Advances in Computational Science and Engineering
Vol. 4, 2025, pp. 57-84
doi:10.3934/acse.2025012

OCTREE-BASED ADAPTIVE MESH REFINEMENT AND THE SHIFTED BOUNDARY METHOD FOR EFFICIENT FLUID DYNAMICS SIMULATIONS

Cheng-Hau Yang[1], Guglielmo Scovazzi[2], Adarsh Krishnamurthy[1] and Baskar Ganapathysubramanian[*1]

[1]Department of Mechanical Engineering, Iowa State University, Ames, IA 50011, USA

[2]Department of Civil and Environmental Engineering, Duke University, Durham, NC 27708, USA

Abstract. This paper presents an adaptive mesh refinement (AMR) framework integrated with the shifted boundary method (SBM) for incompressible flow and coupled thermal-flow simulations. Our framework leverages octree-based AMR, enabling hierarchical and dynamic mesh refinement driven by vorticity magnitude. This strategy enables capturing complex vorticity structures and steep thermal gradients while significantly reducing computational costs compared to traditional uniform refinement approaches, particularly for flows around complex geometries. The octree-based architecture ensures efficient data management, including robust intergrid transfer and load balancing, which is critical for scalability in distributed-memory environments. Dynamic mesh adaptivity is demonstrated for complex geometries where achieving ideal refinement is often non-trivial due to the irregular boundaries. SBM enhances this adaptability by accurately enforcing boundary conditions on intricate and non-conformal geometries without requiring boundary-fitted meshes. Together, these methods address longstanding challenges in computational fluid dynamics, providing a resource-efficient yet accurate approach for capturing critical flow and thermal features. The utility of the framework is demonstrated through numerical experiments, showcasing its ability to adapt dynamically to evolving flow and thermal patterns in diverse and challenging geometries.

1. **Introduction.** The simulation of fluid flows governed by the Navier-Stokes equations remains one of the most challenging problems in computational physics, especially when resolving multi-scale structures like vortices. Traditional approaches using uniform grid refinement, while effective, often impose prohibitive computational costs. AMR emerged as an innovative solution to this issue, fundamentally transforming the landscape of computational fluid dynamics (CFD).

AMR's foundational principles were set forth in the 1980s by Berger and colleagues [12, 11]. Their pioneering methods introduced dynamic grid refinement strategies that adapt to local flow characteristics, especially in regions with steep

2020 *Mathematics Subject Classification.* Primary: 58F15, 58F17; Secondary: 53C35.
Key words and phrases. Shifted boundary method, Immersed Boundary Method, adaptive mesh refinement, computational fluid dynamics, thermal incompressible flow, incomplete octree, weak boundary conditions.
*Corresponding author: Baskar Ganapathysubramanian.

gradients or discontinuities. This adaptability enabled high-resolution computations in critical areas while maintaining efficiency in smoother regions. Colella's contributions further advanced AMR for turbulent and shock-driven phenomena [1], leading to practical implementations as the Chombo framework [21]. Building on these numerical advances, AMR has been successfully deployed across diverse fluid dynamics applications. The versatility of AMR has been demonstrated across numerous fluid dynamics applications, including astrophysical phenomena [17], multiphase flows [79, 68, 46, 73], free surface problems [9], reactive and electrochemical flows [25, 48], and aerodynamics [62, 71], highlighting AMR's capability to efficiently handle multi-scale flow problems while maintaining computational efficiency. AMR is often combined with octree meshes to enable high-performance simulations. Octree meshes were originally introduced as a spatial decomposition method and were initially applied in computer graphics and computational geometry [58]. Over time, they were adopted in scientific computing, particularly in FEM [14, 70] and CFD [20, 47, 67] simulations. More recently, octree meshes have been widely used to perform highly parallel scientific simulations [71, 3, 61].

While AMR has proven invaluable in adapting mesh resolution to fine-scale structures, challenges remain in automatically generating high-quality meshes for complex geometries. This issue becomes especially pronounced in domains involving irregular boundaries, multi-scale features, or evolving interfaces, where traditional body-fitted meshing approaches can be computationally expensive and inflexible. Immersed methods provide an attractive alternative by enabling the discretization of governing equations on structured or Cartesian grids, bypassing the need for complex mesh generation. Originally proposed by Peskin [65] for simulating blood flow around heart valves, IBM has been extended to diverse applications, including three-dimensional fluid-structure interaction (FSI) with robust formulations, as well as biological systems, and thermal transport [60, 87]. However, traditional IBMs often suffer from challenges in imposing accurate boundary conditions, especially in the presence of sharp gradients, and are prone to issues such as spurious oscillations or inaccuracies near boundaries. The SBM [55, 56, 43, 7, 6, 5, 23, 72, 8, 99, 34, 22, 95, 4, 91], within the broader class of Immersed Boundary Methods (IBMs), addresses this issue by enabling accurate boundary condition enforcement on irregularly shaped domains without requiring boundary-fitted meshes or costly cut-element integration. Developed to address the limitations of traditional immersed boundary techniques, such as the small cut-cell and load balancing issues [95], the SBM on of complex boundaries. It has proven robust in handling Poisson equations [55, 22, 95, 4], linear elasticity [6, 22, 95], Navier-Stokes equations [56, 97], and free surface flows [23]. It has also been successfully applied to one-way coupled FSI problems [91]. However, its application to two-way coupled FSI and thin-structure problems is still under development.

To date, SBM and AMR have not been jointly applied in time-dependent flow and thermal simulations. This study presents the first framework integrating AMR with the SBM to efficiently capture both vortical structures and thermal gradients in complex domains. By adaptively refining the mesh in critical regions, AMR optimizes computational resources, while the SBM ensures precise enforcement of boundary conditions on non-conformal meshes, providing an effective and resource-efficient solution.

The paper is structured as follows: Section 2 details the AMR framework we adopt, with specific emphasis on field-transfer operators between octree grids; Section 3 describes the weak formulation of the PDEs and the corresponding SBM formulation; and Section 4 presents a wide range of computational experiments to test the proposed computational framework.

2. Adaptive mesh refinement framework.

2.1. Vorticity refinement/de-refinement indicator.

The AMR framework we propose is employed based on the magnitude of the vorticity. The vorticity $\boldsymbol{\omega}$ is defined as the curl of the velocity field:

$$\boldsymbol{\omega} = \nabla \times \boldsymbol{u} \ .$$

In the two dimensional case, it can be expressed as

$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \ ,$$

with magnitude $|\boldsymbol{\omega}|$ given by

$$|\boldsymbol{\omega}| = \left| \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right| \ .$$

The components of $\boldsymbol{\omega}$ in the three-dimensional case are:

$$\begin{aligned} \omega_x &= \frac{\partial w}{\partial y} - \frac{\partial v}{\partial z}, \\ \omega_y &= \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x}, \\ \omega_z &= \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}. \end{aligned}$$

with corresponding magnitude given by:

$$|\boldsymbol{\omega}| = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2} \ .$$

Let $\omega_{\max}$, $\omega_{\min}$, $l_{\max}$, and $l_{\min}$ be the maximum vorticity, minimum vorticity, maximum adaptive refinement level, and minimum adaptive refinement level, respectively. The refinement level (l) inside the fluid domain is then based on the magnitude of the vorticity $|\boldsymbol{\omega}|$ to obtain the corresponding values:

$$l = \begin{cases} l_{\max}, & \text{if } |\boldsymbol{\omega}| \geq \omega_{\max} \ , \\ \text{round}\left(\frac{l_{\max} - l_{\min}}{\omega_{\max} - \omega_{\min}} (|\boldsymbol{\omega}| - \omega_{\min}) + l_{\min} \right), & \text{if } \omega_{\min} < |\boldsymbol{\omega}| < \omega_{\max} \ , \\ l_{\min}, & \text{if } |\boldsymbol{\omega}| \leq \omega_{\min} \ . \end{cases} \tag{1}$$

The AMR parameters $(\omega_{\max}, \omega_{\min}, l_{\max}, l_{\min})$ were selected empirically. We previously conducted simulations using octree meshes with local mesh refinement in specific regions [96, 97]. Based on these studies, we established the following general principles for parameter selection: (a) For problems that generate time-dependent vortices, we set $\omega_{\max} = 2$ to ensure that vortices are adequately captured and well refined. (b) For $\omega_{\min}$, we typically choose 0.1. (c) The choice of $l_{\max}$ is based on our previous simulations, where we observed that refining the mesh to this level in specific regions, such as the wake region, improves the accuracy of the results. (d) For $l_{\min}$, we generally select a value 3 to 4 levels lower than $l_{\max}$ and ensure that $l_{\min} \geq 5$.

2.2. **Intergrid transfer.** A key aspect of AMR is the efficient transfer of data between the current mesh and the newly refined or coarsened mesh - a process known as intergrid transfer. The DENDRO-KT [36, 72, 34, 32, 82] framework supports these intergrid transfers with traversal-based algorithms that allow for local refinement or coarsening across different regions of the mesh. To ensure computational efficiency, DENDRO-KT restricts local refinement or coarsening to a single level. The intergrid transfer between the current mesh, denoted as mesh A, and the newly refined/coarsened mesh B, is performed by simultaneously traversing both meshes in a synchronized fashion, ensuring proper data alignment. Since A and B differ by only one refinement level, DENDRO-KT handles three possible scenarios: (1) if both A and B are at the same refinement level and are leaf elements, the data is directly copied; (2) if A is at the leaf level but B is further refined, values from A are interpolated onto B using shape functions evaluated at the child nodes; and (3) if B is at the leaf level but A is refined, values from A are projected onto B's parent element.

To track evolving solutions, the mesh must continuously adapt through refinement and coarsening based on vorticity magnitudes. This periodic remeshing in DENDRO-KT also requires repartitioning in distributed memory environments to maintain load balance. After each remeshing step, the newly generated mesh undergoes 2:1 balance enforcement [78, 36]. The remapping of data remains efficient by leveraging space-filling curves (SFCs). Detailed approaches for efficiently implementing these distributed-memory intergrid transfers in DENDRO-KT are discussed in [77].

The *Compression Ratio* quantifies the efficiency of AMR compared to a uniform mesh by expressing the reduction in mesh nodes. Calculated as the ratio of nodes in the uniform mesh to nodes in the AMR mesh, this metric highlights the degree of optimization achieved through AMR:

$$\text{Compression Ratio} = \frac{\text{Number of Nodes in Uniform Mesh}}{\text{Number of Nodes in AMR Mesh}} \tag{2}$$

3. The shifted boundary method.

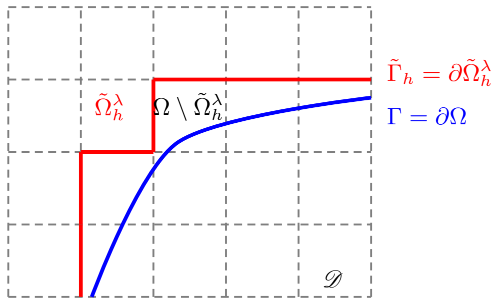
3.1. **Preliminaries: The true domain, surrogate domain, and maps.** Figure 1a depicts a closed region $\mathscr{D}$ such that $\text{clos}(\Omega) \subseteq \mathscr{D}$ (here $\text{clos}(\Omega)$ indicates the *closure* of Ω) and the family $\mathscr{T}_h(\mathscr{D})$ of admissible and shape-regular discrete decompositions (meshes/grids) of $\mathscr{D}$. In this work, we specifically focus on octree grids aligned along the axes of the Cartesian space. Then, we restrict each $\mathscr{T}_h(\mathscr{D})$ by selecting those elements $T \in \mathscr{T}_h(\mathscr{D})$ such that

$$\text{meas}(T \cap \Omega) > (1 - \lambda)\,\text{meas}(T)\ , \qquad \text{for some } \lambda \in [0, 1]\ . \tag{3}$$

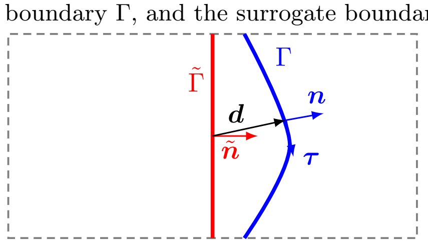
In other words, these are elements that have an intersection with the domain of interest Ω with an area/volume larger than $1-\lambda$ with respect to their total area/volume, respectively in two/three dimensions. For example, choosing $\lambda = 0$ selects the elements that are strictly contained in the computational domain Ω (see, e.g., Figure 1a), choosing $\lambda = 1$ selects the elements that have a non-empty intersection with Ω, and choosing $\lambda = 0.5$ selects elements whose intersection with Ω includes at least 50% of their area/volume.

We define the family of grids that satisfies Eq. 3 as

$$\tilde{\mathscr{T}}_h^{\lambda} := \{T \in \mathscr{T}_h(\mathscr{D}) : \text{meas}(T \cap \Omega) > (1 - \lambda)\,\text{meas}(T)\}\ .$$



(a) The surrogate domain $\tilde{\Omega}_h^\lambda \subset \Omega$, the difference $\Omega \setminus \tilde{\Omega}_h^\lambda$ between the true and surrogate domains, true boundary Γ, and the surrogate boundary.



(b) The distance vector $\boldsymbol{d}$, the true normal $\boldsymbol{n}$, the true tangent $\boldsymbol{\tau}$, and the surrogate normal $\tilde{\boldsymbol{n}}$ (horizontal).

Figure 1 The surrogate domain, its boundary, and the distance vector $\boldsymbol{d}$.

This identifies the *surrogate domain*

$$\tilde{\Omega}_h^\lambda := \mathrm{int}\Big(\bigcup_{T \in \tilde{\mathscr{T}}_h^\lambda} T \Big) ,$$

or, more simply, $\tilde{\Omega}_h$, with *surrogate boundary* $\tilde{\Gamma}_h := \partial\tilde{\Omega}_h$ and outward-oriented unit normal vector $\tilde{\boldsymbol{n}}$ to $\tilde{\Gamma}_h$. Obviously, $\tilde{\mathscr{T}}_h^\lambda$ is an admissible and shape-regular family of decompositions of $\tilde{\Omega}_h$ (see again Figure 1a). Here, we make the optimal choice $\lambda = 0.5$, which minimizes the average distance between the surrogate and true boundaries, and consequently the numerical error in computations, as discussed in detail in [95]. Consider now the mapping, sketched in Figure 1b,

$$\boldsymbol{M}_h : \tilde{\Gamma}_h \to \Gamma \ , \tag{4a}$$

$$\tilde{\boldsymbol{x}} \mapsto \boldsymbol{x} \ , \tag{4b}$$

which associates to any point $\tilde{\boldsymbol{x}} \in \tilde{\Gamma}_h$ on the surrogate boundary a point $\boldsymbol{x} = \boldsymbol{M}_h(\tilde{\boldsymbol{x}})$ on the physical boundary Γ. In this work, $\boldsymbol{M}_h$ is defined as the closest-point projection of $\tilde{\boldsymbol{x}}$ on Γ, as shown in Figure 1b. Through $\boldsymbol{M}_h$, a distance vector function $\boldsymbol{d}_{\boldsymbol{M}_h}$ can be defined as

$$\boldsymbol{d}_{\boldsymbol{M}_h}(\tilde{\boldsymbol{x}}) \ = \ \boldsymbol{x} - \tilde{\boldsymbol{x}} \ = \ [\,\boldsymbol{M} - \boldsymbol{I}\,](\tilde{\boldsymbol{x}}) \ . \tag{5}$$

For the sake of simplicity, we set $\boldsymbol{d} = \boldsymbol{d}_{\boldsymbol{M}_h}$ where $\boldsymbol{d} = \|\boldsymbol{d}\|\boldsymbol{\nu}$ and $\boldsymbol{\nu}$ is a unit vector.

Remark 3.1. There are many strategies to define the map $\boldsymbol{M}_h$ and, correspondingly, the distance vector $\boldsymbol{d}$. Whenever uniquely defined, the closest-point point projection of $\tilde{\boldsymbol{x}}$ upon Γ is a natural choice for $\boldsymbol{x}$ (and, therefore, $\boldsymbol{M}_h$). But other choices may be preferable, such as a level-set description of the true boundary, where $\boldsymbol{d}$ is defined by means of a distance function. See also the discussion in [5, 6] regarding domains with corners.

3.2. **Shifted boundary conditions.** Most of our focus will be on Dirichlet boundary conditions, since to this type belong the no-slip boundary condition enforced on the surface of shapes immersed into a fluid domain. Other types of boundary conditions can also be treated with the SBM, but these are somewhat less interesting in the context of CFD.

Consider then a surrogate Dirichlet boundary $\tilde{\Gamma}_{D,h}$ in proximity of a true Dirichlet boundary Γ_D. Using the construction of the distance between the two boundaries, it is possible to introduce the Taylor expansion of the velocity vector:

$$\boldsymbol{u}(\tilde{\boldsymbol{x}}) + (\nabla \boldsymbol{u} \cdot \boldsymbol{d})(\tilde{\boldsymbol{x}}) + (\boldsymbol{R}_D(\boldsymbol{u}, \boldsymbol{d}))(\tilde{\boldsymbol{x}}) = \boldsymbol{u}_D(\boldsymbol{M}_h(\tilde{\boldsymbol{x}})) \, , \qquad \text{on } \tilde{\Gamma}_{D,h} \, , \tag{6}$$

where the remainder $\boldsymbol{R}_D(\boldsymbol{u}, \boldsymbol{d})$ satisfies $\|\boldsymbol{R}_D(\boldsymbol{u}, \boldsymbol{d})\| = o(\|\boldsymbol{d}\|^2)$ as $\|\boldsymbol{d}\| \to 0$. We can then define, on $\tilde{\Gamma}_{D,h}$, the *extension* operator

$$\mathbb{E}\boldsymbol{u}_D(\tilde{\boldsymbol{x}}) := \boldsymbol{u}_D(\boldsymbol{M}_h(\tilde{\boldsymbol{x}})) \tag{7}$$

and the *shift* operator

$$\boldsymbol{S}_{D,h}\, \boldsymbol{u}(\tilde{\boldsymbol{x}}) := \boldsymbol{u}(\tilde{\boldsymbol{x}}) + \nabla \boldsymbol{u}(\tilde{\boldsymbol{x}})\, \boldsymbol{d}(\tilde{\boldsymbol{x}}) \, . \tag{8}$$

Neglecting the higher-order residual term in Eq. 6, we obtain the final expression of the *shifted* boundary conditions

$$\boldsymbol{S}_{D,h}\, \boldsymbol{u} = \mathbb{E}\boldsymbol{u}_D \, , \qquad \text{on } \tilde{\Gamma}_{D,h} \, . \tag{9}$$

In what follows - for the sake of simplicity and whenever it does not cause confusion - we will omit the symbol $\mathbb{E}$ from $\mathbb{E}\boldsymbol{u}_D(\tilde{\boldsymbol{x}})$ and simply write $\boldsymbol{u}_D(\tilde{\boldsymbol{x}})$.

Remark 3.2. While the SBM provides a convenient approach to enforcing boundary conditions on irregularly shaped domains without the need for boundary-fitted meshes or cut-element integration, we recognize that other IBMs, including direct forcing IBM [59, 75, 2, 33, 15, 64, 29, 74, 76, 27, 39, 40, 93, 88, 35, 26, 101, 92, 41, 10] and continuous forcing IBM [65, 66, 49, 81, 24, 80, 63, 100, 53, 90, 19], also enforce boundary conditions correctly and accurately. Our intent is not to suggest that these alternative methods lack accuracy but rather to highlight the differences in implementation strategies, and ensuing computational overheads.

3.3. **General definitions and notation.** Let us denote by $L^2(\Omega)$ the space of square integrable functions over Ω. Here and in the following, $(v, w)_\omega = \int_\omega v\, w$ denotes the L^2-inner product on a subset $\omega \subseteq \Omega$, and $\langle v, w \rangle_\gamma = \int_\gamma v\, w$ denotes the L^2-inner product on a subset $\gamma \subseteq \Gamma$. Let $H^m(\Omega) = W^{m,p}(\Omega)$ indicate the Sobolev spaces of index of regularity $m \geq 0$ and index of summability $p = 2$, equipped with the (scaled) norm

$$\|v\|_{H^m(\Omega)} = \left(\|v\|^2_{L^2(\Omega)} + \sum_{k=1}^{m} \|l(\Omega)^k \boldsymbol{D}^k v\|^2_{L^2(\Omega)} \right)^{1/2} \, , \tag{10}$$

where $\boldsymbol{D}^k$ is the kth-order spatial derivative operator and $l(\Omega) = (\text{meas}_{n_d}(\Omega))^{1/n_d}$ is a characteristic length of the domain Ω (introduced here to maintain dimensional consistency in the definitions of the Sobolev norms). Note that $H^0(\Omega) = L^2(\Omega)$. As usual, we use a simplified notation for norms and semi-norms, i.e., we set $\|v\|_{m;\Omega} = \|v\|_{H^m(\Omega)}$ and $|v|_{k;\Omega} = \|\boldsymbol{D}^k v\|_{0;\Omega} = \|\boldsymbol{D}^k v\|_{L^2(\Omega)}$. Note that the above definitions are general, and the sets Ω and Γ can be replaced by other sets, depending on the situation.

3.4. **SBM variational formulation.** We now introduce the scalar and vector discrete function spaces

$$\tilde{V}^h(\tilde{\Omega}_h^\lambda) = \left\{ q^h \mid q^h \in C^0(\tilde{\Omega}_h^\lambda) \cap \mathscr{Q}^1(T)\,,\ \text{with } T \in \tilde{\mathscr{T}}_h^\lambda \right\}\ , \tag{11}$$

$$\tilde{\boldsymbol{V}}^h(\tilde{\Omega}_h^\lambda) = \left\{ \boldsymbol{w}^h \mid \boldsymbol{w}^h \in (C^0(\tilde{\Omega}_h^\lambda))^d \cap (\mathscr{Q}^1(T))^d\,,\ \text{with } T \in \tilde{\mathscr{T}}_h^\lambda \right\}\ , \tag{12}$$

which will be instrumental in defining the variational formulations described next. Specifically, $\mathscr{Q}^1(T)$ is the set of functions, defined over T, that are tensor product of linear polynomials along each of the coordinate directions. We note that time integration is performed using the BDF2 time integrator.

3.4.1. *Momentum and mass conservation equations.* The shifted boundary variational formulation of the incompressible Navier-Stokes equations is inspired by the corresponding Nitsche formulation and reads:

Find $\boldsymbol{u}^h \in \tilde{\boldsymbol{V}}^h(\tilde{\Omega}_h^\lambda)$ and $p^h \in \tilde{V}^h(\tilde{\Omega}_h^\lambda)$ such that, for any $\boldsymbol{w}^h \in \tilde{\boldsymbol{V}}^h(\tilde{\Omega}_h^\lambda)$ and $q^h \in \tilde{V}^h(\tilde{\Omega}_h^\lambda)$,

$$\begin{aligned} 0 \ &= \mathrm{NS}[\tilde{\Omega}_h^\lambda\,;\tilde{\mathscr{T}}_h^\lambda](\boldsymbol{w}^h, q^h; \boldsymbol{u}^h, p^h) \\ &\quad - \underbrace{\langle \boldsymbol{w}^h, (\frac{2}{Re}\nabla^s \boldsymbol{u}^h - p^h \boldsymbol{I})\tilde{\boldsymbol{n}}\rangle_{\tilde{\Gamma}_{D,h}}}_{\textit{Consistency term}} \\ &\quad - \underbrace{\langle (\frac{2}{Re}\nabla^s \boldsymbol{w}^h + q^h \boldsymbol{I})\tilde{\boldsymbol{n}}, \boldsymbol{u}^h + \nabla \boldsymbol{u}^h\,\boldsymbol{d} - \boldsymbol{u}_D\rangle_{\tilde{\Gamma}_{D,h}}}_{\textit{Adjoint consistency term}} \\ &\quad + \underbrace{\beta\,\frac{1}{Re}\langle h^{-1}(\boldsymbol{w}^h + \nabla \boldsymbol{w}^h\,\boldsymbol{d}), \boldsymbol{u}^h + \nabla \boldsymbol{u}^h\,\boldsymbol{d} - \boldsymbol{u}_D\rangle_{\tilde{\Gamma}_{D,h}}}_{\textit{Penalty term}} \end{aligned} \tag{13}$$

where the residual of the Navier-Stokes equations is defined as:

$$\begin{aligned} \mathrm{NS}[\tilde{\Omega}_h^\lambda\,;\tilde{\mathscr{T}}_h^\lambda](\boldsymbol{w}^h, q^h; \boldsymbol{u}^h, p^h) &= (\boldsymbol{w}^h, \partial_t \boldsymbol{u}^h + \boldsymbol{u}^h \cdot \nabla \boldsymbol{u}^h)_{\tilde{\Omega}_h^\lambda} + \frac{1}{Re}(\nabla^s \boldsymbol{w}^h, \nabla^s \boldsymbol{u}^h)_{\tilde{\Omega}_h^\lambda} \\ &\quad - (\nabla \cdot \boldsymbol{w}^h, p^h)_{\tilde{\Omega}_h^\lambda} + (q^h, \nabla \cdot \boldsymbol{u}^h)_{\tilde{\Omega}_h^\lambda} - (\boldsymbol{w}^h, \boldsymbol{f}^h)_{\tilde{\Omega}_h^\lambda} \\ &\quad - \sum_{T \in \tilde{\mathscr{T}}_h^\lambda} (\boldsymbol{u}^h \cdot \nabla \boldsymbol{w}^h, \boldsymbol{u'})_T + \sum_{T \in \tilde{\mathscr{T}}_h^\lambda} (\boldsymbol{w}^h, \boldsymbol{u'} \cdot \nabla \boldsymbol{u}^h)_T \\ &\quad - \sum_{T \in \tilde{\mathscr{T}}_h^\lambda} (\nabla \boldsymbol{w}^h, \boldsymbol{u'} \otimes \boldsymbol{u'})_T - \sum_{T \in \tilde{\mathscr{T}}_h^\lambda} (\nabla \cdot \boldsymbol{w}^h, p')_T \\ &\quad - \sum_{T \in \tilde{\mathscr{T}}_h^\lambda} (\nabla q^h, \boldsymbol{u'})_T\ , \end{aligned} \tag{14}$$

with $\nabla^s \boldsymbol{u}^h = (\nabla \boldsymbol{u}^h + (\nabla \boldsymbol{u}^h)^t)/2$ the symmetric part of the gradient of $\boldsymbol{u}^h$. The primed terms (e.g., $\boldsymbol{u'}$, p')] represent fine-scale variables associated with the VMS formulation and are defined, according to the Variational Multiscale (VMS) literature, as follows:

$$\boldsymbol{u'} = -\tau_M \boldsymbol{r}_M(\boldsymbol{u}^h, p^h) \ , \tag{15a}$$

$$p' = -\tau_C r_C(\boldsymbol{u}^h) \ , \tag{15b}$$

where

$$\boldsymbol{r}_M = \partial_t \boldsymbol{u}^h + \boldsymbol{u}^h \cdot \nabla \boldsymbol{u}^h + \nabla p^h - \frac{1}{Re}\nabla^2 \boldsymbol{u}^h - \boldsymbol{f} \ , \tag{15c}$$

$$r_C = \nabla \cdot \boldsymbol{u}^h \ , \tag{15d}$$

$$\tau_M = \left(\frac{4}{\Delta t^2} + \boldsymbol{u}^h \cdot \boldsymbol{G}\boldsymbol{u}^h + \frac{C_M}{Re^2}\boldsymbol{G} : \boldsymbol{G}\right)^{-\frac{1}{2}} \ , \tag{15e}$$

$$\tau_C = (\tau_M \boldsymbol{g} \cdot \boldsymbol{g})^{-1} \ , \tag{15f}$$

$$G_{ij} = \sum_{k=1}^{d} \frac{\partial \xi_k}{\partial x_i}\frac{\partial \xi_k}{\partial x_j} \ , \tag{15g}$$

$$g_i = \sum_{j=1}^{d} \frac{\partial \xi_j}{\partial x_i} \ . \tag{15h}$$

The residuals $\boldsymbol{r}_M$ and r_C are constructed replacing the coarse-scale (numerical) solutions $\boldsymbol{u}^h$ and p^h inside the momentum and continuity equations, respectively. The constant C_M is typically assigned a value of 36. Additionally, the quantities G_{ij} and g_i are related to the isoparametric mapping between the reference element and its physical counterpart in the computational domain. The parameter β in the last term of Eq. 13 is a penalty that, if sufficiently large, ensures the numerical stability of the overall formulation.

3.4.2. *Heat transfer equation.* In this work, we pair the incompressible Navier-Stokes equations with a heat transport equation, function of a non-dimensional temperature θ. The weak form of the heat transfer equation incorporates convection (forced, mixed, or natural) as follows:

Forced or Mixed Convection: *Find* $\theta^h \in \tilde{V}^h(\tilde{\Omega}_h^\lambda)$, *such that, for all* $\phi^h \in \tilde{V}^h(\tilde{\Omega}_h^\lambda)$,

$$\begin{aligned}
0 = {} & (\phi^h, \partial_t \theta^h + \boldsymbol{u}^h \cdot \nabla \theta^h)_{\tilde{\Omega}_h^\lambda} + \frac{1}{Pe}(\nabla \phi^h, \nabla \theta^h)_{\tilde{\Omega}_h^\lambda} - (\phi^h, q)_{\tilde{\Omega}_h^\lambda} \\
& + \sum_{T \in \tilde{\mathscr{T}}_h^\lambda} \tau_{\mathrm{SUPG}}(\boldsymbol{u}^h \cdot \nabla \phi^h, \partial_t \theta^h + \boldsymbol{u}^h \cdot \nabla \theta^h - \frac{1}{Pe}\Delta \theta^h - q)_T \ , \\
& - \underbrace{\langle \phi^h, \frac{1}{Pe}\nabla \theta^h \cdot \tilde{\boldsymbol{n}} \rangle_{\tilde{\Gamma}_{D,h}}}_{\text{Consistency term}} - \underbrace{\langle \frac{1}{Pe}\nabla \phi^h \cdot \tilde{\boldsymbol{n}}, \theta^h + \nabla \theta^h \cdot \boldsymbol{d} - \theta_D \rangle_{\tilde{\Gamma}_{D,h}}}_{\text{Adjoint consistency term}} \\
& + \underbrace{\alpha \frac{1}{Pe} \langle h^{-1}(\phi^h + \nabla \phi^h \cdot \boldsymbol{d}), \theta^h + \nabla \theta^h \cdot \boldsymbol{d} - \theta_D \rangle_{\tilde{\Gamma}_{D,h}}}_{\text{Penalty term}} \ .
\end{aligned} \tag{16}$$

Natural Convection: *Find* $\theta^h \in \tilde{V}^h(\tilde{\Omega}_h^\lambda)$, *such that, for all* $\phi^h \in \tilde{V}^h(\tilde{\Omega}_h^\lambda)$,

$$\begin{aligned}
0 = &(\phi^h, \partial_t\theta^h + \boldsymbol{u}^h \cdot \nabla\theta^h)_{\tilde{\Omega}_h^\lambda} + \frac{1}{\sqrt{Pr \cdot Ra}}(\nabla\phi^h, \nabla\theta^h)_{\tilde{\Omega}_h^\lambda} - (\phi^h, q)_{\tilde{\Omega}_h^\lambda} \\
&+ \sum_{T \in \tilde{\mathcal{T}}_h^\lambda} \tau_{\text{SUPG}}(\boldsymbol{u}^h \cdot \nabla\phi^h, \partial_t\theta^h + \boldsymbol{u}^h \cdot \nabla\theta^h - \frac{1}{\sqrt{Pr \cdot Ra}}\Delta\theta^h - q)_T, \\
&- \underbrace{\langle \phi^h, (PrRa)^{-0.5}\nabla\theta^h \cdot \tilde{\boldsymbol{n}} \rangle_{\tilde{\Gamma}_{D,h}}}_{\text{Consistency term}} - \underbrace{\langle (PrRa)^{-0.5}\nabla\phi^h \cdot \tilde{\boldsymbol{n}}, \theta^h + \nabla\theta^h \cdot \boldsymbol{d} - \theta_D \rangle_{\tilde{\Gamma}_{D,h}}}_{\text{Adjoint consistency term}} \\
&+ \underbrace{\alpha\,(PrRa)^{-0.5} \langle h^{-1}(\phi^h + \nabla\phi^h \cdot \boldsymbol{d}), \theta^h + \nabla\theta^h \cdot \boldsymbol{d} - \theta_D \rangle_{\tilde{\Gamma}_{D,h}}}_{\text{Penalty term}} . \qquad (17)
\end{aligned}$$

Here, θ_D represents the prescribed non-dimensional temperature at Dirichlet boundaries, and the consistency, adjoint consistency, and penalty terms are included to ensure stability and accuracy in the shifted boundary approach for heat transfer. The parameter α in Eq. 16 and Eq. 17 serves as a penalty parameter for the SBM in the heat transfer equation.

Forced convection refers to flow dominated by inertial effects, while natural convection is driven primarily by buoyancy forces due to temperature differences. Mixed convection lies between these two extremes, incorporating aspects of both. Here, Pr represents the Prandtl number, which is the ratio of momentum diffusivity to thermal diffusivity. For the current simulation, $Pr = 0.7$ (air) is used. Pe denotes the Peclet number, which can be expressed as $Re \times Pr$. Ra is the Rayleigh number, a dimensionless number characterizing buoyancy-driven flow. In our framework, which employs linear finite element basis functions, the term $\Delta\theta^h$ is negligible and is therefore omitted in the calculations. The term τ_{SUPG} refers to the SUPG stabilization parameter, which is calculated as:

$$\tau_{\text{SUPG}} = \frac{hz}{2\|\boldsymbol{u}^h\|} , \qquad (18)$$

with h the element length, determined as:

$$h = \frac{2}{\sum_A \frac{|\boldsymbol{u}^h \cdot \nabla N_A|}{\|\boldsymbol{u}^h\|}} , \qquad (19)$$

where ∇N_A denotes the gradient of the shape function N_A, and the index A runs over all the nodes of element T. The parameter z is defined based on the local Reynolds number, $Re_u = \frac{\|\boldsymbol{u}^h\|h}{2\nu}$, as follows:

$$z = \begin{cases} 1 & \text{if } Re_u > 3 , \\ \frac{Re_u}{3} & \text{if } Re_u \leq 3 . \end{cases} \qquad (20)$$

To couple the Navier-Stokes and Heat Transfer equations, we employ a block-iterative strategy [83, 94, 45]. In this approach, the convection-diffusion equation is solved first, and the resulting temperature field is passed to the Navier-Stokes equation, which is then solved for velocity and pressure. Convergence is assessed within this iterative block. If the convergence criterion is not satisfied, the velocity field obtained from the Navier-Stokes solution is fed back into the convection-diffusion equation, and the process repeats. This iterative loop continues until both equations converge to a consistent solution.

When addressing the coupling, it is essential to revisit Eq. 14, where the forcing term $\boldsymbol{f}^h$ depends on the non-dimensional temperature field θ^h. Specifically, the formulation of $\boldsymbol{f}^h$ varies based on the convection regime, as shown below:

$$\boldsymbol{f}^h = \begin{cases} \theta^h \boldsymbol{e}_{\hat{\boldsymbol{g}}}, & \text{Natural convection;} \\ 0, & \text{Forced convection,} \\ \dfrac{Gr}{Re^2}\theta^h \boldsymbol{e}_{\hat{\boldsymbol{g}}} = Ri\,\theta^h \boldsymbol{e}_{\hat{\boldsymbol{g}}}, & \text{Mixed convection,} \end{cases} \tag{21}$$

Here, $\boldsymbol{e}_{\hat{\boldsymbol{g}}}$ represents the unit vector in the direction of gravity, and Ri, the Richardson number, quantifies the relative influence of natural and forced convection.

3.5. **Boundary-fitted implementation.** We will also compare the SBM results with the results obtained with boundary-fitted formulations, where the computational grid fits the shape boundaries and boundary conditions are enforced strongly. For the sake of brevity, we omit the corresponding variational formulations, which are obtained removing the "consistency", "adjoint consistency", and "penalty" terms from Eq. 13, Eq. 16, and Eq. 17, and changing the definitions of the function spaces so that $\boldsymbol{u}^h = \boldsymbol{u}_D$ and $\boldsymbol{w}^h = \boldsymbol{0}$ strongly on the Dirichlet boundary.

3.6. **Backflow stabilization.** Backflow stabilization [30, 50, 31, 37, 16, 13, 28, 54] mitigates instabilities at outflow or open boundaries. Flow recirculation or vortices can lead to backflow, destabilizing the simulation. Backflow stabilization introduces a dissipative boundary term that activates when backflow occurs, maintaining stability. In this work, we implement the methods proposed in [30], and add the following term to the right-hand side of (14):

$$-\langle \boldsymbol{w}^h, \beta_o \min(0, \boldsymbol{u}^h \cdot \boldsymbol{n})\boldsymbol{u}^h \rangle_{\Gamma_o}, \tag{22}$$

where β_o is a stabilization parameter for Navier-Stokes backflow stabilization, and Γ_o represents the outflow boundary. Similarly, backflow stabilization can be applied to the heat transfer equation, adding the following term to the right-hand side of Eq. 16 or Eq. 17:

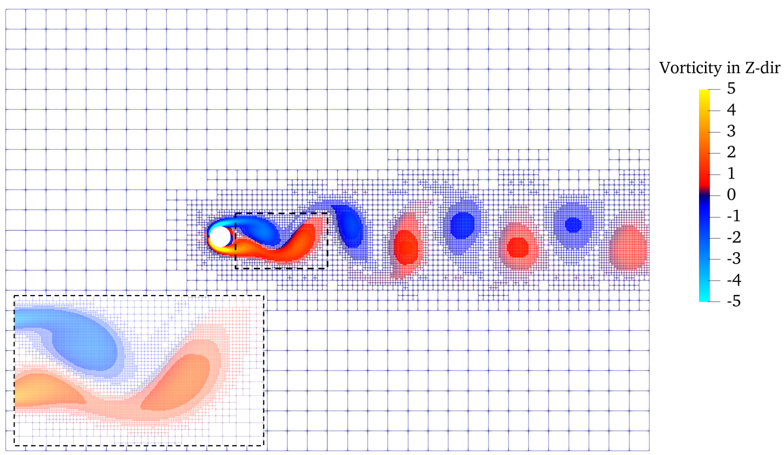
$$-\langle w^h, \beta_\theta \min(0, \boldsymbol{u}^h \cdot \boldsymbol{n})\theta^h \rangle_{\Gamma_o}, \tag{23}$$

where β_θ is a stabilization parameter for heat transfer backflow stabilization.

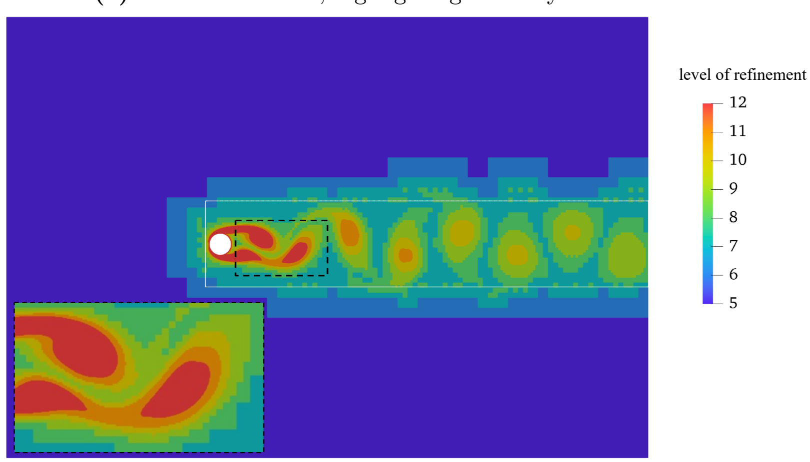
For the simulations in Section 4.4, backflow stabilization for Navier-Stokes and heat transfer equations was applied at the outlet boundary with $\beta_o = 0.5$ and $\beta_\theta = 0.5$. For the simulations in Section 4.5, backflow stabilization for Navier-Stokes equations was applied at the outlet boundary with $\beta_o = 0.5$.

4. **Numerical results.** The results presented in this section are primarily obtained using the SBM with dynamic AMR. However, in certain sections, we compare SBM with dynamic AMR to SBM without dynamic AMR. To distinguish between these cases, we use the term **AMR-SBM** to denote simulations incorporating SBM with dynamic AMR, while **SBM** specifically refers to simulations conducted using SBM on meshes without dynamic AMR. When such comparisons are made, we explicitly indicate this distinction within the respective sections.

4.1. **Two-dimensional unsteady flows past a fixed cylinder.** We study the flow around a circular obstacle at a Reynolds number of 100. The computational domain is defined as a rectangle spanning $[0, 30]$ in the x-direction and $[0, 20]$ in the y-direction, with a circular disk with radius equal to 0.5 centered at $(10, 10)$. At all external boundaries, except for the outlet, the non-dimensional velocity is enforced to match the uniform, freestream velocity $(1, 0, 0)$. At the outlet, the pressure is set to zero, while a no-slip boundary condition is enforced on the circular obstacle using the SBM. The simulation is conducted with a non-dimensional timestep of



(a) Mesh with AMR, highlighting vorticity on the meshes.



(b) Refinement levels during simulation with AMR enabled.

Figure 2 Flow past a two-dimensional circular cylinder at $Re = 100$: mesh with AMR vorticity contours are shown in the top pane, while mesh refinement levels are shown in the bottom pane. The figure highlights the refined mesh regions around the cylinder due to AMR, which optimally reduces mesh elements while retaining flow features.

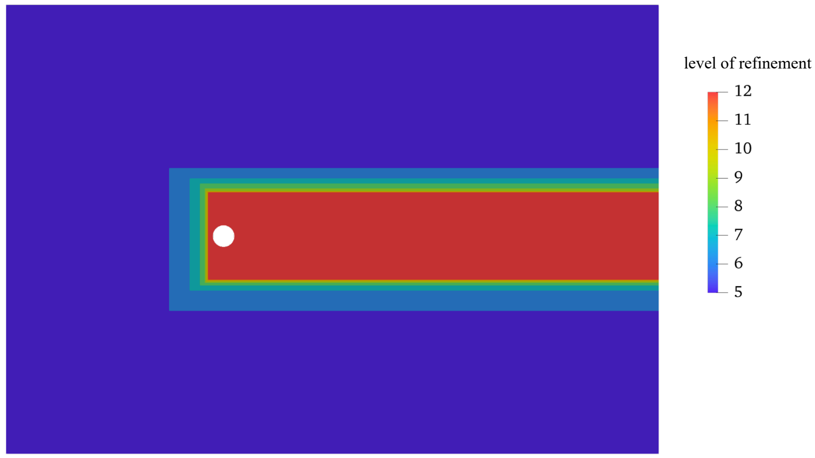


Figure 3 Flow past a two-dimensional circular cylinder at $Re = 100$: mesh with wake refinement (refinement level 12, mesh size 30×2^{-12}).

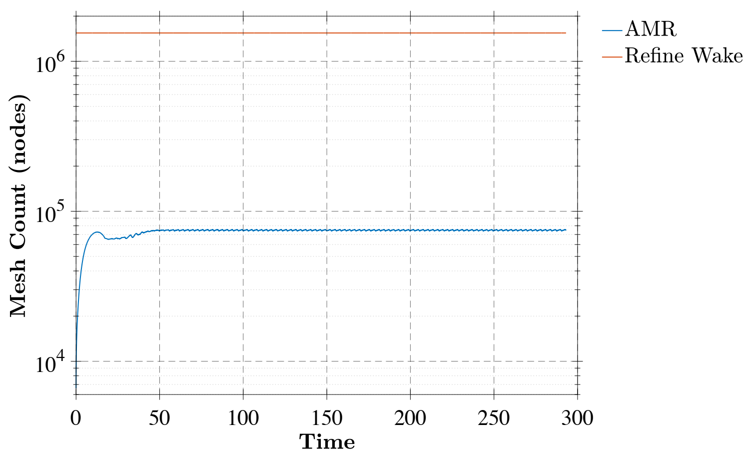


Figure 4 Flow past a two-dimensional circular cylinder at $Re = 100$: history of the number of mesh nodes for AMR versus a regular grid with uniform mesh refinement in the wake region (indicated by the white box in Figure 2b). The significant reduction in node count achieved by AMR emphasizes computational efficiency.

0.01. The base refinement level is set to 5, with boundary layer refinement near the circular disk at levels 12, 13, or 14, to conduct a mesh refinement study. AMR is applied according to equation (1), with parameters $\omega_{\max} = 2$, $\omega_{\min} = 0.1$, $l_{\max} = 12$, and $l_{\min} = 8$.

The AMR grid is shown in Figure 2a and Figure 2b. Figure 4 illustrates the history of mesh sizes for the AMR mesh versus a base mesh (Figure 3) with uniform

TABLE 1. Flow past a two-dimensional circular cylinder at $Re = 100$: comparison of drag coefficient (C_d) and Strouhal number (St). The results of SBM with AMR are consistent with the literature, confirming the accuracy of the proposed approach for different mesh refinement levels.

Study	C_d	St
Liu *et al.* [52]	1.350	0.1650
Lai *et al.* [49]	1.447	0.1650
Uhlmann [85]	1.453	0.1690
Yang *et al.* [98]	1.393	0.1650
Kamensky *et al.* [40]	1.386	0.1700
Current (finest element size = $30/2^{12}$)	1.401	0.1709
Current (finest element size = $30/2^{13}$)	1.404	0.1707
Current (finest element size = $30/2^{14}$)	1.405	0.1707

TABLE 2. Flow past a two-dimensional circular cylinder at $Re = 100$: Comparison of the drag coefficient (C_d) and Strouhal number (St). The term **AMR-SBM** refers to simulations conducted with dynamic AMR and the SBM, while **SBM** denotes simulations using SBM with local mesh refinement. Both AMR-SBM and SBM simulations are performed with a base refinement level of 5 and a boundary layer refinement level of 12 near the circular disk. For SBM (without AMR), we locally refine the region up to $level_{wake}$ just upstream of the circular disk and throughout the wake region, as shown by the white box in Figure 2b. Below, we present SBM simulations with $level_{wake}$ set to 10, 11, and 12, and compare the results with AMR-SBM. We treat the SBM simulation with the highest refinement level in the upstream and wake region as the ground truth and use it to calculate $Error_{C_d} = \frac{|C_d - C_d^{\text{ground truth}}|}{|C_d^{\text{ground truth}}|}$ and $Error_{St} = \frac{|St - St^{\text{ground truth}}|}{|St^{\text{ground truth}}|}$.

Study	Highest Mesh Nodes	C_d	St	$Error_{C_d}$	$Error_{St}$
AMR-SBM	75464	1.401	0.1709	**0.214%**	**0.0%**
SBM ($level_{wake} = 9$)	29053	1.507	0.1685	7.336%	1.404%
SBM ($level_{wake} = 10$)	101694	1.411	0.1717	0.499%	0.468%
SBM ($level_{wake} = 11$)	389987	1.410	0.1717	0.427%	0.468%
SBM ($level_{wake} = 12$)	1544191	1.404	0.1709	0.0%	0.0%

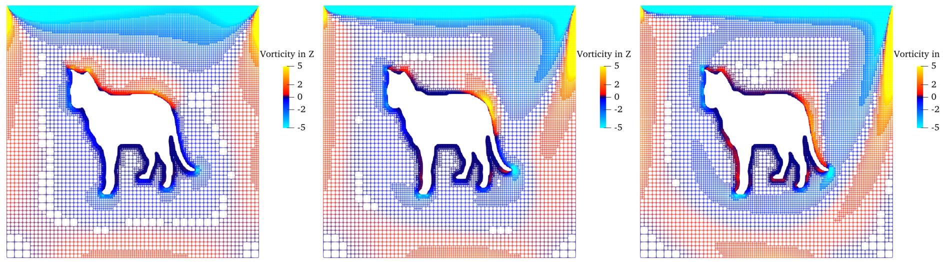
refinement in the wake region (indicated with a white box in Figure 2b). We observe that AMR reduces the number of mesh nodes by over 85%.

We achieve values of the drag coefficient (C_d) and Strouhal number (St) close to those reported in the literature, as shown in Table 1, despite using significantly fewer mesh elements than a uniformly refined mesh in the wake region. Table 1 also presents results from three different refinement levels at the boundary of the circular disk (levels 12, 13, and 14), and we observe mesh convergence. To further evaluate the effectiveness of AMR in reducing computational costs while maintaining accuracy, we compare the results of AMR-SBM with SBM simulations that use local mesh refinement in the wake region (without AMR). Table 2 presents the drag coefficient (C_d) and Strouhal number (St) obtained from these simulations. Despite using significantly fewer mesh nodes, AMR-SBM achieves results that closely match those of SBM with a wake-region refinement level of 12. Notably, AMR-SBM employs fewer mesh nodes than SBM simulations with wake-region refinement levels of 9, 10, and 11 (see Table 2), yet it yields C_d and St values that are closer to those of SBM with a refinement level of 12, which serves as the reference solution in this study. This demonstrates that AMR effectively captures the relevant flow physics without requiring excessive mesh resolution. Thus, AMR proves to be an efficient approach for dynamically refining critical regions, reducing overall computational costs while maintaining accuracy comparable to that of a globally refined mesh.

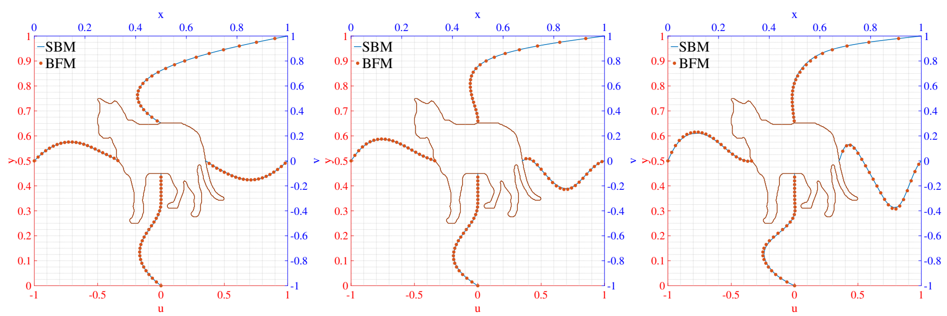
4.2. **Lid-driven cavity flow with obstacles.** We conduct simulations using a complex geometry positioned at the center of a lid-driven cavity. The top wall has a velocity boundary condition of $(1, 0)$, while the other three walls are set to no-slip conditions. The boundary-fitted mesh (for comparing against SBM results) is generated using Gmsh, with an approximate resolution of 2^{-9}. We keep the base refinement level of the mesh as 5 (element size = 2^{-5}) and utilize AMR. As illustrated in Figure 5, the SBM yields a velocity profile that closely matches the results obtained with a boundary-fitted method (BFM), using a quasi-uniform grid near the boundary (grid generation with Gmsh). Compared to the uniform octree mesh with a size of 2^{-9}, the reduction in mesh nodes achieved using AMR is shown in Table 3. We chose a refinement level of 9 (mesh size = 2^{-9}) as the base uniform mesh for comparison based on the mesh refinement study conducted in [97]. As demonstrated in [97], lid-driven cavity flow simulations with a circular disk at the center showed that a uniform mesh with a refinement level of 9 produced grid-independent results. We performed a mesh convergence study using SBM with uniform meshes at different refinement levels, as shown in Figure 6. AMR-SBM results are also included in the plot for comparison. We observe that AMR-SBM outperforms most SBM simulations, except for the case with a uniform mesh refinement level of 9. The number of mesh nodes for a uniform mesh with a refinement level of 8 is 60185. AMR-SBM achieves approximately a 2× reduction in the number of mesh nodes compared to the uniform mesh at refinement level 8, while still producing more accurate results. The AMR parameters are $\omega_{max} = 5$, $\omega_{min} = 0.1$, $l_{max} = 10$, and $l_{min} = 6$. The base refinement level is set to 5.

Table 3. Lid-driven cavity flow with complex internal geometries: grid size comparison (in terms of mesh nodes) for uniform mesh versus AMR.

Reynolds Number	**Uniform Mesh**	**AMR Mesh**	**Compression Ratio**
50		34019	6.99
500	238100	34487	6.90
1000		34676	6.87



(a) AMR, $Re = 50$ **(b)** AMR, $Re = 500$ **(c)** AMR, $Re = 1000$



(d) Velocity, $Re = 50$ **(e)** Velocity, $Re = 500$ **(f)** Velocity, $Re = 1000$

Figure 5 Lid-driven cavity flow with complex internal geometries: velocity profiles. The first row of pictures displays an overlay of the vorticity with the AMR mesh, while the second row displays velocity profiles around a cat-shaped obstacle, comparing results between the SBM and a BFM. These simulations, conducted at Reynolds numbers of $Re = 50$, $Re = 500$, and $Re = 1000$, illustrate how both the obstacle shape and Reynolds number affect the flow dynamics.

4.3. **Mixed convection inside a lid-driven cavity.** This is the first test in our battery of tests in which we couple the Navier-Stokes with the heat transfer equations (NSHT). A circular disk with temperature boundary condition $\theta = 0$ is placed inside a lid-driven cavity. We perform simulations in the regime of mixed convection, in which natural and forced convection effects are equally important. The constant wall temperature boundary condition and no-slip boundary condition on the circular disk are enforced using the SBM. The other boundary conditions are the same as in Section 4.2. The temperature boundary conditions are set as follows: the bottom wall has a non-dimensional temperature of 1, the top wall has a non-dimensional temperature of 0, and the left and right walls are assigned zero-flux boundary conditions. The non-dimensional parameters used are $Re = 100$, $Pr = 0.7$, and Ri values ranging from 0.01 to 5.0, covering the flow regime from forced convection to mixed convection and natural convection. The AMR parameters are $\omega_{max} = 5$, $\omega_{min} = 0.1$, $l_{max} = 7$, and $l_{min} = 5$. The base refinement level is set to 4.

The simulation results for velocity magnitude and temperature, along with the AMR mesh after the flow reaches a steady state, are presented in Figure 7. Furthermore, the temperature profiles along $x = 0.15$, $x = 0.85$, $y = 0.15$, and $y = 0.85$

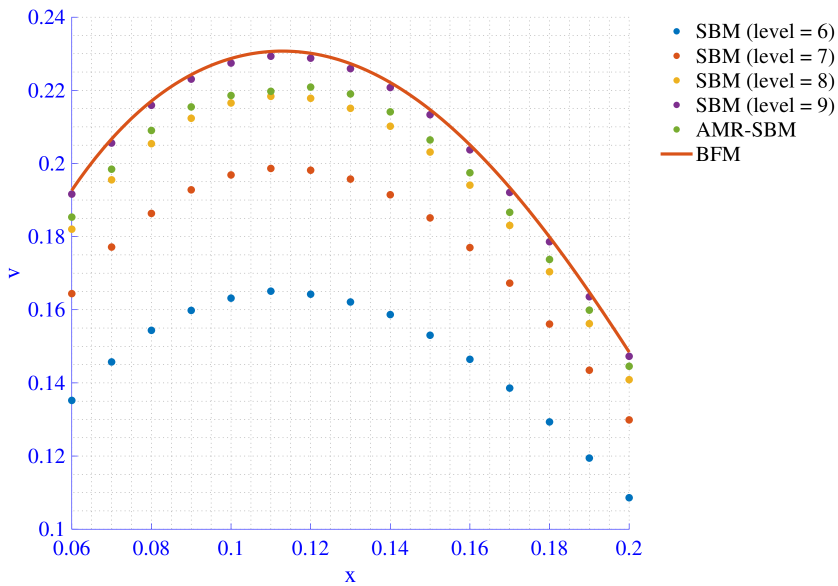


Figure 6 Lid-driven cavity flow with complex internal geometries: Plot of the y-direction velocity (v) along $y = 0.5$, with x ranging from 0.06 to 0.2 inside the flow chamber. Comparison of AMR-SBM and SBM at different uniform refinement levels. **AMR-SBM** refers to simulations using dynamic AMR with SBM, while **SBM** represents simulations with uniform meshes. AMR-SBM outperforms most SBM simulations with uniform meshes, except for the case with a uniform mesh refinement level of 9.

are compared with the literature [18] in Figure 8. The mesh compression ratios achieved using AMR for different Richardson numbers are detailed in Table 4. We selected a uniform mesh with a refinement level of 7 (mesh size $= 2^{-7}$) as the benchmark for comparison in Table 4 based on two previous studies that conducted similar tests. In [44], a boundary-fitted grid with 19520 nodes was shown to produce grid-independent results. The closest equivalent in our uniform mesh setup consists of 14780 nodes, making it a reasonable choice. Additionally, in [18], mixed convection lid-driven cavity flow simulations were performed using a grid size of 251^{-1}. Our uniform mesh size (2^{-7}) is slightly coarser but remains comparable. Furthermore, our results with the 2^{-7} uniform mesh align well with the literature, as demonstrated in [96]. To demonstrate our AMR-SBM framework in comparison with SBM without AMR, we include Figure 9, which shows that AMR-SBM achieves good agreement with SBM without AMR while requiring fewer mesh nodes, as presented in Table 4. These results demonstrate that our AMR approach enables efficient and accurate simulations.

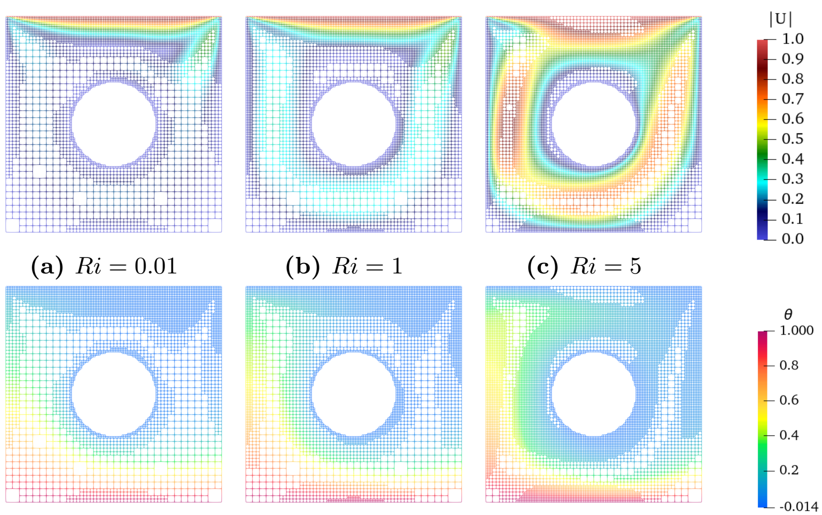
4.4. **Natural convection past a star-shaped domain.** This problem involves natural convection at Rayleigh number of 10^6 in a fluid domain of size $[0, 16] \times [0, 16]$,

Table 4. Mixed convection inside a lid-driven cavity: grid size comparison (in terms of mesh nodes) for uniform mesh versus AMR.

Richardson number	**Uniform Mesh**	**AMR Mesh**	**Compression Ratio**
0.01		4252	3.48
1	14780	5237	2.82
5		9430	1.57

where a hot star-shaped object is placed at coordinates (8, 2.5). The initial conditions are $\theta = 0$ and $\boldsymbol{u}^h = \boldsymbol{0}$. The boundary conditions applied to the chamber are as follows: on the left and right sides, a no-penetration condition (or impermeable wall condition, i.e., $\boldsymbol{u}^h \cdot \boldsymbol{n} = \boldsymbol{0}$ with $\boldsymbol{n}$ the outward unit boundary normal) is imposed for the Navier-Stokes equations.

We employ a zero-flux boundary condition on the left and right walls for the heat transfer equation. On the bottom side, a no-penetration condition is enforced for the Navier-Stokes equations, and a Dirichlet condition with $\theta = 0$ is applied for the heat transfer equation. On the top side, backflow stabilization is applied for both the Navier-Stokes and heat-trasfer equations. A no-slip velocity boundary condition and a temperature $\theta = 1$ are enforced on the star-shaped object using the SBM.



(a) $Ri = 0.01$ **(b)** $Ri = 1$ **(c)** $Ri = 5$

(d) $Ri = 0.01$ **(e)** $Ri = 1$ **(f)** $Ri = 5$

Figure 7 Mixed convection inside a lid-driven cavity: Velocity magnitude distribution (first row of plots) and temperature distribution (second row of plots), for varying Richardson numbers (Ri). These plots demonstrates how changes in Ri affect the temperature field, with the AMR approach capturing all essential temperature gradients and flow patterns.

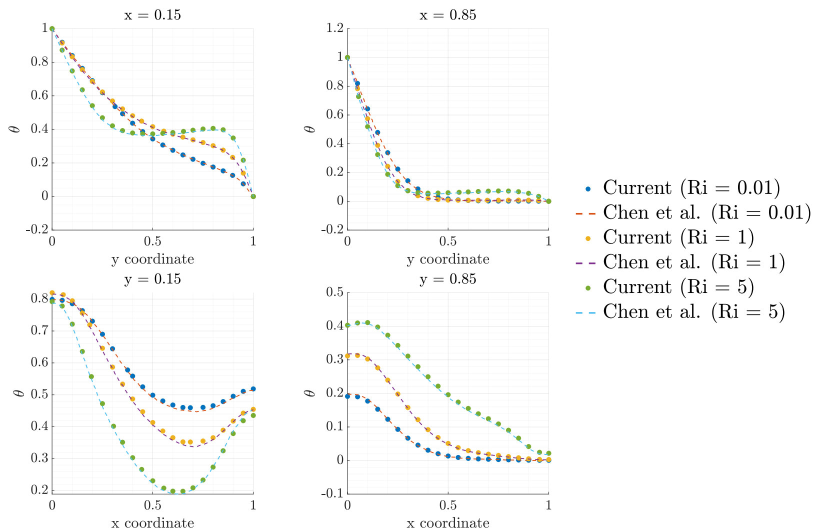


Figure 8 Mixed convection inside a lid-driven cavity: temperature distribution for varying Richardson numbers (Ri). The figure demonstrates how changes in Ri affect the temperature field, with the AMR approach capturing essential temperature gradients.

The region near the star shape is refined to a level of 12, corresponding to an element size of 16×2^{-12}, while the rest of the domain is refined to level 5, with an element size of 16×2^{-5}. The AMR parameters are $\omega_{\max} = 2$, $\omega_{\min} = 0.1$, $l_{\max} = 10$, and $l_{\min} = 7$. Figure 10 and Figure 11 illustrate the vorticity and temperature contours at four non-dimensional times, corresponding to $t = 0$, 10, 20, and 30. This test case demonstrates that vorticity-based refinement for the Navier-Stokes equations also benefits the solution of the heat transfer equation, as regions with strong vorticity coincide with regions of high temperature gradients. The problem in this section is designed to demonstrate the capability of our framework in handling thermal flow simulations involving complex geometries. Due to the complexity of these geometries, direct comparisons with existing literature are limited. For further validation of natural convection simulations using Octree-SBM, we refer the reader to [96].

4.5. **Flow past a sphere at** $Re = 300$**.** To simulate fluid flow past a sphere at a Reynolds number (Re) of 300, we set up a computational domain as described in [2], composed of a rectangular region spanning $[0, 25] \times [0, 10] \times [0, 10]$ with a sphere of radius 0.5 centered at coordinates (3, 5, 5). Our mesh refinement strategy closely mirrors that of [2], in which several mesh refinement layers are progressively laid near the sphere and along its wake. Specifically, we keep the base mesh refinement at level 7 (element size $= 25 \times 2^{-7}$) and use three concentric spheres centered at (3, 5, 5). The innermost sphere has a radius of 1 and a refinement level of 10 (with element size of 25×2^{-10}), the next sphere has a radius of 1.5 and a refinement level of 9 (with an element size of 25×2^{-9}), and the outer sphere has a radius

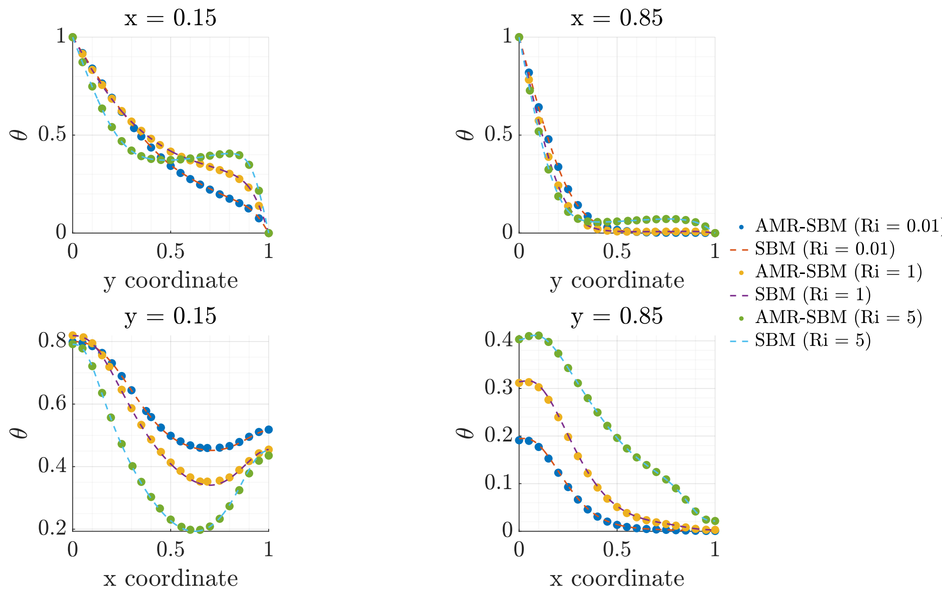


Figure 9 Mixed convection inside a lid-driven cavity: Temperature distribution for different Richardson numbers (Ri). This figure compares results obtained using the SBM with AMR and uniform meshes. The term **AMR-SBM** refers to simulations performed with dynamic AMR and SBM, whereas **SBM** denotes simulations using SBM with uniform octree meshes.

of 2 with a refinement level of 8 (with an element size of 25×2^{-8}). Near the sphere, to ensure a detailed capture of the flow dynamics, we employ a finer mesh with a refinement level of 13, which results in an element size of 25×2^{-13}, as shown in Figure 12. The AMR parameters are $\omega_{max} = 2$, $\omega_{min} = 0.1$, $l_{max} = 10$, and $l_{min} = 8$. The left plot (Figure 12a) displays three concentric spherical slices of the initial refinement setup prior to the application of AMR. The right plot (Figure 12b) focuses on the refined mesh near the sphere boundary, highlighting the level of detail in this region. We use a non-dimensional timestep of 0.025 for the simulation. Backflow stabilization is applied at the outlet, while all other walls are subjected to a uniform non-dimensional freestream velocity of (1, 0, 0). The mesh obtained after applying AMR is shown in Figure 13a, alongside visualizations of the Q-criteria of the flow in Figure 13b. These visualizations demonstrate that in regions of higher flow rotation, the AMR mesh achieves finer resolution, effectively capturing small vortex structures.

When compared to the existing references in terms of time-averaged drag coefficient (C_d) and Strouhal number (St), the present results match closely with the reported values, as shown in Table 5. The time-averaged velocity profile also matches well with the results of [84], as seen in Figure 13c. It is important to note that the x-coordinate in Figure 13c is measured downstream from the sphere's centroid, consistent with the work of [84] and [42]. Both the instantaneous and time-averaged velocity profiles demonstrate that our framework is in strong agreement with findings in the literature.

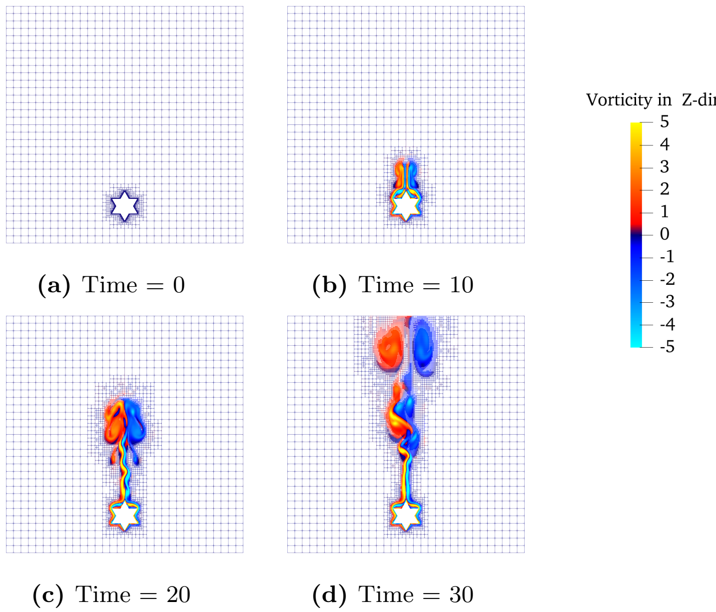


Figure 10 Natural convection past a star-shaped domain: vorticity contours over time. The vorticity-based AMR allows for accurate tracking of vorticity at different non-dimensional times.

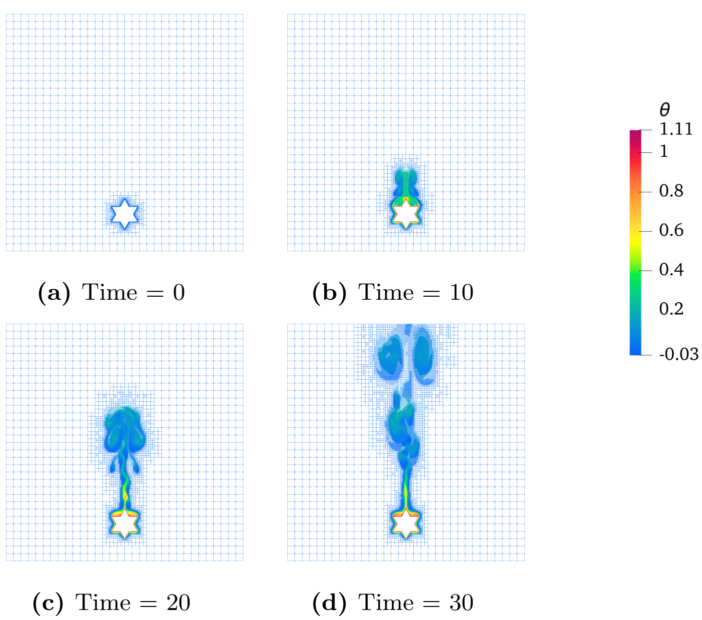
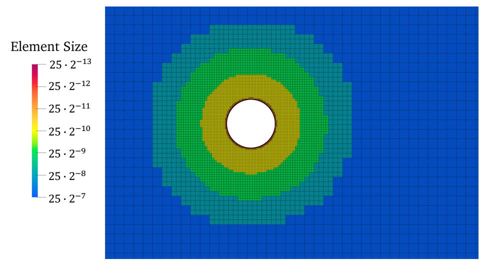
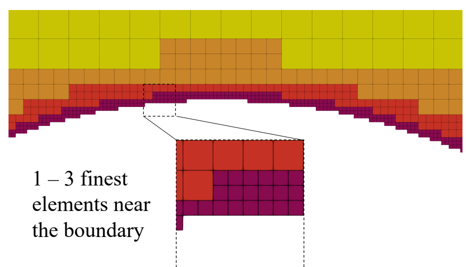


Figure 11 Natural convection past a star-shaped domain: temperature contours at various non-dimensional times. The AMR captures well the evolution of high thermal gradients over time.



(a) Overview of the initial refinement strategy prior to applying AMR.

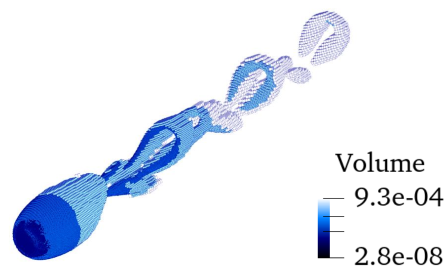


(b) Close-up of the refinement near the boundary.

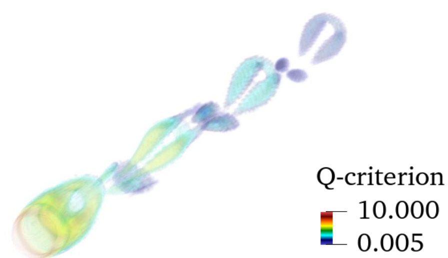
Figure 12 Visualization of the initial mesh refinement setup for the flow past a sphere, highlighting the distribution before adaptive refinement is introduced.

TABLE 5. Flow past a sphere at $Re = 300$: comparison of the drag coefficient (C_d) and Strouhal number (St) against various references. The proposed combined SBM-AMR approach compares well against previous experimental and computational studies.

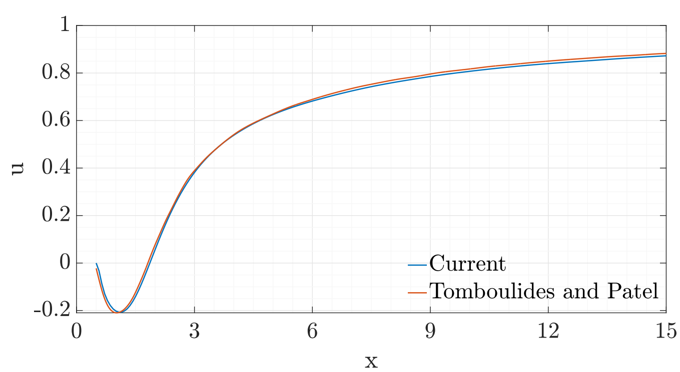
Study	**Cd**	**St**
Roos and Willmarth [69] (interpolated experiment value)	0.629	-
Le Clair *et al.* [51]	0.632	-
Johnson and Patel [38]	0.656	0.137
Marella *et al.* [57]	0.621	0.133
Vanella *et al.* [86]	0.634	0.132
Wang and Zhang [89]	0.680	0.135
Angelidis *et al.* [2]	0.665	0.132
Kang *et al.* [42]	0.663	0.134
Current	0.622	0.134



(a) Octree element volume.



(b) Visualization of the Q-criterion.



(c) Centerline average streamwise velocity profile compared to literature [84] for flow past a sphere at $Re = 300$.

Figure 13 Flow past a sphere at $Re = 300$. (a) and (b) illustrate octree element volumes and the Q-criterion respectively, while (c) compares the centerline average streamwise velocity profile with benchmark data [84].

5. **Conclusions.** In this study, we combined the SBM with AMR to enhance computational efficiency in complex flow simulations. Our case studies, including fluid flow around obstacles and natural/mixed convection problems, demonstrate how AMR effectively concentrates computational resources in critical regions. This targeted approach captures essential flow features while optimizing computational costs. The implementation of vorticity-guided mesh refinement helps preserve both vortical and thermal structures, while the SBM ensures accurate boundary condition enforcement on irregular domains. The combined SBM-AMR approach has proven effective across various fluid dynamics applications. Looking ahead, we plan to extend this framework to handle moving boundary problems and FSI simulations, opening new possibilities for multiphysics applications.

Use of generative-AI tools declaration. The authors used LLMs (ChatGPT) to help with clarity and grammar of the text, and to assist with formatting and consistency in the reference section.

Acknowledgments. This work was partly supported by the National Science Foundation under the grants LEAP-HI 2053760, DMREF 2323715/2323716 (BG, AK, CHY), and DMS 2207164 and DMS 2409919 (GS). BG, AK, and CHY are supported in part by AI Research Institutes program supported by NSF and USDA-NIFA under AI Institute for Resilient Agriculture, grant 2021-67021-35329. We gratefully acknowledge computing support through TACC via the NAIRR and ACCESS programs.

REFERENCES

[1] A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell and M. L. Welcome, A conservative adaptive projection method for the variable density incompressible navier–stokes equations, *Journal of computational Physics*, **142** (1998), 1-46.

[2] D. Angelidis, S. Chawdhary and F. Sotiropoulos, Unstructured cartesian refinement with sharp interface immersed boundary method for 3D unsteady incompressible flows, *Journal of Computational Physics*, **325** (2016), 272-300.

[3] A. Ankit, J. Zhang, S. Eisenträger and C. Song, An octree pattern-based massively parallel pcg solver for elasto-static and dynamic problems, *Computer Methods in Applied Mechanics and Engineering*, **404** (2023), 115779.

[4] N. Antonelli, R. Aristio Andrea Gorgi, R. Zorrilla, R. Rossi, G. Scovazzi and R. Wüchner, The shifted boundary method in isogeometric analysis, *Computer Methods in Applied Mechanics and Engineering*, **430** (2024), 117228.

[5] N. Atallah, C. Canuto and G. Scovazzi, Analysis of the shifted boundary method for the poisson problem in domains with corners, *Mathematics of Computation*, **90** (2021), 2041-2069.

[6] N. Atallah, C. Canuto and G. Scovazzi, The shifted boundary method for solid mechanics, *International Journal for Numerical Methods in Engineering*, **122** (2021), 5935-5970.

[7] N. M. Atallah, C. Canuto and G. Scovazzi, The second-generation shifted boundary method and its numerical analysis, *Computer Methods in Applied Mechanics and Engineering*, **372** (2020), 113341.

[8] N. M. Atallah, C. Canuto and G. Scovazzi, The high-order shifted boundary method and its analysis, *Computer Methods in Applied Mechanics and Engineering*, **394** (2022), 114885.

[9] J. Baiges, R. Codina, A. Pont and E. Castillo, An adaptive fixed-mesh ale method for free surface flows, *Computer Methods in Applied Mechanics and Engineering*, **313** (2017), 159-188.

[10] A. Balu, M. R. Rajanna, J. Khristy, F. Xu, A. Krishnamurthy and M.-C. Hsu, Direct immersogeometric fluid flow and heat transfer analysis of objects represented by point clouds, *Computer Methods in Applied Mechanics and Engineering*, **404** (2023), 115742.

[11] M. J. Berger and P. Colella, Local adaptive mesh refinement for shock hydrodynamics, *Journal of computational Physics*, **82** (1989), 64-84.

[12] M. J. Berger and J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, *Journal of Computational Physics*, **53** (1984), 484-512.

[13] C. Bertoglio and A. Caiazzo, A tangential regularization method for backflow stabilization in hemodynamics, *Journal of Computational Physics*, **261** (2014), 162-171.

[14] J. Bielak, O. Ghattas and E. J. Kim, Parallel octree-based finite element method for large-scale earthquake ground motion simulation, *Computer Modeling in Engineering and Sciences*, **10** (2005), 99-112.

[15] I. Borazjani, L. Ge and F. Sotiropoulos, Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3D rigid bodies, *Journal of Computational Physics*, **227** (2008), 7587-7620.

[16] M. Braack and P. Boguslaw Mucha, Directional do-nothing condition for the Navier-Stokes equations, *Journal of Computational Mathematics*, **32** (2014), 507-521.

[17] G. L. Bryan, M. L. Norman, B. W. O'Shea, T. Abel, J. H. Wise, M. J. Turk, D. R. Reynolds, D. C. Collins, P. Wang, S. W. Skillman, et al, Enzo: An adaptive mesh refinement code for astrophysics, *The Astrophysical Journal Supplement Series*, **211** (2014), 19.

[18] Z. Chen, C. Shu, L. Yang, X. Zhao and N. Liu, Immersed boundary–simplified thermal lattice Boltzmann method for incompressible thermal flows, *Physics of Fluids*, **32** (2020), 01.

[19] J. Cheng, F. Yu and L. T. Zhang, OpenIFEM: A high performance modular open-source software of the immersed finite element method for fluid-structure interactions, *Computer Modeling in Engineering and Sciences: CMES*, **119** (2019), 91.

[20] W. J. Coirier, *An adaptively-refined, Cartesian, Cell-based Scheme for the Euler and Navier-Stokes Equations*, University of Michigan, 1994.

[21] P. Colella, D. T. Graves, T. J. Ligocki, D. F. Martin, D. Modiano, D. B. Serafini and B. Van Straalen, Chombo software package for amr applications design document, *Available at the Chombo Website: http://seesar. lbl. gov/ANAG/chombo/(September 2008)*, **2** (2009).

[22] J. H. Collins, A. Lozinski and G. Scovazzi, A penalty-free shifted boundary method of arbitrary order, *Computer Methods in Applied Mechanics and Engineering*, **417** (2023), 116301.

[23] O. Colomés, A. Main, L. Nouveau and G. Scovazzi, A weighted shifted boundary method for free surface flow problems, *Journal of Computational Physics*, **424** (2021), 109837.

[24] T. Colonius and K. Taira, A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions, *Computer Methods in Applied Mechanics and Engineering*, **197** (2008), 2131-2146.

[25] M. S. Day and J. B. Bell, Numerical simulation of laminar reacting flows with complexchemistry, *Combustion Theory and Modelling*, **4** (2000), 535.

[26] F. de Prenter, C. V. Verhoosel and E. H. van Brummelen, Preconditioning immersed isogeometric finite element methods with application to flow problems, *Computer Methods in Applied Mechanics and Engineering*, **348** (2019), 604-631.

[27] F. de Prenter, C. V. Verhoosel, G. J. van Zwieten and E. H. van Brummelen, Condition number analysis and preconditioning of the finite cell method, *Computer Methods in Applied Mechanics and Engineering*, **316** (2017), 297-327.

[28] S. Dong, A convective-like energy-stable open boundary condition for simulations of incompressible flows, *Journal of Computational Physics*, **302** (2015), 300-328.

[29] A. Düster, J. Parvizian, Z. Yang and E. Rank, The finite cell method for three-dimensional problems of solid mechanics, *Computer Methods in Applied Mechanics and Engineering*, **197** (2008), 3768-3782.

[30] M. Esmily Moghadam, Y. Bazilevs, T.-Y. Hsia, I. Vignon-Clementel, A. Marsden and M. (MOCHA, A comparison of outlet boundary treatments for prevention of backflow divergence with relevance to blood flow simulations, *Computational Mechanics*, **48** (2011), 277-291.

[31] M. Feistauer and T. Neustupa, On the existence of a weak solution of viscous incompressible flow past a cascade of profiles with an arbitrarily large inflow, *Journal of Mathematical Fluid Mechanics*, **15** (2013), 701-715.

[32] D. Gamdha, K. Saurabh, B. Ganapathysubramanian and A. Krishnamurthy, Geometric modeling and physics simulation framework for building a digital twin of extrusion-based additive manufacturing, preprint, `arXiv:2305.07120`, 2023.

[33] L. Ge. and F. Sotiropoulos, A numerical method for solving the 3D unsteady incompressible Navier-Stokes equations in curvilinear domains with complex immersed boundaries, *Journal of Computational Physics*, **225** (2007), 1782-1809.

[34] E. Heisler, C. Yang, A. Deshmukh, B. Ganapathysubramanian and H. Sundar, Generating finite element codes combining adaptive octrees with complex geometries, preprint, arXiv:2305.19398, 2023.

[35] T. Hoang, C. V. Verhoosel, C. Qin, F. Auricchio, A. Reali and E. H. van Brummelen, Skeleton-stabilized immersogeometric analysis for incompressible viscous flow problems, *Computer Methods in Applied Mechanics and Engineering*, **344** (2019), 421-450.

[36] M. Ishii, M. Fernando, K. Saurabh, B. Khara, B. Ganapathysubramanian and H. Sundar, Solving pdes in space-time: 4d tree-based adaptivity, mesh-free and matrix-free approaches, In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, (2019), 1-61.

[37] M. Ismail, V. Gravemeier, A. Comerford and W. Wall, A stable approach for coupling multidimensional cardiovascular and pulmonary networks based on a novel pressure-flow rate or pressure-only neumann boundary condition formulation, *International Journal for Numerical Methods in Biomedical Engineering*, **30** (2014), 447-469.

[38] T. A. Johnson and V. C. Patel, Flow past a sphere up to a Reynolds number of 300, *Journal of Fluid Mechanics*, **378** (1999), 19-70.

[39] J. Jomo, O. Oztoprak, F. de Prenter, N. Zander, S. Kollmannsberger and E. Rank, Hierarchical multigrid approaches for the finite cell method on uniform and multi-level hp-refined grids, *Computer Methods in Applied Mechanics and Engineering*, **386** (2021), 114075.

[40] D. Kamensky, M.-C. Hsu, D. Schillinger, J. A. Evans, A. Aggarwal, Y. Bazilevs, M. S. Sacks and T. J. R. Hughes, An immersogeometric variational framework for fluid-structure interaction: Application to bioprosthetic heart valves, *Computer Methods in Applied Mechanics and Engineering*, **284** (2015), 1005-1053.

[41] D. Kamensky, Open-source immersogeometric analysis of fluid-structure interaction using fenics and tigar, *Computers and Mathematics with Applications*, **81** (2021), 634-648.

[42] S. Kang and A. Masud, A variational multiscale method with immersed boundary conditions for incompressible flows, *Meccanica*, **56** (2021), 1397-1422.

[43] E. N. Karatzas, G. Stabile, L. Nouveau, G. Scovazzi and G. Rozza, A reduced-order shifted boundary method for parametrized incompressible Navier-Stokes equations, *Computer Methods in Applied Mechanics and Engineering*, **370** (2020), 113273.

[44] K. Khanafer and S. M. Aithal, Laminar mixed convection flow and heat transfer characteristics in a lid driven cavity with a circular cylinder, *International Journal of Heat and Mass Transfer*, **66** (2013), 200-209.

[45] M. A. Khanwale, A. D. Lofquist, H. Sundar, J. A. Rossmanith and B. Ganapathysubramanian, Simulating two-phase flows with thermodynamically consistent energy stable cahn-hilliard Navier-Stokes equations on parallel adaptive octree based meshes, *Journal of Computational Physics*, **419** (2020), 109674.

[46] M. A. Khanwale, K. Saurabh, M. Fernando, V. M. Calo, H. Sundar, J. A. Rossmanith and B. Ganapathysubramanian, A fully-coupled framework for solving cahn-hilliard navier-stokes equations: Second-order, energy-stable numerical methods on adaptive octree based meshes, *Computer Physics Communications*, **280** (2022), 108501.

[47] A. M. Khokhlov, Fully threaded tree algorithms for adaptive refinement fluid dynamics simulations, *Journal of Computational Physics*, **143** (1998), 519-543.

[48] S. Kim, K. Saurabh, M. A. Khanwale, A. Mani, R. K. Anand and B. Ganapathysubramanian, Direct numerical simulation of electrokinetic transport phenomena in fluids: Variational multiscale stabilization and octree-based mesh refinement, *Journal of Computational Physics*, **500** (2024), 112747, 25 pp.

[49] M. Lai and C. S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, *Journal of computational Physics*, **160** (2000), 705-719.

[50] M. Lanzendörfer and J. Stebel, On pressure boundary conditions for steady flows of incompressible fluids with pressure and shear rate dependent viscosities, *Applications of Mathematics*, **56** (2011), 265-285.

[51] B. P. Le Clair, A. E. Hamielec and H. R. Pruppacher, A numerical study of the drag on a sphere at low and intermediate Reynolds numbers, *Journal of Atmospheric Sciences*, **27** (1970), 308-315.

[52] C. Liu, X. Zheng and C. H. Sung, Preconditioned multigrid methods for unsteady incompressible flows, *Journal of Computational physics*, **139** (1998), 35-57.

[53] W. K. Liu, Y. Liu, D. Farrell, L. Zhang, X. S. Wang, Y. Fukui, N. Patankar, Y. Zhang, C. Bajaj, J. Lee, et al., Immersed finite element method and its applications to biological systems, *Computer methods in Applied Mechanics and Engineering*, **195** (2006), 1722-1749.

[54] X. Liu, Z. Xie and S. Dong, On a simple and effective thermal open boundary condition for convective heat transfer problems, *International Journal of Heat and Mass Transfer*, **151** (2020), 119355.

[55] A. Main and G. Scovazzi, The shifted boundary method for embedded domain computations. Part I: Poisson and Stokes problems, *Journal of Computational Physics*, **372** (2018), 972-995.

[56] A. Main and G. Scovazzi, The shifted boundary method for embedded domain computations. Part II: Linear advection-diffusion and incompressible Navier-Stokes equations, *J. Comput. Phys.*, **372** (2018), 996-1026.

[57] S. Marella, S. Krishnan, H. Liu and H. Udaykumar, Sharp interface cartesian grid method i: An easily implemented technique for 3D moving boundary computations, *Journal of Computational Physics*, **210** (2005), 1-31.

[58] D. J. R. Meagher, *Octree Encoding: A New Technique for the Representation, Manipulation and Display of Arbitrary 3-d Objects by Computer*, Electrical and Systems Engineering Department Rensseiaer Polytechnic ..., 1980.

[59] R. Mittal, H. Dong, M. Bozkurttas, F. M. Najjar, A. Vargas and A. Von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, *Journal of Computational Physics*, **227** (2008), 4825-4852.

[60] R. Mittal and G. Iaccarino, Immersed boundary methods, *Annual Review of Fluid Mechanics*, **37** (2005), 239-261.

[61] S. Mohammadian, A. S. Kumar and C. Song, A multi-gpu based high-performance computing framework in elastodynamics simulation using octree meshes, *Computer Methods in Applied Mechanics and Engineering*, **436** (2025), 117723.

[62] M. Nemec, M. Aftosmis and M. Wintzer, Adjoint-based adaptive mesh refinement for complex geometries, In *46th AIAA Aerospace Sciences Meeting and Exhibit*, (2008), 725.

[63] K. C. Ong and M.-C. Lai, An immersed boundary projection method for simulating the inextensible vesicle dynamics, *Journal of Computational Physics*, **408** (2020), 109277.

[64] J. Parvizian, A. Düster and E. Rank, Finite cell method: h- and p- extension for embedded domain methods in solid mechanics, *Computational Mechanics*, **41** (2007), 122-133.

[65] C. S. Peskin, Flow patterns around heart valves: A numerical method, *Journal of Computational Physics*, **10** (1972), 252-271.

[66] C. S. Peskin, The immersed boundary method, *Acta Numerica*, **11** (2002), 479-517.

[67] S. Popinet, Gerris: A tree-based adaptive solver for the incompressible euler equations in complex geometries, *Journal of Computational Physics*, **190** (2003), 572-600.

[68] S. Popinet, An accurate adaptive solver for surface-tension-driven interfacial flows, *Journal of Computational Physics*, **228** (2009), 5838-5866.

[69] F. W. Roos and W. W. Willmarth, Some experimental results on sphere and disk drag, *AIAA Journal*, **9** (1971), 285-291.

[70] R. S. Sampath and G. Biros, A parallel geometric multigrid method for finite elements on octree meshes, *SIAM Journal on Scientific Computing*, **32** (2010), 1361-1392.

[71] K. Saurabh, B. Gao, M. Fernando, S. Xu, M. A. Khanwale, B. Khara, M.-C. Hsu, A. Krishnamurthy, H. Sundar and B. Ganapathysubramanian, Industrial scale large eddy simulations with adaptive octree meshes using immersogeometric analysis, *Computers and Mathematics with Applications*, **97** (2021), 28-44.

[72] K. Saurabh, M. Ishii, M. Fernando, B. Gao, K. Tan, M.-C. Hsu, A. Krishnamurthy, H. Sundar and B. Ganapathysubramanian, Scalable adaptive pde solvers in arbitrary domains, In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, (2021), 1-15.

[73] K. Saurabh, M. Ishii, M. A. Khanwale, H. Sundar and B. Ganapathysubramanian, Scalable adaptive algorithms for next-generation multiphase flow simulations, In *2023 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, (2023), 590-601.

[74] D. Schillinger, Q. Cai, R.-P. Mundani and E. Rank, A review of the finite cell method for nonlinear structural analysis of complex CAD and image-based geometric models, In *Advanced Computing*, (2013), 1-23.

[75] J. H. Seo and R. Mittal, A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations, *Journal of Computational Physics*, **230** (2011), 7347-7363.

[76] A. Stavrev, L. H. Nguyen, R. Shen, V. Varduhn, M. Behr, S. Elgeti and D. Schillinger, Geometrically accurate, efficient and flexible quadrature techniques for the tetrahedral finite cell method, *Computer Methods in Applied Mechanics and Engineering*, **310** (2016), 646-673.

[77] H. Sundar, G. Biros, C. Burstedde, J. Rudi, O. Ghattas and G. Stadler, Parallel geometric-algebraic multigrid on unstructured forests of octrees, In *SC'12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, IEEE, (2012), 1-11.

[78] H. Sundar, R. S. Sampath and G. Biros, Bottom-up construction and 2: 1 balance refinement of linear octrees in parallel, *SIAM Journal on Scientific Computing*, **30** (2008), 2675-2708.

[79] M. Sussman and E. G. Puckett, A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows, *Journal of Computational Physics*, **162** (2000), 301-337.

[80] K. Taira, The Immersed Boundary Projection Method and Its Application to Simulation and Control of Flows Around Low-Aspect-Ratio Wings, California Institute of Technology, 2008.

[81] K. Taira and T. Colonius, The immersed boundary method: A projection approach, *Journal of Computational Physics*, **225** (2007), 2118-2137.

[82] K. Tan, B. Gao, C.-H. Yang, E. L. Johnson, M.-C. Hsu, A. Passalacqua, A. Krishnamurthy and B. Ganapathysubramanian, A computational framework for transmission risk assessment of aerosolized particles in classrooms, *Engineering with Computers*, **40** (2024), 235-256.

[83] T. E. Tezduyar, S. Sathe, R. Keedy and K. Stein, Space-time finite element techniques for computation of fluid-structure interactions, *Computer Methods in Applied Mechanics and Engineering*, **195** (2006), 2002-2027.

[84] A. G. Tomboulides and S. A. Orszag, Numerical investigation of transitional and weak turbulent flow past a sphere, *Journal of Fluid Mechanics*, **416** (2000), 45-73.

[85] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, *Journal of Computational Physics*, **209** (2005), 448-476.

[86] M. Vanella, P. Rabenold and E. Balaras, A direct-forcing embedded-boundary method with adaptive mesh refinement for fluid-structure interaction problems, *Journal of Computational Physics*, **229** (2010), 6427-6449.

[87] R. Verzicco, Immersed boundary methods: Historical perspective and future outlook, *Annual Review of Fluid Mechanics*, **55** (2023), 129-155.

[88] C. Wang, F. Xu, M.-C. Hsu and A. Krishnamurthy, Rapid b-rep model preprocessing for immersogeometric analysis using analytic surfaces, *Computer Aided Geometric Design*, **52/53** (2017), 190-204.

[89] S. Wang and X. Zhang, An immersed boundary method based on discrete stream function formulation for two-and three-dimensional incompressible flows, *Journal of Computational Physics*, **230** (2011), 3479-3499.

[90] X. Wang and L. Zhang, Modified immersed finite element method for fully-coupled fluid-structure interactions, *Computer Methods in Applied Mechanics and Engineering*, **267** (2013), 150-169.

[91] D. Xu, O. Colomés, A. Main, K. Li, N. M. Atallah, N. Abboud and G. Scovazzi, A weighted shifted boundary method for immersed moving boundary simulations of stokes' flow, *Journal of Computational Physics*, **510** (2024), 113095.

[92] F. Xu, E. L. Johnson, C. Wang, A. Jafari, C. Yang, M. S. Sacks, A. Krishnamurthy and M. Hsu, Computational investigation of left ventricular hemodynamics following bioprosthetic aortic and mitral valve replacement, *Mechanics Research Communications*, **112** (2021), 103604.

[93] F. Xu, D. Schillinger, D. Kamensky, V. Varduhn, C. Wang and M. Hsu, The tetrahedral finite cell method for fluids: Immersogeometric analysis of turbulent flow around complex geometries, *Computers and Fluids*, **141** (2016), 135-154.

[94] S. Xu, B. Gao, M. Hsu and B. Ganapathysubramanian, A residual-based variational multiscale method with weak imposition of boundary conditions for buoyancy-driven flows, *Computer Methods in Applied Mechanics and Engineering*, **352** (2019), 345-368.

[95] C. Yang, K. Saurabh, G. Scovazzi, C. Canuto, A. Krishnamurthy and B. Ganapathysubramanian, Optimal surrogate boundary selection and scalability studies for the shifted boundary

method on octree meshes, *Computer Methods in Applied Mechanics and Engineering*, **419** (2024), 116686.

[96] C. Yang, G. Scovazzi, A. Krishnamurthy and B. Ganapathysubramanian, Octree-based shifted boundary method for multiphysics simulations using linearized navier-stokes in complex geometries, preprint, arXiv:2501.00143, 2024.

[97] C. Yang, G. Scovazzi, A. Krishnamurthy and B. Ganapathysubramanian, Simulating incompressible flows over complex geometries using the shifted boundary method with incomplete adaptive octree meshes, preprint, arXiv:2411.00272, 2024.

[98] X. Yang, X. Zhang, Z. Li and G. He, A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations, *Journal of Computational Physics*, **228** (2009), 7821-7836.

[99] X. Zeng, G. Stabile, E. N. Karatzas, G. Scovazzi and G. Rozza, Embedded domain reduced basis models for the shallow water hyperbolic equations with the shifted boundary method, *Computer Methods in Applied Mechanics and Engineering*, **398** (2022), 115143.

[100] L. Zhang, A. Gerstenberger, X. Wang and W. K. Liu, Immersed finite element method, *Computer Methods in Applied Mechanics and Engineering*, **193** (2004), 2051-2067.

[101] Q. Zhu, F. Xu, S. Xu, M. Hsu and J. Yan, An immersogeometric formulation for free-surface flows with application to marine engineering problems, *Computer Methods in Applied Mechanics and Engineering*, **361** (2019), 112748.

Received December 2024; revised March 2025.