COBEM 2023 27TH INTERNATIONAL CONGRESS OF MECHANICAL ENGINEERING

ABCM Brazilian Society of Mechanical Sciences and Engineering
27th ABCM International Congress of Mechanical Engineering
December 04-08, 2023, Florianópolis-SC, Brazil

COB-2023-2081
QUADTREE IMPLEMENTATION FOR H-ADAPTIVE MESH REFINEMENT: A CASE STUDY IN MULTI-DOMAIN PROBLEMS

Gabriel Nunes
Elisan dos Santos Magalhães
Guilherme Borges Ribeiro
Aeronautics Institute of Technology (ITA), São José dos Campos-SP
nunesgn@ita.br, elisan@ita.br, guilherme.ribeiro@gp.ita.br

Abstract. *This study presents an implementation of a quadtree-based, h-adaptive algorithm for solving 2D thermal conduction problems involving multiple materials. The algorithm employs Python and the Finite Volume Method (FVM) to dynamically refine and coarsen the computational mesh, improving the efficiency and accuracy of the solution. This approach addresses the challenges associated with computational cost and localization of features in material interface problems. The quadtree structure is used to efficiently manage the hierarchical subdivisions of the domain, allowing for a high degree of local refinement where needed, such as at material interfaces, while keeping coarser grids elsewhere. The quadtree structure is navigated via cardinal direction-based functions to identify neighboring elements and allow for the application of boundary conditions. In this work, we benchmark our algorithm against analytic solutions for one-dimensional problems and observe improved error convergence with increased levels of refinement. For the two-dimensional problem, we propose a setup involving a plate with two subdomains and provide a detailed procedure to analyze the resulting temperature field which obtained solution was successfully validated using a commercial software. Refined meshes outperformed base-level uniform meshes in aligning with asymptotic values. The computational demand of the solver is predominantly driven by element count and represents the most significant effort when compared to the preceding data structuring operations. However, the study provides insights into the restructuring of the quadtree, indicating that this operation might become a bottleneck in inherently transient problems. Overall, integrating mesh adaption with quadtrees delivers a robust solution, underscoring its potential for advanced research in transport phenomena.*

Keywords: *Quadtree, h-Adaptive Mesh, Refinement, Multi-Domain Problems, Case Study*

1. INTRODUCTION

Numerical solutions to complex physical phenomena often involve tackling nonlinear, time-dependent equations. In many instances, these problems manifest intrinsic discontinuities, especially when different materials with varied thermal properties intersect (Peng *et al.*, 2013). The proper and efficient handling of these discontinuities is paramount. Failing to do so can jeopardize both the accuracy and stability of numerical outcomes. Conventional approaches, reliant on static mesh-based methods, frequently grapple with these complexities, leading to extensive computational demands. In this landscape, Adaptive Mesh Refinement (AMR), underpinned by appropriate data structures, emerges as a promising contender, offering notable advantages over traditional techniques (Ainsworth and Oden, 1997).

Heat transfer is undeniably foundational in a myriad of scientific domains, ranging from climate modeling to geothermal energy harnessing and even advanced material processing (Bejan, 2013, 2016). At its core, heat transfer navigates the controlled movement of thermal energy, governed by three predominant mechanisms: conduction, convection, and radiation. Mathematically, these mechanisms materialize in the form of partial differential equations (PDEs), which also play an instrumental role in defining governing equations for mass, momentum, and energy conservation (Bird, 2002). When multi-domain interplays are introduced, especially where material properties exhibit spatial variability, the intricacy of heat transfer problems intensifies (Incropera *et al.*, 1996).

Several numerical strategies, encompassing the finite volume, finite element, and finite difference methods, have been crafted and refined over time to navigate these intricate challenges (LeVeque, 2002). Among these, AMR techniques stand out, particularly due to their proficiency in accurately capturing and representing the abrupt transitions associated with changing material properties.

AMR's strength lies in its adaptive nature. It efficiently refines computational meshes in regions with pronounced gradients or inherent solution complexities, ensuring precision where it's most required (Berger and Oliger, 1984; Berger and Colella, 1989). The evolution of AMR has led to specialized variants, including h-AMR and p-AMR. These adaptations provide the added flexibility of modulating mesh sizes or polynomial degrees based on problem demands, allowing for superior local resolution and optimizing computational resources in areas of lesser criticality (Ainsworth and Oden, 1997).

Conclusively, the Finite Volume Method (FVM) is a stalwart in numerical modeling, adept at resolving PDEs, and shines particularly bright in multi-domain scenarios encompassing both heat transfer and fluid dynamics (Patankar, 2018; Moukalled *et al.*, 2016; Versteeg and Malalasekera, 2007). The FVM's emphasis on fluxes across control volume boundaries not only assures robustness but also guarantees precision—a dual attribute that positions it as a preferred method in Computational Fluid Dynamics (CFD) studies and intertwined disciplines (Moukalled *et al.*, 2016).

The domain of numerical solutions, especially in heat transfer, constantly grapples with the intricacies of multi-domain challenges, especially where intricate discontinuities are at play. The present work has underscored the immense value of Adaptive Mesh Refinement (AMR) techniques in addressing these challenges. By implementing a quadtree-based, h-adaptive algorithm tailored for 2D thermal conduction problems, this study introduces a novel approach that combines the rigor of the Finite Volume Method (FVM) with the dynamism of adaptive mesh refinement. Notably, the employment of the quadtree structure showcases a striking balance between computational efficiency and solution accuracy, particularly in areas of pronounced solution complexities. By benchmarking against both analytical solutions and outputs from commercial software, this study offers an authoritative validation of its methodological advancements. Furthermore, the insights gleaned regarding the potential bottlenecks of the quadtree structure in transient problems underscore the need for ongoing refinement in this direction. In essence, the contribution of this research to the landscape of numerical solutions is twofold: it not only accentuates the efficacy of AMR techniques in handling complex heat transfer problems but also spotlights the potential of quadtrees as a powerful tool for future computational endeavors in the realm of transport phenomena and beyond.

2. METHODOLOGY

2.1 Energy Transport in Multi-Domain Solids

The balance of energy in a continuous medium can be seen in Bejan (2013) and equated using Einstein notation as following:

$$\frac{\partial(\rho e)}{\partial t} + \frac{\partial(\rho e u_i)}{\partial x_i} = \frac{\partial}{\partial x_i}\left(k\frac{\partial T}{\partial x_i}\right) + q''' - p\frac{\partial u_i}{\partial x_i} + \mu\phi \tag{1}$$

where ρ is the material density, e is the specific internal energy, u_i is a component of the velocity field parallel to one of the cartesian coordinate axes, T is the temperature scalar field, k is the thermal conductivity, q''' is the volumetric heat generation (e.g. those that occurs due to exothermic chemical reactions), p is the pressure, μ is the dynamic viscosity, and ϕ is a dissipation function due to viscous effects. The above partial differential equation is function of all cartesian coordinates in space x_i and time t. Repeated indices indicate a summation in this notation.

In the context of multi-domain solid materials, it's important to note that all properties are considered to be dependent on space, time and temperature, except when otherwise noted and also that the eq.(1) can be simplified omitting the velocity terms, the pressure work and viscous dissipation function since their effects are not as influential in the physics of solid problems as they are for fluids. Then we get:

$$\frac{\partial(\rho e)}{\partial t} = \frac{\partial}{\partial x_i}\left(k\frac{\partial T}{\partial x_i}\right) + q''' \tag{2}$$

Assuming that the density, ρ, remains constant over time and space, and that the internal energy, e, is exclusively a function of temperature, T, the relationship between e and T can be described using the specific heat, denoted as c. Specifically, this relationship is given by $\frac{\partial e}{\partial T} = c$

Under these assumptions, the transient term can be rewritten in the equation as:

$$\rho c\frac{\partial T}{\partial t} = \frac{\partial}{\partial x_i}\left(k\frac{\partial T}{\partial x_i}\right) + q''' \tag{3}$$

Such simplifications are often employed when analyzing heat conduction in solids and incompressible fluids where volume changes with temperature are minimal. For example, in "Heat Conduction" by Hahn and Özisik (2012), it is illustrated that assuming constant properties provides reasonable solutions for many practical cases involving metals, ceramics, and some liquids. The term c represents the specific heat of the material, and it describes the material's ability to store thermal energy. Essentially, it defines how much energy is required to raise the temperature of a unit mass of the material by one unit of temperature.

When dealing with a multi-domain material, where properties vary according to space, the modelling becomes more intricate. The material is divided into distinct domains or regions, each characterized by its unique set of properties. Each

domain is modeled separately, and appropriate boundary conditions are applied where domains meet and it is crucial to ensure continuity. For heat conduction problems, this often means ensuring both the temperature and the heat flux are continuous across domain boundaries.

$$T_1 = T_2 \quad \text{and} \quad k_1 \frac{\partial T_1}{\partial n} = k_2 \frac{\partial T_2}{\partial n} \tag{4}$$

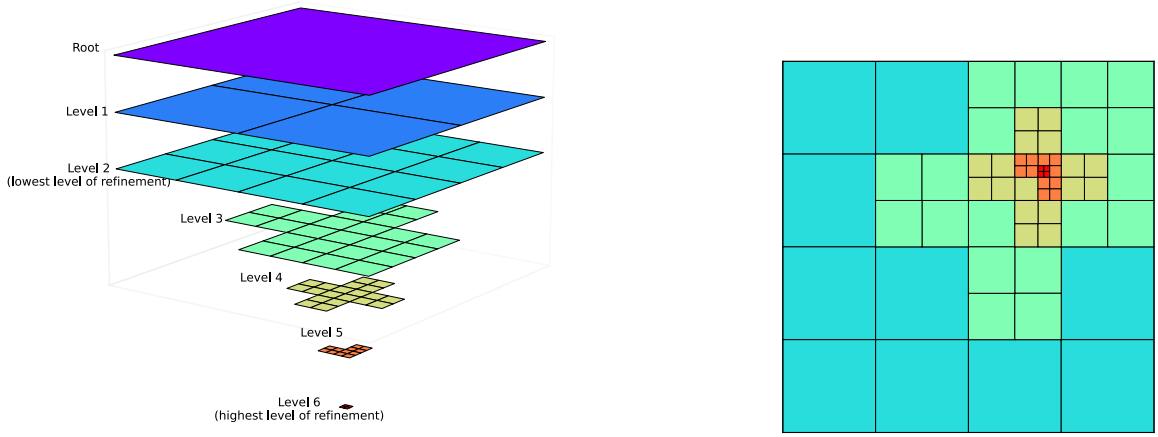
where n is the normal direction to the boundary.

2.2 Quadtree as Hierarchical Adaptive Mesh Refinement Framework

Quadtree is a dynamic data structure utilized in various computational fields, from image processing to computational fluid dynamics (Samet, 1984). This structure is based on the principle of recursive spatial subdivision, where a computational domain starts with a root node that divides into four child nodes, one in each intercardinal points: NW, NE, SW, and SE, explaining its name "Quadtree".

This structure's adaptability is what sets it apart, facilitating precise mesh refinement in areas with significant gradients or complex behaviors, continuing until reaching set depth or error thresholds (Finkel and Bentley, 1974). The hierarchical nature of Quadtree manages heterogeneity in two-dimensional spaces efficiently. Every node represents a rectangular subdomain, with the parent-child relationship conveying spatial containment details, enhancing the tree's spatial sensitivity crucial for adaptive simulations (Zlatanova *et al.*, 2002), with the root encapsulating the entire domain. A visualization of this can be found in Fig 1a.

Within the finite volume method (FVM) framework, Quadtree proves invaluable. Nodes discern their neighbors, enhancing flux computations, while the leaf nodes, shown in Fig. 1b, house key thermodynamic details crucial for thermal analysis.



(a) (b)

Figure 1: Mesh generation and its (a) hierarchical representation of the Quadtree, with each level illustrating adaptive refinement from the root (entire domain) down to localized regions. (b) resulting elements, depicted as leaves from the tree, with colors corresponding to their respective refinement levels.

Utilizing Python's simplicity and object-oriented programming (OOP) support, this study establishes a Quadtree-based discretization system. The setup comprises two Python classes: Node and Quadtree. The Node class embeds a domain rectangle and its thermal features, whereas Quadtree directs the tree's structure and refinement processes.

To initiate the Quadtree, the structure requires three parameters: domain size, and the minimum and maximum levels, dictating recursion depth and computational efficiency. The domain is initially segmented based on the lowest level, setting the basic computation resolution. Refinement is guided by thermal conductivity variations, optimizing computational resource distribution.

Adaptive Mesh Refinement (AMR) techniques, crucial for enhancing computational simulations, often employ h-adaptivity due to its ease of implementation and significant impact on precision and efficiency (Ainsworth and Oden,

1997). This approach varies element sizes based on the intricacy of the physical phenomena: areas with rapid changes use smaller elements, while static regions use larger ones (Oden and Reddy, 2012; Babuška and Suri, 1994).

Refinement areas are determined by error estimators, with various types existing, such as residual-based or gradient-based (Babuška and Miller, 1987). Posteriori error estimates are widely preferred for their dependability (Verführt, 1996). To maintain accuracy, h-adaptivity is used iteratively, adjusting the mesh after each computation until an error is under a set threshold or the solution stabilizes.

Programmatically, h-adaptivity's success hinges on effective mesh data structure management. Tree-based structures, like Quadtree for 2D or Octree for 3D, are favored for their hierarchical adaptability (Samet, 2006). In the presented framework, h-adaptivity is integrated to dynamically modify the Quadtree grid, focusing on thermal gradients between control volumes. Different interpolation schemes, based on neighboring nodes, help estimate these gradients, guiding h-adaptive refinement.

The process involves three steps: coarsening, refinement, and smoothing (also termed balancing). For time complexity analysis, we will use the Big O notation. In this context, let n represent the number of elements in the tree.

1. **Coarsening**: In this step, oversized leaf nodes (those that were previously refined but no longer meet the refinement criterion) are coarsened, or combined back into a larger parent node. This process helps to eliminate unnecessary computational resources where the solution is relatively uniform. Since this operation runs through all leaves performing operations of constant time, the time complexity is $O(n)$.

2. **Refinement**: The refinement step involves subdividing leaf nodes that meet the criterion for further refinement. This is typically based on some measure of error or variation in the physical property of interest, in this case, the thermal conductivity. By the same reason this operation have also time complexity of $O(n)$.

3. **Smoothing /Balancing**: This step is designed to ensure the smooth transition between different levels of refinement in the grid. After the refinement process, it is possible that some leaf nodes have more than two neighboring nodes, causing an irregular distribution of nodes, with large elements adjacent to much smaller ones. To prevent this condition, the smoothing step is implemented. If a leaf node has more than two neighboring nodes, the node is further refined. This ensures that the gradient evaluation schemes of the Finite Volume Method (FVM) work accurately and provides a smoother transition between refined and coarsened regions. The smoothing operation processes every element at least once. Additionally, due to the neighbor search mechanism for each leaf it leads to a time complexity that surpasses $O(n \log_4(n))$.

These steps constitute a cycle that play a pivotal role in fine-tuning computational simulations. As the simulation progresses, the grid dynamically adapts, directing computational resources where they are most beneficial. By prioritizing areas with significant solution variations or high interest, the simulation's overall efficiency and accuracy are amplified.

The mechanism for finding neighbors within the quadtree pivots on using cardinal positions (North, South, East, West) as identifiers for each node. When seeking the northern neighbors of a node:

1. If the node is in its parent's southern half (South-East or South-West), the algorithm checks the parent's northern counterpart. If that's a leaf, it becomes a neighbor. Otherwise, the process iterates through its southern descendants to locate leaf neighbors.Notably, this operation is mostly constant in time, leading to a complexity of $O(1)$.

2. If the node is in its parent's northern half (North-East or North-West), the algorithm climbs the tree until it finds a parent in the southern half of its parent. During this ascent, the traversed path is stored. The algorithm then descends, inverting the stored path's directions, reaching an equivalent depth on the northern side, and collects the southern leaves as the node's northern neighbors. Given the potential tree traversal, the worst-case time complexity for this operation is $O(\log_4(n))$.

This streamlined traversal strategy, in tandem with the quadtree's hierarchy, optimally manages local adaptivity in multi-domain challenges, offering computational efficiency.

2.3 Discretization in Finite Volume Method

In the section 2.1 we obtained the expression 3 for heat conduction in a solid, considering some simplifying hypotheses for materials with multiple regions. We also discussed the compatibilizing equations for the interface so that the solution does not present discontinuities in the solution field or in its spatial derivatives (equations 4). Now let's discretize these equations for the FVM.

To do this, the differential equation 3 is integrated over the finite volume of an element P contained in the mesh that represents the physical domain of the problem. Gauss's theorem is applied to the divergent term, which transforms the domain integral into a surface integral, which will be evaluated on the four faces of our rectangula element. The internal generation term can be approximated as a constant throughout the domain, as can the time derivative. To evaluate the time derivative, finite difference is applied, so the equation can be written as found in Versteeg and Malalasekera (2007):

$$\rho c V \frac{\Delta T_P}{\Delta t} = k_e \frac{\partial T_e}{\partial x} A_e - k_w \frac{\partial T_w}{\partial x} A_w + k_n \frac{\partial T_n}{\partial y} A_n - k_s \frac{\partial T_s}{\partial y} A_s + q''' V \tag{5}$$

The temperature within the control volume, T_P, symbolizes the current state, whereas T_P^O denotes its state from the previous timestep. The change in temperature, depicted as $\Delta T_P = T_P - T_P^O$, over an interval Δt, characterizes the transient behavior of heat conduction. Central geometric quantities in this context are the area A, evaluated at the boundaries, and the volume V of the element. Diving into the diffusive terms, they represent the heat flux across the boundaries of the control volume. This flux arises due to temperature gradients, with the spatial derivative, thermal conductivity k, and area A all being assessed at the boundaries of the current element. The lower case subscripts emphasize their evaluation specifically at these boundaries, elucidating the local directional influence on heat transfer. Furthermore, it's crucial to note that the signs associated with the w and s terms have been adjusted to account for the specific directionality of the area, ensuring accurate representation of heat flux across these boundaries.

As for the evaluation of the diffusive term, in a conventional structured rectangular mesh, a central difference is commonly used (Hirsch, 2007), which can be extracted from a linear interpolation between neighboring terms. On more sophisticated meshes, this evaluation needs to be either corrected or approximated by another approach. The advantage of using a smoothing mechanism in adaptive refinement with quadtrees is that it limits the type of neighborhood that is obtained in the final result, and this made it possible to work on evaluating the derivative of T in relation to the normal direction by limiting it to three situations, which are illustrated in figure 2. Otherwise, multiple situations with even greater level differences should be considered.

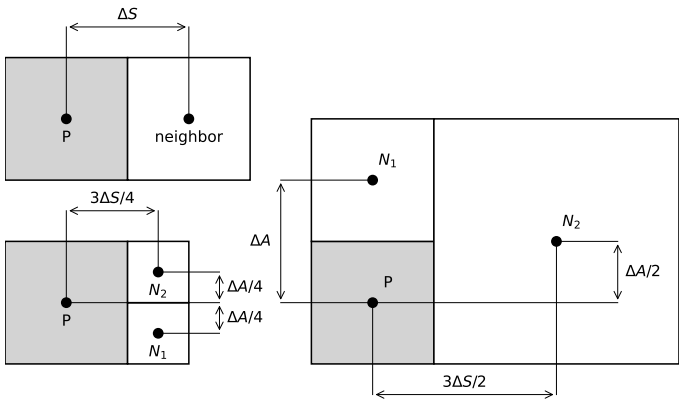


Figure 2: Illustration of the three possible configurations in the directional vicinity of a reference element P. Distances are characterized both parallel (ΔA) and orthogonal (ΔS) to the heat exchange interface.

The illustrated diagram shows the distances between the centers of the P element that is being computed as the current element and its neighbors; the examples are considering the calculation of the derivative in the direction of the east face. But all the expressions remain valid for directions equivalent to the orthogonal rotations of these schemes.

The expressions for these derivatives were obtained by considering a linear variation between the neighbors with the expression $T(x) = ay + bx + c$, by substituting the values of the temperature and the relative position of the centroids of the elements involved in the calculation, it is possible to obtain values for the coefficients and derive the expression in the direction normal to the face.

The number of neighbors is identified in a specific routine; in the absence of neighbors in a given direction, it is inferred that the node is in a border position. In this case, the boundary conditions are applied, either Dirichlet, Neumann or Robin conditions. Thus, the equations can be written in the traditional form of finite volumes.

$$\alpha_P T_P + \alpha_N T_N + \alpha_S T_S + \alpha_E T_E + \alpha_W T_W = \sigma \tag{6}$$

Here, α_P is the coefficient of the central or reference cell (often termed as the pivot), while α_N, α_S, α_E, and α_W are the coefficients corresponding to the neighboring cells in the North, South, East, and West of the pivot cell, respectively. The magnitude and sign of these coefficients are influenced by the chosen discretization scheme and the material properties such as thermal conductivity and specific heat, combined with cell dimensions. Remember, however, that each directional

term can depend on more than one other element in the neighborhood. The term σ symbolizes the source or sink present within the control volume, encompassing internal heat generation or absorption. In the absence of such sources or sinks, $\sigma = 0$. When aggregating the equations across the computational domain, a matrix form can be derived as:

$$\mathbf{AT} = \mathbf{b} \tag{7}$$

In this representation, $\mathbf{A}$ is the matrix constructed from the α_i terms, $\mathbf{T}$ is a column vector detailing the temperatures of each cell or node, and $\mathbf{b}$ contains the respective source term σ for each cell or node.

In our computational framework, to address the challenges posed by asymmetric linear systems, we employed the Generalized Minimal Residual method (GMRES) (Saad and Schultz, 1986). Its effectiveness is particularly pronounced when used in conjunction with the sparse matrix representation from the SciPy (Virtanen *et al.*, 2020) library. Leveraging sparse representations not only reduces memory overhead but also optimizes computational performance, especially crucial given the inherent sparsity of our problem matrix. All computational tasks were executed on a system equipped with an 11th Gen Intel(R) Core(TM) i7-11700 processor and 24 GB of RAM, ensuring reliable and efficient computations.

3. RESULTS

3.1 Analysis of 1D Simulation Results

We examined a 1D heterogeneous bar, length 1 m, with boundary conditions $T(0) = 1^o$C and $T(1) = 0^o$C. In the domain $x < 0.5$ m, thermal conductivity was k while for $x \geq 0.5$ m it was $10k$. The analytical solution for this problem can be expressed as:

$$T(x) = \begin{cases} -\frac{20x}{11} + 1 & \text{if } x \in [0.0,\ 0.5) \\ -\frac{2x}{11} + \frac{2}{11} & \text{if } x \in [0.5,\ 1.0] \end{cases} \tag{8}$$

Steady-state solutions offer validation benefits due to the ease of obtaining analytical solutions in 1D multi-domain cases. For numerical resolution, the GMRES method from the SciPy library was used. In order to solve the system we utilize GMRES setting its convergence criterion as residual norm below a tolerance of 10^{-8}, a revised parameter within SciPy's GMRES.

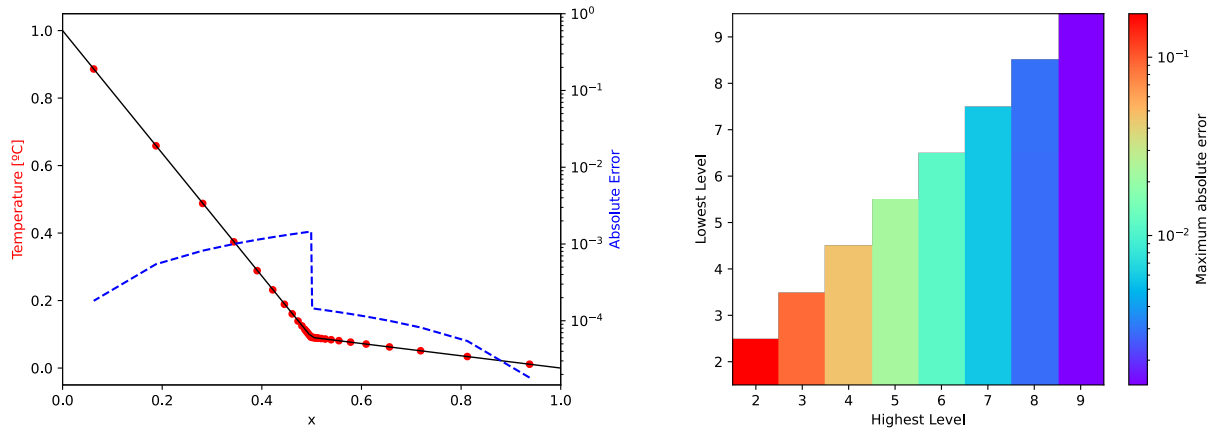
Mesh refinement tests ranged the lowest level from 2 to 9 and highest between the lowest value and 9. The solution for the configuration with lowest level 2 and highest level 9 can be seen on Figure 3a where both analytical (solid black line) and numerical (red dots) solutions for temperature profile have their values corresponding to the vertical axis on the left. The absolute error distribution (blue dashed line) across the domain is addressed to the vertical axis on the right. From the distribution of the absolute error, it can also be inferred that the critical point will be found exactly on the interface surface between the two materials. This must be due to the linear approximations introduced by the finite volume method.

The heatmap in Figure 3b reveals that using coarser meshes outside the interface area, combined with adaptive refinement in the direction of heat transfer, leads to maximum errors comparable to those from fully refined meshes. This suggests that the approach achieves commendable results while conserving computational resources. Interestingly, the heatmap indicates that errors across the level 9 column are closely matched, with a consistent gradient from levels 2 to 9.

3.2 Analysis of 2D Simulation Results

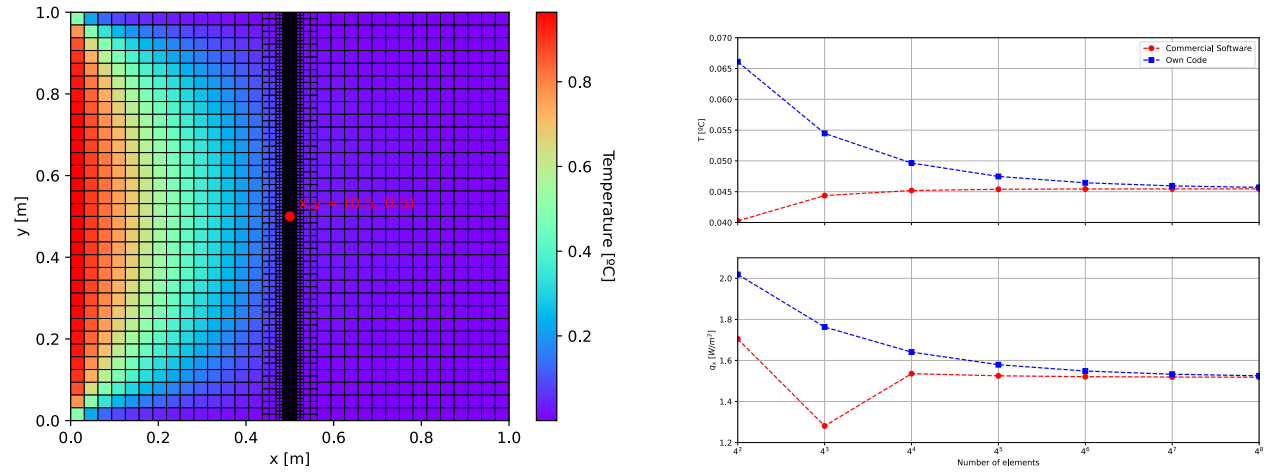
In order to analyze a two-dimensional multi-domain problem, we consider other means of validating the code and ascertaining simulation performance metrics. As the analytical solution for two-dimensional problems with spatially variable properties is more challenging to obtain, we will compare the solution obtained on uniform meshes with commercial software. Considering a plate with the same characteristics as the bar in the previous one-dimensional case, i.e. with thermal conductivity multiplied by 10 to $x \geq 0.5$, let's keep all the edges at zero temperature, except for the edge at $x = 0$, where $T(0, y) = 1^o$C. In this situation, without heat generation, the maximum and minimum temperature points are expected to be on the contour and the profile obtained is two-dimensional and symmetrical. The temperature profile obtained for this problem can be seen in figure 4a for a mesh configuration in which the refinement levels vary between 6 and 9, with a total of 11008 rectangular elements. For the specific case under consideration, the temperature at the centroidal point was determined to be $0.04217\,$ºC, and the heat flux at this point was $1.5204\,\mathrm{W/m^2}$. Drawing from Fourier's law of heat conduction, as expounded in Incropera *et al.* (1996), we're reminded that the heat flux is directly proportional to the derivative of the temperature T in the x-direction, with the material's thermal conductivity k serving as the proportionality constant.

In figure 4a, it can be seen that the temperature profile has greater variation on the left side, where the conductivity is lower, while the right side shows an almost uniform scalar field for temperature. This behavior is similar to the one-dimensional case, in which the rate of temperature decay was greater in the range of lower thermal conductivity.



(a) (b)

Figure 3: (a) Graph comparing 1D heat transfer's analytical and numerical solutions with error values on the right. (b) Map of maximum errors across mesh configurations with best and worst cases indicated by arrows.



(a) (b)

Figure 4: (a) 2D temperature profile of a multidomain heat transfer on a mesh, highlighting the centroid. (b) Temperature and heat flux comparison at the plate's centroid based on mesh element count, contrasting our quadtree code and commercial software.

Various mesh configurations, as was done for the unidmensional case, were used to analyze the effectiveness of the quadtrees implementation. The figures 5 display metrics to guide us in the analysis of this two-dimensional case. Each highlighted point represents a different mesh configuration, characterized by the highest and lowest levels of refinement allowed, plotted as a function of the total number of elements. The colors of the markers indicate a value for the lowest level of refinement, while the solid colored lines are used to indicate configurations that have values of the highest level of refinement in common, making it possible to see a certain pattern in the solution's behavior. Dashed lines connect uniform mesh configurations.

Figure 5a shows the temperature for each of the configurations tested, and figure 5b shows the heat flux, both evaluated at the centroid of the plate. It can be seen that there is an asymptotic behavior, for a value that is later validated using commercial software, this result can be seen in figure 4b.

The figures 5c and 5d are metrics of computational effort for the part that refers to the solution of the linear system assembled using GMRES as a solver. They show, respectively, the number of iterations needed to reach the tolerance for the residue norm and the running time on the current machine. Computational aspects of GMRES are addressed in Saad (2003), including its dependence on the dimension of the Krylov subspace (which grows with iterations, hence the relation to number of iterations) and the matrix-vector multiplication's dependence on its dimension and the sparsity structure.

In the table 1, information on the time taken to assemble the quadtree, assemble the linear system and solve the linear

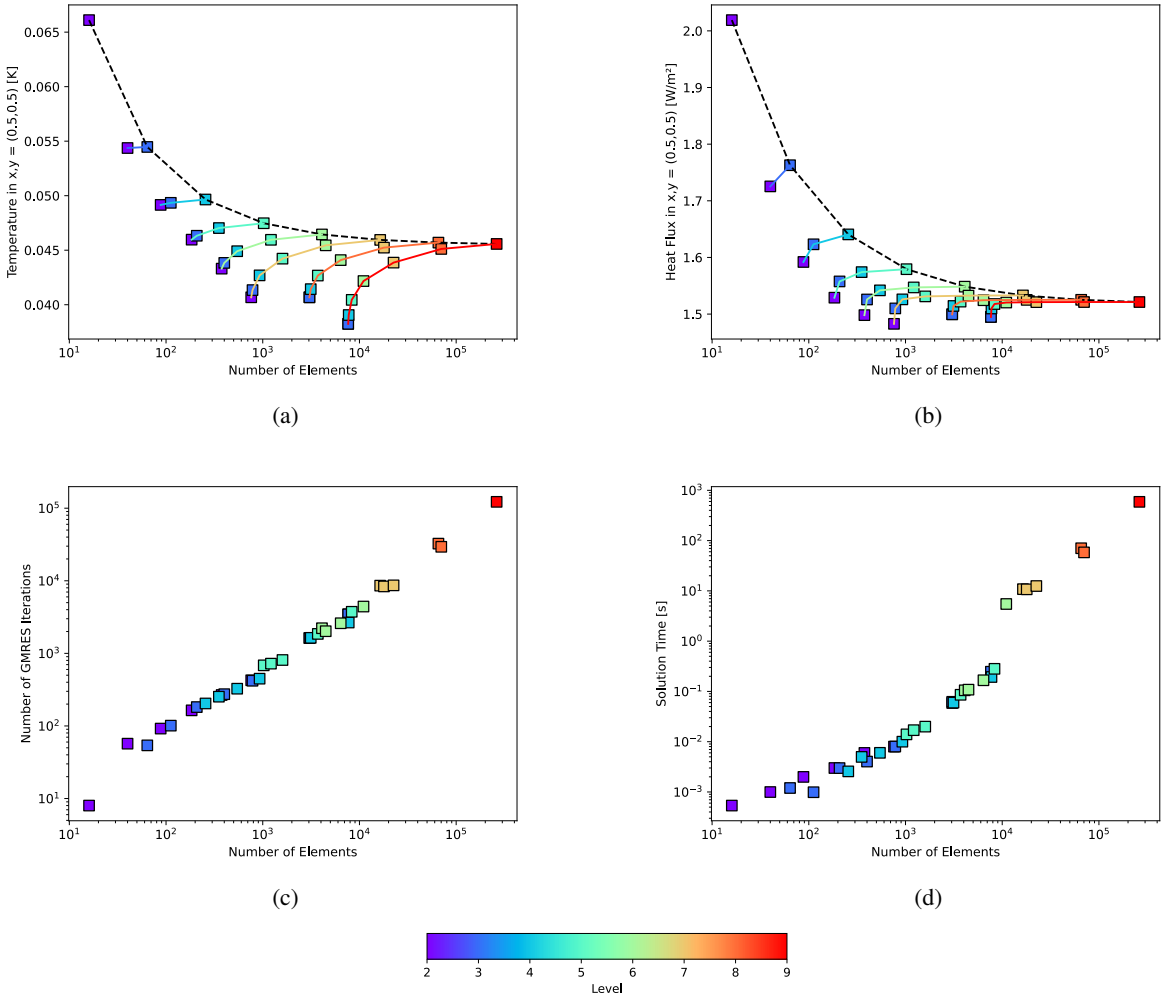


Figure 5: Plots for the 2D plate data where (a) shows temperature convergence at the centroid; (b) indicates heat flux at the same point; (c) illustrates GMRES iteration count to achieve the solution; (d) presents computational time. The x-axis shows element count, markers colors denote lowest refinement levels, solid lines represent highest refinement levels for, and dashed lines link uniform meshes.

system, as well as the number of iterations to solve it, has been presented as a function of the number of elements in the uniform meshes that have been used as a reference for these tests. These meshes can be considered the worst case of each maximum level of refinement, where the entire domain is discretized up to a certain level, forming a complete tree with all the leaves at the last level.

Number of Elements	QuadTree Assembly	Linear System Assembly	Solution	Number of Iterations
4^2	0,0000 (0%)	0,0000 (0%)	0,0005 (100%)	8
4^3	0,0030 (41%)	0,0030 (42%)	0,0012 (17%)	54
4^4	0,0030 (20%)	0,0090 (62%)	0,0026 (18%)	204
4^5	0,0040 (8%)	0,0306 (63%)	0,0140 (29%)	684
4^6	0,0130 (5%)	0,1616 (58%)	0,1053 (38%)	2220
4^7	0,1179 (1%)	0,5612 (5%)	10,821 (94%)	8574
4^8	0,4949 (1%)	2,3169 (3%)	70,452 (96%)	32513
4^9	2,0614 ($\sim$0%)	9,1534 (2%)	590,23 (98%)	122378

Table 1: Times table for quadtree assembly, linear system setup, and GMRES solution. Percentages show time distribution for each mesh setup, with iteration numbers emphasized to correlate with solution time.

It is observed that the solution time asymptotically becomes the most costly, while the quadtree assembly time becomes negligible in comparison. However, we still see a significant increase as the tree's height grows. The linear system

assembly time is readily identified as having an increase proportional to the number of elements, given that the matrix is sparse.

4. CONCLUSION

The numerical resolution of multi-domain heat transfer problems serves as a prime example where adaptive mesh refinement (AMR) can be leveraged. This approach is geared towards optimizing computational resources by focusing more extensively on regions proximal to the inherent discontinuities in the represented physics.

In our unidimensional simulation, it was discerned that the absolute error yielded by adaptively refined meshes is commensurate with those of uniform meshes. The peak error, which coincides with the point of discontinuity, manifested errors of similar magnitude for tests with identical highest levels of refinement. This simulation was executed to draw a direct comparison with an analytical solution.

For the bidimensional case, the solution was juxtaposed against a numerical output from a commercial software. The asymptotic behavior converged to identical values, corroborating the accurate implementation of the Finite Volume Method (FVM) for this scenario as well. Additionally, it was observed that temperature and heat flux oscillated more closely around the asymptotic value when employing a mesh with multiple refinement levels, as opposed to uniform meshes at the base refinement level. This further underscores the advantages conferred by adaptive refinement.

Regarding the solver, both the number of iterations required and the computational time signal that this is the most resource-intensive step in the solution process. However, this operation's intensity doesn't stem from mesh configuration but is strictly contingent on the number of elements. Thus, utilizing a mesh with fewer elements yields quicker solutions.

The construction of the quadtree, initially not perceived as overly burdensome, showed that its restructuring in transient problems might accrue significant costs. This is especially true when superlinear operations, such as smoothing, come into play. The linear system's necessity to be continually reformed is another factor to consider.

In conclusion, this research accentuates the pivotal role of AMR techniques, bolstered by adept data structures like quadtrees, in delivering accurate and resource-efficient solutions to multi-domain heat transfer challenges. Pairing AMR with the Finite Volume Method (FVM) emerges as a promising strategy for future pursuits in heat transfer research and its affiliated fields.

5. ACKNOWLEDGEMENTS

We acknowledge the team of Laboratory of Applied Thermal Engineering at the Aeronautics Institute of Technology for providing computer facilities. This work was funded by CNPq and Petrobras.

6. REFERENCES

Ainsworth, M. and Oden, J.T., 1997. "A posteriori error estimation in finite element analysis". *Computer methods in applied mechanics and engineering*, Vol. 142, No. 1-2, pp. 1–88.

Babuška, I. and Suri, M., 1994. "The p and h-p versions of the finite element method, basic principles and properties". *SIAM Review*, Vol. 36, No. 4, pp. 578–632. doi:10.1137/1036141. URL `https://doi.org/10.1137/1036141`.

Babuška, I. and Miller, A., 1987. "A feedback finite element method with a posteriori error estimation: Part i. the finite element method and some basic properties of the a posteriori error estimator". *Computer Methods in Applied Mechanics and Engineering*, Vol. 61, No. 1, pp. 1–40. ISSN 0045-7825. doi:https://doi.org/10.1016/0045-7825(87)90114-9. URL `https://www.sciencedirect.com/science/article/pii/0045782587901149`.

Bejan, A., 2013. *Convection heat transfer*. John wiley & sons.

Bejan, A., 2016. *Advanced engineering thermodynamics*. John Wiley & Sons.

Berger, M.J. and Colella, P., 1989. "Local adaptive mesh refinement for shock hydrodynamics". *Journal of computational Physics*, Vol. 82, No. 1, pp. 64–84.

Berger, M.J. and Oliger, J., 1984. "Adaptive mesh refinement for hyperbolic partial differential equations". *Journal of computational Physics*, Vol. 53, No. 3, pp. 484–512.

Bird, R.B., 2002. "Transport phenomena". *Appl. Mech. Rev.*, Vol. 55, No. 1, pp. R1–R4.

Finkel, R.A. and Bentley, J.L., 1974. "Quad trees a data structure for retrieval on composite keys". *Acta informatica*, Vol. 4, pp. 1–9.

Hahn, D.W. and Özisik, M.N., 2012. *Heat conduction*. John Wiley & Sons.

Hirsch, C., 2007. *Numerical computation of internal and external flows: The fundamentals of computational fluid dynamics*. Elsevier.

Incropera, F.P., DeWitt, D.P., Bergman, T.L., Lavine, A.S. *et al.*, 1996. *Fundamentals of heat and mass transfer*, Vol. 6. Wiley New York.

LeVeque, R.J., 2002. *Finite volume methods for hyperbolic problems*, Vol. 31. Cambridge university press.

Moukalled, F., Mangani, L., Darwish, M., Moukalled, F., Mangani, L. and Darwish, M., 2016. *The finite volume method*.

Springer.

Oden, J.T. and Reddy, J.N., 2012. *An introduction to the mathematical theory of finite elements*. Courier Corporation.

Patankar, S., 2018. *Numerical heat transfer and fluid flow*. Taylor & Francis.

Peng, H.F., Bai, Y.G., Yang, K. and Gao, X.W., 2013. "Three-step multi-domain bem for solving transient multi-media heat conduction problems". *Engineering Analysis with Boundary Elements*, Vol. 37, No. 11, pp. 1545–1555.

Saad, Y. and Schultz, M.H., 1986. "Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems". *SIAM Journal on scientific and statistical computing*, Vol. 7, No. 3, pp. 856–869.

Saad, Y., 2003. *Iterative methods for sparse linear systems*. SIAM.

Samet, H., 1984. "The quadtree and related hierarchical data structures". *ACM Computing Surveys (CSUR)*, Vol. 16, No. 2, pp. 187–260.

Samet, H., 2006. *Foundations of multidimensional and metric data structures*. Morgan Kaufmann.

Verführt, R., 1996. "A review of a posteriori error estimation and adaptive mesh-refinement techniques".

Versteeg, H.K. and Malalasekera, W., 2007. *An introduction to computational fluid dynamics: the finite volume method*. Pearson education.

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J. *et al.*, 2020. "Scipy 1.0: fundamental algorithms for scientific computing in python". *Nature methods*, Vol. 17, No. 3, pp. 261–272.

Zlatanova, S., Rahman, A.A. and Pilouk, M., 2002. "Trends in 3d gis development". *Journal of Geospatial Engineering*, Vol. 4, No. 2, pp. 71–80.

7. RESPONSIBILITY NOTICE

The authors are solely responsible for the printed material included in this paper.