XLVI CILAMCE
VITÓRIA - ES - BRAZIL - 2025
XLVI IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGENEERING

An ACA accelerated boundary element analysis of viscous flow

L. S. Campos[1], L.A Almeida[1], E.L Albuquerque[2], T.R Nicolaidis[2]

[1]*Department of Mechanical Engineering, UFES*
Avenida Fernando Ferrari, 514, 29060-410, Vitória/ES, Brazil
lucas.s.campos@ufes.br, lucas.almeida.45@edu.ufes.br
[2]*Department of Mechanical Engineering, UnB*
Campus Universitário Darcy Ribeiro, 70910-900, Brasília-DF, Brazil
eder@unb.br, thiago1103@hotmail.com

Abstract.

This paper presents an accelerated boundary element method (BEM) for analyzing incompressible viscous flow. The formulation stems from the continuity and Navier-Stokes equations. By employing the direct interpolation method within the BEM framework, domain integrals are handled without the need for cells. Hierarchical matrices are used to mitigate the computational cost inherent in traditional BEM, leading to significant improvements in simulation time and storage efficiency.

Keywords: Boundary Element Method, Hierarchical matrices , Viscous flow

1 Introduction

The Boundary Element Method (BEM) is a numerical technique that that distinguishes itself from methods like the Finite Element Method (FEM) and the Finite Difference Method (FDM) by it transforming a problem's governing partial differential equations (PDEs) into integral equations that are applied solely over the boundary of the domain. This boundary-only formulation reduces the problem's dimensionality.

While potential flows can be solved with a boundary-only discretization, analyzing Newtonian fluid flows governed by the Navier-Stokes equations, introduces distributed body forces, or non-linearities, which result in domain integrals. Evaluating these integrals typically requires discretizing the entire computational domain, which negates BEM's key advantage of boundary-only formulation.

To address this issue, this study employs the Direct Interpolation Boundary Element Method (DIBEM)[1, 2] to handle the domain integrals that arise from the Navier-Stokes equations. As presented in [3], this approach maintains the boundary-only discretization and leverages the BEM formulation for linear elasticity by using a penalty function method.

Despite its advantages, BEM's computational demands are a major hurdle for large-scale problems. The resulting BEM matrices are dense, meaning both the memory required for storage and the time for matrix-vector products scale quadratically with the number of degrees of freedom. For direct solvers, the computational time scales cubically. To overcome these limitations, Hierarchical Matrices ($\mathcal{H}$-matrices)[4] are utilized in this work. This method reduces the computational complexity to a nearly linear scale.

A further consideration is that as the Reynolds number increases, a greater number of iterations are necessary for convergence. A key advantage of the $\mathcal{H}$-matrix approach, compared to the Fast Multipole Method[5], is that the compressed matrix can be reused in each iterative step. This reusability significantly enhances the overall computational efficiency of the proposed method.

2 Governing equations

The isothermal fluid dynamics of a Newtonian incompressible fluid can be represented, as presented in [6], by an equivalent homogeneous and isotropic elastostatic problem. Once this correspondence is made, the integral

equation from elasticity [7] is used:

$$c_{kl}(d)\,u_l(d) = -\int_\Gamma t_{kl}(x,d)\,u_l(x)\,d\Gamma + \int_\Gamma u_{kl}(x,d)\,t_l(x)\,d\Gamma - \int_\Omega u_{kl}(x,d)\,u_j(x)u_{l,j}(x)\,d\Omega \tag{1}$$

where $u_{ij}(x,d)$ and $t_{kl}(x,d)$ are the Kelvin's solutions, r is the distance between the load point x and field point d. u_i is the fluid velocity and $-(u_j(x)u_{l,j}(x)$ is the convective term. The coefficient $c_{kl}(d)$ is determined by the load point position: $c_{kl}(d) = \delta_{kl}$ for $d \in \Omega$ and $c_{kl}(d) = \frac{1}{2}\delta_{kl}$ for $d \in \Gamma$. As linear discontinuous elements are used, the boundary Γ is considered smooth in this work.

3 Domain Integrals - DIBEM

DIBEM is used to calculate the domain integral from equation 1. The entire integrand is interpolated by radial basis functions. Since it is singular, a regularization must be applied:

$$I = \int_\Omega u_{kl}(x,d)\,b_l(x)\,d\Omega = \int_\Omega u_{kl}(x,d)\,(b_l(x) - b_l(d))\,d\Omega + \int_\Omega u_{kl}(x,d)\,b_l(d)\,d\Omega \tag{2}$$

and then it is interpolated:

$$u_{kl}(x,d)\,(b_l(x) - b_l(d)) = \sum_{m=1}^{M} \gamma_m f_m. \tag{3}$$

where γ_m are the unknown coefficients, f_m is the radial basis function (r^3 is used) and all boundary nodes and internal points, a total of M points, are used as interpolation points. Substituting this interpolation in equation 2 results in:

$$\int_\Omega u_{kl}(x,d)\,(b_l(x) - b_l(d))\,d\Omega = \sum_{m=1}^{M} \gamma_m \int_\Omega f_m d\Omega, \tag{4}$$

where the domain integral can be rewritten in polar coordinates as:

$$\sum_{m=1}^{M} \gamma_m \int_\theta \int_0^r f_m \rho d\rho d\theta, \tag{5}$$

where r is the value of ρ in a point of the boundary Γ. By defining $F_m(Q)$ as the integral in the radial direction and using the geometric relation $d\theta = \frac{\mathbf{n_i r_i}}{r} d\Gamma$, equation (2) can be reformulated as a boundary integral:

$$\sum_{m=1}^{M} \gamma_m \int_\Gamma \frac{F_m(Q)}{r} n_i r_i d\Gamma. \tag{6}$$

Applying equation (6) to each boundary node and internal point, the following equation, in matrix form, is obtained for the regularized integral:

$$\mathbf{I_r} = \mathbf{sF}^{-1} u_{kl}(x,d)\,(b_l(x) - b_l(d)). \tag{7}$$

where

$$\mathbf{s} = \left[\int_\Gamma \frac{F_1(Q)}{r}\mathbf{n}.\mathbf{r}d\Gamma \quad \int_\Gamma \frac{F_2(Q)}{r}\mathbf{n}.\mathbf{r}d\Gamma \quad \ldots \quad \int_\Gamma \frac{F_M(Q)}{r}\mathbf{n}.\mathbf{r}d\Gamma \right]. \tag{8}$$

Naming $\mathbf{sF^{-1}}$ as $[S_1 S_2 \ldots S_m]$ and considering the source and interpolating points as coincident, equation (7) can be rewritten for all collocation points in matrix form as:

$$\mathbf{I_r} = \begin{bmatrix} S_{11} & S_2 u_{kl}(X_2, X_1) & \ldots & S_M u_{kl}(X_M, X_1) \\ S_1 u_{kl}(X_1, X_2) & S_{22} & \ldots & S_M u_{kl}(X_M, X_2) \\ \vdots & \vdots & \ddots & \vdots \\ S_1 u_{kl}(X_1, X_M) & S_2 u_{kl}(X_2, X_M) & \ldots & S_{MM} \end{bmatrix} \begin{bmatrix} b_l(X_1) \\ b_l(X_2) \\ \vdots \\ b_l(X_3) \end{bmatrix} \tag{9}$$

where S_{ii} is minus the sum of all other terms in its line. To calculate the singular integral on the right-hand side of equation (2), RIM is employed. As $b_l(d)$ is constant, the integral depends only on the fundamental solution.

Therefore, the integration along the radial direction can be performed analytically, effectively reducing the domain integral to a boundary integral. This singular integral, calculated for each source point, is represented as:

$$\mathbf{I_s} = \begin{bmatrix} \int_\Omega u_{kl}(x, X_1)\, d\Omega & 0 & \dots & 0 \\ 0 & \int_\Omega u_{kl}(x, X_2)\, d\Omega & & \dots \quad 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \int_\Omega u_{kl}(x, X_M)\, d\Omega \end{bmatrix} \begin{bmatrix} b_l(X_1) \\ b_l(X_2) \\ \vdots \\ b_l(X_M) \end{bmatrix}. \tag{10}$$

Finally the domain integral is written as:

$$\mathbf{Mb} = \mathbf{I_r} + \mathbf{I_s}. \tag{11}$$

The body forces b_i in this work consists of the convective term $-u_j u_{i,j}$ and contains the velocity gradient $u_{i,j}$. The velocity gradient is calculated by interpolating the velocity using augmented radial basis functions:

$$u_i(P) = \sum_{m=1}^{M} \gamma_i^m f_m + a_i\, x(P) + b_i\, y(P) + c_i \tag{12}$$

and applying equation (12) to all source points, resulting in:

$$\mathbf{u_i} = \mathbf{F}\gamma_\mathbf{i}, \tag{13}$$

where F is a matrix that stores the radial basis functions at each point and γ are the unknown coefficients. γ is be computed as:

$$\gamma_\mathbf{i} = \mathbf{F^{-1}u_i}. \tag{14}$$

Equation (12) derivative is finally given by:

$$u_{i,j} = \mathbf{f}_{,j}\mathbf{F}^{-1}\mathbf{u_i} \tag{15}$$

4 Numerical implementation

Equation 1 can be represented, after discretization and considering each source point, in matrix form, as:

$$-\mathbf{Hu} + \mathbf{Gt} + \mathbf{Mb} = \mathbf{0}. \tag{16}$$

Applying boundary conditions results in:

$$\mathbf{Ax} = \mathbf{Cy} + \mathbf{Mb} = \mathbf{c} + \mathbf{Mb} \tag{17}$$

where $\mathbf{y}$ is known and $\mathbf{x}$unknown.

Once the matrices of the problem ($\mathbf{A}$, $\mathbf{M}$, and $\mathbf{c}$) are computed an iterative process is repeated until the velocity error criterion is satisfied. The solution vector ($\mathbf{x}$) in each step is calculated by:

$$\mathbf{x} = \mathbf{A}\backslash(\mathbf{c} + \mathbf{Mb}) \tag{18}$$

As this step is repeated hundreds of times, specially for large Re, a LU decomposition is applied to A and this decomposition is then used in each step. A relaxation factor (η) is used and is adjusted based on the Reynolds number (Re). Specifically, larger Re values necessitate a lower η:

$$\mathbf{x_{new}} = \mathbf{x_{new}}\eta + \mathbf{x_{old}}(\mathbf{1} - \eta) \tag{19}$$

Iterations continue until either the mean absolute velocity error falls below 10^{-6}.

4.1 Hierarchical matrices

The $\mathcal{H}$-matrix process starts by dividing the geometric domain as a tree data structure. This division creates, in each tree node, a cluster of points points that are close to each other. When the distance between a pair of collocation and control points is large enough, the block defined from those entries is considered admissible and

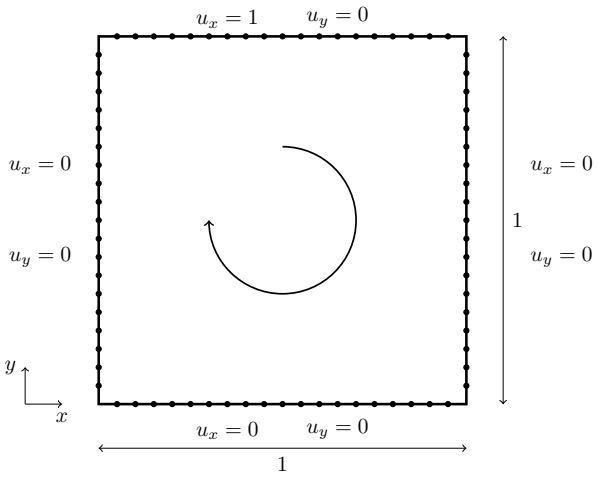


Figure 1. Geometry and boundary conditions of the Cavity case

is calculated by the Adaptive Cross Approximation and represented by a low rank approximation. The remaining blocks are generated in the usual BEM manner and are stored as full matrices. A detailed presentation of ACA can be found in [8] and [9].

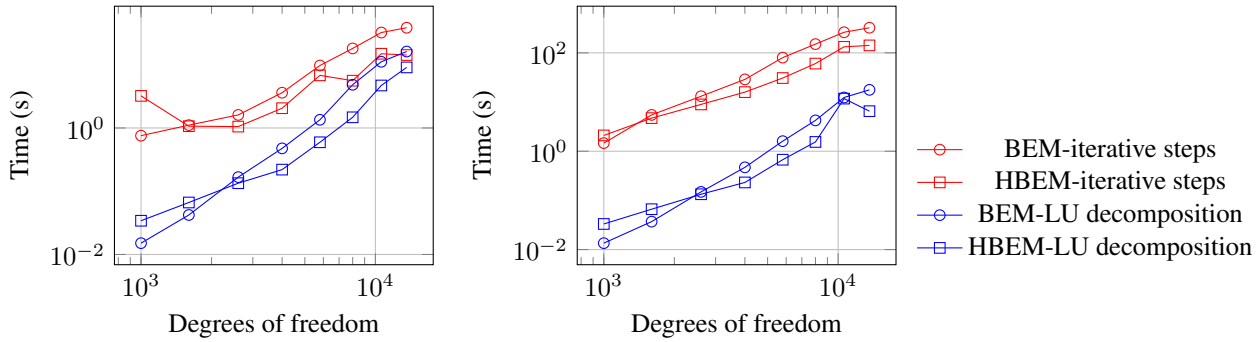
The savings in memory and time are due to the low rank approximations. This format needs less entries than a full matrix and its matrices operations are much faster. To get the full benefits of $\mathcal{H}$-matrices for storage and computation time, all final matrices must be represented this way. In this work, this applies to the matrix A, which represents the boundary integrals; matrix M, which represents the domain integrals; and matrices F_j which are used to calculate the velocity gradient. The LU decomposition of the $\mathcal{H}$-matrix A, as presented in [Hackbusch] is used in each step.

5 Numerical examples

To demonstrate the effectiveness of the proposed method, a lid-driven recirculation flow within a square cavity, as shown in Figure 1, is considered.

Figure 2 present the computational times for the iterative steps and the LU decomposition for $Re = 200$ and $Re = 300$, respectively. In both cases, the Hierarchical-BEM (HBEM) outperforms the standard approach in most discretizations. Only for the coarsest discretizations does this advantage diminish. This is expected, as the overhead of the hierarchical approach is more significant when the problem size is small.

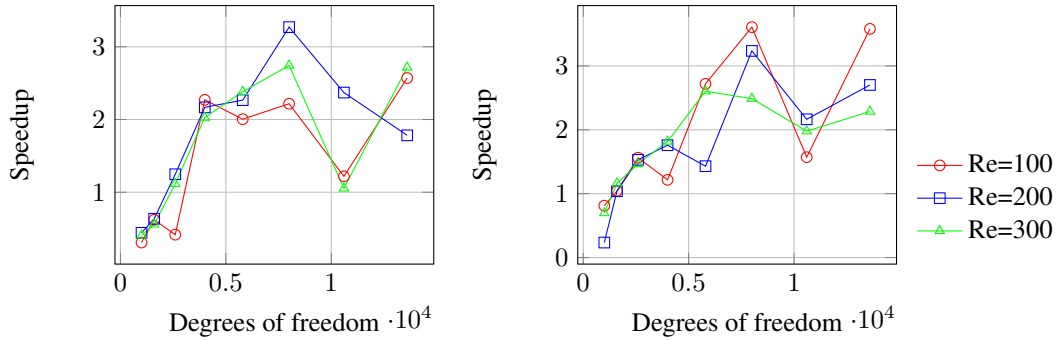
It can also be observed that, as Re increases, the iterative steps become more time-consuming relative to the LU decomposition, due to the larger number of required iterations. Furthermore, the slope of the LU decomposition time with respect to the discretization size is less steep for HBEM.



(a) $Re = 200$ (b) $Re = 300$

Figure 2. Elapsed time for the iterative steps and LU decomposition for two Reynolds numbers.

Figure 3 show the speed-up achieved by HBEM for the LU decomposition and the non-linear iterative procedure, respectively. In both cases, significant speed-up is observed for most discretizations, with peak values exceeding $3\times$.



(a) LU decomposition (b) iterative procedure

Figure 3. Speed-up in the LU decomposition and the non-linear iterative procedure

The ratio of uncompressed memory usage for the AA matrix between BEM and HBEM is presented in Table 1. As expected, the compression ratio increases with the discretization size. This highlights a key limitation of BEM when run on a standard PC: it eventually exhausts the available memory. By contrast, HBEM mitigates this issue, enabling the solution of substantially larger problems with the same hardware.

Degrees of freedom	Compression ratio
1000	2.779
1600	3.522
2600	4.933
4000	6.325
5800	8.753
8000	10.677
10600	13.849
13600	14.648

Table 1. Compression ratio for different discretizations.

6 Conclusion

The results confirm that the Hierarchical-BEM consistently outperforms the standard approach in memory and time consumption for most discretizations. In non-linear problems, these advantages become even more significant, as the linear system must be solved multiple times and the performance gains are effectively multiplied.

References

[1] C. F. Loeffler, A. L. Cruz, and A. Bulcao. Direct use of radial basis interpolation functions for modelling source terms with the boundary element method. *Eng. Anal. Boundary Elem.*, vol. 50, pp. 97–108, 2015.
[2] C. F. Loeffler and W. J. Mansur. A regularization scheme applied to the direct interpolation boundary element technique with radial basis functions for solving eigenvalue problem. *Engineering Analysis with Boundary Elements*, vol. 74, 2017.
[3] L. A. d. Almeida, L. S. Campos, de E. L. Albuquerque, and T. R. Nicolaidis. A Direct Interpolation Boundary Element Method (Dibem) Approach for Incompressible Newtonian Flow. SSRN, 2025.
[4] S. Börm, L. Grasedyck, and W. Hackbusch. Introduction to hierarchical matrices with applications. *Eng. Anal. Boundary Elem.*, vol. 27, n. 5, pp. 405–422, 2003.
[5] Y. Liu. *Fast multipole boundary element method: theory and applications in engineering*. Cambridge university press, 2009.

[6] K. Kitagawa, C. A. Brebbia, L. C. Wrobel, and M. Tanaka. Boundary element analysis of viscous flow by penalty function formulation. *Engineering Analysis*, vol. 3, n. 4, pp. 194–200, 1986.
[7] J. H. Kane. *Boundary Element Analysis in Engineering Continuum Mechanics*. Pearson College Div, 1994.
[8] M. Bebendorf. *Hierarchical matrices: a means to efficiently solve elliptic boundary value problems*, volume 63. Springer, 2008.
[9] S. Kurz, O. Rain, and S. Rjasanow. Fast Boundary Element Methods in Computational Electromagnetism. In *Boundary Element Analysis*, pp. 249–279. Springer, 2007.
[Hackbusch] W. Hackbusch. *Hierarchical Matrices: Algorithms and Analysis*. Springer, Berlin, Germany.