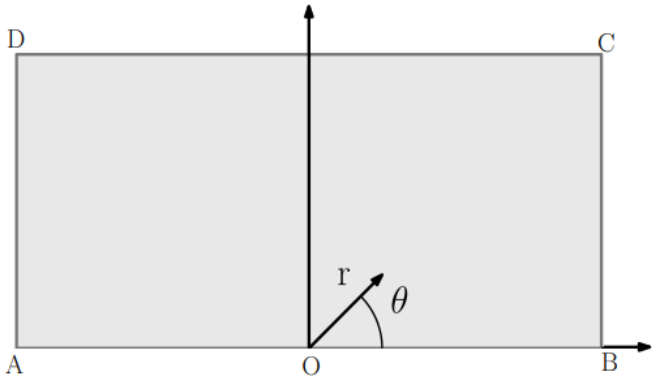


Figure 9: Domain for the Moulton problem.

The boundary, although simple, has the boundary conditions now variable point by point and are given as follows, where u represents the temperature and q the heat flux in terms of the polar coordinates r and θ:

$$q_1(r,\theta) = -\frac{1}{2\sqrt{r}}\left(\cos\left(\frac{\theta}{2}\right)\cos(\theta)+\sin\left(\frac{\theta}{2}\right)\sin(\theta)\right) \text{ in BC}, \tag{47}$$

$$q_2(r,\theta) = -\frac{1}{2\sqrt{r}}\left(\cos\left(\frac{\theta}{2}\right)\cos(\theta)-\sin\left(\frac{\theta}{2}\right)\sin(\theta)\right) \text{ in CD}, \tag{48}$$

$$q_3(r,\theta) = \frac{1}{2\sqrt{r}}\left(\cos\left(\frac{\theta}{2}\right)\cos(\theta)+\sin\left(\frac{\theta}{2}\right)\sin(\theta)\right) \text{ in DA} \tag{49}$$

$$q_4(r,\theta) = 0 \text{ in OB}. \tag{50}$$

and

$$u(r,\theta)=0 \text{ in } \mathrm{AO}. \tag{51}$$

The analytical solution to this problem is given by:

$$u(r,\theta)=\sqrt{r}\cos\left(\frac{\theta}{2}\right), \tag{52}$$

$$q_x(r,\theta)=\frac{\cos\left(\frac{\theta}{2}\right)}{2\sqrt{r}}, \tag{53}$$

and

$$q_y(r,\theta)=\frac{\sin\left(\frac{\theta}{2}\right)}{2\sqrt{r}}. \tag{54}$$

The approximation error was calculated, by the L^2 norm, given in Eq. (44), for the temperature in the segment OB, whose result can be seen in Figure 10, and for the heat flux in the segment AO, whose result can be seen in Figure 11. For the two proposed formulations, NURBS of order 2, degree 1, were used and for the conventional BEM constant elements were used. It is worth noting that the boundary conditions imposed are not constant, but vary from point to point, including an important singularity at point O for the boundary conditions of heat flux. It should be noted that the boundary geometry is quite simple, and can be represented exactly by low order elements such as NURBS degree 1 for the isogeometric and constant elements for the conventional BEM.

For the temperature result, it can be seen in Figure 10 that the three methods converge asymptotically to the analytical solution. Convergence is a little slower for the case of the heat flux given in Figure 11, which can be explained by the singularity at point O. In terms of accuracy and precision, here we also highlight the isogeometric formulations in relation to the conventional BEM.

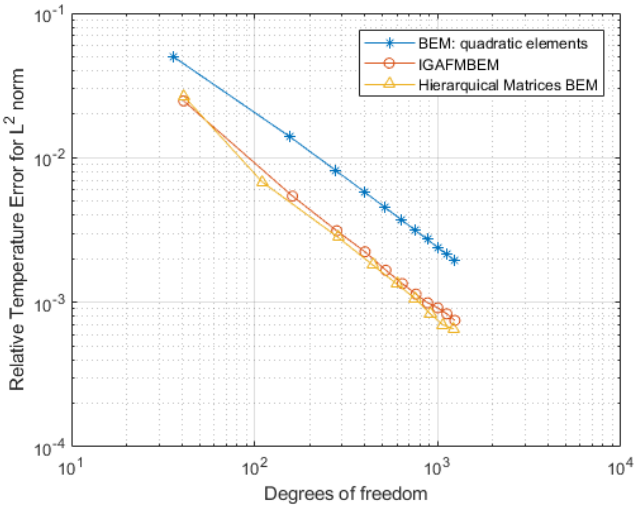


Figure 10: Moulton Problem: temperature results.

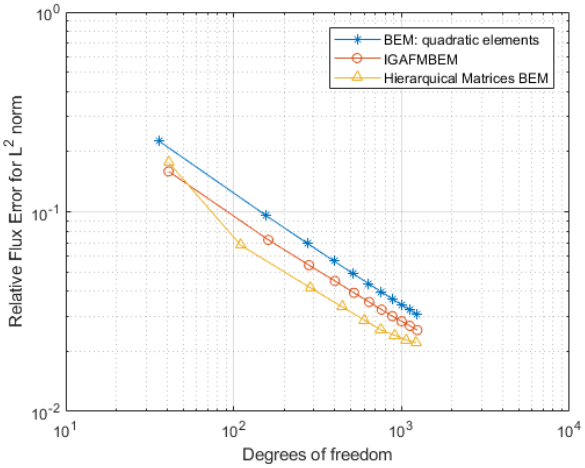


Figure 11: Moulton Problem: flux results.

6.4 Problem of heat transfer in a square plate with many holes

In order to complement the verification of the efficiency of the presented formulations, it is proposed to simulate, in order to access processing time, the problem of heat transfer in a square plate of $1\ \mathrm{m}^2$ area and constant thermal conductivity of $1\ \mathrm{W}\cdot\mathrm{m}^{-1}\cdot\mathrm{K}^{-1}$, with an increasing number of evenly distributed circular holes. The size of the holes is modified so that the sum of the areas of the holes makes up the proportion of 12.47% of the plate area, which proportion is kept constant throughout the simulation. This problem has been analyzed in the reference (Liu, 2009), using Fast Multipole Accelerated BEM (FMBEM) with constant elements.

Boundary conditions are applied with thermal insulation on the lower and upper outer boundary and temperatures $\mathrm{T}=0$ and $\mathrm{T}=1$ for the left and right outer boundary, respectively. Zero flux is also defined on the inner boundary. For an example of a plate with 16 holes see Figure 12.

The simulation was performed by varying the number of holes between 1 and 10000. It was observed the need to maintain a uniformity in the length of the elements between the NURBS patches of the external and internal boundary. Otherwise, there would be two problems to deal with: first, there would be a need to use different amounts of Gauss-Legendre points on the outer and inner boundary for numerical integration, in order to prevent loss of precision; second, non-uniformity could weaken the validity conditions of FMM. The h –refinement was applied only on the external boundary to guarantee the uniformity mentioned above, while in each hole the amount of source points and boundary elements were kept constant. Thus, the number of degrees of freedom is given as a function of the number of holes and the refinement of the outer boundary.

The processing time was considered from the assembly of the matrices until the end of GMRES. The evaluation involved the conventional BEM, the FMBEM with constant elements (Liu, 2009) and the two proposed formulations in this article. Results can be seen in Figure 13. It is worth noting the suitability of fast formulations for large problems. The changes in the slopes of the curves show the improvement in speed for the proposed formulations: $O(N)$ for Fast Multipole Accelerated Isogeometric BEM (IGAFMBEM) and FMBEM; $O(N\cdot\log N)$ for the formulation with hierarchical matrices and $O(N^2)$ for the conventional BEM. The time difference between IGAFMBEM and FMBEM can be explained by the complexity of numerical operations performed.

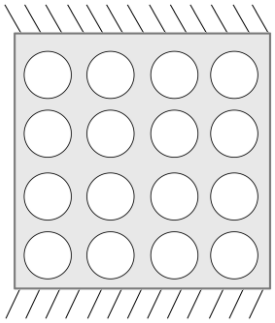


Figure 12: Plate with $16=4\times4$ holes.

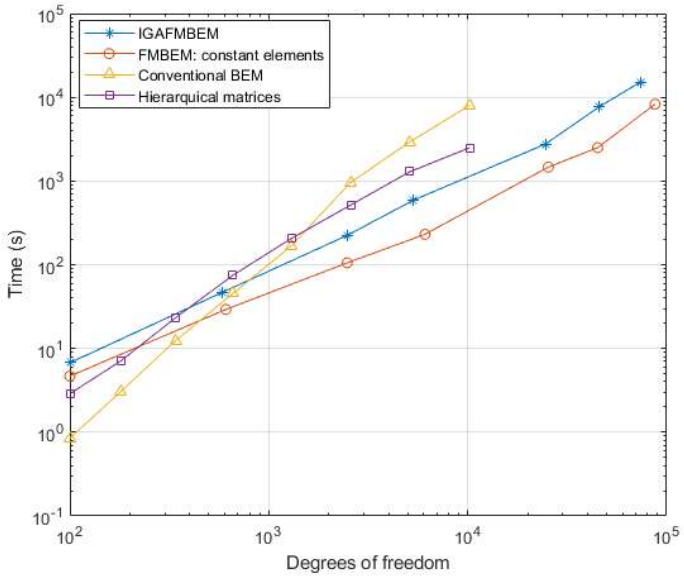


Figure 13: Execution time between assembly and GMRES.

7 CONCLUSIONS

In this work two fast isogeometric formulations of the accelerated boundary element method were presented, one by the Fast Multiple Method and the other by the Hierarchical Matrices Method with low rank CUR approximation coupled to a geometric sampling method.

The NURBS were used as shape functions and generated by the Bézier extraction operator obtained by the Bézier decomposition which, in turn, reduced the computational cost and brought simplicity in the implementation, making the isogeometric formulations very similar to the formulation of the conventional BEM regarding the structure of elements.

The results show that the proposed formulations have high accuracy and efficiency while dealing with generic geometries and boundary conditions. The effectiveness of the boundary conditions imposition method is also shown.

Regarding the processing time, the method coupled to FMM presented a complexity order close to $O(N)$ as it is known in the literature. While the complexity order of operations of the isogeometric formulation of BEM with the Hierarchical Matrices Method was estimated close to $O(N \cdot \log N)$. It is worth mentioning that using $\mathcal{H}^2$ matrices, a better result is expected for the hierarchical matrices in terms of memory savings and computational complexity (Hackbusch et al., 2000; Hackbusch and Börm, 2002; Löhndorf, 2003; Börm and Hackbusch, 2004; Börm, 2006 and 2013; Börm and Garcke, 2007; Hackbusch, 2016). The use of $\mathcal{H}^2$ matrices will be investigated in future work.

Therefore, the proposed formulations in this work show a promising horizon for scientific computing regarding mesh generation, precision, accuracy, storage cost and processing speed of large-scale problems.

Author's Contribuitions: Conceptualization, E Bastos, EL Albuquerque, LS Campos and LC Wrobel; Formal Analysis, E Bastos, EL Albuquerque, LS Campos and LC Wrobel; Methodology, E Bastos, EL Albuquerque, LS Campos and LC Wrobel; Investigation, E Bastos, EL Albuquerque, LS Campos; Software, E Bastos, EL Albuquerque, LS Campos; Supervision, E Bastos, EL Albuquerque, LS Campos and LC Wrobel; Writing - original draft, E Bastos, EL Albuquerque, LS Campos; Writing - review & editing, E Bastos, EL Albuquerque, LS Campos.

Editor: Marco L. Bittencourt

References

Abramowitz, M. and Stegun, I., (1972). Handbook of mathematical functions: with formulas, graphs and mathematical tables. Eighth Dover Printing. New York: Dover.

Ayala, A., Claeys, X. and Grigori, L., (2020). Linear-time cur approximation of bem matrices. Journal of Computational and Applied Mathematics, 368:1–29, Elsevier.

Barnes, J. and Hut, P., (1986). A hierarquical o(nlogn) force-calculation algorithm. Nature, 324:446–449.

Barra, L. P. S., Coutinho, A. L. G. A., Mansur, W. J. and Telles, J. C. F., (1992). Iterative solution of bem equations by gmres algorithm. Computer and Structures, 44:1249–1953.

Bebendorf, M., (2000). Approximation of boundary element matrices. Numerishce Mathematik, 86:565–589.

Bebendorf, M., (2008). Hierarchical matrices. S.L. Springer.

Bebendorf, M. and Rjasanow, S., (2003). Adaptive low-rank approximation of collocation matrices. Computing, 70:1–24.

Beer, G., Marussig, B. and Duenser, C., (2019). The Isogeometric Boundary Element Method. Springer.

Borden, M. J., Scott, M. A., Evans, J. A. and Hughes, T. J. R., (2011). Isogeometric finite element data structures based on bézier extraction of nurbs. International Journal for Numerical Methods in Engineering, 7:15–47.

Börm, S., (2006). $\mathcal{H}^2$ -matrices - an efficient tool for the treatment of dense matrices. Habilitation thesis, Universität.

Börm, S., (2010). Efficient Numerical Methods for Non-local Operators: $\mathcal{H}^2$ matrix Compression, Algorithms and Analysis. European Mathematical Society.

Börm, S., (2013). Efficient Numerical Methods for Non-local Operators. EMS, Zürich (2010). Corrected 2nd printing.

Börm, S. and Garcke, J., (2007). Approximating gaussian processes with $\mathcal{H}^2$ -matrices. Machine Learning ECML, pages 42–53.

Börm, S. and Hackbusch, W., (2004). Approximation of boundary element operators by adaptive $\mathcal{H}^2$ -matrices. In Foundations of computational mathematics: Minneapolis, London Math. Soc. Lecture Note Ser., 312:58–75.

Boutsidis, C., Mahoney, M. and Drineas, P., (2009). An improved approximation algorithm for the column subset selection problem. Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 968–977.

Bucher, H. F., Wrobel, L. C., Mansur, W. and Magluta, C., (2002). A novel approach to applying fast wavelet transforms in the boundary element method. Electron J Bound Elem, 2:187–95.

Cabral, J. J. S. P., Wrobel, L. C. and Brebbia, C. A., (1990). A bem formulation using b-splines: I - uniform blending functions. Engineering Analysis with Boundary Elements, 7:n. 3, 136–144.

Cabral, J. J. S. P., Wrobel, L. C. and Brebbia, C. A., (1991). A bem formulation using b-splines: II - multiple knots and non-uniform blending functions. Engineering Analysis with Boundary Elements, 8:51–55.

Campos, L. S., Albuquerque, E. L. and Wrobel, L. C., (2017). An aca accelerated isogeometric boundary element analysis of potential problems with non-uniform boundary conditions. Engineering Analysis with Boundary Elements, 80:108– 115.

Collier, N., Pardo, D., Dalcin, L., Paszynski, M. and Calo, V. M., (2012). The cost of continuity: a study of the performance of isogeometric finite elements using direct solvers. Comput Methods Appl Mech Eng, 61:213:353.

Cottrell, J. A., Hughes, T. J. R. and Bazilevs, Y., (2009). Isogeometric Analysis: Toward Integration of CAD and FEA. Wiley: Chichester, U. K.

Crotty, J., (1982). A block equation solver for large unsymmetric matrices arising in the boundary integral equation method. Int J Numer Methods Eng, 18(7):997–1017.

Dongarra, J. and Sullivan, F., (2000). The top 10 algorithms of the twentieth century. Computing in Science and Engineering, 2(1):22–23.

Farin, G. E., (1996). Curves and Surfaces for Computer-Aided Geometric Design: A Practical Code. Academic Press, Inc.

Fong, W. and Darve, E., (2009). The black-box fast multipole method. Journal of Computational Physics, 228:8712–8725.

Golub, G. H. and Loan, C. F. V., (2013). Matrix Computations. The Jhons Hopkins University Press Baltimore.

Goreinov, S. and Tyrtyshnikov, E., (2001). The maximal-volume concept in approximation by low-rank matrices. Contemporary Mathematics, 280:47–51.

Goreinov, S., Oseledets, I., Savostyanov, D., Tyrtyshnikov, E. and Zamarashkin, N., (2008). How to find a good submatrix. Institute for Compuatational Mathematics Hong Kong Baptist University, 08-10:1–9.

Greenbaum, A., (1997). Iterative Methods for Solving Linear Systems. Frontiers in Applied Mathematics, v. 17. Theorem 10.1.3. SIAM.

Greengard, L. F. and Rokhlin, V., (1987). A fast algorithm for particle simulations. Journal of Computational Physics, 73:325–348.

Gu, M. and Eisenstat, S., (1996). Efficient algorithms for computing a strong rankrevealing qr factorization. SIAM J. Matrix Anal. Appl., 17(4):848–869.

Hackbusch, W., (2016). Hierarchical matrices: algorithms and analysis. Springer Series in Computational Mathematics. Springer, 3rd edition.

Hackbusch, W. and Börm, S., (2002). Data-sparse approximation by adaptive $\mathcal{H}^2$ -matrices. Computing, 69:1–35.

Hackbusch, W., Khoromskij, B. N. and Sauter, S. A., (2000). On $\mathcal{H}^2$ -matrices. In Lectures on applied mathematics, pages 9–29, Springer-Verlag, Berlin.

Halko, N., Martinsson, P. G. and Tropp, J. A., (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. SIAM Review, 53(2):217–288.

Han, J., Kamber, M. and Pei, J., (2012). Data Mining: Concepts and Techniques. 3rd. Ed. Amsterdam: Morgan Kaufmann Publishers.

Hughes, T. J. R., Cottrell, J. A. and Bazilevs, Y., (2005). Isogeometric analysis: Cad, finite elements, nurbs, exact geometry, and mesh refinement. Computer Methods in Applied Mechanics and Engineering, 194:4135–4195.

Johnston, P. R. and Elliott, D., (2004). A sinh transformation for evaluating nearly singular boundary element integrals. Int. J. Numer. Meth. Engng, 62:564–578.

Kagan, P. and Fisher, A., (2000). Integrated mechanically based cae system using b-spline finite elements. Computer Aided Design, 32:n. 8, 539–552.

Kane, J., Kumar, B., and Kashava, S. S., (1990). An arbitrary condensing, noncondensing solution strategy for large scale, multi-zone boundary element analysis. Comput Methods Appl Mech Eng, 79(2):219–44.

Kapur, S. and Long, D., (1998). N-body problems: Ies3: Efficient electrostatic and electromagnetic simulation. IEEE Computational Science and Engineering, 5(4):60–67.

Kumar, N. K. and Schneider, J., (2017). Literature survey on low rank approximation of matrices. Linear and Multilinear Algebra, 65(11):2212–2244.

Li, K. and Qian, X., (2011). Isogeometric analysis and shape optimization via boundary integral. Computer-Aided Design - Elsevier Science, 43:1427–1437.

Liu, Y. J., (2009). The fast multipole boundary element method - Theory and Applications in Engineering. Cambridge - University Press.

Löhndorf, M., (2003). Effiziente behandlung von integralgleichungen mit $\mathcal{H}^2$ -matrizen variabler ordnung. Doctoral thesis, Universität Leipzig.

Loyola, F. M., Doca, T., Campos, L. S., Trevelyan, J. and Albuquerque, E. L., (2022). Analysis of 2D contact problems under cyclic loads using IGABEM with Bézier decomposition. Engineering Analysis with Boundary Elements, 139:246-263.

MacQueen, J. B., (1967). Some methods for classification and analysis of multivariate observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press: 281–297.

Mahoney, M. and Drineas, P., (2009). Cur matrix decompositions for improved data analysis. Proceedings of the National Academy of Sciences, 106(3):697–702.

Mansur, W. J., Araujo, F. C. and Malaghini, J. E. B., (1992). Solution of bem systems of equations via iterative techniques. International Journal for Numerical Methods in Engineering, 33:1823–1841.

March, W. and Biros, G., (2017). Far-field compression for fast kernel summation methods in high dimensions. Applied and Computational Harmonic Analysis, 43(1):39– 75.

Matsumoto, T. and Takahashi, T., (2012). An application of fast multipole method to isogeometric boundary element method for laplace equation in two dimensions. Engineering analysis with boundary elements, 36:1766–1775.

Ozdemir, N. A. and Lee, J. F., (2004). A low-rank ie-qr algorithm for matrix compression in volume integral equations. IEEE Transactions on Magnetics, 40(2):1017– 1020.

Peigl, L. and Tiller, W., (1996). The NURBS Book. Monographs in Visual Communication 2nd ed. Berlim, Springer-Verlag.

Rigby, R. and Aliabadi, M., (1995). Out-of-core solver for large, multi-zone boundary element matrices. Int J Numer Methods Eng, 38(9):1507–33.

Rogers, D. F., (2000). An Introduction to NURBS: with historical perspective. Elsevier.

Rokhlin, V., (1985). Rapid solution of integral equations of classical potencial theory. Journal of Computational Physics, 60:187–207.

Saad, Y. and Schultz, M. H., (1986). Gmres: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J Sci Stat Comput, 7(3):856–869.

Scott, M. A., Simpson, R. N., Evans, J. A., Lipton, S., Bordas, S. P. A., Hughes, T. J. R. and Sederberg, T. W., (2013). Isogeometric boundary element analysis using unstructured t-splines. Computer Methods in Applied Mechanics and Engineering, Elsevier, 254:197–221.

Shene, C. K., (2011). Introduction to computing with geometry notes. Michigan Technological University.

Simpson, R. N. and Liu, Z., (2016). Acceleration of isogeometric boundary element analysis through a black-box fast multipole method. Engineering Analysis with Boundary Elements, 66:168–182.

Simpson, R. N., Bordas, S. P. A., Trevelyan, J. and Rabczuk, T., (2012). A two-dimensional isogeometric boundary element method for elastostatic analysis. Computer Methods in Applied Mechanics and Engineering, Elsevier, 209:87–100.

Sommerville, D. M. Y., (1958). An Introduction to the Geometry of n Dimensions. Dover, New York. New Academic Science Ltd.

Voronin, S. and Martinsson, P. G., (2017). Efficient algorithms for cur and interpolative matrix decompositions. Advances in Computational Mathematics, 43(3):495– 516.

Wang, Q., Zhou, W., Cheng, Y., Ma, G., Chang, X. and Liu, B., (2019). A nurbs-enhanced improved interpolating boundary element-free method for 2d potential problems and accelerated by fast multipole method. Engineering analysis with boundary elements, 98:126–136.

Wang, Y. and Benson, D., (2015). Multi-patch nonsingular isogeometric boundary element analysis in 3d. Computer Methods in Applied Mechanics and Engineering, Elsevier, 293:71–91.

Ying, L., Biros, G. and Zorin, D., (2004). A kernel-independent adaptive fast multipole algorithm in two and three dimensions. Journal of Computational Physics, 196:591–626.

Appendix A. $k-$ **Means algorithm**

University of California researcher James B. Macqueen was the first to use the term "$k-$means" in 1967, although ten years earlier the conceptual idea was conceived by Polish mathematician Wladyslaw Hugo Dyonizy Steinhaus. Since then, the concept has gone through many heuristic developments making it computationally viable. Today, it is an unsupervised learning and clustering algorithm used to partition n data into distinct k clusters. It groups data that share important and similar characteristics so that, empirically, any two groups are different with respect to these characteristics. In this work, the characteristic or metric that will define the clusters will be the Euclidean distance.

To fix the ideas, let's consider a dataset with n points, $\mathrm{P}=\{\mathrm{x}_i\}_{i=1}^{n}$. The goal is to partition P into k subsets $\mathrm{C}=\{\mathrm{C}_1,\mathrm{C}_2,\ldots,\mathrm{C}_k\}$, where each C_i is called a cluster of the set. For each C_i we will associate a y_i, named centroid, which will represent and redefine the cluster, being calculated as:

$$y_i=\frac{1}{n_i}\sum_{x_j\in \mathrm{C}_j} x_j\,, \tag{A.1}$$

where n_i will be the number of elements of C_i.

To verify the fit of the clusters with their respective elements, a function, called similarity function, is defined, given as:

$$F_S(P)=\sum_{i=1}^{k}\sum_{x_j\in \mathrm{C}_i} D(x_j,y_i)\,, \tag{A.2}$$

where D is the Euclidean distance. The F_S function represents the sum of all distances between each element and the centroid of its cluster, but it can also be seen as a measure of clustering dispersion. As an ultimate goal, the algorithm seeks to minimize F_S by finding a local solution. Given a set of points, the algorithm steps are as follows:

1. Initially, all points in the set P are distributed into random k clusters.

2. Calculates through Eq. (A.1), the centroid of each cluster C_i.

3. Assigns each point $x_j\in\mathrm{P}$ to a cluster C_{i^*}, with centroid y_{i^*} closest to the point, that is, $i^*=\underset{i=1,2,\ldots,k}{\arg\min}\, D^2(x_j,y_i)$.

4. After the previous step, many points will have changed groups, so it is necessary to update the centroids of each cluster, so the 2º step is repeated, where we will find a new centroid y_i for the cluster C_i.

5. The last two steps will be repeated iteratively, until the respective centroids no longer change or satisfy the established precision, then this iteration will be the local minimum.

6. The stopping test is based on the analysis of the differences between the centroids of the current iteration and the previous one, that is, $\sum_{i=1}^{k}\left\|y_i^t-y_i^{t-1}\right\|\le\varepsilon$, y_i^t represents the centroid of the current iteration and ε the fixed precision.

Appendix B. Pivoted partial QR factoring

Every matrix $\mathrm{A\,m\times n}$ of rank k, with $k<\min\{m,n\}$, admits a decomposition $\mathrm{A=QR}$, where Q is a matrix $m\times k$ with orthonormal columns and R is a $k\times n$ upper quasi-triangular matrix (Golub and Loan, 2013). For R to be upper triangular, the decomposition will require pivoting columns of the matrix A and this introduces a permutation matrix P, hence $\mathrm{AP=QR}\ \Leftrightarrow\ \mathrm{A=QRP^{T}}$. The permutation matrix P is such that the diagonal elements of R are non-increasing, thereby improving precision and providing a basis for a more accurate knowledge of the numerical rank of A with lower computational cost than SVD.

To calculate a pivoted partial QR decomposition, successive orthogonalizations with pivoting over the columns of the A matrix one by one are performed. This task is terminated when the Frobenius norm of the remaining columns is less than a given computational tolerance ε. Let ℓ be the smallest number of steps for which this tolerance is reached, hence the process results in partial factoring

$$\mathrm{A=QR+E},$$

where Q is an orthonormal $m\times\ell$ matrix, R is an $\ell\times n$ upper triangular matrix, and E is a residue matrix satisfying $\left\|\mathrm{E}\right\|_{\mathrm{F}}\le\varepsilon$. The computational cost of this partial decomposition is $O(\ell\cdot m\cdot n)$, which in turn is less than $O(m\cdot n\cdot\min\{m,n\})$ for the SVD decomposition (Halko et al., 2011).