2D Fast Multipole Boundary Element Method Algorithm Implementation and Analysis on CAE

Juan C Gutiérrez Urrego
Universidad EAFIT
Colombia
jcgutierru@eafit.edu.co

Simón Marín
Universidad EAFIT
Colombia
smaring1@eafit.edu.co

Mauricio Toro
Universidad EAFIT
Colombia
mtorobe@eafit.edu.co

ABSTRACT
This paper provides an analysis of the Fast Multipole Boundary Element Method, and a comparison with the classical Boundary Element Method and the Finite Element Method in terms of algorithm complexity in time and memory, advantages, and disadvantages of them in Engineering applications.

The algorithm proposed to be programmed was the Fast Multipole Boundary Element Method, as it powers up the classical BEM, reducing its complexity and making possible to solve large scale problems without needing a supercomputer.

Keywords
Boundary Element Method, Fast Multipole Method, Algorithm Complexity, Computed aided engineering

1. INTRODUCTION
Nowadays, numerical methods for solving initial value problems in engineering such as structural, fracture and fluid mechanics, heat transfer and electromagnetism for example, are one of the main tools that an engineer must approximate to large scale solutions. As a result, studying time and memory complexity of these algorithms and methods is a constant concern for us. In this paper, I will analyze and discuss, a couple of algorithms that are used to solve this kind of problems and will compare their results.

1.1. Problem
In this semester, I will be analyzing some engineering 2D plane stress problems by modeling and running them with different numerical methods/algorithms, to compare and conclude about the advantages and disadvantages that comes with them in terms of time and memory complexity, and precision.

1.2 Solution

The algorithm of interest, and the chosen one to be developed is the Fast Multipole Boundary Element Method, as it has the advantages of the initial discretization and the complexity improvement of the Fast Multipole Method.

1.3 Article structure

In what follows, in Section 2, will be presented the related work to the problem. Later, in Section 3, the data sets and methods used in this research. In Section 4, the algorithm design. After, in Section 5, the results. Finally, in Section 6, will be discussed the results and some future work directions will be proposed.

2. RELATED WORK

In what follows, I will explain four related works on the domain of engineering numerical methods and Fast Multipole method implementation in one of them.

3.1 Finite Element Method
This article widely explains the process that a problem goes through when you plan to apply FEM on an engineering problem. From it, can be concluded that this method is the most used nowadays for solving engineering problems, due to its versability and as it is capable of being used in non-linear problems. But on the other hand, it needs a lot of time and memory to create the mesh needed to solve the differential equations formulation. This mesh needs to cover the volume for 3D problems, and the entire area for 2D problems, this observation is made for a later comparison with the other methods.

This article also shows and explains that the first requirement of any analysis is the selection of an appropriate mathematical model to represent the physical problem; as well an example in a cantilever beam with a whole. [2]

3.2 The fast multipole boundary element method for potential problems: A tutorial
This paper is an introduction to the Fast Multipole BEM for potential problems, first explaining how the Boundary Integral Equations are responsible of these applications and bringing some historical context to the reader. It points out the fact that the classic boundary element method owes its fame to the easy meshing process, as for solving problems only meshing the boundary is required, reducing 3D problems to 2D surface faces, and 2D problems to 1D borders of the object.

Also, the conventional boundary element equations are shown, and the way the classic BEM works is exposed and explained, to finally go through the fast multipole method formulation where the complexity reduction comes in, basically because the solver works with an **octree-quadtree** initial discretization, and iterative solvers do not need to store the entire matrix in memory. It finally shows a graphic where the difference in CPU time for FMMBEM and BEM can be clearly seen. [3]

3.3 The Fast Multipole Method: Numerical Implementation

This paper explained the fundamentals of fast multipole method, and how it is one of the most efficient methods used to perform matrix-vector products and accelerate the resolution of the linear system. In which a problem of N degrees of freedom may be solved in CN^{iter} N log N floating operations, whew C is a constant depending on the implementation of the method. The main advantage of this method is that implies a lower total CPU time and a large range of applications, as I will discuss later with the Boundary Element Method. The paper in my opinion is dense but covers a lot of theorical explanation and equation examples of this method's application. [4]

3.4 A new fast multipole boundary element method for solving large-scale two-dimensional elastostatic problems.

This paper shows in a more detailed way the implementation of the Fast Multipole Method and compares results of stress and displacement between BEM and FEM, in order to show the precision between the methods.

The main idea of this article is to explain another way to implement Fast Multipole Method equations in BEM problems written in complex form. It shows some time complexity comparations and some numerical examples implementing this method. [5]

3. MATERIALS AND METHODS

In this section, will be described the different stress-deformation problems that were used to compare the results of the implemented methods. And so, the software that was used to execute these methods.

3.1 Data Collection and Processing

To compare the different alternatives, three problems were defined for testing them. To completely define a mechanical problem of interest for this application, the thing that must be defined are: The geometry of the problems, loads, and material properties.

The material used for all developed solutions was a 2014-T6 aluminum alloy, with an Elastic modulus (E) equal to 72400 N/mm², a Shear modulus (G) equal to 28000 N/mm² and a Poisson's ratio (Nu) equal to 0.33.

The first problem formulation will be a cantilever beam, with a 100 N/mm² distributed force at the free end of it, with a length of 200mm, a height of 20mm and a base of 1mm as for the 2D simplification it needed to be unitary.

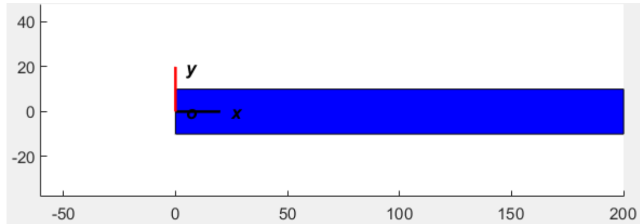

Figure 1. Cantilever Beam

The second one will be a semicircular tunnel, with a uniformly radial distributed pression of 100 N/mm², a radius of 15m and a thickness of 3m, and the displacement in the base will be zero.

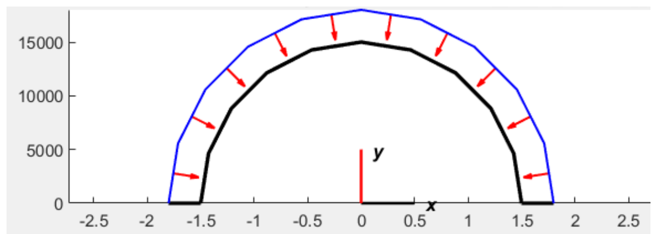

Figure 2. Tunnel

For the last problem, I will define a cylindrical tank, and the analyzed part will be a cross-section showing the circular shell, which will have a uniformly radial distributed pression of 250 N/mm², a radius of 5 meters, and a thickness of 5 centimeters.

\

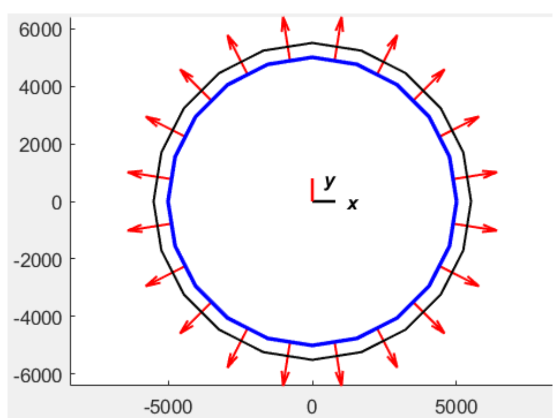

Figure 3. Tank cross-section

3.2 Numerical method alternatives

In what follows, I will present different algorithm/ methods used for solving the problems.

3.2.1 Finite Element Method

For this method I used the very known software SolidWorks, which brings a high reliability as it is a commercial software used and recommended around the world. For solving the 2D problems, I will be using the 2D simplification option that SolidWorks provides, and that I am familiar with because of my past work using this method.

3.2.2 Indirect Boundary Element Method

For this method I will be using a software developed by the Professor Juan Diego Jaramillo on FORTRAN, which was brought to us his students for a first approaching to the BEM method. As this algorithm needs a first problem “handmade” parametrization, I will be using MATLAB and Excel as well to define all these parameters, such as, material properties, nodes, type of nodes, loads, etc.

3.2.3 Boundary Element Method

For this method I developed my own code in Python applying the Elastodynamic representation theorem and Green’s functions and will be analyzing the time and memory complexity that it gets.

3.2.4 Fast Multipole Boundary Element Method

I will be developing this software for my next work, as I might not have enough time to complete it, but I will compare theoretical results and will explain and work on the process to achieve it.

3.3 Analytical Solution

For two of these three problems, the analytical solution is available on academic records, so it will be possible to compare the numerical methods and algorithms against the goal solution, but for most cases in engineering, there is not such a thing as the analytical solution for a complex problem, basically that is why numerical solutions are used. So, I will be comparing for the other case the approximations between the methods themselves.

4. ALGORITHM DESIGN AND IMPLEMENTATION

In what follows, we explain the data structures and the algorithms used in this work. The implementations of the data structures and algorithms are available at GitHub[1].

4.1 Data Structures

For this work, I will use mainly one type of data structure, static vectors, as for the Boundary Element Method, the solution for the displacement field is achieved by solving a system of equations with 2xN unknowns, where N is the number of nodes that the problem has. See figure_.

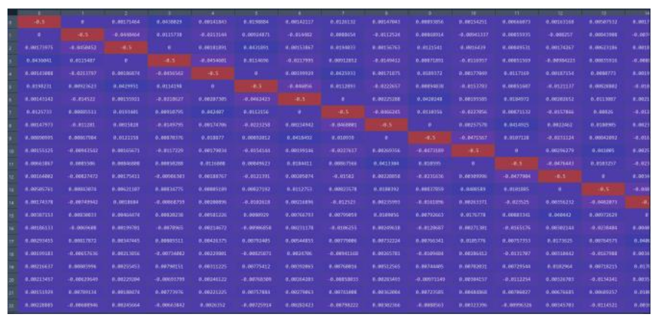

Figure 4. Dense matrix example of BEM

Also, I will be working in an Quadtree-Octree code, which will later be used in the Fast Multipole formulation. See figure_.

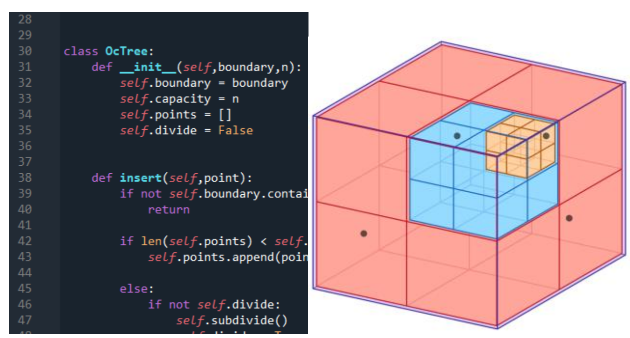

Figure 5. Part of Octree code and Octree discretization example

[1]https://github.com/Jcgutierru/ST0245-001/tree/master/proyecto

4.2 Algorithms

In this work, I propose a compression algorithm which is a combination of the Fast Multipole Method and the Boundary Element Method.

In this semester, one of the algorithms that I will study, and implement will be the classic Boundary Element Method (BEM), which is a semi-analytical method for solving problems of many fields, and the second one will be the Fast Multipole Boundary Element Method (FMBEM), which main goal is to reduce the classic BEM complexity to make it a more competitive method.

4.2.1 Boundary Element Method algorithm

For this method I needed some theoretical explanation of some continuum mechanics concepts and applications which I learned this semester in the BEM course.

This method is a semi-analytical solution for many mechanical problems, as it uses Boundary Integral Equations, so it achieves great precision. The main advantage of this method is the domain discretization dimension reduction and its applications on infinite domains.

To implement the algorithm is based on the Elastodynamic representation theorem:

$$U_{ij}(\xi) = \int_S G_{ij}(x,\xi) * t_i(x) * ds(x) - \int_S H_{ij}(x,\xi) * u_i(x) * ds(x) + \int_V G_{ij}(x,\xi) * P_i(x) * ds(x)$$

To implement this algorithm, Green's functions of G_{ij} and H_{ij} in 2D were also used:

$$H_{ij}^{2D} = \frac{-1}{4\pi G(1-\nu)R^2}\left[R_k n_k\left((1-2\nu)\delta_{ij} + \frac{2r_i r_j}{r^2}\right) + (1 - 2\nu)(r_i n_j - r_j n_i)\right]$$

$$G_{ij}^{2D} = \frac{1}{8\pi G(1-\nu)}\left[(3-4\nu)\delta_{ij}\ln\left(\frac{1}{r}\right) + \frac{r_i r_j}{r^2}\right]$$

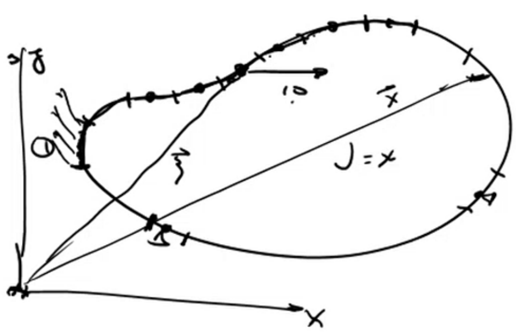

Figure 6. Body segmentation for BEM example

The algorithm works by finding the known and unknown terms of the problem. For solving any problem, one needs the geometry of it, the material, and the loads or displacements in each direction, even if they are equal to zero.

Then the system of equations is assembled by adding all the terms and leaving the unknowns with their respective coefficient, so the Ax = b lineal problem is solved by the program, and finally the displacement and stress field of the points of interest is calculated applying the same elastodinamic theorem.

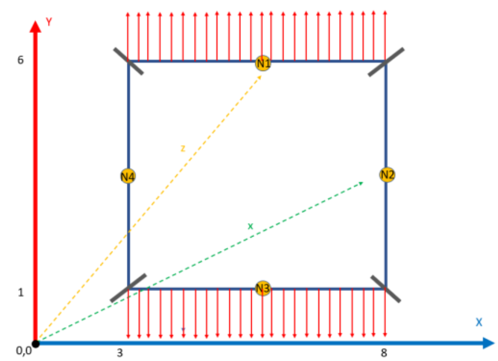

Figure 7. Four node simplified BEM discretization for system of equation initial analisis

4.2.2 Fast Multipole Boundary Element Method algorithm

The fast multipole method is applied using the same equations as in BEM, but the main idea is applying iterative solver (GMRES) and accelerate matrix-vector multiplications by replacing element-element interactions with cell-cell interactions.

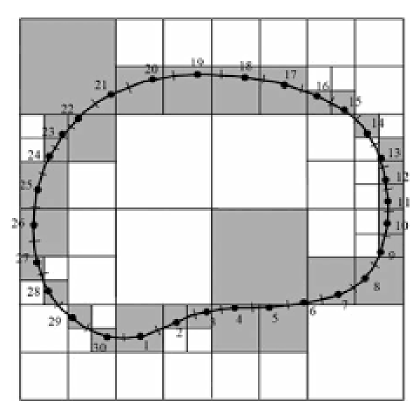

Figure 8. Fast Multipole BEM Octree parametrization

The point of this algorithm is to accelerate the matrix-vector product using these solvers and by using the Octree data structure to do the multiplication (Aλ) in each iteration, without ever forming the entire matrix A explicitly.

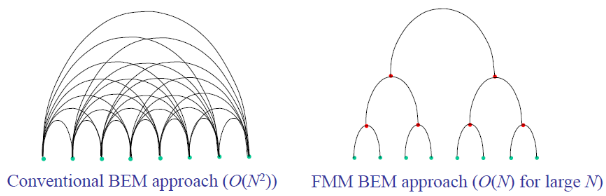

Figure 9. BEM vs FMBEM approach comparison

As can be seen in figure 9, the comparison for this system of equations is reduced, and as it is aimed to be used in large scaled nonsymmetrical matrices, the precision trends not to be importantly affected.

4.3 Complexity analysis of the algorithms

For programming the classic BEM in Python, I used squared matrices that depended on the number of nodes that my domain has, and then I needed to assemble the system of equations which has two unknowns per node. Then this NxN matrix needs to be solved by against the independent vector to find the unknowns; this step raises the complexity of the problem.

Algorithm	Time Complexity
Boundary Element Method	$O(N^2)$
Fast Multipole Boundary Element Method	O(N)

Table 1: Time Complexity of the BEM and FMBEM, where N is the number of nodes that conform the boundary.

Algorithm	Memory Complexity
Boundary Element Method	$O(N^2)$
Fast Multipole Boundary Element Method	O(N)

Table 2: Memory Complexity of the BEM and FMBEM, where N is the number of nodes that conform the boundary.

4.4 Design criteria of the algorithm

The classic Boundary Element Method that I programmed in Python was mainly stored in static vectors, in order to always insert data in O(1) instead of O(N). Also, I used the least possible python functions for which I did not know their complexity, so I would not increase it by mistake.

5. RESULTS

5.1 Model evaluation

5.1.1 Evaluation on tank data set IBEM vs BEM

In what follows, I will present the evaluation metrics for the tank data set in Table 3.

	Training data set
% Error u1	0.026031%
% Error u2	0%
% Total error	0.026031%

Table 3. Boundary Element Method vs Finite Element Method.

5.1.2 Evaluation on tank data set FEM vs BEM

In what follows, I will present the evaluation metrics for the testing dataset in Table 4 without compression and, in Table 5, with compression.

	Testing data set
% Error u1	11.7120%
% Error u2	12.5796%
% Total error	12.5796%

Table 4. Binary image-classification model evaluation on the testing data set without image compression.

6. DISCUSSION OF THE RESULTS

The main advantages that I would point out are that the classic and fast multipole BEM, are important methods that nowadays are not as used as they should in terms of precision and infinite domain applications. The simplification and discretization dimension reduction implies that large scale problems not only would be solved faster, but also its initial partition will be much easier and with less memory demands.

6.1 Future work

I am planning to implement the Fast Multipole Boundary Element Method for other engineering applications such as heat transfer, fluid dynamic or large scale problems related with porous materials and explore the potential of this algorithm for modern problems.

ACKNOWLEDGEMENTS

This work was supported by Prof. Juan Diego Jaramillo, as he was my Boundary Element Method instructor. Also partially supported by Prof. Oscar Eduardo Ruiz, as he is my laboratory supervisor.

I thank them both, and all the class assistants that helped me through the semester to understand the course topics.

REFERENCES

2. Klaus-Jürgen Bathe. 2008. Finite Element Method. (June 2008). Retrieved May 17, 2021 from https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470050118.ecse159

3. Y.J. Liu and N. Nishimura. 2006. The fast multipole boundary element method for potential problems: A tutorial. (March 2006). Retrieved May 17, 2021 from https://www.sciencedirect.com/science/article/abs/pii/S0955799706000105

4. Eric Darve. 2002. The Fast Multipole Method: Numerical Implementation. (May 2002). Retrieved May 22, 2021 from https://www.sciencedirect.com/science/article/pii/S0021999100964519

5. Yijun Liu. 2005. A new fast multipole boundary element method for solving large-scale two-dimensional elastostatic problems. (September 2005). Retrieved May 17, 2021 from https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1474

6. Yijun Liu 2009. Fast Multipole Boundary Element Method. Theory and Applications in Engineering. Cambridge University Press, 2009-08-24.