High-Order Fast Parallel Surface Integral Equation Solvers for Electromagnetic Scattering on PEC Targets

Omid Babazadeh

This Thesis submitted to the Faculty of Graduate and Postdoctoral Studies
of the University of Manitoba
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Electrical and Computer Engineering - Engineering Faculty
University of Manitoba

March 2026

© Copyright by Omid Babazadeh, 2026

Contents

List of Figures

List of Tables

Abstract

This thesis introduces a robust, high-order computational framework for modelling electromagnetic scattering from electrically large perfect electric conductor (PEC) objects. The framework rests on two point-based discretization methods: it is implemented using either an enhanced Locally Corrected Nyström (LCN) scheme or a Chebyshev-based boundary integral equation (CBIE) formulation, accelerated by the hierarchical matrix ($\mathcal{H}$-matrix) framework, which together provide an accurate, fast and memory-efficient kernel-aware discretization of surface-integral equations over complex geometries.

An adaptive hp-refinement strategy selects specific element sizes h_i and polynomial orders p_i for different regions in the geometry, capturing edge singularities to the limited digits of accuracy while maintaining high-order accuracy in smooth regions. Storage and runtime are reduced by orders of magnitude via $\mathcal{H}$-matrix compression, whose high-order far-field expansions accelerate matrix–vector products. We implement both direct $\mathcal{H}$-**LU** and $\mathcal{H}$-matrix preconditioned iterative solvers; blockwise (heterogeneous) tolerances in the $\mathcal{H}$-**LU** factorization further shorten time to solution without degrading accuracy. Hybrid parallelism, MPI across computing nodes and OpenMP within nodes, maximises data locality and resource utilisation, delivering near-linear scalability on modern HPC platforms. Extensive Radar Cross Section (RCS) benchmarks on canonical and complex PEC targets

validate the robustness, precision, and built-in error control in the framework, demonstrating speedups exceeding an order of magnitude and memory reductions of over 90% compared to conventional dense solvers.

By integrating high-order integral equation formulations with *hp*-refinement strategy, $\mathcal{H}$-matrix compression, and hybrid parallel computing, this work delivers a practical, scalable toolset for high-fidelity electromagnetic scattering analysis and advances the state of the art in surface integral equation solvers.

Acknowledgment

First and foremost, I express my deepest gratitude to my supervisor, Professor Vladimir Okhmatovski, for his invaluable guidance, encouragement, and support throughout this research. His insight, high standards, and deep expertise were instrumental to this work.

I also sincerely thank my co-advisor, Professor Ian Jeffrey, for his thoughtful feedback, constructive discussions, and continuous support. His technical advice and mentorship greatly enriched this thesis.

I am grateful to Dr. Emrah Sever for his guidance, helpful discussions, and steady support at every stage of this work.

Finally, I am deeply thankful to my family for their unwavering love, patience, and support during this journey. Their encouragement and belief in me have been a constant source of motivation.

This work would not have been possible without all of your contributions.

Contribution of Authors

This thesis is organized as a "sandwich" or grouped-manuscript thesis. Chapters 4 and 6 contain material from journal articles that have been published or are currently under peer review. The following lists the titles of the research articles, their publication status, and the specific contributions of each author.

1. **Chapter 4** includes the content published in the *IEEE Journal on Multiscale and Multiphysics Computational Techniques (IEEE J-MMCT)*:

   Title: Optimal Preconditioners for Hybrid Direct-Iterative -Matrix Solvers in Boundary Element Methods

   Status: Published

   DOI: 10.1109/JMMCT.2025.3547827

   Author Contributions:

   - **Omid Babazadeh (Thesis Author):** Lead researcher. Responsible for Conceptualization, Methodology, Software development, Data acquisition, Formal analysis, and Writing – original draft.
   - **Vladimir Okhmatovski (Advisor):** Supervision, Conceptualization, and Writing – review and editing.

- **Ian Jeffrey (Co-advisor):** Supervision, Methodology, and Writing – review and editing.
- **Emrah Sever (Aselsan Turkey):** Technical consultation and Writing – review and editing.
- **Constantine Sideris and Jin Hu (University of Southern California):** Technical consultation and Writing – review and editing.

2. **Chapter 6** includes the content under review for publication in the *IEEE Open Journal of Antennas and Propagation (IEEE OJAP)*:

   Title: A High-Order Fast Boundary Element Method as a Benchmarking Computational Framework for RCS Analysis of PEC Targets
   Status: Under review
   ID: OJAP-0139-2026

   Author Contributions:

   - **Omid Babazadeh (Thesis Author):** Lead researcher. Responsible for Conceptualization, Mathematical derivation, Software implementation, Validation, and Writing – original draft.
   - **Vladimir Okhmatovski (Advisor):** Supervision, Project administration, and Writing – review and editing.
   - **Ian Jeffrey (Co-advisor):** Supervision and Writing – review and editing.
   - **Emrah Sever:** Technical collaboration and Writing – review and editing.
   - **Constantine Sideris:** Technical collaboration and Writing – review and editing.

Note: As the primary author of these manuscripts, I performed the majority of the theoretical development, numerical implementation, and manuscript preparation, with guidance and feedback provided by the co-authors via editorial reviews.

Chapter 1

Introduction

The field of computational electromagnetics (CEM) has become indispensable in accurately simulating complex electromagnetic phenomena, essential for various engineering and scientific applications. By providing a framework to model electromagnetic fields, CEM has enabled advancements in antenna design, electromagnetic interference mitigation, radar cross-section (RCS) calculations, and biomedical imaging [1, 2]. The mathematical backbone of CEM lies in the solution of Maxwell's equations, which can be approached using either partial differential equation (PDE) methods or integral equation formulations. PDE methods, such as the finite element method (FEM) [3, 4] and the finite-difference time-domain (FDTD) method [2, 5], are widely used for their flexibility in handling complex media and boundary conditions. On the other hand, integral equation formulations like the Magnetic Field Integral Equation (MFIE), Electric Field Integral Equation (EFIE), and Combined Field Integral Equation (CFIE) [6, 7, 8] are particularly advantageous for problems involving open-region scattering from Perfect Electric Conductor (PEC) structures. However, the computational demands increase significantly as these problems scale up in size and complexity, creating a

need for robust, accurate, and scalable computational methods [9, 10].

In contrast to differential equation based solvers such as the Finite Element Method (FEM) and the Finite Difference Time Domain (FDTD) method, Surface Integral Equation (SIE) techniques formulate electromagnetic problems entirely on the surface of the object, reducing dimensionality and naturally enforcing radiation conditions. In the following discussion, we restrict our attention to IE-based formulations, which form the foundation of this work. Traditional low-order methods, such as those employing Rao – Wilton – Glisson (RWG) basis functions within the Method of Moments (MoM) [11, 12], have long been employed for solving such surface integral equations. These methods, based on piecewise linear basis functions, are relatively straightforward and computationally efficient for simple geometries. However, they typically provide limited accuracy and exhibit convergence rates that depend significantly on mesh refinement, especially in complex scattering scenarios. With convergence rates on the order of $\mathcal{O}(h)$, where h is the mesh size, low-order methods necessitate exceedingly fine discretization for achieving precision, resulting in prohibitively high computational costs and extended processing times for large-scale problems [12].

In contrast, high-order methods provide a powerful alternative, allowing for more accurate approximations with fewer degrees of freedom. By employing higher-degree polynomials or other enriched basis functions, high-order methods achieve rapid convergence with far less computational overhead. Techniques like the Locally Corrected Nyström (LCN) method and Chebyshev-Based Integral Equation (CBIE) formulations exemplify high-order methods that yield better accuracy for complex boundary conditions and achieve high convergence rates, often on the order of $\mathcal{O}(h^p)$ for a polynomial basis of order p. This capability renders high-order methods invaluable for high-frequency problems and applications where precision is paramount [13].

To overcome the efficiency and accuracy limitations of classical Boundary Element Methods (BEMs), this thesis develops a parallel, high-order framework for electrically large PEC scatterers discretized with quadrilateral Bézier patches, i.e., curved surface elements defined by control points [14]. An adaptive *hp*-refinement engine raises resolution only where edge singularities demand it, while higher-order expansions on coarser patches efficiently capture smooth regions. Dense system matrices are compressed into hierarchical low-rank form, then solved in two complementary ways: a direct $\mathcal{H}$-**LU** factorization, and iterations preconditioned by $\mathcal{H}$-matrix approximations that use heterogeneous blockwise tolerances, tight on diagonal and near-field blocks, relaxed on far-field interactions, to cut memory and runtime [15, 16]. The workflow is parallelised with MPI for inter-node distribution and OpenMP for intra-node concurrency, delivering high fidelity, reduced memory footprint, and near-linear scalability across modern HPC platforms.

1.1 Research Questions

This research is driven by the following key questions:

1. What are the advantages and limitations of high-order versus low-order numerical methods in computational electromagnetics, and how can their effectiveness be evaluated with respect to different problem scales and complexities?

2. What efficiency can be gained by applying $\mathcal{H}$-matrix acceleration to high-order integral equations, and what challenges or limitations arise when combining high-order discretization with $\mathcal{H}$-matrix techniques?

3. How can the principles of Rokhlin's dogma, the simultaneous pursuit of robustness,

efficiency, and scalability, inform the selection between $\mathcal{H}$-matrix preconditioned iterative solvers and direct solvers in this context? Under what conditions do these methods optimize performance, accuracy, and computational resources?

4. How does the adaptive hp–refinement LCN framework capture edge-singular interactions and preserve solution precision in complex electromagnetic scattering problems involving PEC geometries?

1.2 Objectives

The overall goal of this thesis is to create a high-fidelity electromagnetic-scattering framework that remains efficient and scalable as problem size and geometric complexity grow. Specifically, the work aims to:

- **Develop an Adaptive hp–Refinement LCN Framework:** Introduce a point-based LCN discretization with variable polynomial order and adaptive element sizing, fine, low-order patches at edges; one–two layers of transition patches with intermediate (h, p) to taper; and coarse, high-order patches in smooth regions capturing edge singularities while preserving $\mathcal{O}(h^p)$ accuracy in smooth regions and overall efficiency.

- **Implement $\mathcal{H}$-matrix Acceleration and Preconditioning:** Apply hierarchical low-rank compression to shrink memory usage and speed up matrix–vector products, then leverage the resulting $\mathcal{H}$-matrix factors both as efficient direct solvers and as powerful preconditioners for iterations, further improving convergence and overall scalability.

- **Enhance Scalability via Parallel Computing:** Design and implement parallel assembly, compression, and solver routines to ensure performance scales across multicore and distributed-memory systems, facilitating simulations of electrically large structures.

- **Validate Through Benchmarking and RCS Computations:** Demonstrate accuracy, efficiency, and scalability by computing radar cross sections (RCS) for canonical and complex PEC geometries at high frequencies, comparing against reference solutions and assessing computational costs.

- **Perform Complexity Analysis:** Quantify computational complexity within the RCS benchmarking framework, showing that high-order formulations exhibit near-quadratic growth in the number of unknowns N to preserve $\mathcal{O}(h^p)$ accuracy, in contrast to the exponential growth observed in traditional low-order methods.

Together, these objectives define a comprehensive framework capable of tackling electrically large PEC scattering problems with a balanced blend of accuracy, speed, and parallel scalability.

1.3 Contributions

This thesis advances high–order, integral equation based CEM by delivering:

- **Hybrid direct/iterative preconditioning strategy** for large–scale PEC scatterers, combining $\mathcal{H}$–**LU** factorization with preconditioned iterative solvers to achieve compact storage, fast matrix–vector products, and high accuracy.

- **Heterogeneous, preconditioned $\mathcal{H}$–matrix framework:** a unified infrastructure that (i) applies block–adapted $\mathcal{ACA}$ thresholds and rank controls, tight on non-leaf blocks, relaxed on leaf blocks, to minimize storage and factorization time; and (ii) reuses the factors both as lightweight direct solvers and as powerful preconditioners that cut iteration counts on very large problems.

- **Complexity analysis of high–order formulations:** a quantitative study within the RCS benchmarking framework demonstrating that high–order methods exhibit near–quadratic growth in the number of unknowns N required to preserve $\mathcal{O}(h^p)$ accuracy, in contrast to the exponential scaling observed in traditional low–order formulations, thereby establishing the computational advantage of the proposed approach.

- **RCS benchmarking and performance validation:** high–frequency scattering tests on canonical and realistic PEC geometries paired with cost–versus–accuracy studies, demonstrating improved accuracy, smaller memory footprint, and faster runtimes than existing baselines, with quantified scaling versus problem size and tolerance.

- **Adaptive hp–refinement LCN framework:** a point–based LCN discretization on quadrilateral Bézier–patch meshes that uses fine, low–order patches at edges, one–two layers of transition patches with intermediate (h, p), and coarse, high–order patches in smooth regions, capturing edge singularities while preserving target $\mathcal{O}(h^p)$ behavior and overall efficiency.

- **Local near–interaction radius selection:** a per–observation–point near–set construction that adapts R_{near} to local geometry and discretization (h, p) (including edge proximity and curvature), meeting prescribed error tolerances while avoiding overload of local corrections and reducing near–assembly wall time.

- **Accuracy preserving framework with adaptive hp–refinement strategy:** practical design rules that preserve high–order accuracy to the limited digits of convergence under tapering, including monotone separations $h_{\text{edge}} \ll h_{\text{transition}} \ll h_{\text{smooth}}$ and $p_{\text{edge}} < p_{\text{transition}} < p_{\text{smooth}}$, and quadrature/correction settings $q \geq p+1$ that maintain $\mathcal{O}(h^{p_{\text{smooth}}})$ convergence in smooth regions while controlling conditioning.

- **Scalable parallel software suite:** for matrix assembly, $\mathcal{H}$–matrix compression, and solver kernels on shared– and distributed–memory systems (MPI/OpenMP), delivering robust performance on electrically large problems.

- **Open, extensible CEM toolkit:** encapsulating the above methods for academic research and future industrial applications.

The aim of this research is to develop a robust and efficient hybrid solver that addresses the challenges of computational electromagnetics, particularly for large-scale PEC problems. Each aspect of this work is discussed in subsequent chapters, culminating in a robust toolset for high-fidelity simulations.

1.4 Scope and Limitations

In this section, we introduce the limitations and scope of the framework developed.

Problem class considered

The numerical experiments and performance studies in this thesis focus on PEC scattering benchmarks of smooth to sharp–edged geometries with radar cross section (RCS) analysis.

The thesis emphasizes efficiency, controllable accuracy, and scalability of the proposed $\mathcal{H}$–matrix accelerated high–order SIE framework with adaptive mesh refinement and polynomial order change.

Extensions beyond the thesis scope

Several natural extensions are not implemented in the present thesis and are left for future work. The present thesis focuses on establishing a rigorous, scalable PEC surface integral equation foundation, upon which volume integral equation/volume surface integral equation capabilities can be built in subsequent research.

Furthermore, the application of this framework to antenna simulations, which involve localized feed models and different port-matching considerations and the treatment of multi-scale problems, where high-order *hp*-refinement can be further leveraged across vast differences in geometric detail, represent key directions for future exploration.

Integral equation formulations

The primary SIE formulations emphasized in this thesis are MFIE- and CFIE-type operators, chosen for their favourable conditioning properties on closed PEC surfaces and their robustness in large–scale computations with hierarchical compression and hybrid solvers. A full EFIE-centered performance study is not included here. EFIE is well known to be more challenging numerically (first-kind behaviour and stronger conditioning demands), and incorporating EFIE in the same high–order, $\mathcal{H}$–accelerated and adaptive *hp* refinement setting is a valuable direction for future work, including the study of specialized preconditioning strategies.

Choice of acceleration strategy

This thesis adopts the $\mathcal{H}$–matrix framework because it supports both (i) efficient matrix–vector products for iterative methods and (ii) compressed approximate factorization ($\mathcal{H}$–**LU**) enabling robust direct and hybrid direct–iterative solution strategies. For the purposes of this thesis, comparisons against dense baselines, analytic solutions, measurements and MLFMM accelerated commercial software are used to quantify compression benefits and scaling trends.

Scale of numerical examples

The numerical investigations in this thesis are primarily centered on the advantages of high-order methods, which are characterized by their ability to achieve superior accuracy with a significantly lower number of degrees of freedom (DoF) compared to traditional low-order approaches. Consequently, the largest problem sizes reported (50λ) are intended to demonstrate controllable error convergence and rigorous verification against reference results, alongside the practical efficiency of the $\mathcal{H}$-matrix accelerated framework. While this work establishes the foundational scaling of memory and runtime, demonstrations at substantially larger electrical sizes, where the absolute problem size increases, are reserved for future work, enabled by larger computational allocations or dedicated HPC deployments.

Bibliography

[1] A. F. Peterson, S. L. Ray, and R. Mittra, *Computational Methods for Electromagnetics*. New York: IEEE Press, 1998.

[2] A. Taflove and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*. Artech House, 2000.

[3] J. Jin, *The Finite Element Method in Electromagnetics*, 2nd ed. Wiley, 2002.

[4] P. Monk, *Finite Element Methods for Maxwell's Equations*. Oxford University Press, 2003.

[5] D. M. Sullivan, *Electromagnetic Simulation Using the FDTD Method*, 2nd ed. Wiley-IEEE Press, 2013.

[6] W. C. Chew, M. S. Tong, and B. Hu, *Integral Equation Methods for Electromagnetic and Elastic Waves*. Morgan Claypool, 2009.

[7] Gedney, S. D. (2003). The Combined Field Integral Equation (CFIE) for Scattering Problems. *IEEE Transactions on Antennas and Propagation*, 51(7), 1710–1717.

[8] R. F. Harrington and J. R. Mautz, "Theory of characteristic modes for conducting bodies," *IEEE Transactions on Antennas and Propagation*, vol. 19, no. 5, pp. 622–628, 1967. doi:10.1109/TAP.1967.1138995.

[9] Gedney, S. D. (2014). An Overview of the Combined Field Integral Equation. *IEEE Antennas and Propagation Magazine*, 56(4), 19–32.

[10] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," *Journal of Computational Physics*, vol. 73, no. 2, pp. 325–348, Dec. 1987.

[11] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Transactions on Antennas and Propagation*, vol. 30, no. 3, pp. 409-418, May 1982.

[12] R. F. Harrington, *Field Computation by Moment Methods*. New York: Wiley-IEEE Press, 1993.

[13] V. Rokhlin, "Rapid solution of integral equations of classical potential theory," *Journal of Computational Physics*, vol. 60, no. 2, pp. 187–207, 1987.

[14] L. Piegl and W. Tiller, *The NURBS Book*. Springer, 1997.

[15] W. Hackbusch, "A Sparse Matrix Arithmetic Based on $\mathcal{H}$-Matrices," *Computing*, vol. 62, no. 2, pp. 89–108, Apr. 1999.

[16] R. Kriemann, "H-matrix Methods for Integral Equation Solutions," *Numerical Mathematics: Theory, Methods and Applications*, vol. 8, no. 1, pp. 13–36, 2015.

Chapter 2

Background

The solution of large-scale electromagnetic scattering problems often requires advanced numerical techniques to achieve both accuracy and efficiency, especially when dealing with electrically large or geometrically complex structures. High-order methods, such as the Locally Corrected Nyström (LCN) method and Chebyshev-Based Integral Equation (CBIE) formulations, provide a promising approach by improving the accuracy and convergence rate with respect to discretization, achieving comparable precision with far fewer degrees of freedom. These methods offer considerable improvements over traditional low-order approaches, particularly in applications where high precision is required, such as the analysis of Radar Cross Section (RCS) for large Perfect Electric Conductor (PEC) targets.

High-order methods address the limitations of low-order techniques by employing higher-degree polynomial basis or interpolation functions that represent fields and boundary conditions more accurately over larger surface regions. The Locally Corrected Nyström (LCN) method, for instance, incorporates essential local correction terms that enable accurate high-order discretization of surface integral equations. Similarly, the Chebyshev-Based Integral

Equation (CBIE) formulation employs Chebyshev polynomial expansions to achieve compact and high-fidelity representations of integral operators. Both approaches thus provide high-order, point-based formulations that can reach a given accuracy with far fewer degrees of freedom than traditional low-order schemes, particularly for geometrically complex or high-frequency scattering scenarios.

While high-order methods provide superior accuracy and faster error convergence with respect to mesh refinement, the resulting system matrices tend to be large and dense, and their conditioning can deteriorate with increasing polynomial order. To address the associated computational cost, $\mathcal{H}$-matrix techniques offer an efficient framework for compressing and accelerating the dense coefficient matrices arising from high-order discretizations. By approximating far-field interactions with low-rank structures, $\mathcal{H}$-matrices substantially reduce memory requirements and computational complexity, enabling scalable solutions for electrically large problems.

In addition, $\mathcal{H}$-matrix based preconditioners can substantially improve the conditioning of these systems, accelerating the convergence of iterative solvers. This approach is crucial when direct methods become infeasible due to memory and computational constraints, as is often the case with high-frequency, large-scale PEC scattering problems. The combination of high-order discretization, $\mathcal{H}$-matrix compression, and preconditioned iterative solvers thus forms a comprehensive solution framework that balances accuracy, efficiency, and scalability.

This chapter provides an in-depth review of the theoretical and computational foundations underlying the high-order methods and $\mathcal{H}$-matrix techniques employed in this work. Section 2.1 begins with a discussion of NURBS surface representation for geometries, which forms the basis for accurately modeling complex structures in electromagnetic simulations. Sections 2.2 - 3.2 then delve into the principles and implementation of high-order methods,

with a focus on the Locally Corrected Nyström (LCN) formulation.

Following this, Section 2.3 introduces $\mathcal{H}$-matrix compression, emphasizing its critical role in enabling efficient direct and iterative solutions by reducing memory footprint and computational cost. Section 2.4 presents both direct solvers and $\mathcal{H}$-matrix preconditioned iterative methods, highlighting their respective advantages and trade-offs when applied to the dense systems arising from high-order discretizations. Finally, Section 2.4.3 delves into the *hp*–refinement LCN framework, detailing how variable polynomial orders and adaptive element sizing resolve edge-singular behaviors and maintain solution continuity across sharp PEC edges.

2.1 Geometry and Surface Current Representation with NURBS and Bézier Quadrilaterals

A high-order solution framework benefits from high-order geometric models capable of efficiently representing complicated geometries. In this work, Non-Uniform Rational B-Spline (NURBS) are used for geometry descriptions, which are sophisticated mathematical constructs designed to describe arbitrary shapes [1]. NURBS offer significant advantages over polygonal surface representations and polynomial spline models (e.g., Bézier and B-spline surfaces) due to their flexibility and precision, making them widely applicable across various fields, including computer-aided design (CAD), computer graphics, and manufacturing. In particular, the ability of NURBS to represent complex geometries with a relatively small number of control points is crucial for computational efficiency in electromagnetic applications.

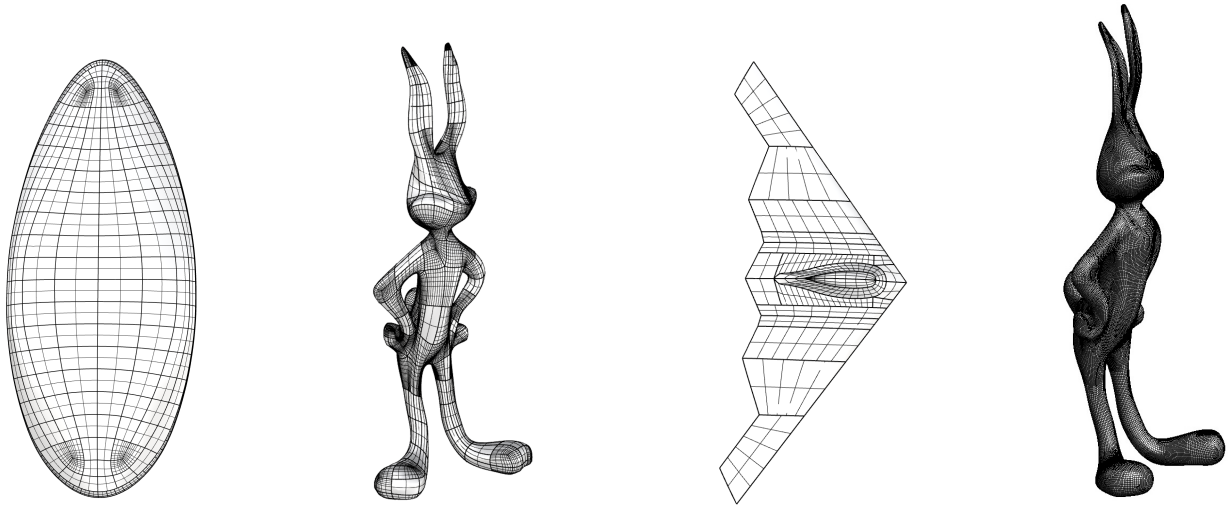
In the context of electromagnetic scattering, particularly when calculating the Radar

Cross Section (RCS) of Perfect Electric Conductor (PEC) objects, NURBS provide an accurate geometric representation that can be discretized into quadrilateral Bézier patches. Bézier surfaces are advantageous in computational electromagnetics (CEM) because their sizes can be tailored to match the desired electrical mesh size, allowing for a more precise approximation of surface currents. The approximation error typically follows an order of convergence of $\mathcal{O}(h^p)$, where p represents the polynomial order of the Bézier surface, and h represents the mesh size. This high-order approximation capability is essential when dealing with complex boundary conditions typical in RCS computations.

To facilitate the analysis, general NURBS surfaces are tessellated into smaller patches until each can be accurately represented by a quadrilateral Bézier surface. This tessellation process is a standard geometric preprocessing step [1], ensuring that the geometry is represented adequately without introducing excessive computational costs. Quadrilateral Bézier patches are preferred for their computational efficiency and ease of design. However, converting NURBS to Bézier patches may introduce minor geometric approximation errors, particularly in regions with high curvature or intricate features. The inherent differences between rational NURBS and polynomial Bézier representations therefore require careful control of tessellation density to preserve acceptable geometric accuracy in the final model.

In Fig. 2.1a–Fig. 2.1c, three representative geometries considered in this work are illustrated. The models, referred to as the Almond, the Bunny, and the B-2 aircraft, were modeled using NURBS surfaces in Rhino [1]. The almond geometry shown here corresponds to a smoothed representation for visualization purposes; however, the RCS simulations also include the sharp-edged NASA almond benchmark to ensure consistency with standard validation studies. As an example, the discretization of the Bunny model is shown in Fig. 2.1d, highlighting the Bézier quadrilateral patches ultimately used to implement the adaptive

integration strategy and solve the Combined Field Integral Equation (CFIE).



(a) (b) (c) (d)

Figure 2.1: NURBS representations of (a) the Almond, (b) the Bunny, and (c) the B-2 Aircraft. (d) Bézier patch representation of the Bunny.

Once the Bézier patches have been defined, they provide a convenient basis for representing the unknown surface current density $\mathbf{J}(\mathbf{r})$. On each Bézier patch (equivalently referred to as an element), the current is approximated as

$$\mathbf{J}(\mathbf{r}) = J^1(\mathbf{r})\frac{\mathbf{a}_1(\mathbf{r})}{\mathcal{J}(\mathbf{r})} + J^2(\mathbf{r})\frac{\mathbf{a}_2(\mathbf{r})}{\mathcal{J}(\mathbf{r})}, \tag{2.1}$$

where $\mathbf{a}_1(\mathbf{r}) = \partial\mathbf{r}/\partial\xi$ and $\mathbf{a}_2 = \partial\mathbf{r}/\partial\eta$ are the covariant unit vectors tangential to the surface of a particular element, defined parametrically in terms of coordinates $\xi \in [-1, 1]$ and $\eta \in [-1, 1]$ within a unit square [2]. Here, $J^1(\mathbf{r})$ and $J^2(\mathbf{r})$ denote the scalar current components along the ξ- and η directions of each patch, respectively. This representation is illustrated in Fig. 6.2. The quantity $\mathcal{J}(\mathbf{r})$ in (2.1) denotes the determinant of the Jacobian matrix associated with the mapping from the unit square to the surface element. This term

accounts for the geometric scaling in the transformation and is included when performing numerical integration [3].

In summary, high-order methods such as LCN and CBIE not only enhance the accuracy of integral equation solvers but also address the computational challenges associated with large-scale electromagnetic problems. Their ability to effectively handle complex geometries and boundary conditions makes them indispensable tools in the analysis of RCS for electrically large structures.

2.2 Locally Corrected Nyström Discretization of Integral Equations

The Locally Corrected Nyström (LCN) method serves as an effective approach for solving integral equations associated with electromagnetic scattering problems, specifically those represented by the Magnetic Field Integral Equation (MFIE), Electric Field Integral Equation (EFIE), and the Combined Field Integral Equation (CFIE). This method is particularly beneficial for applications involving complex geometries and large electrical sizes, allowing for high-order accuracy and efficient numerical solutions.

2.2.1 Magnetic Field Integral Equation (MFIE)

By virtue of the PEC boundary condition $\hat{\mathbf{n}}(\mathbf{r}) \times \mathbf{E}(\mathbf{r}) = \mathbf{0}$ on S, the equivalent magnetic current $\mathbf{J}^m = \hat{\mathbf{n}} \times \mathbf{E}$ vanishes. Using the equivalence principle, the magnetic field in the exterior region is $\mathbf{H}(\mathbf{r}) = \displaystyle\int_S \nabla \times \big(\mathbf{J}(\mathbf{r}')\, G_0(\mathbf{r}, \mathbf{r}')\big)\, dS' + \mathbf{H}^{\text{inc}}(\mathbf{r}), \;\; \mathbf{r} \notin V$. Enforcing the zero tangential magnetic field on a surface S' just inside the conductor, $\hat{\mathbf{n}}(\mathbf{r}) \times \mathbf{H}(\mathbf{r}) = \mathbf{0}$ for $\mathbf{r} \in S'$, yields the magnetic field integral equation. Choosing the observation surface S' to

coincide with the current-carrying surface S produces the MFIE in second-kind form (with the $\frac{1}{2}\mathbf{J}$ term).

$$\frac{1}{2}\,\mathbf{J}(\mathbf{r}) - \hat{\mathbf{n}}(\mathbf{r}) \times P.V. \int_S \bar{\mathbf{G}}_{HJ}(\mathbf{r},\mathbf{r}') \cdot \mathbf{J}(\mathbf{r}')\, dS' = \hat{\mathbf{n}}(\mathbf{r}) \times \mathbf{H}^{\text{inc}}(\mathbf{r}), \quad \mathbf{r} \in S, \tag{2.2}$$

where $\hat{\mathbf{n}}$ is the unit normal vector to the surface S, and $\bar{\mathbf{G}}_{HJ}(\mathbf{r},\mathbf{r}')$ is the magnetic field dyadic Green's function, which characterizes the magnetic interactions due to the surface currents. The MFIE formulation is particularly useful in capturing the magnetic responses of the PEC surfaces, allowing for a more comprehensive characterization of surface currents in scattering phenomena [4].

2.2.2 Electric Field Integral Equation (EFIE)

The EFIE provides a complementary approach focused on the electric fields. It is expressed as:

$$\hat{\mathbf{n}}(\mathbf{r}) \times \int_S \bar{\mathbf{G}}_{EJ}(\mathbf{r},\mathbf{r}') \cdot \mathbf{J}(\mathbf{r}')\, dS' = -\,\hat{\mathbf{n}}(\mathbf{r}) \times \mathbf{E}^{\text{inc}}(\mathbf{r}), \quad \mathbf{r} \in S, \tag{2.3}$$

where $\bar{\mathbf{G}}_{EJ}(\mathbf{r},\mathbf{r}')$ is the electric field dyadic Green's function. The EFIE is advantageous for certain geometrical configurations where the electric fields play a dominant role, particularly in the analysis of radiation and scattering from electrically large objects. However, it may suffer from numerical instability in specific scenarios, particularly when the geometry involves sharp edges or corners [1].

2.2.3 Combined Field Integral Equation (CFIE)

The CFIE integrates the strengths of both the MFIE and EFIE to provide a more balanced formulation that mitigates numerical issues associated with each individual equation. It is defined as follows:

$$\text{CFIE} = \alpha\text{EFIE} + (1-\alpha)\eta_0\text{MFIE}, \tag{2.4}$$

$$\begin{aligned}
&(1-\alpha)\,\eta_0\left[\frac{1}{2}\,\hat{\mathbf{t}}(\mathbf{r})\cdot\mathbf{J}(\mathbf{r}) - \hat{\mathbf{t}}(\mathbf{r})\cdot\hat{\mathbf{n}}(\mathbf{r}) \times P.V.\int_S \bar{\mathbf{G}}_{HJ}(\mathbf{r},\mathbf{r}')\cdot\mathbf{J}(\mathbf{r}')\,\mathrm{d}S'\right] \\
&\quad + \alpha\,\hat{\mathbf{t}}(\mathbf{r})\cdot\int_S \bar{\mathbf{G}}_{EJ}(\mathbf{r},\mathbf{r}')\cdot\mathbf{J}(\mathbf{r}')\,\mathrm{d}S' \\
&= (1-\alpha)\,\eta_0\,\hat{\mathbf{t}}(\mathbf{r})\cdot\hat{\mathbf{n}}(\mathbf{r})\times\mathbf{H}^{\text{inc}}(\mathbf{r}) - \alpha\,\hat{\mathbf{t}}(\mathbf{r})\cdot\mathbf{E}^{\text{inc}}(\mathbf{r}), \qquad \mathbf{r}\in S,
\end{aligned} \tag{2.5}$$

where $\hat{\mathbf{t}}$ is a unit vector tangential to the surface, η_0 is the intrinsic impedance of free space, and α is a parameter used to balance the contributions from the EFIE and MFIE. This formulation enables improved stability and accuracy in the computation of surface currents $\mathbf{J}$ when dealing with complex geometries, effectively minimizing the numerical errors encountered in pure EFIE or MFIE formulations [5].

Next, we proceed to discretize the CFIE formulation, which combines both the MFIE and EFIE components. Once an appropriate geometric representation is established, the discretization of the CFIE follows directly. The integral equation in (2.6) is decomposed into a sum of integrals over N_e Bézier elements. To evaluate the integral on each element, a suitable quadrature rule is applied; herein, the same rule is used for all elements. On the mth patch, the quadrature points are denoted by $\mathbf{r}_{q_m}$, $q = 1, 2, \ldots, Q$, where Q is the number of points in the chosen quadrature rule, and the corresponding weights are w_{q_m}.

Substituting the current expansion (2.1) into (2.6), and evaluating the CFIE at each

quadrature point gives a set of scalar integral equations:

$$\begin{aligned}
&\mathbf{a}_i^{q_m} \cdot \Big[\alpha\, \mathbf{E}^{\rm inc}(\mathbf{r}_{q_m}) + (1-\alpha)\,\eta_0\, \hat{\mathbf{n}}^{q_m} \times \mathbf{H}^{\rm inc}(\mathbf{r}_{q_m})\Big] \;= \\
&\quad - \;\mathbf{a}_i^{q_m} \cdot \sum_{e=1}^{N_e} \int\limits_{S_e} \Big[\alpha\, \bar{\mathbf{G}}_{EJ}(\mathbf{r}_{q_m}, \mathbf{r}') + (1-\alpha)\,\eta_0\, \hat{\mathbf{n}}^{q_m} \times \bar{\mathbf{G}}_{HJ}(\mathbf{r}_{q_m}, \mathbf{r}')\Big] \cdot \\
&\qquad \left(\frac{J^1(\mathbf{r}')\,\mathbf{a}_1(\mathbf{r}')}{\mathcal{J}(\mathbf{r}')} + \frac{J^2(\mathbf{r}')\,\mathbf{a}_2(\mathbf{r}')}{\mathcal{J}(\mathbf{r}')} \right) dS' \;+\; \frac{(1-\alpha)\,\eta_0}{2}\, \mathbf{a}_i^{q_m} \cdot \left(\frac{J^1(\mathbf{r}_{q_m})\,\mathbf{a}_1^{q_m}}{\mathcal{J}(\mathbf{r}_{q_m})} + \frac{J^2(\mathbf{r}_{q_m})\,\mathbf{a}_2^{q_m}}{\mathcal{J}(\mathbf{r}_{q_m})} \right),
\end{aligned} \tag{2.6}$$

where, at each quadrature point $\mathbf{r}_{q_m}$, the tangential testing directions are the covariant unit vectors evaluated there: $\hat{\mathbf{t}}_i(\mathbf{r}_{q_m}) = \mathbf{a}_i(\mathbf{r}_{q_m}),\ i = 1, 2.$

Numerically integrating over each element S_e leads to a system of linear equations in terms of each of the two current components at the quadrature points:

$$\begin{aligned}
\mathbf{a}_i^{q_m} \cdot \Big[\alpha \mathbf{E}^{\rm inc}(\mathbf{r}_{q_m}) + (1-\alpha)\,\eta_0\, \hat{\mathbf{n}}^{q_m} \times \mathbf{H}^{\rm inc}(\mathbf{r}_{q_m})\Big] = \\
\sum_{e=1}^{N_e} \sum_{q=1}^{Q} \Big[\, w^{i1}_{q_m,q_e}\, J^1(\mathbf{r}_{q_e}) + w^{i2}_{q_m,q_e}\, J^2(\mathbf{r}_{q_e})\Big],
\end{aligned} \tag{2.7}$$

where, $w^{ij}_{q_m,q_e}$ accounts for the integration weight and integrand as follows:

$$w^{ij}_{q_m,q_e} = -\, w_{q_e}\, \mathbf{a}_i^{q_m} \cdot \Big[\alpha \bar{\mathbf{G}}_{EJ}(\mathbf{r}_{q_m}, \mathbf{r}_{q_e}) \cdot \mathbf{a}_j^{q_e} + (1-\alpha)\,\eta_0\, \hat{\mathbf{n}}^{q_m} \times \big(\bar{\mathbf{G}}_{HJ}(\mathbf{r}_{q_m}, \mathbf{r}_{q_e}) \cdot \mathbf{a}_j^{q_e}\big)\Big], \tag{2.8}$$

where $i = 1, 2$ is the testing component index and $j = 1, 2$ is the current component index. Due to the singular behavior of $G_0(\mathbf{r}_{q_m}, \mathbf{r}_{q_e})$, the standard weights in (2.8) are not suitable when the observation point $\mathbf{r}_{q_m}$ is in, or near, the integration element. To accurately evaluate (hyper-) singular integrals we adopt a local correction procedure [3, 7]. Simply put, local corrections compute a set of integral weights that can be used to accurately evaluate a specific singular integral. These new weights, $\tilde{w}^{ij}_{q_m,q_e}$ are obtained by solving a small system

of equations for each element:

$$\begin{aligned}&\sum_{q=1}^{Q} \tilde{w}^{ij}_{q_m,q_e}\, \mathbf{F}_k(\mathbf{r}_{q_e}) = \\ &-\ \mathbf{a}_i^{q_m} \cdot \int_S \Big[\alpha\, \bar{\mathbf{G}}_{EJ}(\mathbf{r}_{q_m}, \mathbf{r}') + (1-\alpha)\, \eta_0 \hat{\mathbf{n}}^{q_m} \times \bar{\mathbf{G}}_{HJ}(\mathbf{r}_{q_m}, \mathbf{r}')\Big] \cdot \Big(\frac{\mathbf{F}_k(\mathbf{r}')}{\mathcal{J}(\mathbf{r}')}\, \mathbf{a}_j(\mathbf{r}')\Big)\, dS' \,,\end{aligned} \tag{2.9}$$

Here, $k = 1, \ldots, Q$, and $\{\mathbf{F}_k\}$ denotes a set of 2-D Legendre polynomials, complete to a prescribed polynomial order. The essential idea is that solving this system of equations will produce a set of quadrature weights that will exactly integrate the CFIE kernel for any 2-D polynomial current component (up to the prescribed order) over the element. The matrix arising on the left-hand-side of (2.9) is a $Q \times Q$ Vandermonde matrix [6]. To fill the right-hand-side vector, it is necessary to accurately evaluate the (hyper-) singular integral over S_e. These local correction systems are solved to produce the locally corrected weights for all singular (or nearly singular) integrals. Once those weights are known, they are introduced as required into (2.7) resulting in an accurate global discretization of the CFIE.

Assuming we decompose the interactions for a particular quadrature point $\mathbf{r}_{q_m}$ into a near interaction set $\mathcal{S}^{\text{near}}_{\mathbf{r}_{q_m}}$ (requiring local corrections) and a far interaction set $\mathcal{S}^{\text{far}}_{\mathbf{r}_{q_m}}$ (captured accurately with the standard quadrature rule), in the end, the LCN discretization of the CFIE results in the system of linear equations:

$$\mathbf{Z}_{\text{LCN}} \cdot \begin{bmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \end{bmatrix}. \tag{2.10}$$

The impedance matrix $\mathbf{Z}_{\text{LCN}}$ is a $2N_eQ \times 2N_eQ$ dense matrix of the form

$$\mathbf{Z}_{\text{LCN}} = \begin{bmatrix} \mathbf{Z}_{1,1} & \mathbf{Z}_{1,2} \\ \mathbf{Z}_{2,1} & \mathbf{Z}_{2,2} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_{1,1}^{\text{far}} & \mathbf{Z}_{1,2}^{\text{far}} \\ \mathbf{Z}_{2,1}^{\text{far}} & \mathbf{Z}_{2,2}^{\text{far}} \end{bmatrix} + \begin{bmatrix} \mathbf{Z}_{1,1}^{\text{near}} & \mathbf{Z}_{1,2}^{\text{near}} \\ \mathbf{Z}_{2,1}^{\text{near}} & \mathbf{Z}_{2,2}^{\text{near}} \end{bmatrix}, \tag{2.11}$$

where the natural decomposition into blocks results from the two basis components used for representing the surface current. Our goal now is to efficiently solve (2.10) for the unknown coefficient vectors $\mathbf{J}_1$ and $\mathbf{J}_2$ resulting from a particular incident field that results in the right-hand-side coefficients $\mathbf{H}_1$ and $\mathbf{H}_2$.

2.3 $\mathcal{H}$-matrix Framework

In the context of solving large-scale linear systems resulting from high-order point-based discretizations, namely, the Locally Corrected Nyström (LCN) and Chebyshev-Based Integral Equation (CBIE) methods, where the latter employs Chebyshev polynomial interpolation instead of local corrections, conventional storage and direct factorization of the coefficient matrices can lead to substantial memory consumption and computational overhead. Even for moderately sized problems (i.e., a few thousand unknowns), the demands associated with storing and directly factorizing these matrices can quickly become prohibitive, making the process infeasible. To address these computational challenges, acceleration techniques based on $\mathcal{H}$-matrices have been developed [8, 9]. These techniques enable efficient compression and hierarchical storage of the matrices, significantly reducing memory usage and computational costs, while preserving the high-order accuracy that characterizes the original LCN and CBIE methods. The $\mathcal{H}$-matrix approach leverages recursive block structures and low-rank approximations, facilitating efficient matrix operations that are crucial for handling

complex, large-scale problems.

$\mathcal{H}$-matrices provide a hierarchical, tree-based framework for efficient matrix representation and operations. Unlike conventional dense or sparse matrix formats, $\mathcal{H}$-matrices utilize a recursive block-structured approach that allows the matrix to be decomposed into smaller sub-blocks, each of which can be analyzed for compressibility. This hierarchical organization facilitates the identification of patterns and structures within the matrix, particularly enabling the detection of blocks that can be efficiently approximated. Specifically, $\mathcal{H}$-matrices take advantage of the fact that sub-blocks corresponding to interactions between far-separated observation and source points exhibit low-rank characteristics. By representing these blocks in a compressed, low-rank format, $\mathcal{H}$-matrices can significantly reduce the storage requirements and computational complexity associated with large-scale problems. This compression capability is especially beneficial in applications where the underlying physical phenomena lead to well-separated interactions, as it reduces the overall data footprint and speeds up matrix operations without sacrificing accuracy [10]. Furthermore, the recursive nature of $\mathcal{H}$-matrices allows for efficient algorithms for matrix multiplication, factorization, and inversion, which are critical in the iterative and direct solution of large linear systems.

The construction of $\mathcal{H}$-matrices begins by organizing the source and observation basis functions into a hierarchical cluster tree, which is efficiently achieved using Orthogonal Recursive Bisection (ORB) [11]. As depicted in Fig. 2.2, the ORB technique can be implemented based on either geometric bisection or cardinality-based (algebraic) partitioning, leading to a binary cluster structure that serves as the foundation for the hierarchical matrix representation. In this process, the domain containing the basis functions is recursively divided, with each step of the bisection creating two child clusters from a parent cluster. Each node in the resulting cluster tree represents a subset of the total basis functions, effectively

grouping them based on spatial proximity or algebraic similarity. This recursive subdivision continues until a specified criterion is met, typically when each leaf node contains a number of basis functions below a pre-determined maximum threshold. By organizing the basis functions into this hierarchical structure, the $\mathcal{H}$-matrix framework can more readily identify low-rank approximations within the matrix blocks, thereby facilitating efficient compression and enabling scalable computation for large-scale problems.

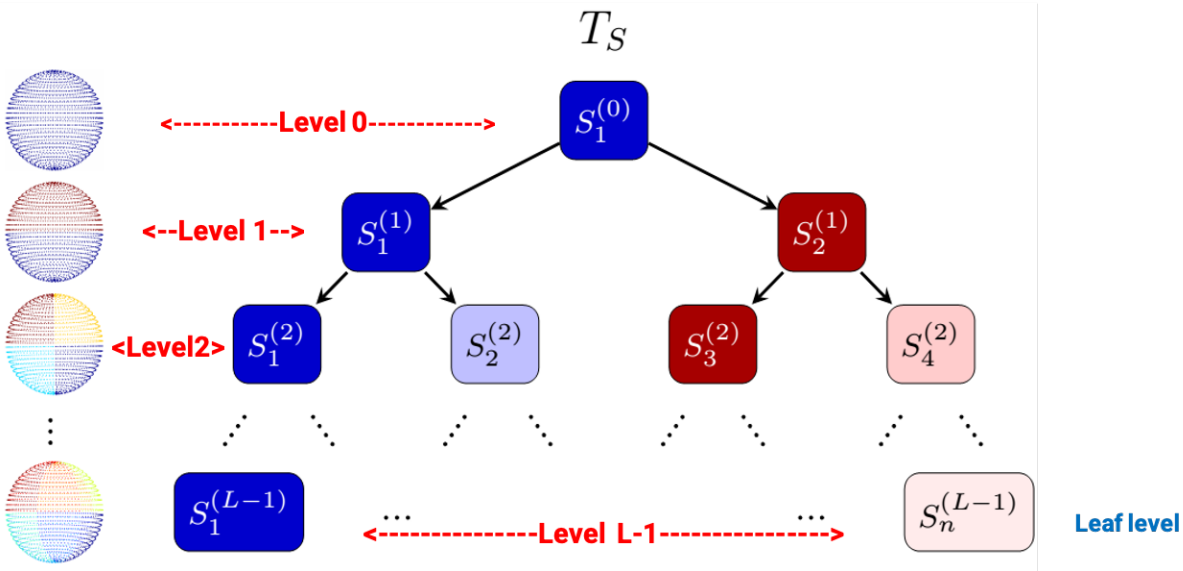


Figure 2.2: Construction of a binary cluster tree for the LCN basis functions. Each bisection of the basis functions corresponds to two children in the tree.

From the hierarchical cluster tree, a block cluster tree is constructed by pairing nodes from the source and observation cluster trees, as illustrated in Fig. 2.3. Each pair of clusters undergoes an assessment based on a specific admissibility condition, which determines whether the corresponding matrix block can be approximated in a compressed, low-rank format. The admissibility condition is defined as:

$$\min\{\mathrm{diam}(S_{\mathrm{obs}}), \mathrm{diam}(S_{\mathrm{src}})\} \leq \hat{\eta}\ \mathrm{dist}(S_{\mathrm{obs}}, S_{\mathrm{src}}), \tag{2.12}$$

where S_{obs} and S_{src} represent clusters of observation points and source points, respectively, and $\hat{\eta}$ is a predefined admissibility parameter that controls the threshold for compression. The condition essentially states that if the diameter (size) of the smaller cluster is less than or equal to a fraction $\hat{\eta}$ of the distance between the centers of the two clusters, the block is considered admissible. This enables selective compression of the matrix: admissible blocks are stored using low-rank approximations, significantly reducing memory requirements, while inadmissible blocks, where the approximation is not feasible, are stored in their full, uncompressed form [10].

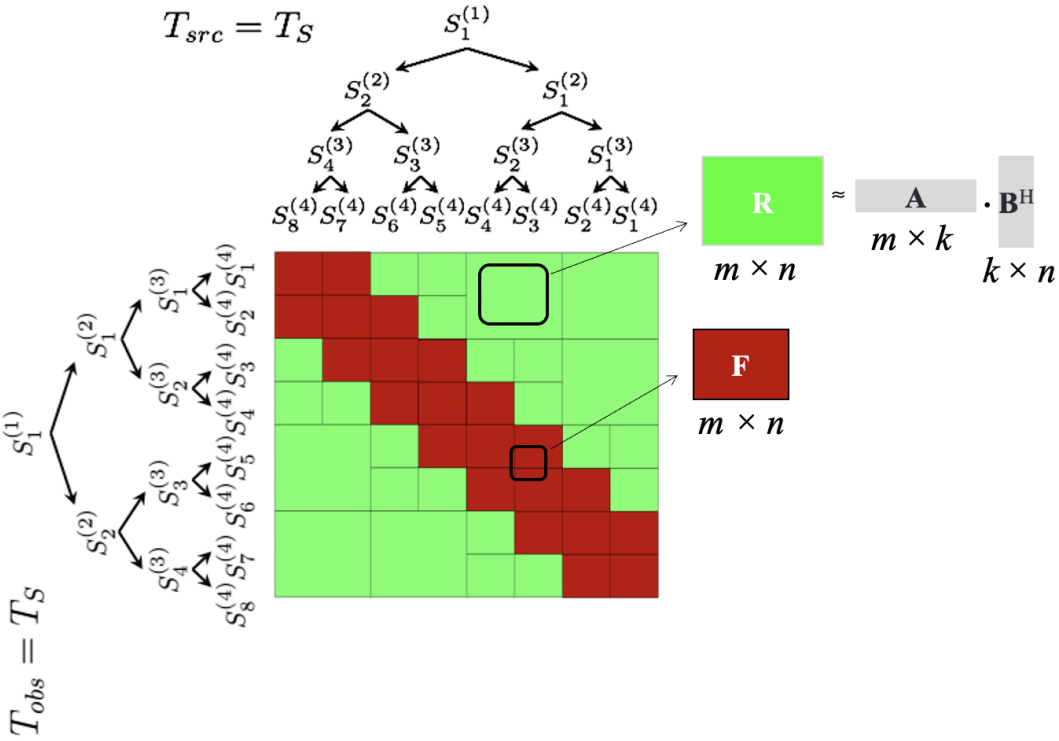


Figure 2.3: Illustration of admissible (compressible) and inadmissible blocks in an $\mathcal{H}$-matrix. The source and observation cluster trees are used to construct a block cluster tree partitioning the matrix into admissible (green) and inadmissible (red) blocks. Admissible blocks are stored in a rank-compressed format.

The recursive structure of $\mathcal{H}$-matrices makes them particularly suitable for efficiently handling large-scale problems. This hierarchical approach divides the matrix into smaller sub-blocks, some of which can be represented in a compressed, low-rank form. By systematically identifying and compressing the low-rank blocks across the hierarchy, $\mathcal{H}$-matrices reduce the need for storing and processing large, dense matrices, leading to substantial reductions in both computational cost and memory usage. This is especially beneficial in applications involving boundary element methods or integral equation solvers, where traditional dense matrix representations become infeasible due to the prohibitive resource demands.

The key advantage of $\mathcal{H}$-matrices lies in their ability to leverage the block cluster tree, a hierarchical structure that pairs clusters of observation and source points. Each block is assessed to determine whether it can be approximated using a low-rank representation, based on an admissibility condition that considers the size of the clusters and the distance between them. This selective compression strategy allows $\mathcal{H}$-matrices to efficiently store only the admissibl blocks in a compressed format, while the inadmissible blocks, which cannot be accurately approximated, are stored in full. By selectively compressing matrix blocks, $\mathcal{H}$-matrices enable the management of very large datasets and complex systems that would otherwise be impractical with traditional dense storage methods.

Another feature of $\mathcal{H}$-matrices is their support for fast arithmetic operations. Unlike conventional dense or sparse matrices, $\mathcal{H}$-matrices can perform matrix-vector multiplications, matrix additions, and even more complex operations such as inversion and factorization in a hierarchical, recursive manner that retains the low-rank structure where possible. For example, the $\mathcal{H}$-matrix framework enables efficient $\mathcal{H}$-$\mathbf{LU}$ decomposition, which is crucial for solving linear systems and can be employed as part of a direct solver. Additionally, $\mathcal{H}$-forward and $\mathcal{H}$-backward substitution methods facilitate fast preconditioning of iterative

solvers [12]. These capabilities make $\mathcal{H}$-matrices particularly attractive for large-scale problems, where iterative methods like GMRES[13] or Conjugate Gradient[14] require effective preconditioners to ensure fast convergence.

The flexibility of the $\mathcal{H}$-matrix format also extends to hybrid direct-iterative solvers. In scenarios where a purely iterative approach may converge too slowly, $\mathcal{H}$-matrices can be used to create preconditioners that accelerate convergence by approximating the inverse of the system matrix, thereby reducing the condition number of the linear system. This hybrid approach combines the robustness of direct solvers with the scalability of iterative methods, enabling the efficient solution of very large systems.

The hierarchical nature of $\mathcal{H}$-matrices is particularly advantageous for problems where the kernel interactions between well-separated source and observation domains exhibit asymptotic smoothness, that is, the Green's function entries vary slowly with respect to spatial separation. In such regions, the corresponding subblocks of the system matrix can be efficiently approximated by low-rank representations, while near-field interactions, which contain strong local variations and singular behavior, are stored and computed in full precision. This hierarchical separation between low-rank far-field and dense near-field blocks drastically reduces computational cost while preserving overall solution accuracy.

As illustrated in Fig. 2.4, the $\mathcal{H}$-matrix representation of an LCN impedance matrix demonstrates how a combination of compressed and uncompressed structures is used to solve a large-scale problem. The blocks representing interactions between well-separated basis functions are stored in a compressed, low-rank form, while the blocks involving nearby basis functions are stored without compression. This hybrid storage format allows $\mathcal{H}$-matrices to efficiently handle problems with millions of unknowns, which would otherwise be impossible with traditional approaches.

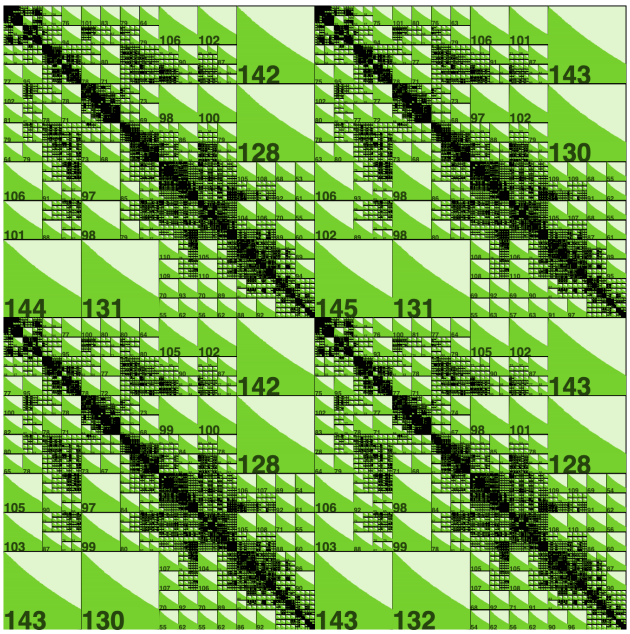


Figure 2.4: The $\mathcal{H}$-matrix for a large-scale LCN problem is shown, where numbers in a block indicate the rank used for low-rank approximation [12].

In summary, $\mathcal{H}$-matrices offer a powerful framework for managing and solving large-scale linear systems, particularly those arising from integral equation discretizations. Their hierarchical, recursive approach not only reduces storage and computational costs but also enables the development of efficient solvers and preconditioners. By maintaining a balance between compression and exact representation, $\mathcal{H}$-matrices ensure that large-scale problems can be addressed with high accuracy and efficiency, providing a robust tool for a wide range of applications in computational science and engineering.

2.4 Solving Linear System of Equations: Direct and Iterative Approaches

Efficiently solving large linear systems of equations is a fundamental challenge in scientific computing, particularly in the field of integral equations. The choice between direct and iterative methods depends on factors such as the size of the system, memory constraints, and the desired accuracy of the solution. In this section, we explore both approaches and how $\mathcal{H}$-matrices are utilized to accelerate these solutions, providing insights into their benefits and trade-offs.

2.4.1 Direct Solution Methods

Direct methods for solving linear systems, such as Gaussian elimination and **LU** decomposition, aim to find the exact solution by systematically reducing the system to a simpler form. These methods are often preferred when high accuracy is required and the system matrix is relatively small or well-structured. The **LU** decomposition, for example, involves factorizing the matrix $\mathbf{A}$ into a product of lower ($\mathbf{L}$) and upper ($\mathbf{U}$) triangular matrices such that $\mathbf{A} = \mathbf{LU}$. The solution can then be obtained through forward and backward substitution steps.

In the context of $\mathcal{H}$-matrices, direct methods are adapted to leverage the hierarchical structure, leading to $\mathcal{H}$-**LU** decompositions. These decompositions approximate the exact **LU** factors using low-rank representations, significantly reducing the computational complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(N \log^2 N)$ or $\mathcal{O}(N \log^3 N)$, where N is the number of unknowns [8]. Despite the efficiency improvements, $\mathcal{H}$-**LU** decompositions can still be computationally intensive for extremely large systems due to the need to accurately represent the entire ma-

trix, which is why they are sometimes used as preconditioners instead of standalone solvers.

2.4.2 Iterative Solution Methods

Iterative methods are often the preferred choice for solving large sparse systems where direct methods would be computationally prohibitive. Unlike direct methods, iterative solvers do not attempt to find the exact solution in a single step; instead, they generate a sequence of approximate solutions that converge to the true solution over multiple iterations. The efficiency of these methods depends strongly on the spectral properties of the system matrix and the quality of the initial guess.

Among the most widely used iterative solvers is the Generalized Minimal Residual (GMRES) method [13]. GMRES is a Krylov subspace method that iteratively minimizes the residual $||\mathbf{Ax} - \mathbf{b}||$ over a sequence of subspaces, where $\mathbf{A}$ is the system matrix, $\mathbf{x}$ is the solution vector, and $\mathbf{b}$ is the excitation vector. One of its advantages is the ability to handle non-symmetric and non–positive-definite matrices, making it applicable to a broad range of integral equation formulations. However, GMRES can experience slow convergence or stagnation when the spectral condition number of $\mathbf{A}$ is large.

To accelerate convergence, preconditioners are typically introduced. Preconditioning transforms the system $\mathbf{Ax} = \mathbf{b}$ into $\mathbf{M}^{-1}\mathbf{Ax} = \mathbf{M}^{-1}\mathbf{b}$, where $\mathbf{M}$ approximates $\mathbf{A}$ and is designed such that $\mathbf{M}^{-1}\mathbf{A}$ exhibits more favorable spectral properties.

In the present work, $\mathcal{H}$-matrices are employed both for compressing the dense system matrix and for constructing efficient preconditioners. Two distinct $\mathcal{H}$-matrix representations are used: (1) a high-accuracy representation of the system matrix, denoted $\mathcal{Z}$, constructed with a tight compression tolerance ϵ_z; and (2) a lower-accuracy preconditioner, denoted $\mathcal{P}$, built with a relaxed tolerance $\epsilon_p > \epsilon_z$. This approach enables fine-grained control over

computational cost and precision, while $\mathcal{Z}$ retains the accuracy of the original discretization, $\mathcal{P}$ provides an efficient approximation that significantly accelerates iterative convergence without excessive storage or factorization cost.

The resulting preconditioned system takes the form of Eq. (4.7), where the inverse of the preconditioner is applied to the impedance matrix. The corresponding hierarchical structure is illustrated in Fig. 4.3, showing the recursive $\mathcal{H}$-**LU** factorization applied to $\mathcal{P}$. When $\epsilon_p = \epsilon_z$, the preconditioner becomes equivalent to the system matrix, and the iterative solver converges in a single step (equivalent to a direct $\mathcal{H}$-**LU** solution). Increasing ϵ_p reduces the cost of building and factoring $\mathcal{P}$, though at the expense of requiring additional GMRES iterations. The trade-off between preconditioner accuracy, memory consumption, and overall runtime efficiency is systematically analyzed in Chapter 4.

2.4.3 Adaptive *hp*–Refinement LCN Overview

In electromagnetic scattering from perfectly conducting surfaces, sharp edges and corners introduce singularities in the surface current density. These singularities manifest as unbounded field behavior near geometric discontinuities, causing standard high-order discretizations to lose accuracy and stability if uniform polynomial orders are used across the entire model. In particular, over-refinement in smooth regions leads to excessive computational cost, while insufficient resolution near edges fails to capture the steep current gradients. To address these issues, adaptive *hp*–refinement strategies are employed, combining local mesh refinement (*h*–refinement) with variable polynomial order (*p*–refinement) to resolve edge singularities efficiently while maintaining high-order accuracy over smooth surfaces.

The Locally Corrected Nyström (LCN) formulation provides a suitable framework for

such adaptation, as its point-based discretization and local correction mechanism naturally support non-uniform element sizes and polynomial orders. When coupled with hierarchical $\mathcal{H}$–matrix acceleration, these adaptive discretizations can be efficiently applied to large-scale scattering problems, enabling accurate modeling of complex geometries that include both sharp features and smooth surfaces.

2.4.4 Computational Efficiency and Parallelization

Modern large-scale electromagnetic simulations require efficient use of high-performance computing (HPC) resources to manage the substantial memory and computational demands of dense integral equation formulations. Two widely used parallel programming paradigms are the Message Passing Interface (MPI) and Open Multi-Processing (OpenMP). MPI provides a distributed-memory model in which data and computations are divided among multiple compute nodes that communicate through message passing, allowing simulations to scale across clusters and supercomputers. In contrast, OpenMP supports shared-memory parallelism within a single node, distributing loop iterations or independent tasks across multiple processor cores.

Combining these paradigms, known as hybrid MPI/OpenMP parallelism, enables efficient utilization of both inter-node and intra-node resources, balancing communication overhead and workload distribution. Such parallelization strategies are essential for accelerating matrix assembly, compression, and solver routines in boundary element and integral equation solvers, particularly when problem sizes reach millions of unknowns. A detailed implementation of this hybrid strategy in the proposed solver is discussed later in chapter 3.

2.5 Summary

This chapter has established the theoretical groundwork for the high-order solver developed in later chapters. The key background topics reviewed are:

- **Geometry representation** — Non-Uniform Rational B-Spline (NURBS) surfaces tessellated into quadrilateral Bézier patches provide a smooth, high-order geometric description suitable for complex PEC scatterers.

- **High-order integral equation discretizations** — The Locally Corrected Nyström (LCN) method was introduced as high-order, point-based discretization technique, highlighting its ability to achieve high accuracy with fewer degrees of freedom than traditional low-order MoM approaches.

- **Edge singularities and adaptive *hp* refinement** — The behavior of surface currents near sharp edges and corners was discussed, emphasizing the need for adaptive *hp*-refinement to capture singular current distributions efficiently. The concept of combining local mesh refinement (h) with variable polynomial order (p) was presented as a general strategy for maintaining accuracy and stability across geometrically complex regions.

- **$\mathcal{H}$-matrix fundamentals** — The principles of hierarchical matrix compression were reviewed, including cluster-tree construction, admissibility conditions, adaptive cross approximation ($\mathcal{ACA}$), and $\mathcal{H}$-**LU** factorization, together with their computational complexity scaling of approximately $\mathcal{O}(r^2 N \log^2 N)$.

- **Parallelization concepts** — The fundamental ideas of distributed-memory (MPI) and shared-memory (OpenMP) parallelization were summarized, explaining how hy-

brid MPI/OpenMP models can efficiently exploit inter-node and intra-node resources in large-scale electromagnetic simulations.

Together, these topics provide the theoretical basis for understanding the development and evaluation of the $\mathcal{H}$-matrix–accelerated, *hp*-refined integral equations solver presented in the subsequent chapters.

Bibliography

[1] L. Piegl and W. Tiller, *The NURBS Book*. Springer, 1997.

[2] J. A. Stratton, *Electromagnetic Theory*. Piscataway, NJ, USA: IEEE Press, 2007.

[3] S. D. Gedney, "The combined field integral equation (CFIE) for scattering problems," *IEEE Transactions on Antennas and Propagation*, vol. 51, no. 7, pp. 1710–1717, 2003.

[4] R. F. Harrington and J. R. Mautz, "Theory of characteristic modes for conducting bodies," *IEEE Transactions on Antennas and Propagation*, vol. 19, no. 5, pp. 622–628, 1967. doi:10.1109/TAP.1967.1138995.

[5] S. D. Gedney, "An overview of the combined field integral equation," *IEEE Antennas and Propagation Magazine*, vol. 56, no. 4, pp. 19–32, 2014.

[6] E. Bleszynski, M. Bleszynski, and T. Jaroszewicz, "AIM: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems," *Radio Science*, vol. 31, no. 5, pp. 1225–1251, Sep. 1996.

[7] L. S. Canino, D. R. Jackson, D. R. Wilton, N. J. Champagne, and F. Reitich, "A hybrid method for solving the problem of electromagnetic scattering from a cavity in a ground

plane," *IEEE Transactions on Antennas and Propagation*, vol. 44, no. 3, pp. 292–300, Mar. 1996.

[8] W. Hackbusch, *Tensor Spaces and Numerical Tensor Calculus*. Berlin, Germany: Springer-Verlag, 2002.

[9] R. Kriemann, *HLIBpro C Language Interface*. Max Planck Institute for Mathematics in the Sciences, 2008.

[10] W. Hackbusch, B. Khoromskij, and R. Kriemann, "Hierarchical matrices based on a weak admissibility criterion," *Computing*, vol. 73, pp. 207–243, 2004. doi:10.1007/s00607-004-0080-4.

[11] R. Gholami, J. Mojolagbe, A. Menshov, F. Sheikh Hosseini Lori, and V. Okhmatovski, "$\mathcal{H}$-matrix arithmetic for fast direct and iterative method of moment solution of surface-volume-surface EFIE for 3-D radiation problems," *Progress in Electromagnetics Research B*, vol. 82, pp. 189–210, 2018.

[12] R. Kriemann, "**LU** factorization on many-core systems," *Computing and Visualization in Science*, vol. 16, pp. 105–117, 2013. doi:10.1007/s00791-014-0226-7.

[13] Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing*, 1986.

[14] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409–436, 1952. doi:10.6028/jres.049.044.

Chapter 3

Overview of the Proposed Framework

The methodology presented in this chapter outlines the design and implementation of the framework developed in this thesis. By integrating high-order methods, $\mathcal{H}$-matrix compression, and advanced parallel computing techniques, this framework addresses the challenges associated with solving large-scale electromagnetic problems involving complex PEC geometries. The following sections detail each component of the framework, including the high-order discretization methods, $\mathcal{H}$-matrix techniques, solver strategies, and the parallelization approach, culminating in a robust and scalable solution platform.

3.1 High-Order Discretization Methods

This section describes the high-order numerical methods employed to achieve high accuracy in discretizing integral equations for PEC geometries. We focus on the Locally Corrected Nyström (LCN) method and Chebyshev-Based Integral Equation (CBIE) formulation, both of which enhance convergence and precision in solving electromagnetic scattering problems.

3.1.1 Locally Corrected Nyström (LCN) Method

The Locally Corrected Nyström (LCN) method provides an efficient framework for discretizing integral equations by incorporating locally defined correction terms that enhance accuracy near complex boundaries, such as edges and corners, without the need for prohibitively fine meshes. These local corrections are particularly advantageous for high-frequency problems, where traditional low-order methods often encounter significant numerical errors in regions with geometric complexity.

In this work, the LCN method is applied to the Magnetic Field Integral Equation (MFIE), Electric Field Integral Equation (EFIE), and Combined Field Integral Equation (CFIE) formulations across a variety of geometric configurations. By leveraging local corrections, the LCN method allows for stable and accurate approximations of the electromagnetic fields on perfectly electrically conducting (PEC) surfaces, even in the presence of sharp transitions and high-frequency oscillations.

The implementation of LCN in this framework is optimized to maintain computational efficiency while providing robust solutions. This optimization is achieved through a systematic selection of sampling points and correction terms, tailored for each type of geometry and frequency range. Additionally, the LCN method is integrated with $\mathcal{H}$-matrix techniques to further reduce computational complexity, enabling efficient matrix assembly and solution processes even for large-scale geometries. The use of local corrections within this framework allows the solution to converge more rapidly, particularly in complex regions, enhancing both precision and stability.

3.1.2 Adaptive Integration of the Integral Equations Kernel

When the observation point is within the integration patch, calculating the locally corrected weights using (2.9) involves a (hyper)-singular integral due to the singularity of the free-space Green's function $G_0(\mathbf{r}, \mathbf{r}')$.

$$G_0(\mathbf{r}, \mathbf{r}') = \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|}, \tag{3.1}$$

where k is the wavenumber in free space. While a combination of manipulating the integral expression [1] and a Duffy transformation [2] enables a reasonable numerical evaluation of the integral, adaptive integration is required to compute the integral to a prescribed accuracy. Our adaptive approach recursively divides the integration area (Bézier quadrilateral) into four sub-quadrilaterals. At each recursive step/level, the integral is computed and compared to the previous level. Once the values agree to the desired accuracy, the recursion stops. In this way, new levels are introduced in areas where more accuracy is required.

This same adaptive integration process has value for other near (but non-singular) interactions. As an example, Fig. 3.1 illustrates the adaptive integration process for a non-singular integration, where the observation point is outside, but close to the source patch. The need for subsequent levels of integration shows that the integral with the standard weights (at Level 1) was not sufficiently accurate, despite the integral being non-singular.

3.1.3 Computing the near and far interaction sets

As evaluating the singular integrals needed to compute the locally corrected weights is significantly more expensive than using the regular quadrature rule, it is imperative to partition the interaction set into a minimal set of near interactions $S^{\text{near}}_{\mathbf{r}_{q_m}}$ (singular or near-singular

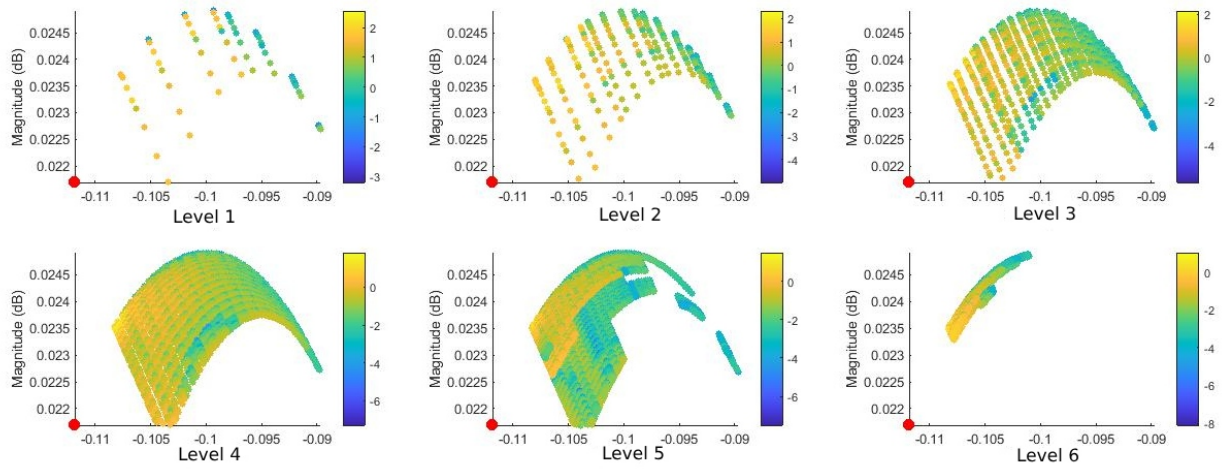


Figure 3.1: Adaptive integration for a near, but non-singular, interaction.

with locally corrected weights) and the remaining set of far interactions $S^{\text{far}}_{\mathbf{r}_{q_m}}$ (non-singular with standard weights) that ensures that the CFIE is discretized to the desired precision. To more precisely define a "near" interaction we compute an LCN radius parameter R_{LCN}, defined as the minimum distance $|\mathbf{r}_{q_m} - \mathbf{r}|$ for a given observation point $\mathbf{r}_{q_m}$, such that the integral (2.6) using the standard quadrature rule matches the integral computed with the locally corrected weights (to a desired accuracy). Rather than recompute this distance for each element, we consider the largest mesh element (thus having the largest potential for varying surface current) and gradually move it away from a single, fixed, observation point. Once the patch is far enough from the point to achieve the desired accuracy, that radius is adopted as R_{LCN}, representing a worst-case scenario for all possible numerical integrals encountered in the discretization. Once the radius is computed, we employ a fast tree-based search to efficiently calculate the near and far interaction sets [3]. As depicted in Fig. 3.2, the centroids of all Bézier patches are pushed into an oct-tree. Subsequently, the near neighbor Bézier patches of a given observation point can be found by exploring a progressively

increasing radius surrounding that patch's leaf node in the oct-tree. The overall CPU time complexity of determining the near and far interaction sets for all points is $\mathcal{O}(N_e Q \log(N_e))$, as documented in [3].

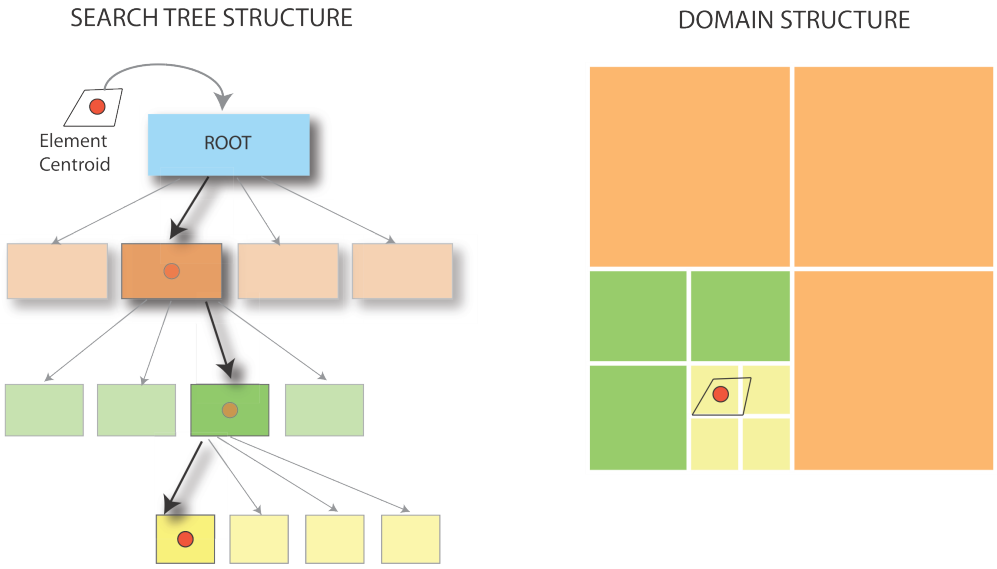


Figure 3.2: Near neighbour search tree. Each element is pushed into the tree so that its geometric proximity to other elements can be determined efficiently.

3.1.4 Efficient calculation of LCN discretized integrals

In LCN discretization, achieving precision without excessive computational cost requires advanced techniques. We leverage Bézier patches to maintain high geometric fidelity while managing surface integrals effectively. To evaluate the integral expressions, we consider transformations like the Duffy transform [2] for handling singularities and adaptive integration when necessary. The approach helps ensure that the accuracy of the integral matches the precision required by CFIE formulation, minimizing computational effort.

Furthermore, the integration scheme must account for the influence of edges and other geometrical features, particularly when they introduce local variations in the fields. The

proposed adaptive integration, combined with the oct-tree structure, can adapt to these variations by refining patches near such features. This ensures a balance between computational efficiency and integral accuracy, making the approach scalable for larger problems.

The combination of these steps within the LCN framework allows for an efficient and accurate solution to the integral equations governing electromagnetic scattering, paving the way for high-order methods to be applied in practical engineering problems.

3.2 Chebyshev-Based High-Order Accurate Magnetic Field Integral Equation(MFIE)

Chebyshev-based approaches, explored extensively in [4], provide another avenue for achieving high-order accuracy. These methods utilize Chebyshev polynomials $T_n(x)$ to approximate the surface current density and the integrand functions in the integral equations, leading to reduced numerical errors and improved convergence rates. Again, with a suitable geometric representation chosen, the discretization of the MFIE using Chebyshev polynomials and weights is obtained. Let's consider the MFIE formulation as:

$$\frac{\mathbf{J}}{2} + \mathcal{K}\,\mathbf{J} \;=\; \hat{\mathbf{n}} \times \mathbf{H}^{\text{inc}}, \tag{3.2}$$

where $\mathbf{J}$ is the surface current density, $\mathbf{H}^{\text{inc}}$ is the incident magnetic field, $\hat{\mathbf{n}}$ is the unit normal vector, and $\mathcal{K}$ is the integral operator

$$\mathcal{K}[\mathbf{a}](\mathbf{r}) \;=\; \hat{\mathbf{n}}(\mathbf{r}) \times \int_S \mathbf{a}(\mathbf{r}') \times \nabla G_0(\mathbf{r} - \mathbf{r}')\,\mathrm{d}S'), \tag{3.3}$$

where ∇ represents the gradient with respect to the observation coordinates $\mathbf{r}$, and G_0 denotes the free-space scalar Green's function for the Helmholtz equation. We now focus on the discretization of the operator $\mathcal{K}$. It is evident that any integral over integrations domain can be decomposed into a sum of integrals over each of the M patches:

$$\mathcal{K}[\mathbf{J}](\mathbf{r}) = \sum_{p=1}^{M} \mathcal{K}[\mathbf{J}^p](\mathbf{r}), \tag{3.4}$$

$$\mathcal{K}[\mathbf{J}^p](\mathbf{r}) = \hat{\mathbf{n}}(\mathbf{r}) \times \int_S \mathbf{J}^p(\mathbf{r}') \times \nabla G_0(\mathbf{r} - \mathbf{r}')\, \mathrm{d}S'). \tag{3.5}$$

This can be further expressed as

$$\begin{aligned} \mathcal{K}[\mathbf{J}^p](\mathbf{r}) = \hat{\mathbf{n}}(\mathbf{r}) \times \int_{-1}^{1}\int_{-1}^{1} \Big(& J^{p,\xi}(\xi,\eta)\, \mathbf{a}_\xi^p(\xi,\eta) + J^{p,\eta}(\xi,\eta)\, \mathbf{a}_\eta^p(\xi,\eta) \Big) \\ & \times \nabla G_0\big(\mathbf{r} - \mathbf{r}^p(\xi,\eta)\big) \Big)\, \mathrm{d}\xi\, \mathrm{d}\eta. \end{aligned} \tag{3.6}$$

Note that $\sqrt{|g^p(\xi,\eta)|}$ in the denominator of the expansion cancels with the Jacobian $\sqrt{|g^p(\xi,\eta)|}$ present in the integral. In its current formulation, Eq. (3.6) involves the hyper-singular kernel ∇G_0; however, it can be re-expressed using the BAC–CAB vector identity as:

$$\begin{aligned} \mathcal{K}[\mathbf{J}^p](\mathbf{r}) = & \int_{-1}^{1}\int_{-1}^{1} J^{p,\xi}(\xi,\eta) \cdot \Big(\mathbf{a}_\xi^p(\xi,\eta)\, \frac{\partial G_0\big(\mathbf{r} - \mathbf{r}^p(\xi,\eta)\big)}{\partial \hat{\mathbf{n}}(\mathbf{r})} \\ & - \nabla G_0\big(\mathbf{r} - \mathbf{r}^p(\xi,\eta)\big)\, \hat{\mathbf{n}}(\mathbf{r}) \cdot \mathbf{a}_\xi^p(\xi,\eta) \Big)\, \mathrm{d}\xi\, \mathrm{d}\eta \\ & + \int_{-1}^{1}\int_{-1}^{1} J^{p,\eta}(\xi,\eta) \cdot \Big(\mathbf{a}_\eta^p(\xi,\eta)\, \frac{\partial G_0\big(\mathbf{r} - \mathbf{r}^p(\xi,\eta)\big)}{\partial \hat{\mathbf{n}}(\mathbf{r})} \\ & - \nabla G_0\big(\mathbf{r} - \mathbf{r}^p(\xi,\eta)\big)\, \hat{\mathbf{n}}(\mathbf{r}) \cdot \mathbf{a}_\eta^p(\xi,\eta) \Big)\, \mathrm{d}\xi\, \mathrm{d}\eta, \end{aligned} \tag{3.7}$$

where $\mathbf{a}_\xi^p(\xi,\eta)$ and $\mathbf{a}_\eta^p(\xi,\eta)$ represent the covariant tangent vectors along the ξ- and η-directions, respectively, on the Mth Bézier patch. Eq. 3.7 is weakly singular since $\hat{\mathbf{n}}(\mathbf{r})\cdot\mathbf{a}_\ell^p$ approaches 0 as $\mathbf{r}^p(\xi,\eta)\to\mathbf{r}$. We must obtain $2\sum_{p=1}^{M} N_\ell^p N_0^p$ linearly independent equations in order to obtain a uniquely solvable linear system for approximating $\mathbf{J}$ on S, integration domain. This is achieved by using a collocation method and testing at the same points as the unknowns.

Hence, the discretized versions of Eq. 3.7 become (with $a=\{\xi,\eta\}$ and $b=\{\xi,\eta\}$ to represent the ξ and η contravariant components):

$$K_{ba}^{qp}[J^{p,a}](\xi',\eta') = \sum_{k=0}^{N_v^p-1}\sum_{l=0}^{N_u^p-1} A_{ba}^{qp}(\xi',\eta',\xi_l,\eta_k)\,\sqrt{|g^q(\xi',\eta')|}\,w_l w_k\,J^{p,a}(\xi_l,\eta_k), \tag{3.8}$$

with:

$$\begin{aligned} A_{ba}^{qp}(\xi',\eta',\xi_l,\eta_k) = {}& \mathbf{a}^{q,b}(\xi',\eta')\cdot\mathbf{a}_a^p(\xi_l,\eta_k)\,\frac{\partial G_0\big(\mathbf{r}^q(\xi',\eta')-\mathbf{r}^p(\xi_l,\eta_k)\big)}{\partial\hat{\mathbf{n}}^{\,q}(\xi',\eta')} \\ & - \hat{\mathbf{n}}^{\,q}(\xi',\eta')\cdot\mathbf{a}_a^p(\xi_l,\eta_k)\,\mathbf{a}^{q,b}(\xi',\eta')\cdot\nabla G_0\big(\mathbf{r}^q(\xi',\eta')-\mathbf{r}^p(\xi_l,\eta_k)\big), \end{aligned} \tag{3.9}$$

where the variables ξ_l and η_k represent discretization points on a Chebyshev grid, corresponding to the nodes x_i. Specifically, $\xi_l = x_l$ for $l=0,\ldots,N_\xi^p-1$ and $\eta_k = x_k$ for $k=0,\ldots,N_\eta^p-1$. The weights w_l and w_k are the quadrature weights in the ξ and η directions, respectively.

Since $J^{p,a}(\xi,\eta)$ is expressed in a series of Chebyshev polynomials and these polynomials exhibit a discrete orthogonality property on the Chebyshev grid points, the action of the operator K_{ba}^{qp} on each Chebyshev polynomial can be efficiently pre-computed:

$$K_{ba}^{qp}[T_{mn}](\xi',\eta') = \sqrt{|g^q(\xi',\eta')|}\sum_{k=0}^{N_\beta^\eta-1}\sum_{l=0}^{N_\beta^\xi-1} w_l w_k\, A_{ba}^{qp}\big(\xi',\eta',\tilde{\xi}_{\xi'}(s_l),\tilde{\eta}_{\eta'}(t_k)\big) \times \frac{\partial\xi}{\partial s}(s_l)\,\frac{\partial\eta}{\partial t}(t_k)\; T_{mn}\big(\tilde{\xi}_{\xi'}(s_l),\tilde{\eta}_{\eta'}(t_k)\big), \quad (3.10)$$

where $T_{mn}(\xi,\eta) \equiv T_n(\xi)T_m(\eta)$, and where $(\partial\xi)/(\partial s) \to 0$ and $(\partial\eta)/(\partial t) \to 0$ as $\tilde{\xi}_{\xi'}(s) \to \xi'$ and $\tilde{\eta}_{\eta'}(t) \to \eta'$, respectively, canceling the singularity in A up to a degree $(d-1)$. It is important that $N_\beta^{\xi,\eta}$ is chosen sufficiently large to accurately compute each of the precomputation integrals in Eq. 3.10. Finally, on the basis of these precomputations, the action of each of these operators on any $J^{p,a}$ can be readily computed using the Chebyshev expansion of the density, e.g.,

$$K_{ba}^{qp}[J^{p,a}](\xi',\eta') = \sum_{m=0}^{N_\eta^p-1}\sum_{n=0}^{N_\xi^p-1}\gamma_{m,n}^{p,a}\, K_{ba}^{qp}[T_{mn}](\xi',\eta'), \quad (3.11)$$

where $\gamma_{m,n}^{p,a}$ are the Chebyshev expansion coefficients defined as:

$$J^{p,a}(\xi,\eta) = \sum_{m=0}^{N_\eta^p-1}\sum_{n=0}^{N_\xi^p-1}\gamma_{n,m}^{p,a}\, T_n(\xi)\, T_m(\eta), \quad \text{for } a=\xi,\eta. \quad (3.12)$$

The combination of Chebyshev-Based integral equation with $\mathcal{H}$-matrix framework, offers an effective, fast and accurate means of solving complex electromagnetic problems, enabling precise modeling of electrically large and complex PEC structures.[4, 5].

3.3 $\mathcal{H}$-matrix Framework

In this work, we employ the HLib and HLibPro libraries developed at the Max Planck Institute for Mathematics in the Sciences to implement $\mathcal{H}$-matrix functionality [6, 7]. HLibPro extends HLib by providing optimized arithmetic routines and built-in OpenMP parallelization for key operations such as adaptive cross approximation ($\mathcal{ACA}$), blockwise **LU** factorization, and matrix–vector multiplication, thereby improving performance on multicore architectures.

To integrate HLibPro with the high-order solvers, a custom interface layer was developed to map the discretized impedance matrix into hierarchical block-cluster structure of HLibPro. This required defining admissibility criteria based on geometric separation of Bézier patches, generating bounding boxes for each surface element, and constructing consistent index mappings between local element numbering in the LCN assembly and global cluster tree in HLibPro. For high-order elements, additional care was taken to correctly evaluate near-field and far-field separation thresholds to prevent over-compression of strongly coupled basis functions.

A few implementation adjustments were also necessary to achieve robust performance:

- Numerical kernels were wrapped in `C++` interfaces compatible with `TBlockClusterTree` class in HLibPro to ensure efficient data access and alignment.
- Memory layout and tolerance settings were fine-tuned to balance compression accuracy (ϵ_z) and speed.

These implementation refinements enabled stable integration of $\mathcal{H}$-matrix compression and factorization within the proposed framework, supporting scalable simulations with millions of unknowns.

3.4 Solver Strategies: Direct and Iterative Approaches

This framework incorporates both direct and iterative solvers, with the choice of method guided by the specific characteristics of the problem, such as system size, matrix density, and required precision. For small to medium-sized systems, direct solvers are generally preferred due to their robustness and ability to provide exact solutions (up to numerical precision). In contrast, for large-scale systems, iterative solvers are favored due to their scalability and memory efficiency, especially when paired with effective preconditioning.

3.4.1 Direct Solution via $\mathcal{H}$-LU Decomposition

For small and moderately sized systems, we utilize $\mathcal{H}$-**LU** decomposition as the direct solution method. The $\mathcal{H}$-**LU** decomposition provides an efficient factorization of the matrix by taking advantage of the $\mathcal{H}$-matrix structure to represent large dense matrices in a compressed format. This approach allows for significant reductions in both memory usage and computational time compared to a standard dense **LU** decomposition.

The $\mathcal{H}$-**LU** decomposition is particularly advantageous in scenarios where high accuracy is required, as it avoids the convergence issues associated with iterative solvers. Additionally, the decomposition enables efficient forward and backward substitutions, facilitating rapid solution of the linear system once the factorization is complete. Built-in functions HLibPro for $\mathcal{H}$-**LU** decomposition are employed to maximize efficiency, with OpenMP parallelization accelerating certain stages of the factorization process.

3.4.2 Iterative Solution with $\mathcal{H}$-matrix Preconditioners

For large systems, where the memory and computational demands of direct solvers become prohibitive, iterative methods are employed. In this framework, we utilize iterative solvers such as Generalized Minimal Residual (GMRES)[8], Conjugate Gradient (CG)[9], and Bi-Conjugate Gradient Stabilized (BiCGSTAB)[10], which are well-suited for handling large-scale problems with sparse or structured matrices.

$\mathcal{H}$-Matrices as Effective Preconditioners

Direct $\mathcal{H}$-matrix solutions, while efficient, can still present significant computational challenges due to the high costs of storing, factorizing, and manipulating the entire $\mathcal{H}$-matrix for very large problems. To mitigate these costs, $\mathcal{H}$-matrices are used as preconditioners within iterative solvers rather than being solved directly, thereby improving computational efficiency without sacrificing solution quality.

Preconditioning transforms the system of equations into a form that enhances convergence rates in iterative methods. Specifically, the preconditioner $\mathbf{M}$ is constructed as an $\mathcal{H}$-matrix representation with reduced accuracy relative to the main system matrix $\mathbf{A}$. By approximating $\mathbf{A}$ with $\mathbf{M}$, which captures essential problem characteristics at a lower computational cost, the iterative solver achieves faster convergence. This approach effectively balances computational efficiency with solution accuracy, making it possible to address large-scale problems that would be infeasible with traditional direct solvers.

The iterative process is defined by:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{M}\backslash(\mathbf{A}\mathbf{x}_i - \mathbf{b}), \tag{3.13}$$

where $\mathbf{x}$ represents the solution vector, $\mathbf{A}_{\mathcal{H}}$ is the system matrix represented in $\mathcal{H}$-matrix form with a tight compression tolerance ϵ_z, $\mathbf{b}$ is the right-hand side vector, and $\mathbf{M}_{\mathcal{H}}$ is the preconditioner, also stored as an $\mathcal{H}$-matrix but constructed with a looser tolerance $\epsilon_p > \epsilon_z$. The preconditioner $\mathbf{M}_{\mathcal{H}}$ undergoes an $\mathcal{H}$-**LU** factorization, which is computationally less expensive than factoring $\mathbf{A}_{\mathcal{H}}$ due to its relaxed accuracy constraints. This tolerance-based trade-off allows the iterative solver to converge efficiently, often within a few iterations, while maintaining an optimal balance between computational cost and solution accuracy.

The effectiveness of this approach depends on selecting appropriate parameters for $\mathcal{H}$-matrix construction, particularly the compression tolerances used for $\mathbf{A}_{\mathcal{H}}$ and $\mathbf{M}_{\mathcal{H}}$. Fine-tuning parameters such as the maximum admissible block rank, the admissibility parameter $\hat{\eta}$, and the adaptive cross approximation ($\mathcal{ACA}$) tolerance $\tau_{ACA}^{Precond.}$ is crucial for managing the trade-off between accuracy and computational load. Increasing the preconditioner tolerance ϵ_p or the $\mathcal{ACA}$ threshold can accelerate factorization and reduce memory requirements, but at the cost of reduced precision. Hence, empirical tuning of these parameters is necessary to ensure that $\mathbf{M}_{\mathcal{H}}$ provides sufficient accuracy to accelerate convergence without incurring excessive computational overhead.

3.4.3 Heterogeneous $\mathcal{H}$-matrix Tolerances

Within the $\mathcal{H}$-matrix preconditioner, each admissible block can be compressed to a rank determined by an adaptive cross approximation ($\mathcal{ACA}$) tolerance. Assigning distinct tolerances to leaf and non-leaf admissible blocks provides additional flexibility: the non-leaf blocks interactions are kept at higher precision, whereas the leaf blocks, representing local interactions, can be compressed more aggressively without affecting overall accuracy. This heterogeneous tolerance strategy reduces the cost of $\mathcal{H}$-**LU** factorization while preserving

the preconditioner's effectiveness.

A detailed quantitative study of this approach, including parameter sweeps over $\epsilon_{p,\text{leaf}}$ and $\epsilon_{p,\text{nonleaf}}$, is presented in Chapter 4 (see Section 4.4.5 and Fig. 4.12). That analysis, based on a 4λ PEC sphere with $N = 200{,}000$ unknowns, demonstrates that using separate tolerances for leaf and non-leaf blocks reduces preconditioner wall time by roughly 40 % compared with a uniform tolerance, without any measurable loss of accuracy. The heterogeneous tolerance configuration identified in that study is therefore adopted as the default strategy in the proposed large-scale solver.

Summary and Practical Impact

In summary, $\mathcal{H}$-matrices serve as efficient and scalable preconditioners for accelerating Locally Corrected Nyström (LCN)-based integral equation solvers, particularly for large-scale problems where traditional methods are impractical. By organizing matrices hierarchically and implementing multi-level operations, $\mathcal{H}$-matrices significantly reduce computational time and memory requirements. Strategies like heterogeneous tolerance settings further optimize the process, making it feasible to tackle high-dimensional problems with practical computational resources. As a result, $\mathcal{H}$-matrix preconditioning enables efficient and accurate solutions to complex, large-scale problems that would otherwise be intractable.

3.4.4 Selection Criteria for Solver Approach

The decision to use a direct or iterative solver is based on the size and conditioning of the matrix, as well as the computational resources available. For systems that are small to medium in size and require high precision, $\mathcal{H}$-**LU** decomposition is generally preferred due to its reliability and exact solution. However, for larger systems, iterative solvers equipped

with $\mathcal{H}$-matrix preconditioners are selected to leverage the scalability and reduced memory footprint, making them ideal for large-scale simulations where memory and time are limiting factors.

3.4.5 Adaptive *hp*–Refinement Framework (Detailed)

We present an *adaptive hp*–refinement framework for point–based boundary integral solvers (LCN) on quadrilateral Bézier–patch meshes that restores the target $\mathcal{O}(h^p)$ accuracy on sharp–edged PEC scatterers [11]. The surface is partitioned into three zones: (i) an *edge* zone with fine h and low p to resolve singular currents; (ii) a one–two–layer *transition* band with intermediate (h, p); and (iii) a *smooth* zone with coarse h and high p. The tapered assignment

$$h_{\text{edge}} \ll h_{\text{transition}} \ll h_{\text{smooth}}, \qquad p_{\text{edge}} < p_{\text{transition}} < p_{\text{smooth}},$$

where

$$p_{\text{transition}} = \text{round}\left(\tfrac{p_{\text{edge}}+p_{\text{smooth}}}{2}\right)$$

mitigates edge–induced oscillations and conditioning issues while preserving efficiency. Accuracy controls are fixed before assembly and chosen per patch so that

$$q_K \;\geq\; p_K + 1 \quad \text{(patch quadrature)}, \qquad q_i \;\geq\; p_i \quad \text{(local corrections)},$$

ensuring stable weak/principal–value kernel evaluation. A per–observation-point near radius $r_i = r_i(h_i, p_i, kh_i)$ replaces a single global threshold, localising singular/quasi-singular treatment, preventing near-set overload, and improving near-assembly balance for $\mathcal{H}$-compression.

See also Fig. 5.1 and §5.3 and §5.4 for the full methodology and initialization workflow.

3.5 Parallel Computing Strategy

Large–scale, high-order simulations demand more memory and arithmetic throughput than any single core, or even a single node can provide. Accordingly, the framework exploits parallelism at two distinct tiers:

- **Distributed memory (MPI).** Coarse–grain task and data distribution across nodes [12, 13, 14].
- **Shared memory (OpenMP).** Fine–grain loop parallelism inside each node [14, 15, 16].

A third tier, *StarPU*, is slated for future work so that the same task graph can be scheduled over CPUs and accelerators without code changes [17, 18].

3.5.1 Distributed Memory: MPI for Near Interactions and RHS Assembly

The costliest stage in a high-order LCN / CBIE solve is the evaluation of near–interaction integrals and the right-hand side (RHS) excitation vector, both of which are localised and therefore embarrassingly parallel across elements or quadrature points. We partition the global list of near interactions (and corresponding RHS collocation points) into non-overlapping blocks and scatter those blocks to MPI ranks. Each rank:

1. pulls the geometry and basis data for its block from a replicated read-only buffer;

2. evaluates all required kernel integrals, including adaptive quadrature where necessary;

3. accumulates its portion of the dense near-field matrix and RHS;

4. contributes to a global `MPI_Reduce` that assembles the full matrix block on rank 0 (or on a designated master in block-cyclic fashion for out-of-core runs).

MPI collective operations are limited to the final reduction, keeping communication volume low and overlapping well with computation. Load balancing is handled by a simple greedy scheduler that assigns a roughly equal number of source–observation patch pairs to every rank; this proves sufficient because the cost per pair is strongly correlated with the pair count.

3.5.2 $\mathcal{H}-$matrix Phase: OpenMP Only

All $\mathcal{H}$-matrix tasks including $\mathcal{ACA}$ block compression, block-cluster traversals, and the hierarchical **LU** factorization are carried out by HLibPro[7]. The library is strictly *thread-parallel*: it uses OpenMP (up to four cores) and is not MPI-aware. Although the factorization scales as $\mathcal{O}\big(r^2 N \log^2 N\big)$ and, can become the dominant cost at extreme problem sizes, the built-in four-thread OpenMP support still delivers a useful speed-up and, for the problem sizes considered here, prevents the $\mathcal{H}$-**LU** step from becoming a bottleneck.

3.5.3 Outlook: Task-Based Runtime

To remove the explicit split between MPI and OpenMP, the task graph will be ported to the StarPU runtime in future work[17, 18]. StarPU can schedule tasks across heterogeneous resources (CPUs, GPUs, NUMA domains) and automatically choose between data replication

or distributed data-flow, providing a single-source solution that adapts to emerging hardware while preserving the high-order, error-controllable nature of the solver.

3.6 Summary

Chapter 3 presented the complete architecture of the proposed high-order solver for electrically large PEC scattering. The method rests on three pillars—*adaptive hp* discretization, hierarchical low-rank algebra, and hybrid (MPI/OpenMP) parallelism—with the following defining features:

- **Adaptive *hp* LCN discretization** — fine, low-order edge patches; one–two layers of *transition* patches with intermediate (h, p); and coarse, high-order smooth patches. This tapered design controls edge singularities while preserving target $\mathcal{O}(h^p)$ accuracy and efficiency. In the *hp*-refinement framework, single global threshold is replaced with a per–observation-point near radius r_i to balance the near interactions.

- **Accuracy-preservation rules for adaptive *hp*-refinement** — practical conditions on $(h_{\text{edge}}, h_{\text{transition}}, h_{\text{smooth}})$ and $(p_{\text{edge}}, p_{\text{transition}}, p_{\text{smooth}})$ plus quadrature/correction choices $(q \geq p{+}1)$ that maintain the smooth-region rate $\mathcal{O}(h^{p_{\text{smooth}}})$; see the error-balancing hypothesis in §5.3.

- **Hybrid direct/iterative $\mathcal{H}$-matrix solver** — a rank-adaptive $\mathcal{H}$-**LU** factorization acts as (i) a standalone direct solver for mid-sized problems and (ii) a powerful preconditioner that limits GMRES iterations in million-unknown cases.

- **Heterogeneous $\mathcal{ACA}$ tolerances** — distinct thresholds/rank controls for leaf and non-leaf blocks shorten factorization time (empirically $\sim$40%), as confirmed on the

heterogeneous tolerance tests reproduced in Chapter 4.

- **Parallel execution** — MPI distributes RHS and near-field assembly across nodes; the HLibPro back-end supplies OpenMP concurrency for $\mathcal{H}$-kernels, giving a practical hybrid that scales to millions of unknowns on commodity clusters.

Collectively, these ingredients realise fast, high-order, error-controllable algorithms whose performance claims are tied to explicit accuracy metrics. The framework introduced here forms the computational backbone for the RCS benchmarks and applications in the following chapters.

Bibliography

[1] S. D. Gedney, "The combined field integral equation (CFIE) for scattering problems," *IEEE Trans. Antennas Propag.*, vol. 51, no. 7, pp. 1710–1717, 2003.

[2] G. Xie, Y. Zhong, H. Li, W. Du, S. Feng, W. He, L. Wang, Y. Dong, and J. Cheng, "Near singularity cancellation in weakly singular integrals of three-dimensional boundary element method," *Eng. Anal. Bound. Elem.*, vol. 118, pp. 54–59, 2020.

[3] I. Jeffrey, J. Aronsson, M. Shafieipour, and V. Okhmatovski, "Error controllable solutions of large scale problems in electromagnetics: MLFMA accelerated locally corrected Nyström solutions of CFIE in 3D," *IEEE Antennas Propag. Mag.*, vol. 55, no. 3, pp. 294–308, Jun. 2013.

[4] J. Hu, E. Garza, and C. Sideris, "A Chebyshev-based high-order-accurate integral equation solver for Maxwell's equations," *IEEE Trans. Antennas Propag.*, vol. 67, no. 7, pp. 4313–4325, 2019.

[5] J. Choi, D. D. H. Yang, and Y. Kim, "A high-order integral equation solver for electromagnetic scattering using a hybrid approach," *IEEE Trans. Antennas Propag.*, vol. 68, no. 10, pp. 7950–7960, 2020.

[6] S. Börm and L. Grasedyck, *HLib: A program library for hierarchical and $\mathcal{H}^2$-matrices*. (Developed in part at MPI-MIS). [Online]. Available: `https://www.hlib.org/`. Accessed: Nov. 6, 2025.

[7] R. Kriemann, *HLIBpro: A software library for hierarchical matrices*. Max Planck Institute for Mathematics in the Sciences (MPI-MIS). [Online]. Available: `https://www.hlibpro.com/`. Accessed: Nov. 6, 2025.

[8] Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput.*, vol. 7, no. 3, pp. 856–869, 1986. doi:10.1137/0907058.

[9] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *J. Res. Nat. Bur. Stand.*, vol. 49, no. 6, pp. 409–436, 1952. doi:10.6028/jres.049.044.

[10] H. A. van der Vorst, "Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput.*, vol. 13, no. 2, pp. 631–644, 1992. doi:10.1137/0913035.

[11] O. Babazadeh *et al.*, "Optimal preconditioners for hybrid direct–iterative H-matrix solvers in boundary element methods," *IEEE J. Multiscale Multiphys. Comput. Techn.*, to appear, 2025. doi:10.1109/JMMCT.2025.3547827.

[12] Message Passing Interface Forum, "MPI: A message-passing interface standard, Version 4.1," 2023. [Online]. Available: `https://www.mpi-forum.org/docs/`.

[13] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, 2nd ed. Cambridge, MA, USA: MIT Press, 1999.

[14] R. Rabenseifner, G. Hager, and G. Jost, "Hybrid MPI/OpenMP parallel programming on clusters of multi-core SMP nodes," in *Proc. 17th Euromicro Int. Conf. Parallel, Distributed and Network-Based Processing (PDP)*, 2009, pp. 427–436.

[15] OpenMP Architecture Review Board, "OpenMP application programming interface, Version 5.2," Nov. 2021. [Online]. Available: `https://www.openmp.org/specifications/`.

[16] L. Dagum and R. Menon, "OpenMP: An industry-standard API for shared-memory programming," *IEEE Comput. Sci. Eng.*, vol. 5, no. 1, pp. 46–55, 1998.

[17] C. Augonnet, S. Thibault, R. Namyst, and P.-A. Wacrenier, "StarPU: A unified platform for task scheduling on heterogeneous multicore architectures," *Concurrency Comput. Pract. Exp.*, vol. 23, no. 2, pp. 187–198, 2011.

[18] C. Augonnet, S. Thibault, R. Namyst, and P.-A. Wacrenier, "Scheduling tasks over multicore machines with GPU accelerators using the StarPU runtime system," in *Euro-Par 2009*, LNCS 5704, 2009, pp. 863–874.

Chapter 4

Optimal Preconditioners for Hybrid Direct-Iterative $\mathcal{H}$-matrix Solvers in Boundary Element Methods

In this chapter, we address several of the key challenges in computational electromagnetics as outlined in the earlier research questions. Specifically, we integrate the $\mathcal{H}$-matrix framework with the Locally Corrected Nyström (LCN) discretization method, aiming to explore the advantages of high-order versus low-order methods (Research Question 1). By applying high-order techniques, such as LCN, and comparing them to traditional low-order approaches, we evaluate their suitability for complex geometries while considering computational efficiency and accuracy. This exploration provides insights into the benefits of advanced techniques for precision and reduced computational costs in large-scale electromagnetic problems.

Further, to tackle efficient use of $\mathcal{H}$-matrix acceleration (Research Question 2), this study develops and evaluates the $\mathcal{H}$-matrix as a preconditioner within a hybrid direct-iterative

solver. We explore the practical challenges of combining high-order LCN discretization with $\mathcal{H}$-matrix compression, assessing the impact on memory efficiency and computational demands, particularly for PEC objects with varying complexities and electrical sizes.

Guided by Rokhlin's principles—robustness, efficiency, and scalability (Research Question 3), we analyze the balance between direct and iterative solvers within the hybrid approach. By leveraging $\mathcal{H}$-matrix preconditioning, we aim to combine the robustness of direct methods with the efficiency of iterative solvers, exploring conditions under which the hybrid approach optimizes performance across diverse problem scales.

Finally, we incorporate parallel computing strategies using MPI and OpenMP to address scalability and efficient parallelism. These strategies are employed to optimize wall-time performance and enhance the efficiency of the solver, ensuring that the proposed hybrid solver framework is scalable and adaptable across different problem complexities.

The results of this research have been compiled into a paper, which has been published in the *IEEE Journal of Multiscale and Multiphysics Computational Techniques (J-MMCT)*. The remainder of this chapter presents the paper.

Abstract

The paper proposes a new approach to the fast solution of matrix equations resulting from boundary element discretization of integral equations. By hybridizing fast iterative $\mathcal{H}$-matrix solvers with a fast direct $\mathcal{H}$-matrix preconditioner factorization, we create a framework that can be tuned between the extremes of a direct solver and a unpreconditioned iterative solver. This tuning is largely achieved using a single numerical parameter representing the preconditioner tolerance. A more complicated scheme involving two different tolerances is also briefly considered. The proposed framework is demonstrated on a high-order accurate Locally Corrected Nyström solution of surface integral equations for PEC targets. Examples consider various scattering problems including those featuring strong physical resonances. We show that appropriately choosing the preconditioner tolerance achieves the prescribed solution accuracy with minimal CPU time. Expanding from one to two tolerance parameters further enhances the framework by providing the flexibility to dynamically adjust tolerance, enabling higher compression while maintaining accuracy and fast convergence. This adaptive strategy offers significant potential for optimizing the balance between memory usage and CPU time in the future.

4.1 Introduction

Electromagnetic scattering problems are fundamental in various fields, including antenna design, radar cross-section analysis, and wireless communication [1, 2, 3]. The Combined Field Integral Equation (CFIE) is a widely used formulation to solve such problems due to its ability to eliminate spurious resonances [4, 5]. However, the dense matrix equations resulting from CFIE discretization pose significant computational challenges for large-scale scattering problems on realistic targets, especially for simulations involving physical resonances, over-sampling, low-frequencies or other factors that degrade the numerical conditioning of the pertinent matrix equation [6].

Traditional direct matrix solvers, while robust and accurate, are computationally prohibitive due to their $\mathcal{O}(N^3)$ time complexity and $\mathcal{O}(N^2)$ memory usage, where N is the number of unknowns [7]. Hierarchical matrix ($\mathcal{H}$-matrix) techniques have emerged as a leading approach to achieving reduced computational complexity by leveraging data-sparse representations of dense matrices [8]. As $\mathcal{H}$-matrices provide a mechanism for the efficient direct (non-iterative) solution of scattering problems, they have the potential to maintain robustness in poorly conditioned matrix scenarios that would otherwise be challenging for iterative solvers applied to large-scale realistic targets [9]. Direct $\mathcal{H}$-matrix solutions such as those based on an $\mathcal{H}$-**LU** factorization of the pertinent $\mathcal{H}$-matrices, results in a computational bottleneck [10] and ultimately limits the size of problems and the speed at which they can be solved.

In this context, we propose to hybridize direct and iterative $\mathcal{H}$-matrix solvers. In our approach, a direct $\mathcal{H}$-**LU** factorization is applied to an approximate system matrix that serves as a preconditioner in an iterative scheme. The numerical tolerance used to create and factor the preconditioner provides a balance between the amount of work/time required

to create and factor the preconditioner and the amount of work/time resulting from the number of iterations required to compute the solution. Control of the overall solution accuracy is maintained through accurate construction of the BEM's impedance $\mathcal{H}$-matrix, and iterative efficiency is obtained by performing inexpensive $\mathcal{H}$-matrix-vector products during the iterative process [11, 12, 13].

The computational control enabled by this framework begs the question: what preconditioner tolerances should be used to get the best performance (measured in time-to-solution or memory consumption) out of the hybrid scheme? In an attempt to shed light on the trade-off between preconditioner time and iteration time, herein we study, for a variety of examples, the computational expense of the proposed hybrid framework as a function of the preconditioner parameters. We demonstrate a range of preconditioner parameters that can be used to minimize computational time for various problems, and how the existence of resonances shifts these parameters as the balance between preconditioner factorization time and iteration time is affected. With appropriately chosen parameters, the iterative solution approach becomes largely insensitive to the condition number of the matrix, enabling a solution framework with robustness comparable to a direct solution while reducing the overall time-to-solution.

This paper extends our previous work presented at the 2024 IEEE MTT-S International Microwave Symposium (IMS) [14]. Therein, an error-controllable, high-order, fast direct $\mathcal{H}$-matrix solver was used to accelerate the locally corrected Nyström (LCN) solution of the CFIE. In this extension of that work, we apply the proposed $\mathcal{H}$-matrix preconditioner scheme to the LCN CFIE formulation and study its behaviour.

The paper is structured as follows: Section §4.2 summarizes the $\mathcal{H}$-matrix formulation of the LCN solution to the CFIE. Section §4.3 formulates the preconditioned iterative so-

lution to the resulting system of equations. Numerical results demonstrating the relationship between preconditioner parameters and time-to-solution are presented in Section §4.4. Conclusions, and recommendations for making efficient use of the proposed framework are provided in Section §4.5.

4.2 The $\mathcal{H}$-matrix Accelerated LCN Solution of CFIE

This section outlines discretization of CFIE with LCN method and acceleration of the resulting matrix equation solution with the framework of hierarchical matrices.

4.2.1 CFIE formulation of the scattering problem

The time-harmonic combined field integral equation (CFIE) relating the (unknown) electric surface current density $\mathbf{J}$ on a PEC surface $S-$ (as a limit from inside PEC volume) to the incident electric field $\mathbf{E}^{\text{inc}}$ and magnetic $\mathbf{H}^{\text{inc}}$ in free space is [15, 16]:

$$\begin{aligned}\hat{\mathbf{t}}(\mathbf{r}) \cdot \Big[\alpha\, \mathbf{E}^{\text{inc}}(\mathbf{r}) + (1-\alpha)\, \eta_0\, \hat{\mathbf{n}}(\mathbf{r}) \times \mathbf{H}^{\text{inc}}(\mathbf{r})\Big] \;=\; \\ -\; \hat{\mathbf{t}}(\mathbf{r}) \cdot \int\limits_S \Big[\alpha\, \bar{\mathbf{G}}_{EJ}(\mathbf{r},\mathbf{r}') + (1-\alpha)\, \eta_0\, \hat{\mathbf{n}} \times \bar{\mathbf{G}}_{HJ}(\mathbf{r},\mathbf{r}')\Big] \cdot \mathbf{J}(\mathbf{r}')\, \mathrm{d}S', \\ \mathbf{r} \in S-,\end{aligned} \tag{4.1}$$

where ω is the radial frequency of the time-harmonic fields, $\hat{\mathbf{t}}$ is a unit-vector tangential to S, $\hat{\mathbf{n}}$ is the unit-normal to S, $\bar{\mathbf{G}}_{HJ}(\mathbf{r},\mathbf{r}') = \nabla G_0(\mathbf{r},\mathbf{r}') \times \bar{\mathbf{I}}$ is the free-space magnetic field dyadic Green's function, and $\bar{\mathbf{G}}_{EJ}(\mathbf{r},\mathbf{r}') = -j\omega\mu_0[\nabla\nabla G_0(\mathbf{r},\mathbf{r}')/k_0^2 + \bar{\mathbf{I}}]$ is the free-space electric field dyadic Green's function, where $\bar{\mathbf{I}}$ is the identity dyadic (a.k.a. idem-factor). These Green's functions rely on the free space scalar Green's function $G_0(\mathbf{r},\mathbf{r}') = e^{-ik_0|\mathbf{r}-\mathbf{r}'|}/(4\pi|\mathbf{r}-\mathbf{r}'|)$

where observation and source locations are respectively denoted by $\mathbf{r}$ and $\mathbf{r}'$. The physical parameters μ_0, ϵ_0, $\eta_0 = \sqrt{\mu_0/\epsilon_0}$ and $k_0 = \omega\sqrt{\epsilon_0\mu_0}$ represent the permeability, permittivity, intrinsic impedance and wavenumber of free space. The parameter α is used to balance the contributions of the electric field integral equation (EFIE) and magnetic field integral equation (MFIE) to the CFIE [16].

4.2.2 Locally Corrected Nyström Discretization of the CFIE and $\mathcal{H}$-matrix Acceleration

After defining the model geometry with NURBS (as shown in Fig. 4.1), the geometric representation and discretization of the CFIE involves its partitioning into a sum of integrals over N_e Bézier elements. Each integral is assessed using a quadrature rule, with quadrature points $\mathbf{r}_{q_m}$ and weights w_{q_m} for each element m. The current expansion [16]

$$\mathbf{J}(\mathbf{r}) = J^1(\mathbf{r})\frac{\mathbf{a}_1(\mathbf{r})}{\mathcal{J}(\mathbf{r})} + J^2(\mathbf{r})\frac{\mathbf{a}_2(\mathbf{r})}{\mathcal{J}(\mathbf{r})}, \tag{4.2}$$

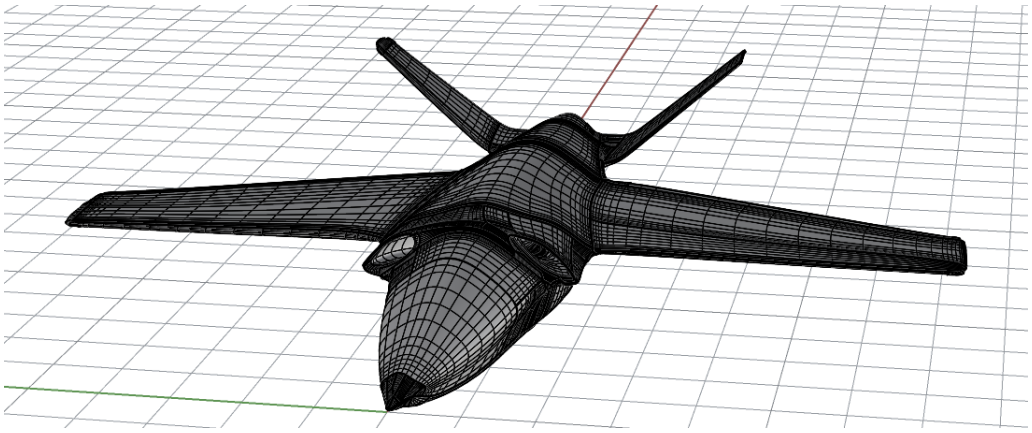


Figure 4.1: NURBS representation of an aircraft model from Austin Benchmarks [17].

is substituted into (4.1) and tested at each quadrature point:

$$\begin{aligned}\mathbf{a}_i^{q_m} \cdot \left[\alpha\, \mathbf{E}^{\text{inc}}(\mathbf{r}_{q_m}) + (1-\alpha)\, \eta_0\, \hat{\mathbf{n}}^{q_m} \times \mathbf{H}^{\text{inc}}(\mathbf{r}_{q_m})\right] \\ = \sum_{e=1}^{N_e} \sum_{q=1}^{Q} \left[w_{q_m,q_e}^{i1}\, J^1(\mathbf{r}_{q_e}) + w_{q_m,q_e}^{i2}\, J^2(\mathbf{r}_{q_e})\right],\end{aligned} \tag{4.3}$$

where

$$\begin{aligned}w_{q_m,q_e}^{ij} = -\, w_{q_e}\, \mathbf{a}_i^{q_m} \cdot \Big[\alpha\, \bar{\mathbf{G}}_{EJ}(\mathbf{r}_{q_m}, \mathbf{r}_{q_e}) \\ +(1-\alpha)\, \eta_0\, \hat{\mathbf{n}}^{q_m} \times \bar{\mathbf{G}}_{HJ}(\mathbf{r}_{q_m}, \mathbf{r}_{q_e})\Big] \cdot \mathbf{a}_j^{q_e}.\end{aligned} \tag{4.4}$$

To handle the singular behavior of $G_0(\mathbf{r}_{q_m}, \mathbf{r}_{q_e})$, local corrections are applied which locally correct (modify) the integration weights. Details can be found in [16]. The result of this discretization process is a system of linear algebraic equations (SLAE):

$$\underbrace{\begin{bmatrix}\mathbf{Z}_{1,1} & \mathbf{Z}_{1,2} \\ \mathbf{Z}_{2,1} & \mathbf{Z}_{2,2}\end{bmatrix}}_{\mathbf{Z}} \cdot \underbrace{\begin{bmatrix}\mathbf{J}_1 \\ \mathbf{J}_2\end{bmatrix}}_{\mathbf{J}} = \underbrace{\begin{bmatrix}\mathbf{V}_1 \\ \mathbf{V}_2\end{bmatrix}}_{\mathbf{V}}, \tag{4.5}$$

where $\mathbf{Z}$ is the BEM impedance matrix, $\mathbf{J}_1$ and $\mathbf{J}_2$ are the surface current expansions using $\mathbf{a}_1$ and $\mathbf{a}_2$ unitary vectors [16]. This dense $2N_eQ \times 2N_eQ$ system of equations must be solved to determine the surface current coefficients.

4.2.3 $\mathcal{H}$-matrix Acceleration

Naïve direct factorization of the LCN impedance matrix is computationally prohibitive for scattering problems of interest. $\mathcal{H}$-matrices provide an alternative formulation that alleviates the computational burden by hierarchically compressing regions of the impedance matrix [11, 8]. In our implementation, the $\mathcal{H}$-matrix is constructed by organizing source and observation

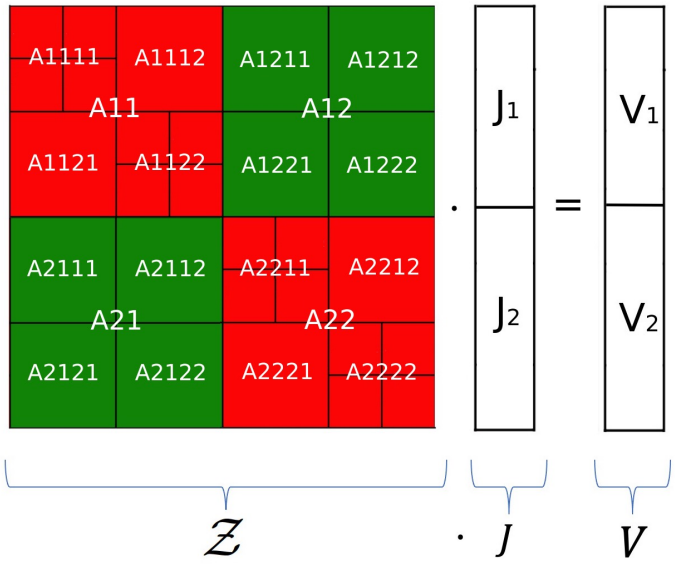


Figure 4.2: SLAE formulated using an $\mathcal{H}$-matrix approximation.

basis functions into a hierarchical cluster tree via Orthogonal Recursive Bisection (ORB) [44].

Interactions between observation and source points are categorized efficiently by means of a block cluster tree [11, 8]. If a source node is sufficiently far from an observation node, the pair is labeled "admissible" and compressed using low-rank approximations; otherwise, it is stored as a dense matrix. Compression is achieved by applying adaptive cross approximation ($\mathcal{ACA}$) to compressible blocks, with error controlled by an error-level parameter ϵ such that smaller ϵ means higher accuracy. The $\mathcal{H}$-matrix representation of the LCN impedance matrix is illustrated in Fig. 4.2.

This compression scheme is supported by a full mathematical theory of $\mathcal{H}$-matrix arithmetic and linear algebra [11, 8] that enables accelerated matrix-vector-products, matrix factorizations, and solutions, all while mitigating memory and CPU requirements, making it suitable for solving complex problems efficiently. The interested reader can find additional details in [11, 8]. We denote the $\mathcal{H}$-matrix approximation to $\mathbf{Z}$ by $\mathcal{Z}$, such that the

hierarchical matrix representation of (4.5) is:

$$\begin{bmatrix} \mathcal{Z}_{1,1} & \mathcal{Z}_{1,2} \\ \mathcal{Z}_{2,1} & \mathcal{Z}_{2,2} \end{bmatrix} \begin{bmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix}. \tag{4.6}$$

This approximation is built using a tolerance ϵ_z.

The compressed system (4.6) can be solved efficiently using $\mathcal{H}$-matrix solvers such as the $\mathcal{H}$-**LU** approach adopted to integral equations [10]. However, the same robustness can be maintained for a fraction of the cost if we adopt the proposed hybrid direct/iterative preconditioned solution to the system of equations.

4.3 Hybrid Direct/Iterative Framework

We now focus on solving a preconditioned version of (4.6). The preconditioner is a $\mathcal{H}$-matrix approximation to Z, denote by $\mathcal{P}$, constructed with a generally larger tolerance ϵ_p:

$$\begin{bmatrix} \mathcal{P}_{1,1} & \mathcal{P}_{1,2} \\ \mathcal{P}_{2,1} & \mathcal{P}_{2,2} \end{bmatrix}^{-1} \underbrace{\begin{bmatrix} \mathcal{Z}_{1,1} & \mathcal{Z}_{1,2} \\ \mathcal{Z}_{2,1} & \mathcal{Z}_{2,2} \end{bmatrix} \begin{bmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \end{bmatrix}}_{\mathbf{V}} = \underbrace{\begin{bmatrix} \mathcal{P}_{1,1} & \mathcal{P}_{1,2} \\ \mathcal{P}_{2,1} & \mathcal{P}_{2,2} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix}}_{\mathcal{J}}. \tag{4.7}$$

Fig. 4.3 depicts the $\mathcal{H}$-matrix representation of (4.7). Note that the $\mathcal{H}$-**LU** factorization applies to the preconditioner $\mathcal{P}$ and is thus less expensive than factoring $\mathcal{Z}$ when $\epsilon_p > \epsilon_z$.

4.3.1 Iteratively Solving the Preconditioned System

Iterative methods are applied to efficiently solve the preconditioned linear system, converging to an accurate solution with reduced computational overhead. When $\epsilon_p = \epsilon_z$, the precondi-

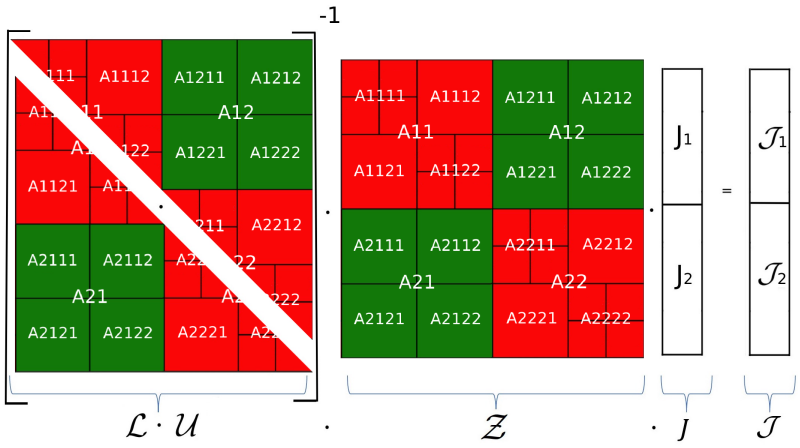


Figure 4.3: Preconditioned SLAE in $\mathcal{H}$-matrix format.

tioner $\mathcal{P}$ equals the system impedance matrix $\mathcal{Z}$, resulting in formulation that should achieve the same accuracy as a direct solution to the system and converge in a single iteration. When $\epsilon_p > \epsilon_z$, less effort is required to factor the preconditioner, but more iterations are required to solve the problem. The trade-off between computations required to factor the preconditioner and computations required to compute iterations is the focus of our numerical study.

The preconditioner, represented as an $\mathcal{H}$-matrix, is a large, dense (but compressed) linear system that can be efficiently factored using an $\mathcal{H}$-**LU** decomposition into the product of a lower triangular $\mathcal{H}$-matrix $\mathcal{L}$ and an upper triangular $\mathcal{H}$-matrix $\mathcal{U}$ [19, 20]:

$$\mathcal{P} = \begin{bmatrix} \mathcal{P}_{1,1} & \mathcal{P}_{1,2} \\ \mathcal{P}_{2,1} & \mathcal{P}_{2,2} \end{bmatrix} = \begin{bmatrix} \mathcal{L}_{1,1} & 0 \\ \mathcal{L}_{2,1} & \mathcal{L}_{2,2} \end{bmatrix} \begin{bmatrix} \mathcal{U}_{1,1} & \mathcal{U}_{1,2} \\ 0 & \mathcal{U}_{2,2} \end{bmatrix}. \tag{4.8}$$

The decomposition is performed recursively, leveraging $\mathcal{H}$-matrix arithmetic [11].

In this work, we use the GMRES (Generalized Minimal Residual) iterative solution method[21], due to its applicability to the general matrix equations. In this context, the convergence rate will depend on the accuracy of the preconditioner, thus the number of it-

erations should decrease as ϵ_p decreases sufficiently. Care is taken to account for the fact that we do not compute or store the inverse of the preconditioner. As iterative methods applied without a preconditioner require matrix-vector-products $\mathcal{Z}\mathbf{J}^{(i)}$ at iteration i, preconditioned systems require matrix-vector-products $\mathcal{P}^{-1}(\mathcal{Z}\mathbf{J}^{(i)})$. In the case of GMRES with preconditioning, we solve the preconditioned system:

$$\mathcal{P}^{-1}\mathcal{A}\mathbf{J} = \mathcal{P}^{-1}\mathbf{V}, \tag{4.9}$$

where $\mathcal{P}$ is the preconditioner. Instead of explicitly computing $\mathcal{P}^{-1}$, we apply the preconditioner indirectly by solving a system of the form:

$$\mathcal{P}\mathbf{X} = \mathbf{Y}. \tag{4.10}$$

to compute $\mathbf{X} = \mathcal{P}^{-1}\mathbf{Y}$. This approach incorporates the effect of the preconditioner without directly inverting $\mathcal{P}$.

In GMRES, the solution $\mathbf{J}$ is updated iteratively by minimizing the residual over the Krylov subspace generated by $\mathcal{P}^{-1}\mathcal{A}$. Each iteration involves constructing and solving a least-squares problem to find the optimal coefficients for the basis vectors of the Krylov subspace.

As illustrated in Fig. 4.4, we let $\mathcal{U}\mathbf{X} = \mathbf{D}$ and apply forward substitution in the lower triangular hierarchical matrix equation $\mathcal{L}\mathbf{D} = \mathbf{Y}$

$$\begin{bmatrix} \mathcal{L}_{1,1} & 0 \\ \mathcal{L}_{2,1} & \mathcal{L}_{2,2} \end{bmatrix} \begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \end{bmatrix}, \tag{4.11}$$

is solved for $\mathbf{D}$. Next, back substitution is applied to

$$\begin{bmatrix} \mathcal{U}_{1,1} & \mathcal{U}_{1,2} \\ 0 & \mathcal{U}_{2,2} \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \end{bmatrix}, \tag{4.12}$$

to obtain $\mathbf{X}$ satisfying $\mathbf{X} = \mathcal{P}^{-1}\mathbf{Y}$.

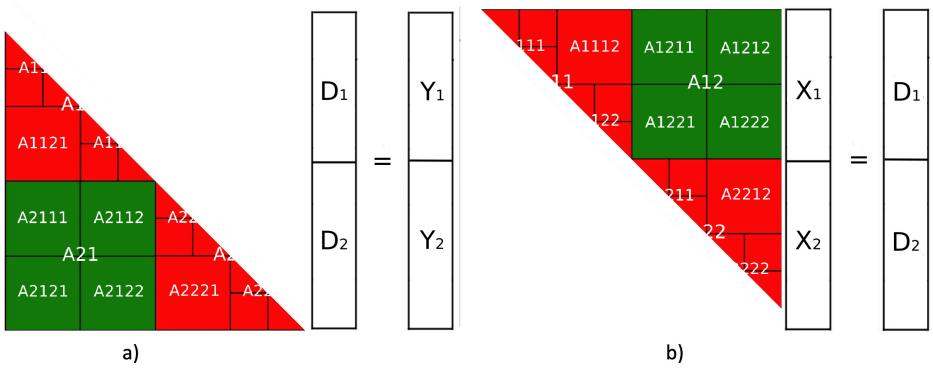


Figure 4.4: a) Forward substitution applies to the lower triangular $\mathcal{H}$-matrix equation $\mathcal{L}D = Y$, b) backward substitution applies to the upper triangular $\mathcal{H}$-matrix equation $\mathcal{U}X = D$.

4.3.2 Tuning Time-to-Solution with the $\mathcal{H}$-matrix Preconditioner Tolerance

The proposed $\mathcal{H}$-matrix preconditioned framework adopts a direct factorization of the preconditioner and an iterative solver. Thus, the total computational time consists of the time to build and factor the preconditioner and the time to iteratively update the solution. The demands of these computations can be analyzed to expose the role of the preconditioner tolerance ϵ_p in the total computational time. In what follows we adopt the bounds in d-dimensional space from [11, 12], but note that the analysis used to produce these bounds leaves them potentially loose, indicating higher requirements than necessary, especially as

we are solving problems on $(d-1)$-dimensional surfaces within $d=3$ dimensional space.

Given a tolerance ϵ, an $\mathcal{H}$-matrix approximation $\mathcal{A}$ to a matrix $\mathbf{A}$ that satisfies $\|\mathbf{A} - \mathcal{A}\|_F < \epsilon\|\mathbf{A}\|_F$ has a build time:

$$t_{build} \in O((\log(1/\epsilon))^{2d} N \log N). \tag{4.13}$$

The $\mathcal{H}$-matrix-vector product, illustrated in Fig. 4.5, leverages the compression scheme to enable a time requirement

$$t_{mvp} \in O((\log(1/\epsilon))^{d} N \log N). \tag{4.14}$$

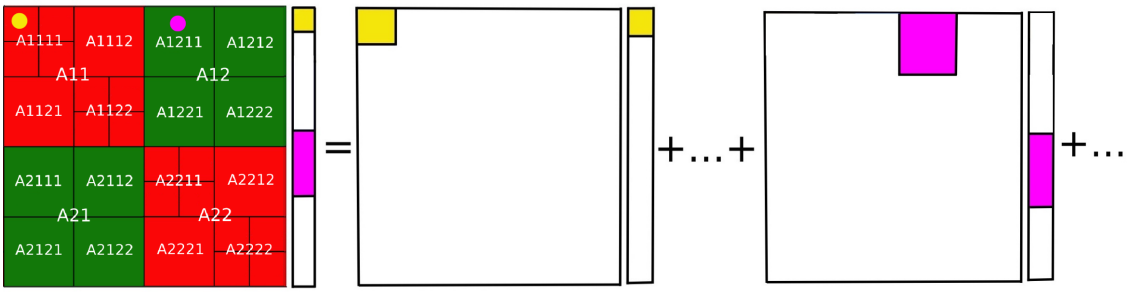


Figure 4.5: Matrix-Vector Product in $\mathcal{H}$-matrix format.

The $\mathcal{H}$-**LU** decomposition, illustrated in Fig. 4.4 also leverages the compression scheme by maintaining the hierarchical structure of the $\mathcal{H}$-matrix while further leveraging low-rank approximations. The required time is

$$t_{lu} \in O((\log(1/\epsilon))^{2d} N \log^2 N). \tag{4.15}$$

Similarly, the time required for forward or backward substitution is

$$t_{sub} \in O((\log(1/\epsilon))^{d} N \log N). \tag{4.16}$$

while the time to evaluate m iterations of a standard iterative method with an $\mathcal{H}$-**LU** preconditioner is

$$t_{iter} \in O(mt_{mvp}) + O(mt_{sub}). \tag{4.17}$$

The number of iterations depends on the effectiveness of the preconditioner, i.e., it depends on ϵ_p. We assume a model

$$m \approx \gamma \cdot \left(\log\left(\frac{1}{\epsilon_p}\right)\right)^{-\beta}. \tag{4.18}$$

In our proposed scheme, we are building two $\mathcal{H}$-matrices $\mathcal{Z}$ and $\mathcal{P}$ with respective tolerances ϵ_z and ϵ_p. We factor and perform substitution with $\mathcal{P}$. We perform matrix vector products with $\mathcal{Z}$. Using the complexities as bounds for the wall time, we can write the following expression for the overall time:

$$\begin{aligned} t_{total} \approx\ & \overbrace{C_1(\log(1/\epsilon_z))^{2d} N \log N}^{\text{Build } \mathcal{Z}} \\ & + \overbrace{C_2(\log(1/\epsilon_p))^{2d} N \log N}^{\text{Build } \mathcal{P}} \\ & + \overbrace{C_3(\log(1/\epsilon_p))^{2d} N \log^2 N}^{\mathcal{H}-\mathbf{LU} \text{ Factor } \mathcal{P}} \\ & + m \times \overbrace{C_4(\log(1/\epsilon_p))^{d} N \log N}^{\text{Fwd/Back Sub per iteration}} \\ & + m \times \overbrace{C_5(\log(1/\epsilon_z))^{d} N \log N}^{\text{MVP per iteration}}, \end{aligned} \tag{4.19}$$

where C_i are constants. We note that i) we cannot avoid building $\mathcal{Z}$, and ii) the **LU** decomposition time dominates the time to build the preconditioner. Thus we can drop the times required to build the systems from the time budget. Applying (4.18) then leads to a

'solve' time as a function of the preconditioner tolerance ϵ_p:

$$
\begin{aligned}
t_{solve}(\epsilon_p) \approx \; & \overbrace{\tilde{C}_1(\log(1/\epsilon_p))^{2d} N \log^2 N}^{\mathcal{H}-\mathbf{LU}\text{ Factor }\mathcal{P}} \\
& + \overbrace{\left(\log\left(\frac{1}{\epsilon_p}\right)\right)^{-\beta}}^{\propto \#\text{ iterations } m} \times \overbrace{\tilde{C}_2(\log(1/\epsilon_p))^{d} N \log N}^{\text{Fwd/Back Sub per iteration}} \\
& + \left(\log\left(\frac{1}{\epsilon_p}\right)\right)^{-\beta} \times \overbrace{\tilde{C}_3(\log(1/\epsilon_z))^{d} N \log N}^{\text{MVP per iteration}},
\end{aligned}
\tag{4.20}
$$

where $\tilde{C}_i$ are constants.

Equation (4.20) clarifies the tradeoff between the accuracy of the preconditioner and the computational burden of $\mathcal{H}$-matrix operations (which increases as ϵ_p decreases), and the number of iterations (which decreases as ϵ_p decreases). The interested reader can jump ahead to Fig. 4.7 for an illustration.

As the analysis used to derive (4.20) is not rigorous in its evaluation of tightness of bound nor constants used to evaluate the time, it cannot be practically used to optimize the overall solve time; it simply provides a bound. However, it clearly expresses the role that ϵ_p plays to balance the work between the preconditioner and the number of iterations, suggesting that in practical situations there is an opportunity to improve time-to-solution performance through appropriate choices of ϵ_p.

4.3.3 Heterogeneous tolerance possibilities

In the $\mathcal{H}$-matrix framework, assigning different tolerances to various blocks of the block cluster tree may improve the efficiency of the solver while maintaining solution accuracy. Specifically, in the preconditioner we may assign different tolerances to leaf admissible blocks

($\epsilon_{p,leaf}$) and non-leaf admissible blocks ($\epsilon_{p,nonleaf}$) as these different blocks contribute in different ways throughout algorithms like $\mathcal{H}$-**LU**. These tolerance could also vary by tree level and block cluster size. A comprehensive analysis of adaptive tolerance assignment across the block cluster tree is beyond the scope of this paper. In what follows we will primarily investigate the effects of a single preconditioner tolerance for a series of examples. As a preliminary look at a more general tolerance framework, we will present a single example using different constant tolerances for the leaf and non-leaf admissible nodes.

4.4 Numerical Results

We implemented the proposed scheme by using in-house preconditioned iterative GMRES solver with HLibPro-enabled matrix-vector products [10] and HLibPro-enabled fast direct $\mathcal{H}$-**LU** decomposition of LCN matrices (previously we used HLibPro for purely direct $\mathcal{H}$-**LU** based solutions [14]).

The solver was tested on a variety of PEC geometries, including irregular shapes with sharp corners and objects with deep cavities. In all cases, the system matrix $\mathcal{Z}$ was constructed with a tolerance of $\epsilon_z = 10^{-4}$, based on our experience that this value is sufficient to achieve accurate third- and fourth-order solutions on sufficiently refined meshes for the problems considered. For order 5 solution, to keep up with $\mathcal{O}(h^p)$ behavior of the high-order solutions, we need to set the tolerance of impedance matrix at least $\epsilon_z = 10^{-5}$.

Examples 1-4 were run on the Digital Research Alliance of Canada's Narval cluster[25] with AMD Rome 7532 @ 2.40 GHz 256M cache L3 processors. All jobs used 10 cores in a custom combination of MPI (near-interactions) and OpenMP (HLibPro-based [10] $\mathcal{H}$-matrix) through various parts of the computations. Example 5 was run on the Digital

Research Alliance of Canada's Niagara cluster[26] with Intel Skylake cores at 2.4GHz or Cascaselake cores at 2.5GHz. This job was run using 80 cores in parallel.

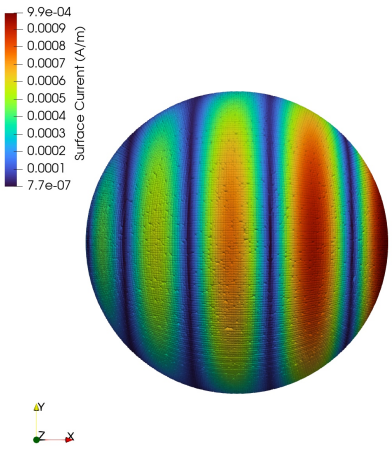
Examples 1-4 use a single preconditioner tolerance ϵ_p. Example 5 uses two tolerances $\epsilon_{p,leaf}$ and $\epsilon_{p,nonleaf}$.

4.4.1 Example 1 - Sphere

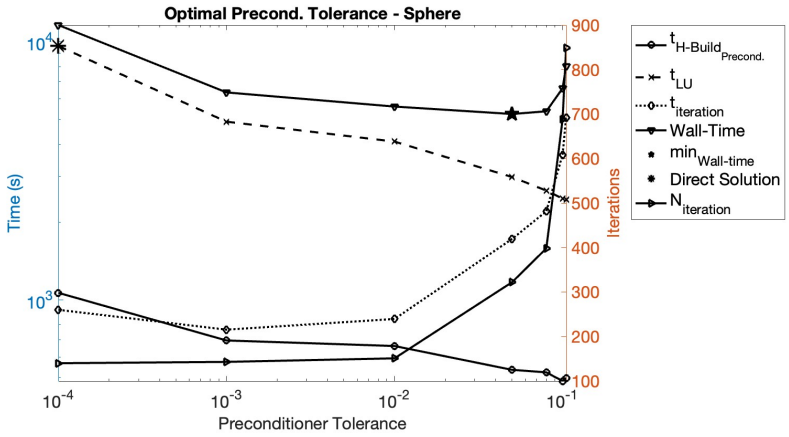
As a first example, we solve a 2λ sphere with $N = 120,544$ unknowns arising from a 4th-order LCN discretization of the CFIE ($\alpha = 0.2$). In this example, the incident field is an x-polarized electric dipole positioned far from the object along the z axis. Fig. 4.6a illustrates the solution (surface current). This problem was solved for a variety of different preconditioner tolerances ϵ_p resulting in the computational times shown in Fig. 4.6b.

The figure illustrates the effects of ϵ_p on total computation time (in blue) and number of iterations (in orange). At one extreme for the 'low' error $\epsilon_p = 10^{-4}$, the computational time is high, despite the number of iterations being low. This result is obviously expected due to the effort that must be put into factoring the preconditioner. In fact, as $\epsilon_z = 10^{-4}$, factoring the preconditioner amounts to solving the system directly, confirmed by the proximity of the direct solve time (outside of an iterative framework) also shown in Fig. 4.6b. At the other extreme, choosing a 'high' error $\epsilon_p = 10^{-1}$ results in an increased time-to-solution due to the large number of iterations required for convergence as a result of the weak preconditioning of the system.

While the iteration model in (4.18) should be applicable to any example, it is illustrative to compare the model to the number of required iterations as a function of tolerance for this example. The comparison shown in shown in Fig. 4.7 shows good agreement.



(a)

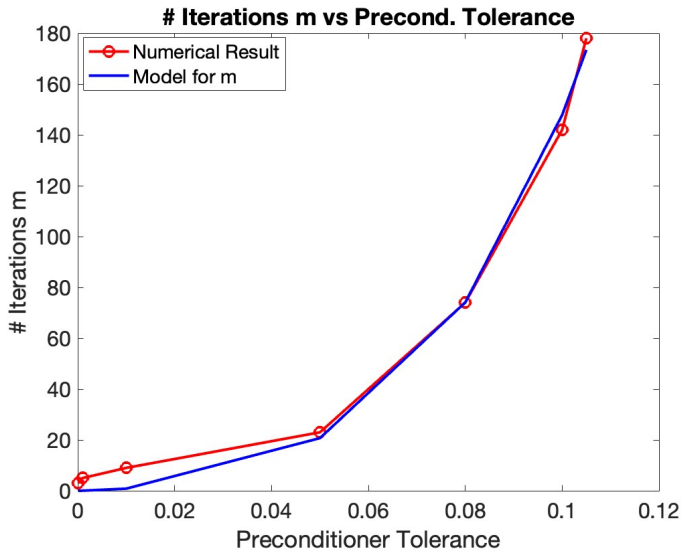


(b)

Figure 4.6: 2λ Sphere Example 1: (a) Surface current for LCN order 4 with $N = 120,544$ unknowns, (b) wall-time study for GMRES with different tolerances for preconditioner $\mathcal{H}$-matrix in $N = 120,544$ unknowns.

4.4.2 Example - B2 Aircraft

We next tested the algorithm on a B2 aircraft model. Figure 4.8a shows the 4th-order MFIE ($\alpha = 0$) solution for an 18λ model with $N = 92,096$ unknowns in the presence of the same electric dipole used in the previous example. The electrical size and sharp edges of this model make it more challenging than the sphere. Figure 4.8b again demonstrates the



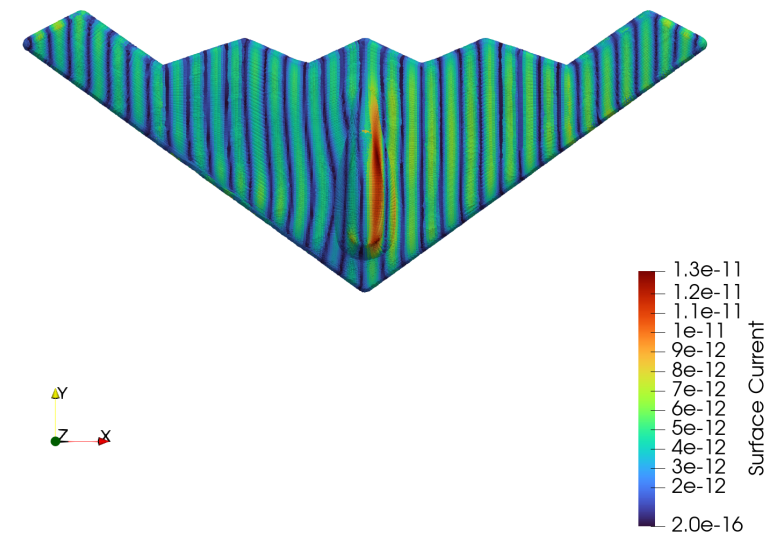
(a)

Figure 4.7: Curve fitting (4.18) in Example 1 for the sphere with $N = 120,544$ unknowns

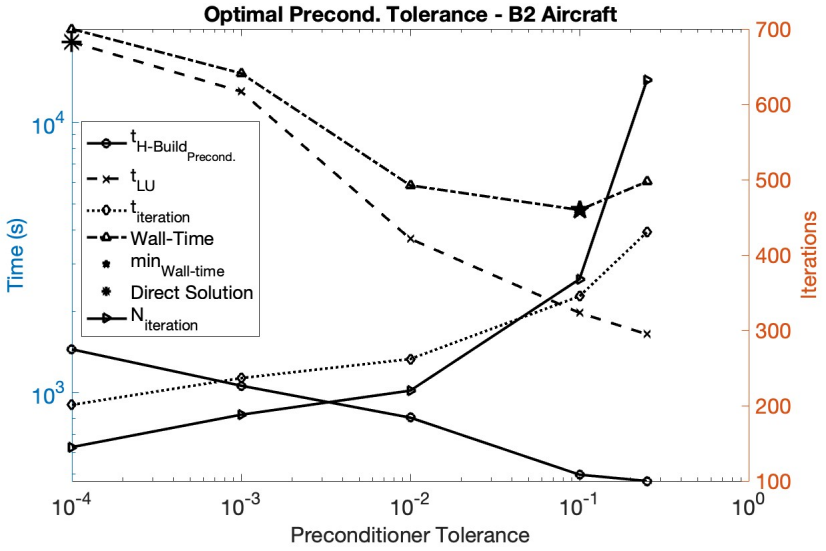
relationship between solve time and preconditioner accuracy. For this particular example, the optimal preconditioner tolerance (of those we tested) is near $\epsilon_p = 10^{-1}$. Lower error again have increased time-to-solution due to the effort required to factor the preconditioner, while higher error increase the solution time due to the number of iterations.

4.4.3 Example 3 - Open Cylindrical Resonator (OCR)

We next test an Open Cylindrical Resonator with a cavity depth of $\ell = 8.6$ m and a radius $r = 3.3$ m. The cavity is exposed to a z-polarized electric dipole placed far from the object along the $-y$ axis. Problems involving cavities are of interest in this study due to the effects of cavity resonances on the conditioning of the resulting SLAE. To investigate resonance effects, we ran a coarse frequency sweep, solving the problem at 100-500 MHz in 100 MHz steps. In all cases the problem involved $N = 23,544$ unknowns for a 3rd-order LCN solution of the CFIE ($\alpha = 0.2$). The surface currents at 300 MHz ($\ell = 8.6\lambda$) are shown in Fig. 4.9a



(a)



(b)

Figure 4.8: 18λ B2 Aircraft: (a) LCN order 4 Surface current solution with $N = 92,096$ unknowns. (b) Optimality study using GMRES with different preconditioner tolerances ϵ_p.

while time-to-solution as a function of ϵ_p is shown in Fig. 4.9b.

The general trend is that increasing the frequency increases the time-to-solution. This is expected as higher frequency problems lead to larger electrical sizes for admissible blocks (for a fixed number of unknowns) and $\mathcal{H}$-matrix approximations require larger rank in these blocks to achieve the same error level. While the optimal choice of preconditioner tolerance still lies between the two extremes, the range of near-optimal tolerances is frequency dependent. This suggests that additional iterative cost is potentially being incurred due to frequencies being near resonance. To further investigate this effect, Fig. 4.9c shows the wall time as a function of frequency for various tolerances ϵ_p. Of particular interest is the fact that the best wall time for 500 MHz was observed using $\epsilon_p = 2 \times 10^{-2}$, but at 300 MHz (which from a frequency content perspective should be an easier problem) this tolerance did not lead to convergence. Rather, a lower error of 10^{-2} was needed to ensure convergence. This suggests the presence of resonances near this frequency.

To further investigate resonance effects, we ran a finer frequency sweep around $f = 300$ MHz. Fig. 4.10 shows, for a fixed tolerance $\epsilon_p = 10^{-2}$, the number of iterations required for convergence. Both 300 and 320 MHz exhibit jumps in the number of required iterations, indicating proximity to resonances.

4.4.4 Example 4 - Aircraft with Inlet/Exhaust Cavities

As a final example we consider an aircraft with cavities, combining the challenges of complicated geometries with those of cavity resonances. A 2nd-order MFIE solution was computed at a frequency corresponding to the aircraft being 15λ long. The incident field is a z-polarized electric dipole placed far from the object along the $-x$ axis. The model, with $N = 243,664$ unknowns, includes three cavities: two inlets at the front ($\ell_{inlet} = 5.5\lambda$

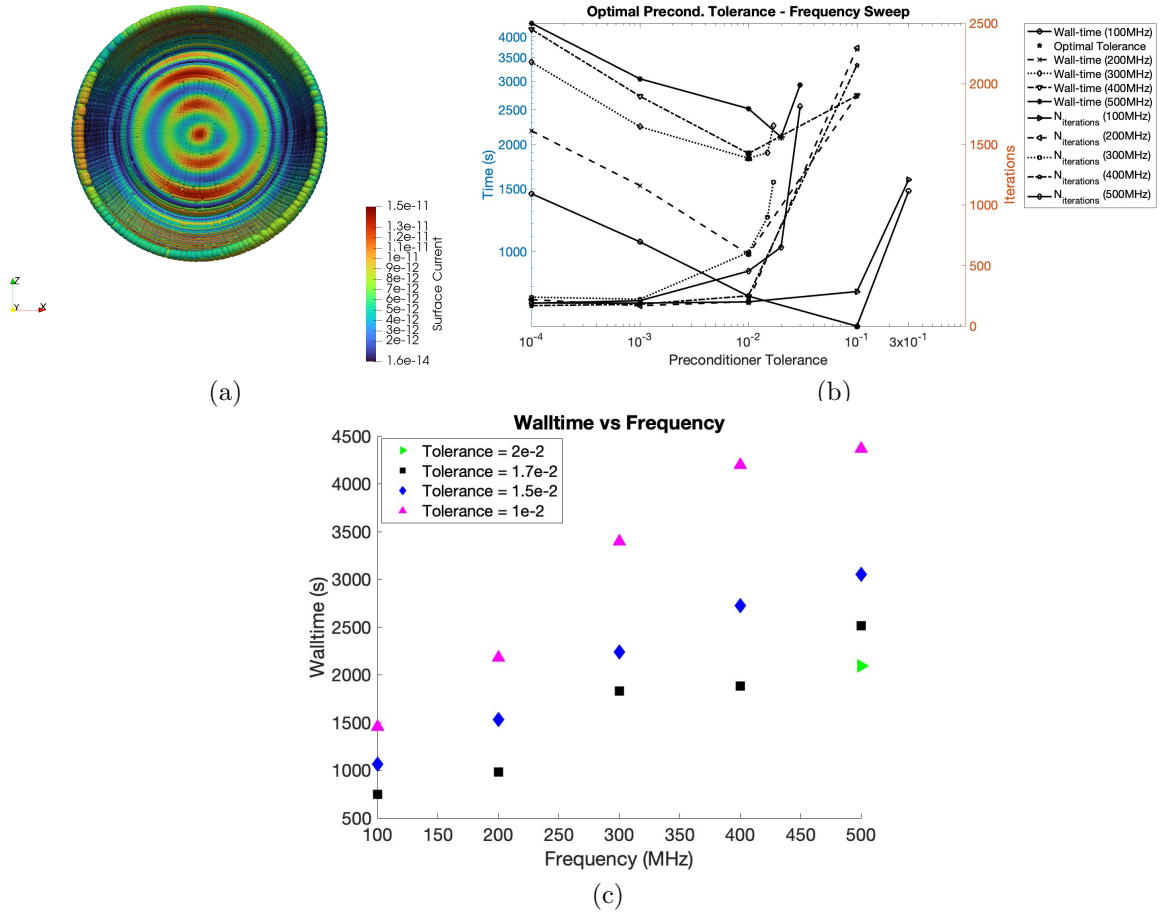


Figure 4.9: $\ell = 8.6\lambda$ OCR: (a) surface current at $f = 300\,\text{MHz}$; (b) optimality study using GMRES with different preconditioner tolerances ϵ_p at various frequencies; (c) wall time vs. frequency for different ϵ_p.

and a radius $r_{inlet} \approx 0.75\lambda$), and an exhaust at the back ($\ell_{exhaust} = 2.5\lambda$ and a radius $r_{exhaust} \approx 1.8\lambda$).

Fig. 4.11a-4.11b show the geometry and solution. The solution obtained using FEKO [27] is provided in Fig. 4.11c for reference. A discrepancy is observed between the FEKO and LCN results, particularly in the $\theta = 90°$ cut. This difference arises from the input geometry. FEKO accepts mesh-based geometries, such as STL files, whereas the LCN solver requires NURBS-

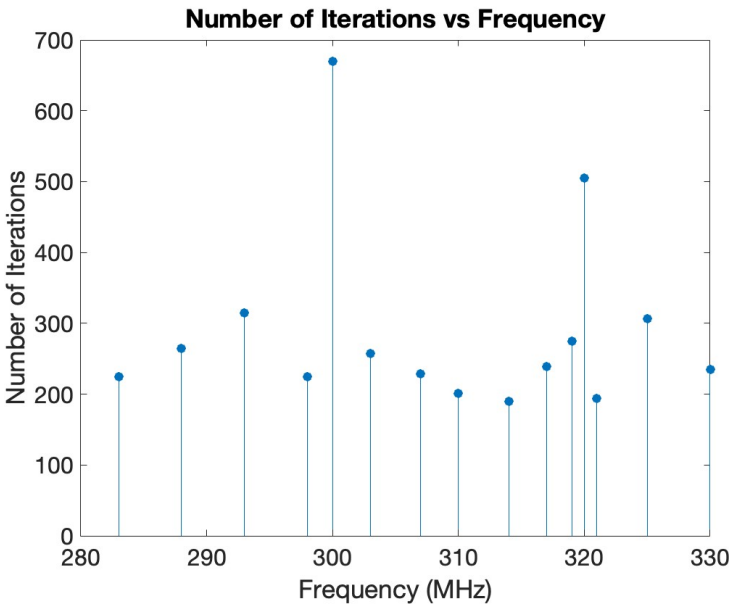
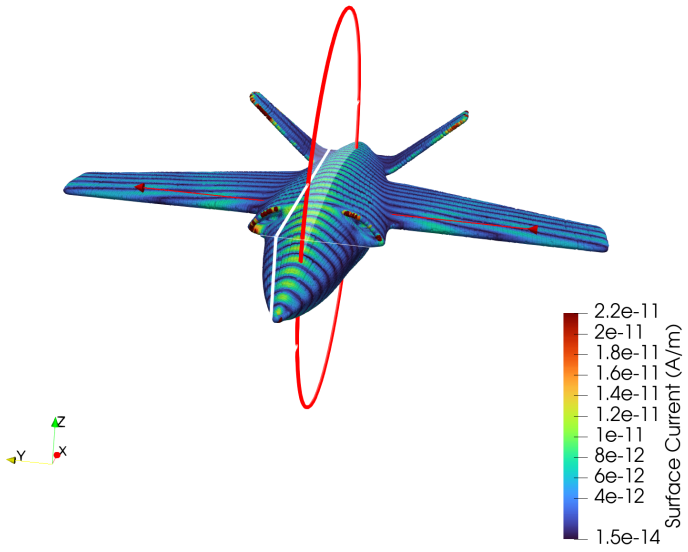


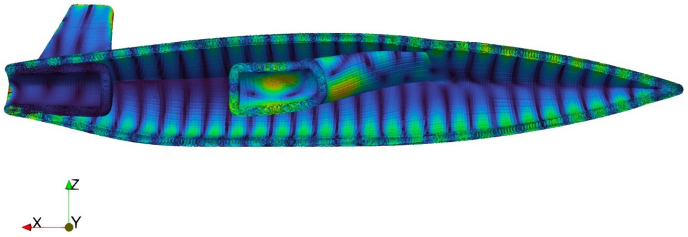
Figure 4.10: $\ell = 8.6\lambda$ OCR Example: Number of iterations versus frequency around $f = 300$ MHz.

based geometries to generate quadrilateral Bézier patches. Converting NURBS geometries in Rhino to STL introduces a minor error, which likely accounts for this discrepancy.

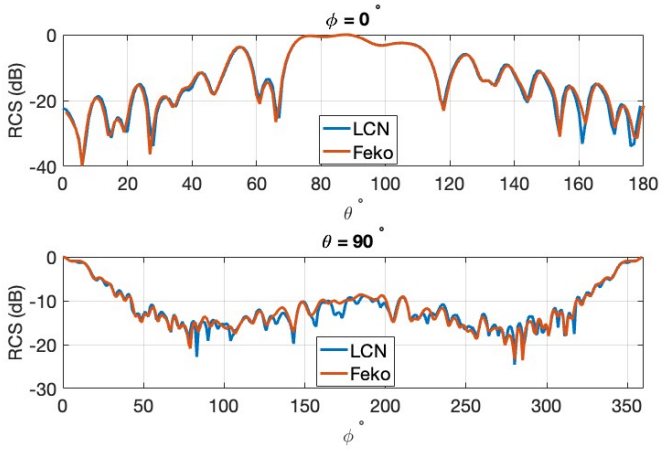
Given the level of detail in the model and experience with the previous three examples, one might guess that an error level of $\epsilon_p = 10^{-3}$ might be necessary for convergence. This is indeed the case: the performance summary in Table 4.1 includes the fact that $\epsilon_p = 10^{-2}$ did not lead to a converging iterative process. Therein, one also finds that the computational burden is dominated by the $\mathcal{H}$-**LU** factorization time. Efforts to reduce this time through efficient distributed parallelism should be sought in the future.



(a)



(b)



(c)

Figure 4.11: 15λ Aircraft: (a) 2nd-order LCN surface current for $N = 243,664$ unknowns. (b) xz-cut depicting the inlet and the exhaust (c) Radar Cross Section (RCS) $\theta = 90°$ cut is compared to Feko solution.

Table 4.1: Performance summary for Experiment 4 (values aligned numerically).

Task	$\epsilon_p = 1 \times 10^{-2}$	$\epsilon_p = 1 \times 10^{-3}$
$\mathcal{H}$–**LU** Memory (GB)	18	30
$\mathcal{H}$–**LU** Time (min)	313	560
GMRES Time (min)	–	205
GMRES Iterations (iters)	–	748

4.4.5 Example 5 - Sphere with Heterogeneous Preconditioner Tolerance

In this final example, we consider the problem of scattering from a 4λ sphere with $N = 200,000$ unknowns (a 4th-order solution). Unlike Example 1 we now consider two preconditioner tolerances $\epsilon_{p,leaf}$ and $\epsilon_{p,nonleaf}$. Admissible leaf nodes in the block cluster tree are assigned the tolerance $\epsilon_{p,leaf}$, while admissible non-leaf nodes are assigned $\epsilon_{p,nonleaf}$ The optimal preconditioner for minimizing the walltime now lies somewhere in a two dimensional optimization space in $\epsilon_{p,leaf}$ and $\epsilon_{p,nonleaf}$. Fig. 4.12 shows the preconditioner walltime for $\epsilon_{p,leaf}$ varying from 1×10^{-4} to 3×10^{-1} and $\epsilon_{p,nonleaf}$ varying from 1×10^{-4} to 2×10^{-1}. The optimal walltime of 162.74 seconds occurs when $\epsilon_{p,leaf} = 1 \times 10^{-1}$ and $\epsilon_{p,nonleaf} = 5 \times 10^{-2}$, i.e., there is a benefit to having two different tolerances (the best result when $\epsilon_{p,leaf} = \epsilon_{p,nonleaf}$ is at the tolerance 5×10^{-2} and requires 280.3 seconds).

4.4.6 Example 6 - Sphere with Block Diagonal and Near-Field Preconditioner $\mathcal{H}$-matrix

In this example, we tested block diagonal and near-field $\mathcal{H}$matrices as the preconditioner matrix. Fig. 4.13a shows the structure of the block diagonal matrix when it is built at the leaf level with the block size 2,000. In this test, which was performed for a 4λ sphere with

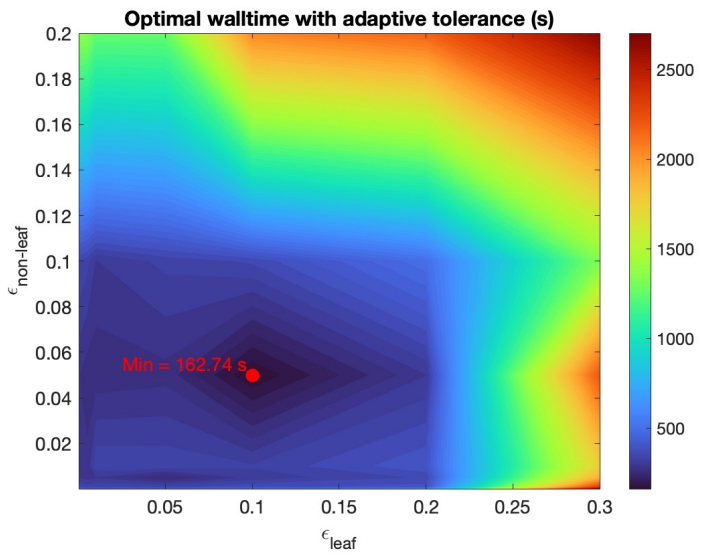


Figure 4.12: Optimal preconditioner walltime in adaptive arithmetic tolerance.

$N = 50,000$ unknowns, the solver did not converge to the solution. We repeated the test for the largest block size and also, more inadmissible blocks(low compression), and it did not converge again. We also tested the near-field matrix as the preconditioner by deleting all admissible blocks at the leaf level and keeping only the inadmissible blocks, as depicted in Fig. 4.13b. In the near-field test, we achieved the same solution accuracy as with a full $\mathcal{H}$-matrix implementation for the preconditioner matrix, but the number of iterations for GMRES was $m = 7,356$. The preconditioners implemented in this example are not optimal for this system of equations, as indicated by the wall time results when compared to the previous examples presented in this paper.

4.4.7 Discussion and Recommendations

The proposed $\mathcal{H}$-matrix-preconditioned computational framework effectively allows a user to tune a numerical solution between the extremes of a direct solver ($\epsilon_p = \epsilon_z$) and an iterative

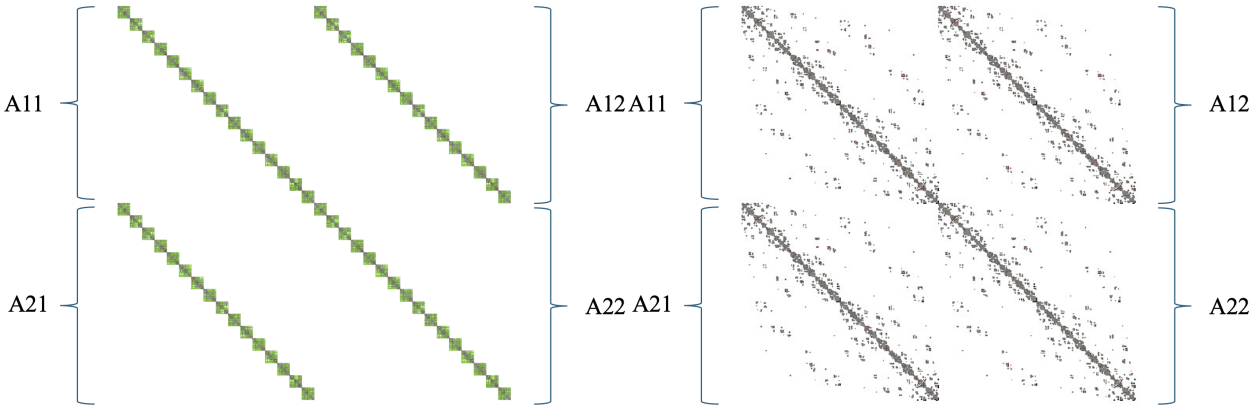


Figure 4.13: Preconditioners for a 4λ sphere: (a) Block diagonal matrix. (b) Near-field matrix.

solver without any preconditioning by continuously varying the preconditioner tolerance. The robustness offered by this scheme makes it applicable to resonant problems, provided the tolerance is appropriately adapted to the excitation frequency. Examples 1-4 (single tolerance ϵ_p) have shown that a problem-dependent optimal value of ϵ_p exists, and that, for the problems tested, a range of values can lead to near equal time-to-solution. Example 5 showed that by providing more preconditioner flexibility (in terms of two tolerances $\epsilon_{p,leaf}$ and $\epsilon_{p,nonleaf}$) even better performance is possible, demonstrating about 75% improvement in CPU time.

While our limited testing is not sufficient to determine optimal preconditioner tolerance policies that will be applicable to any example, we offer the following heuristics based on our experience.

In our experience with the tested examples using a single tolerance ϵ_p, a tolerance of 10^{-2} is typically sufficient for geometries without cavities, while 10^{-3} is recommended for those with cavities. These values have proven effective when $\epsilon_z = 10^{-4}$. However, if a lower overall

error (i.e., a smaller ϵ_z) is desired, further investigation into the relationship with ϵ_p would be necessary. For geometries with sharp edges or larger electric sizes, stricter tolerances for ϵ_p may be required to ensure accuracy. For example, choosing $\epsilon_p = 10^{\frac{\log_{10}(\epsilon_z)}{2}}$ may be a reasonable choice. In general, overshooting the accuracy requirement slightly by lowering ϵ_p can be beneficial, as faster convergence can offset increased $\mathcal{H}$-**LU** costs. Predicting the optimal value for ϵ_p is challenging, but understanding the preconditioner factorization and substitution times could lead to meaningful optimization of the solve time.

For heterogeneous tolerance configurations, significantly more investigation is required before making any concrete recommendations. In general, performance is similar to the single tolerance case, but with increased compression. That is, a less accurate preconditioner can still maintain a low number of iterations. Overall, optimal performance will most likely require a heterogeneous configuration.

As a final discussion point we note that while we have focused on computational time, memory usage is also of interest. Fig. 4.14 summarizes the memory use for Examples 1-4. For users that are memory-bound, increasing the preconditioner tolerance when possible will improve memory consumption, further demonstrating the flexibility of the proposed framework.

4.5 Conclusion

We have presented a robust, unified framework for solving scattering problems that leverages $\mathcal{H}$-matrices for the efficient storage and computations of the both the system matrix and the preconditioner. Either a single error tolerance ϵ_p, or a pair of tolerances $\epsilon_{p,leaf}$ and $\epsilon_{p,nonleaf}$ tunes the performance, balancing time and memory requirements.

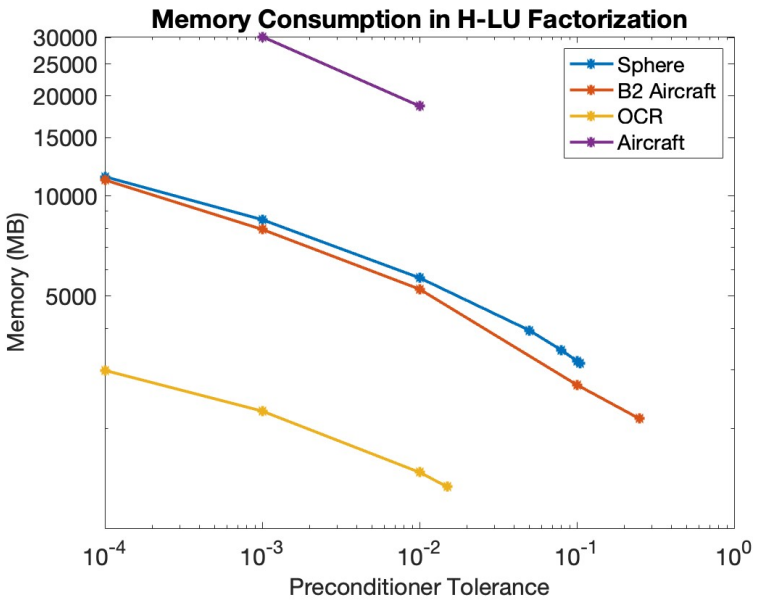


Figure 4.14: $\mathcal{H}$-**LU** memory consumption as a function of error level ϵ_p for the examples studied. Cylindrical resonator memory is shown for 300 MHz.

Our investigation has shown that balancing the accuracy of preconditioner $\mathcal{H}$-matrix and computational cost incurred by it (including preconditioner construction, its $\mathcal{H}$-**LU** factoring and cost of iterations) leads to optimal solution wall time of the preconditioned GMRES iterative solver. Moderate reductions in preconditioner accuracy can achieve significant savings in memory usage and CPU time. However, excessive reductions increase the number of iterations required, ultimately leading to higher CPU time. Examples involving resonant cavities demonstrated the potential for the proposed framework to handle performance tradeoffs as a function of excitation frequency.

We observed that introducing a more flexible heterogeneous tolerance presents a promising extension to our approach. It allows for flexibility in adjusting tolerance on a block-wise basis, facilitating higher compression (resulting in a more compact preconditioner matrix) while still ensuring high accuracy and fast convergence. This method could potentially lead

to further optimizations in balancing memory usage and CPU time. However, it requires more in-depth investigation with complex, large-scale, and resonant geometries to draw a more definitive conclusion.

Future work should focus on building experience in choosing optimal preconditioner tolerances, and task-based parallelization of $\mathcal{H}$-**LU** factorization and substitution. Both of these efforts have the potential to greatly reduce the time-to-solution for solving general scattering problems, further pushing our capabilities for solving realistic large-scale problems for real-world applications.

Bibliography

[1] R. F. Harrington, *Field Computation by Moment Methods*. Hoboken, NJ, USA: Wiley–IEEE Press, 2001.

[2] C. A. Balanis, *Antenna Theory: Analysis and Design*, 4th ed. Hoboken, NJ, USA: Wiley, 2016.

[3] J. D. Kraus and R. J. Marhefka, *Antennas: For All Applications*, 3rd ed. New York, NY, USA: McGraw–Hill, 2002.

[4] D. Colton and R. Kress, *Inverse Acoustic and Electromagnetic Scattering Theory*, 2nd ed. Berlin, Germany: Springer, 1998.

[5] R. F. Harrington, *Field Computation by Moment Methods*. Hoboken, NJ, USA: Wiley–IEEE Press, 2001.

[6] J. C. Nédélec, *Acoustic and Electromagnetic Equations*. Berlin, Germany: Springer, 2001.

[7] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed. Baltimore, MD, USA: Johns Hopkins Univ. Press, 2013.

[8] W. Hackbusch, *Hierarchical Matrices: Algorithms and Analysis*. Berlin, Germany: Springer, 2015.

[9] S. Börm, L. Grasedyck, and W. Hackbusch, "Introduction to hierarchical matrices with applications," *Eng. Anal. Bound. Elem.*, vol. 27, no. 5, pp. 405–422, 2003. [Online]. Available: `https://doi.org/10.1016/S0955-7997(02)00152-2`

[10] R. Kriemann, *HLIBpro C Language Interface*. Leipzig, Germany: Max Planck Institute for Mathematics in the Sciences, 2008.

[11] M. Bebendorf, *Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems*, Lecture Notes in Computational Science and Engineering, vol. 63. Berlin, Germany: Springer, 2008.

[12] M. Bebendorf, "Approximation of boundary element matrices," *Numer. Math.*, vol. 103, no. 3, pp. 389–417, 2006.

[13] M. Bebendorf, "Approximation of boundary element matrices," *Numer. Math.*, vol. 86, no. 4, pp. 565–589, Dec. 2000.

[14] O. Babazadeh, J. Hu, E. Sever, I. Jeffrey, C. Sideris, and V. Okhmatovski, "Fast direct error-controlled solution of scattering problem via $\mathcal{H}$-matrix accelerated locally corrected Nyström method of combined field integral equation," in *Proc. IEEE MTT-S Int. Microwave Symp. (IMS)*, 2024.

[15] S. D. Gedney, "On deriving a locally corrected Nyström scheme from a quadrature sampled moment method," *IEEE Trans. Antennas Propag.*, vol. 51, no. 9, pp. 2402–2412, Sep. 2003.

[16] S. D. Gedney and J. C. Young, "The locally corrected Nyström method for electromagnetics," in R. Mittra, Ed., *Computational Electromagnetics*. New York, NY, USA: Springer, 2014.

[17] [Online]. Available: `https://utlists.utexas.edu/sympa/subscribe/austincembenchmarks`

[18] R. Gholami, J. Mojolagbe, A. Menshov, F. Sheikh Hosseini Lori, and V. Okhmatovski, "$\mathcal{H}$-matrix arithmetic for fast direct and iterative method of moment solution of surface-volume-surface EFIE for 3-D radiation problems," *Prog. Electromagn. Res. B*, vol. 82, pp. 189–210, 2018.

[19] M. Bebendorf and S. Rjasanow, "Adaptive low-rank approximation of collocation matrices," *Computing*, vol. 70, pp. 1–24, 2003.

[20] K. Wang, M. Li, D. Ding, and R. Chen, "A parallelizable direct solution of integral equation methods for electromagnetic analysis," *Eng. Anal. Bound. Elem.*, vol. 85, pp. 158–164, 2017. [Online]. Available: `https://doi.org/10.1016/j.enganabound.2017.10.005`

[21] Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput.*, vol. 7, no. 3, pp. 856–869, Jul. 1986.

[22] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. Philadelphia, PA, USA: SIAM, 2003.

[23] W. Hackbusch, *Iterative Solution of Large Sparse Systems of Equations*. Berlin, Germany: Springer, 1994.

[24] R. S. Varga, "On the iterative solution of linear systems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 2, pp. 53–58, Jun. 1957.

[25] [Online]. Available: https://docs.alliancecan.ca/wiki/Narval/en

[26] [Online]. Available: https://docs.alliancecan.ca/wiki/Niagara/en

[27] [Online]. Available: https://help.altair.com/feko/pdf/Altair_Feko_User_Guide.pdf

Chapter 5

Controlling Solution Accuracy via Adaptive hp-Refinement in High-Order Integral Equation Solvers for Sharp-Edged PEC Scatterers

This chapter presents an adaptive hp-refinement framework for point–based boundary integral solvers targeting sharp–edged PEC scatterers. The core discretization employs the Locally Corrected Nyström (LCN) method supplemented by a region–aware design: (i) aggressive h–refinement with low p in edge zones to resolve singular behavior; (ii) smoothly tapered transition patches; and (iii) coarse h with high p on smooth patches. This monotone taper from fine/low to coarse/high preserves the target $\mathcal{O}(h^p)$ convergence in smooth patches while avoiding over–resolution at edges (Research Question 4).

Beyond the discretization, we introduce a per–observation–point near–set construction

that selects near interactions locally and schedules quadrature/correction orders to meet a prescribed tolerance without overloading the local corrections. In practice, this reduces the walltime for near–interaction assembly at fixed accuracy relative to uniform high–order meshes on the same geometry. The formulation integrates seamlessly with our $\mathcal{H}$–matrix accelerated solver and hybrid direct–iterative workflow ($\mathcal{H}$–**LU** preconditioning), and is implemented with hybrid MPI/OpenMP parallelism on the *Trillium* supercomputer (Alliance Canada) [1].

Comprehensive benchmarks on canonical and complex scatterers (disk, cube, and a B2–like aircraft) confirm that the adaptive *hp* strategy recovers surface currents with accuracy comparable to uniform high-order, while offering a more favorable accuracy–cost trade–off: near–field assembly is notably cheaper at a given tolerance, and end–to–end memory/runtime scale competitively at large N.

Abstract

We propose an adaptive hp-refinement strategy for sharp-edged PEC scatterers in point-based boundary integral solvers. The method couples aggressive h-refinement with low polynomial order near edges to resolve singular behavior in current, with p-enrichment on smooth patches, and inserts transition patches that taper monotonically from fine/low to coarse/high resolution. This design tries to preserve the target $\mathcal{O}(h^p)$ convergence of the underlying scheme. In addition, we introduce a per–observation-point near-set construction that selects near interactions locally to meet error tolerances while controlling the cost of local corrections. Validation on canonical scatterers, a disk, a B2-like aircraft, a cube, and a missile with sharp corners shows that, the adaptive hp strategy attains the performance of uniform high-order at substantially lower memory and runtime with a better accuracy in solution. The framework yields accurate surface current recovery and well-conditioned linear systems, enabling robust and efficient high-order boundary integral analyses for radar cross section prediction, antenna design, and electromagnetic compatibility.

5.1 Introduction

Accurate evaluation of surface currents on perfectly conducting (PEC) scatterers is central to a wide range of electromagnetic analyses, from antenna characterization to compatibility testing. Boundary integral formulations, such as the Electric Field Integral Equation (EFIE), Magnetic Field Integral Equation (MFIE) and Combined Field Integral Equation[2, 3, 4], reduce three-dimensional scattering problems to surface discretizations, resulting in dense but well-conditioned systems that capture full-wave physics without volumetric meshing [5, 6, 7].

Traditionally, low-order discretizations like the RWG–MoM (triangular basis functions) have dominated practical solvers due to their simplicity and robustness near geometric singularities [5]. These schemes achieve linear convergence in surface currents and quadratic convergence in far-field quantities, but attaining high accuracy requires excessively fine meshes, inflating both memory usage and solver time. Fast algorithms such as the Fast Multipole Method (FMM) [8, 9], Adaptive Integral Method (AIM) [10] and $\mathcal{H}$-matrix [11] mitigate the increase in computational cost, yet the core limitation remains: low-order methods waste degrees of freedom in smooth regions and struggle to fully exploit modern high-performance hardware.

To overcome this, point-based high-order methods in integral equations have been developed. The Locally Corrected Nyström (LCN) technique [12, 13] employs high-order quadrature and interpolation to achieve algebraic convergence rates ($\mathcal{O}(h^p)$), while Chebyshev-based Integral Equation (CBIE) solvers uses global change of variables through orthogonal polynomial expansions [14]. When combined with fast-solver frameworks, such as $\mathcal{H}$-Matrix compression, AIM, or FMM, these approaches deliver reference-grade solutions for smooth geometries with dramatically fewer unknowns than low-order methods such as RWG–MoM.

However, the very characteristic that endows high-order schemes with efficiency is their reliance on smooth basis and quadrature points, which becomes a challenge at sharp edges and corners. Field singularities manifest as reduced convergence rates, spurious oscillations in the reconstructed currents, and localized integration errors. Traditional remedies include: (i) Local h-refinement, which resolves singularities by subdividing elements near edges but requires mesh-dependent heuristics and can destroy the uniform high-order convergence elsewhere [15]. (ii) Singular basis enrichment, which embeds known asymptotic edge behavior into the approximation space but complicates coupling with fast solvers and increases implementation complexity [16]. (iii) Hybrid low/high-order schemes, which treat edges with low-order patches and smooth regions with high-order, yet often suffer from order-mismatch artifacts at the interface [17].

Despite these advances, a unified framework that preserves high-order convergence in smooth patches and control the error in sharp edges, handles geometric singularities robustly, and integrates seamlessly with existing fast-solver engines remains an open challenge. In this work, we address this gap by developing a *adaptive hp-refinement* methodology for point-based integral equation solvers. Our strategy builds on prior hybrid concepts but introduces smoothly varying transition patches that interpolate both mesh density and polynomial order, ensuring compatibility of quadrature and interpolation across refinement interfaces. The result is a solver that retains $\mathcal{O}(h^p)$ convergence in smooth regions, achieves accurate current recovery at edges, and integrates directly with fast algorithms such as FMM and $\mathcal{H}$-matrix frameworks.

We demonstrate the effectiveness of the proposed approach through numerical experiments on canonical PEC geometries with sharp corners, comparing against analytic solution, uniform high-order and locally refined low-order baselines. The results confirm that adap-

tive *hp*-refinement attains high-order accuracy with significantly reduced degrees of freedom, memory footprint, and runtime, paving the way for efficient, large-scale boundary integral analyses of complex scatterers.

5.2 Surface Integral Equations for PEC Targets

We consider a perfectly electrically conducting (PEC) body with surface $S = \partial V$ in a homogeneous medium with $k_0 = \omega\sqrt{\mu_0\varepsilon_0}$, $\eta_0 = \sqrt{\mu_0/\varepsilon_0}$, and

$$G_0(\mathbf{r}, \mathbf{r}') = \frac{e^{-ik_0|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|}.$$

Let $\mathbf{J}$ denote the equivalent electric surface current. The integral equation formulations used below for EFIE, MFIE, and CFIE are explained in detail in [12, 14, 18]. We reproduce here only the essentials needed for our error-controlled *hp*-refinement framework.

5.2.1 Electric Field Integral Equation (EFIE)

For $\mathbf{r} \notin V$, the total electric field is

$$\mathbf{E}(\mathbf{r}) = \int_S \Big(-i\omega\mu_0\, \mathbf{J}(\mathbf{r}')\, G_0(\mathbf{r}, \mathbf{r}') + \frac{\nabla\nabla\cdot}{i\omega\varepsilon_0} \big[\mathbf{J}(\mathbf{r}')\, G_0(\mathbf{r}, \mathbf{r}')\big] - \nabla \times \big[\mathbf{J}^m(\mathbf{r}')\, G_0(\mathbf{r}, \mathbf{r}')\big] \Big) \mathrm{d}S' + \mathbf{E}^{inc}(\mathbf{r}). \tag{5.1}$$

On a PEC target, $\mathbf{J}^m = \mathbf{0}$ and the boundary condition requires the tangential component of the electric field to vanish on S

$$\hat{\mathbf{n}}(\mathbf{r}) \times \mathbf{E}(\mathbf{r}) = \mathbf{0}, \qquad \mathbf{r} \in S.$$

Hence the EFIE is

$$\begin{aligned}\hat{\mathbf{n}}(\mathbf{r}) \times \Big[- i\omega\mu_0 \int_S \mathbf{J}(\mathbf{r}')\, G_0(\mathbf{r},\mathbf{r}')\, \mathrm{d}S' + \frac{\nabla\nabla\cdot}{i\omega\varepsilon_0} \int_S \mathbf{J}(\mathbf{r}')\, G_0(\mathbf{r},\mathbf{r}')\, \mathrm{d}S' \Big] \\ = - \hat{\mathbf{n}}(\mathbf{r}) \times \mathbf{E}^{inc}(\mathbf{r}), \quad \mathbf{r} \in S.\end{aligned} \tag{5.2}$$

5.2.2 Magnetic Field Integral Equation (MFIE)

By the equivalence principle, for $\mathbf{r} \notin V$,

$$\mathbf{H}(\mathbf{r}) = \int_S \nabla \times \big(\mathbf{J}(\mathbf{r}')\, G_0(\mathbf{r},\mathbf{r}')\big)\, \mathrm{d}S' + \mathbf{H}^{inc}(\mathbf{r}). \tag{5.3}$$

Enforcing zero tangential magnetic field on a surface S^- inside the conductor,

$$\hat{\mathbf{n}}(\mathbf{r}) \times \mathbf{H}(\mathbf{r}) = \mathbf{0}, \qquad \mathbf{r} \in S^-,$$

gives the MFIE in the form of

$$\hat{\mathbf{n}}(\mathbf{r}) \times \int_S \nabla \times \big(\mathbf{J}(\mathbf{r}')\, G_0(\mathbf{r},\mathbf{r}')\big)\, \mathrm{d}S' = - \hat{\mathbf{n}}(\mathbf{r}) \times \mathbf{H}^{inc}(\mathbf{r}), \qquad \mathbf{r} \in S^-. \tag{5.4}$$

5.2.3 Combined Field Integral Equation (CFIE)

The CFIE integrates the strengths of both the MFIE and EFIE to provide a more balanced formulation that mitigates numerical issues associated with each individual equation. It is defined as follows:

$$\mathrm{CFIE} = \alpha \mathrm{EFIE} + (1-\alpha)\eta_0 \mathrm{MFIE} \tag{5.5}$$

$$
\begin{aligned}
&(1-\alpha)\,\eta_0\left[\frac{1}{2}\,\hat{\mathbf{t}}(\mathbf{r})\cdot\mathbf{J}(\mathbf{r}) - \hat{\mathbf{t}}(\mathbf{r})\cdot\hat{\mathbf{n}}(\mathbf{r})\times P.V.\int_S \bar{\mathbf{G}}_{HJ}(\mathbf{r},\mathbf{r}')\cdot\mathbf{J}(\mathbf{r}')\,\mathrm{d}S'\right] \\
&\quad + \alpha\,\hat{\mathbf{t}}(\mathbf{r})\cdot\int_S \bar{\mathbf{G}}_{EJ}(\mathbf{r},\mathbf{r}')\cdot\mathbf{J}(\mathbf{r}')\,\mathrm{d}S' \\
&= (1-\alpha)\,\eta_0\,\hat{\mathbf{t}}(\mathbf{r})\cdot\hat{\mathbf{n}}(\mathbf{r})\times\mathbf{H}^{\mathrm{inc}}(\mathbf{r}) - \alpha\,\hat{\mathbf{t}}(\mathbf{r})\cdot\mathbf{E}^{\mathrm{inc}}(\mathbf{r}), \qquad \mathbf{r}\in S,
\end{aligned}
\tag{5.6}
$$

where $\hat{\mathbf{t}}$ is a unit vector tangential to the surface, η_0 is the intrinsic impedance of free space, and α is a parameter used to balance the contributions from the EFIE and MFIE. This formulation enables improved stability and accuracy in the computation of surface currents $\mathbf{J}$ when dealing with complex geometries, effectively minimizing the numerical errors encountered in pure EFIE or MFIE formulations.

Equations (5.2), (5.4), and (5.6) are discretized by a high-order point-based method such as the Nyström scheme with Local Correction (LCN); self/near interactions use locally corrected weights to recover the target polynomial order.

5.2.4 Final System of Linear Algebraic Equations (SLAE)

Any point-based discretization ultimately assembles the same linear system on the common collocation grid; only the evaluation of self and near interactions differs. In all cases, the discretized operator acts on the two tangential components of the surface current at every collocation node.

$$
\mathbf{Z}(h,p)\,\mathbf{J}(h,p) \;=\; \mathbf{H}(h,p), \tag{5.7}
$$

where $\mathbf{Z}(h,p)$ is the system matrix depending on the element size h and local polynomial order p, $\mathbf{J}$ collects the unknown current samples, and $\mathbf{H}$ is the excitation vector derived from the incident fields.

For a specific formulation, $\mathbf{Z}$ and $\mathbf{H}$ are assembled from the corresponding integral equation(s). For instance:

$$\mathbf{Z} \;=\; \alpha\,\mathbf{Z}_E + (1-\alpha)\,\eta_0\,\mathbf{Z}_M, \qquad \mathbf{H} \;=\; \alpha\,\mathbf{H}_E + (1-\alpha)\,\eta_0\,\mathbf{H}_M, \tag{5.8}$$

where $\mathbf{Z}_E$, $\mathbf{Z}_M$ are the EFIE and MFIE discretizations, $\mathbf{H}_E$, $\mathbf{H}_M$ are the corresponding right-hand sides, $\eta_0 = \sqrt{\mu_0/\varepsilon_0}$ balances units, and $0 < \alpha < 1$ is the CFIE blend. Setting $\alpha = 1$ or $\alpha = 0$ recovers the pure EFIE or MFIE system, respectively.

Thus, the solver accommodates any choice of surface integral equation (EFIE, MFIE, CFIE), with differences between point-based schemes arising only in the treatment of singular and near-field terms in $\mathbf{Z}$.

In conventional implementations, the near-interaction radius (R_{near}) is fixed and uniform across the mesh, since the elements are of comparable size. Local corrections are applied within this radius, while far interactions are approximated using one-dimensional Gauss Legendre quadrature. In the present solver, both iterative and direct solution strategies are employed, accelerated by the $\mathcal{H}$-matrix framework (HLibPro) [19], to reduce complexity and memory usage. For iterative solutions, a hybrid direct $\mathcal{H}$-matrix scheme is adopted [20].

In what follows, we introduce a new adaptive *hp*-refinement framework designed to preserve exponential error convergence, i.e., the $\mathcal{O}(h^p)$ behavior of the solution.

5.3 Adaptive hp-Refinement for High-Order Integral Equation Solvers in Sharp-Edged PEC Scatterers

In this section, we introduce an adaptive hp-refinement framework for point-based discretization methods, designed to accurately resolve electromagnetic scattering from PEC geometries with sharp edges and corners. This strategy specifically targets the loss of convergence typically observed near geometric singularities and aims to preserve $\mathcal{O}(h^p)$ error behavior away from sharp edges by sacrificing high-order accuracy in the current reconstruction near edges [20].

Motivating Hypotheses and Observations. The heuristic hp strategy developed in this work is guided by practical observations and hypothesis of the numerical behavior of high-order surface integral formulations and supported by insights from classical MoM analyses:

1. **Polynomial blow-up near edges.** The surface current density exhibits singular behavior near sharp edges and corners (typically $r^{-\alpha}$, where α depends on wedge angle) [21, 22, 23, 24]. We hypothesize that attempting to represent such singularities using purely high-order polynomial bases leads to large coefficient magnitudes that dominate coefficients in regions of smoother current density and, thus: a) reduces available digits of precision for surface regions with small level of current relative to current at the edge elements, b) degrades the conditioning of the global system matrix. Hence, excessive polynomial orders near edges should be avoided.

2. **h-refinement mitigates coefficient amplification.** We hypothesize that decreasing the element size h near edges reduces the area of local testing and integration domains, effectively lowering the influence of large current coefficients on the global system. This

behavior helps preserve the dynamic range and numerical stability of the solver even in the presence of edge singularities.

3. **Edge resolution through h rather than p.** In conventional RWG-based MoM formulations, accuracy near edges can only be improved through finer discretization. We hypothesize that the same principle extends to high-order point-based methods: refining h near edges is more effective for resolving singular fields than increasing p.

4. **Balanced accuracy via combined h and moderate p.** Reducing p alone without corresponding h-refinement would under-resolve the edge region and propagate local errors to the rest of the surface. Through testing, we have observed that a combined strategy (small h with moderate p_{edge}) provides a balanced trade-off between local resolution and overall accuracy.

Although the foundation for these hypotheses/observations is empirical, the hypotheses provide a physical and numerical rationale for the adaptive policy described below. They also align with prior studies on edge singularities in surface integral equations [25, 26, 27]. The refinement mechanism proposed in the following subsection tunes the framework towards a practical and accurate solution in complex PEC targets.

5.3.1 Heuristic hp-Refinement Policy Near Edges

Based on the above hypotheses, an adaptive refinement procedure was designed to keep the solution high-order on smooth patches while containing the impact of singular neighborhoods through targeted h refinement and capped p values. The procedure can be understood in two steps: mesh modifications, and order assignment. Together, these two steps constitute a pre-processing of the mesh in order to assign values of h and p appropriately.

Mesh-modifications. Let $i \in \{s, t, e\}$ denote the region type (smooth, transition, edge). Starting from a baseline smooth size h_{s}, define the mesh size h_i and refinement ratio $\rho_i := h_{\mathrm{s}}/h_i$. We use

$$\rho_{\mathrm{e}} \in \{4, 8\}, \qquad \rho_{\mathrm{t}} \in \{2, 4\}, \qquad \rho_{\mathrm{s}} = 1,$$

rounded to powers of two for conformity and multilevel acceleration. Based on experiments we adopt the following order-to-refinement mapping:

target p_s	p_{e}	ρ_{t}	ρ_{e}
3–4	$1-2$	2	4
5–6	$2-3$	2	8
≥ 7	> 3	4	8

These ratios bias the mesh toward finer h near edges as p increases. We begin by identifying edge elements on the initial mesh. These elements are refined by a factor ρ_{t}, yielding two mesh-size levels. Edge detection is then repeated on the refined mesh, and the newly detected edge elements are further refined by ρ_{e}. Elements refined by ρ_{e} constitute the final *edge* set; those refined only by ρ_{t} but not flagged in the second pass form the *transition* set.

Figure 5.1 summarizes the two–stage refinement pipeline described above. Starting from the baseline smooth size h_{s}, we (i) detect sharp candidates on the initial mesh and refine only those by ρ_{t} to obtain a first partition; (ii) repeat the edge test on this refined subset and refine only the confirmed edges by ρ_{e}.

The refinement process is illustrated in Fig. 5.2. In (a), we start from the basic Bézier-patch tessellation with no tagging. In Stage 1 (b), only the transition set (t) is activated: patches neighboring candidate edges, identified by curvature/dihedral-angle tests but not on

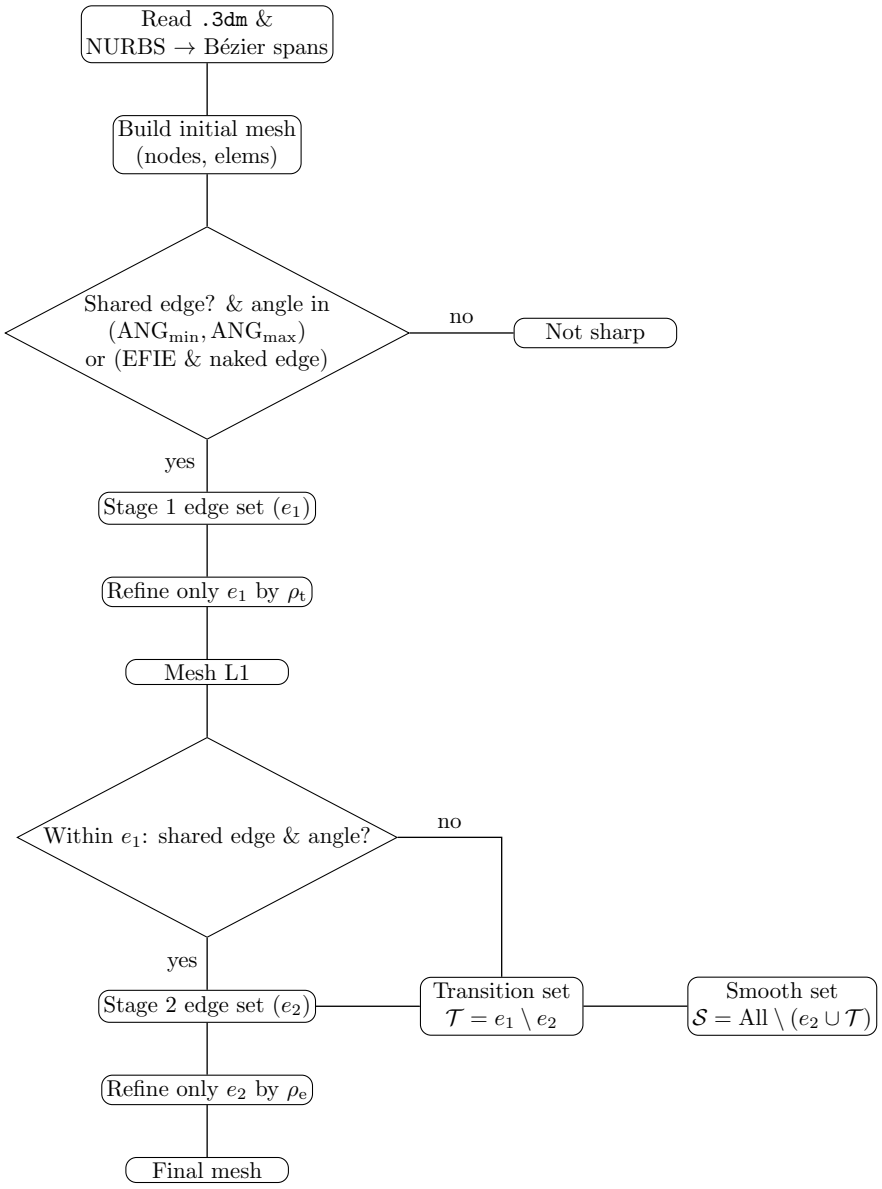


Figure 5.1: Two-stage edge detection and refinement pipeline: Stage-1 → refine by ρ_t; Stage-2 → refine by ρ_e; sets e_2 (edge), t (transition), and s (smooth).

the edge itself, are locally refined to mediate the jump between coarse smooth regions and the eventual edge zones. In Stage 2 (c), edge detection is finalized and the mesh is partitioned into edge (e), transition (t), and smooth (s) regions, shown respectively in green, blue, and

orange. This sequencing isolates the buffering role of (t), concentrates resolution where singular behavior is expected along (e), and preserves coarser elements in (s) for efficiency.

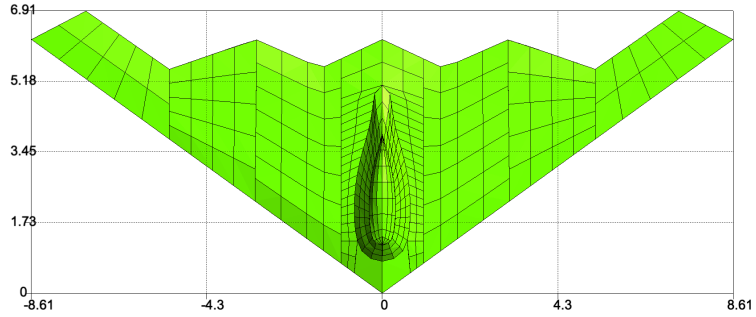
Together with the remaining *smooth* elements, this produces three regions with distinct mesh sizes. The next step is to assign distinct polynomial orders to each of the mesh regions.

Order assignment. To have a smooth transition in polynomial orders between different regions, our assumptions generally start with setting p_{s} to be the target polynomial order on smooth patches, where p_{s} is chosen so as to satisfy an expected error behaviour in smooth regions. We then set

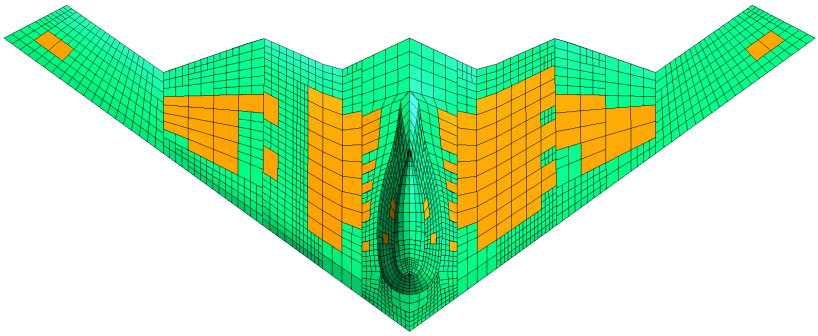
$$p_{\mathrm{e}} = \mathrm{round}\big(p_s/2\big), \qquad p_{\mathrm{t}} = \mathrm{round}\big((p_{\mathrm{e}} + p_s)/2\big).$$

Orders for smooth and edge elements, p_{s} and p_{e}, are specified in the input file, while the transition order p_{t} is computed by the solver. Final order assignment is performed only after the patches have been classified into edge, transition, and smooth regions. Lowering p_s at edges reduces sensitivity to the singular current, while maintaining high-order in smooth regions preserves overall accuracy and efficiency.

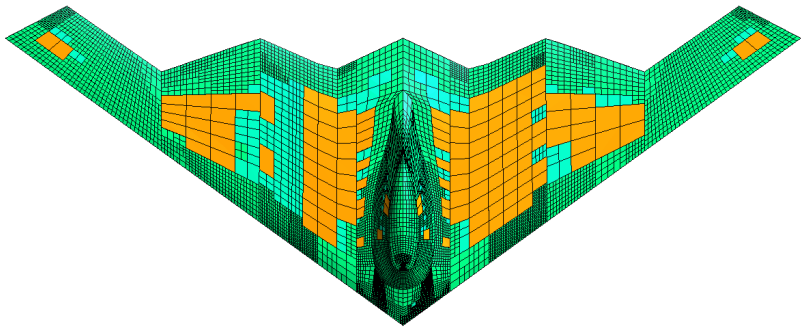
Remark. Because h and p vary across the mesh, a single global near radius is insufficient. The neighborhood must be *local and adaptive* (depending on h_i, p_i, and kh_i) to maintain both accuracy and compression efficiency.



(a) Initial Bézier-patch mesh (no tagging).



(b) Stage 1: transition set (t) refined and highlighted only.



(c) Stage 2: final region partitioning — edge (e), transition (t), smooth (s).

Figure 5.2: Refinement workflow on the B2-like geometry. (a) Initial basic Bézier-patch mesh. (b) Intermediate Stage 1 where only transition patches are refined/tagged. (c) Final Stage 2 after edge detection and full region partitioning. Colors: edge (green), transition (blue), smooth (orange).

5.4 Adaptive Local Near-Set Selection within the *hp*-Refinement Framework

For each quadrature point $\mathbf{x}_i$, we identify near source interactions, where the Helmholtz kernel behaves weakly or in the principal-value sense and requires singular/local correction from far patches where standard high-order rules suffice. Without adaptive mesh refinement it is often sufficient to consider a single global input for the near interaction radius, and to define the set of near interactions $\mathcal{N}$ for point i as

$$\mathcal{N}(i) \;=\; \big\{\, S_e :\; \mathrm{dist}(\mathbf{x}_i, S_e) \le R_{\mathrm{near}} \,\big\} \;\cup\; \{S_{e_i}\}, \tag{5.9}$$

where S_{e_i} is the patch containing $\mathbf{x}_i$.

However, in adaptive or highly nonuniform meshes, a single global R_{near} is either too conservative (too many neighbors for small patches) or unsafe (too few for large ones). We therefore assign each observation point its own radius r_i, depending on the local patch size h_i, polynomial order p_i, and electrical size kh_i. The heuristically chosen near radius equation used in this work is

$$r_i \;=\; \min\big(\tau_0 h_i + \tau_1 p_i + \tau_2 k h_i,\; \tau_\lambda \lambda\big), \tag{5.10}$$

with pairwise symmetry enforced by

$$r_{ij} \;=\; \max(r_i, r_j). \tag{5.11}$$

Here $\tau_0, \tau_1, \tau_2, \tau_\lambda$ are tunable constants (typically $\tau_0 \in [1, 2]$, $\tau_1 \approx 0.1$–0.3, $\tau_2 \approx 0.05$–0.1, $\tau_\lambda \approx 0.1$). Each component of (5.10) is motivated by an underlying analytic constraint:

- *Patch size h_i:* the kernel $G \sim 1/R$ is near-singular when $R \sim h$, so a base term $c\,h_i$ ensures smoothness.

- *Polynomial order p_i:* quadrature remainders scale like $(h/R)^{p+1}$; to keep them below the discretization error $\mathcal{O}(h^p)$, the radius must grow with p, giving the $\tau_1 p_i$ factor.

- *Electrical size kh_i:* phase variation e^{ikR} introduces oscillatory error scaling with kh_i; admissibility and quadrature stability require an additional buffer proportional to kh_i, leading to the $\tau_2 kh_i$ term.

- *Wavelength cap:* beyond a fixed fraction of $\lambda = 2\pi/k$, oscillations aid smoothness and low-rank compression rather than hinder them. Thus, the radius is capped at $\tau_\lambda \lambda$.

Together, these terms yield a balanced adaptive radius that scales with geometry, order, and frequency, while remaining bounded by physics. Patches in $\mathcal{N}(i)$ use LCN local corrections while patches outside use standard high-order quadrature. With (5.9)–(5.11), near/self interactions are captured only where necessary, and accuracy near edge singularities is limited so as to preserve accuracy in smooth regions. At the same time, the median neighbor count remains moderate, keeping $\mathcal{H}$-matrix/$\mathcal{ACA}$ ranks tractable and solver cost balanced.

5.5 Numerical Results and Validation

Computational Setup

All experiments were executed on the *Trillium* supercomputer (Alliance Canada)[1]. Near–field (non-admissible) interactions were evaluated in distributed memory using 192 MPI processors. $\mathcal{H}$-matrix assembly for admissible blocks and the $\mathcal{H}$–**LU** factorization/solve

were performed in shared memory with limited OpenMP parallelism (typically $T \in \{4, 8\}$ threads) using the HLibPro library from the Max Planck institute to balance memory locality and task granularity [19]. In this work, double precision and admissibility $\hat{\eta} = 2$ is used for all of examples.

$\mathcal{H}$-matrix configuration (heterogeneous tolerances) We employ heterogeneous $\mathcal{ACA}$ tolerances to accelerate the preconditioner while preserving current accuracy: coefficients build $\varepsilon_{\mathcal{ACA}}^{\text{coeff}} = 10^{-4}$; preconditioner *leaf* blocks $\varepsilon_{\mathcal{ACA}}^{\text{prec,leaf}} = 10^{-2}$; preconditioner *non-leaf* blocks $\varepsilon_{\mathcal{ACA}}^{\text{prec,nonleaf}} = 10^{-3}$. Near (non-admissible) blocks are stored as dense matrices.

All cases are illuminated by either x or y-polarized plane waves propagating in $+z$.

5.5.1 Thin PEC Disk

A circular, electrically thin PEC disk of radius $a = 2\,\text{m}$ (numerical thickness $0.01\,a$), was solved with the MFIE LCN formulation. For comparison we use the analytic reference solution $\mathbf{J}_{\text{ref}}^{an}$ for an infinitesimal disk.

The mesh refinement strategy was applied to the disk obtained by setting the refinement ratios (ρ_t,ρ_e) = (2,4) for each region. This process produced the mesh shown in Figure 5.3.

Order selection followed (p_s, p_t, p_e) = (7,5,3) for the *hp*-refinement case. The resulting discretization results in N = 109,192 unknowns. However, the regular LCN is solved with $p = p_s = 7$ everywhere and mesh size equal to initial basic mesh ($h = h_s$), which results in N = 156,800 unknowns. The LCN MFIE and analytic surface currents are shown in Fig. 5.4. In the shaded mid-radius annulus, the adaptive solution preserves region contrast and location, while the uniform-order case appears smoothed and shows mild rim ripples; along the edge, the adaptive setup yields a sharper, more stable current layer with no visible

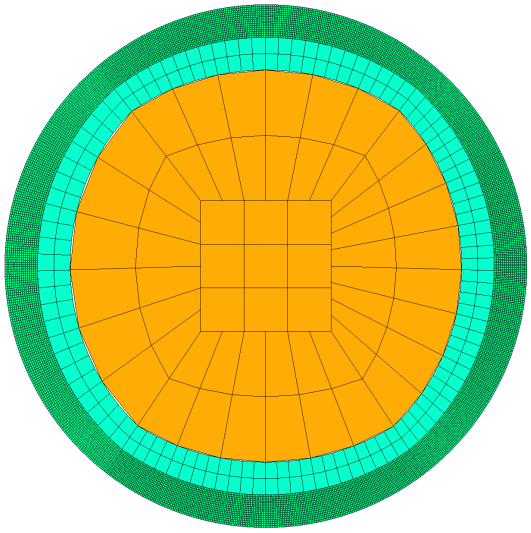
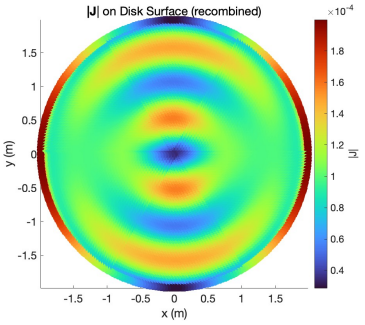


Figure 5.3: Disk mesh classification: edge elements (green), transition patches (blue), and smooth interior (orange).

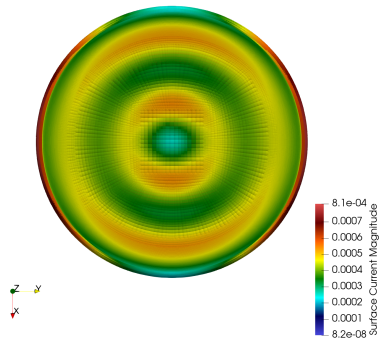
order-mismatch artifacts.

To quantify the error in PEC disk, we benchmark both discretizations against an internal best solution $\mathbf{J}_{\text{ref}}^{LCN}$ computed with the same formulation and geometry but on a much finer mesh (575,552 unknowns) and with tightened numerical controls as shown in Fig. 5.5. We evaluate each candidate and $\mathbf{J}_{\text{ref}}^{LCN}$ on the *same quadrature nodes* and report the mean relative error. The two curves correspond to: (i) *adaptive hp* LCN with $(p_s, p_t, p_e) = (7, 5, 3)$ and smooth-patch sizes $h_s \in \{8, 4, 2\}$ cm; and (ii) *regular* LCN with uniform order 7 and $h_s \in \{4, 2, 1\}$ cm. enlarged near-field radius, higher acceleration settings, increased near-block quadrature, a stricter $\mathcal{ACA}$ tolerance for admissible blocks ($\leq 10^{-6}$), a tighter $\mathcal{H}$–**LU** arithmetic tolerance (down to 10^{-6}), and a stronger preconditioner.

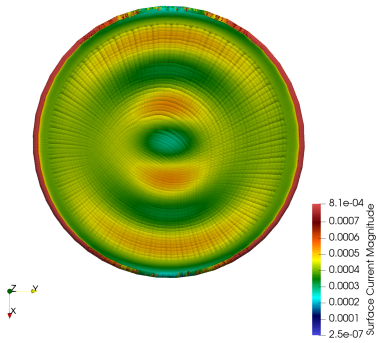
The adaptive curve exhibits a markedly steeper, near-algebraic decay with h and gets closer to the desired $\mathcal{O}(h^p)$ behavior for smooth region, whereas the regular curve remains in a pre-asymptotic regime dominated by higher error in solution.



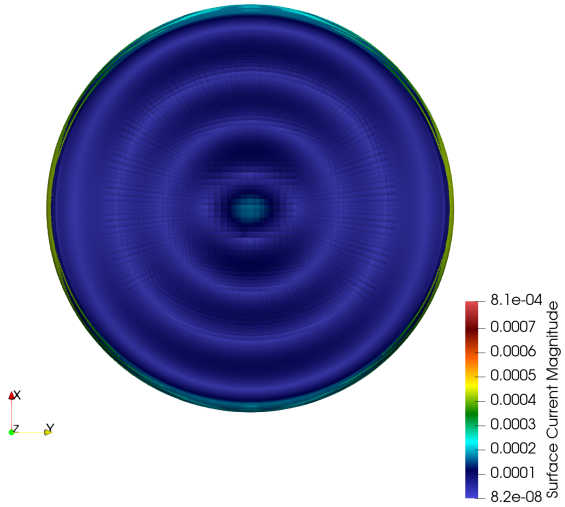
(a) Analytic (top view)



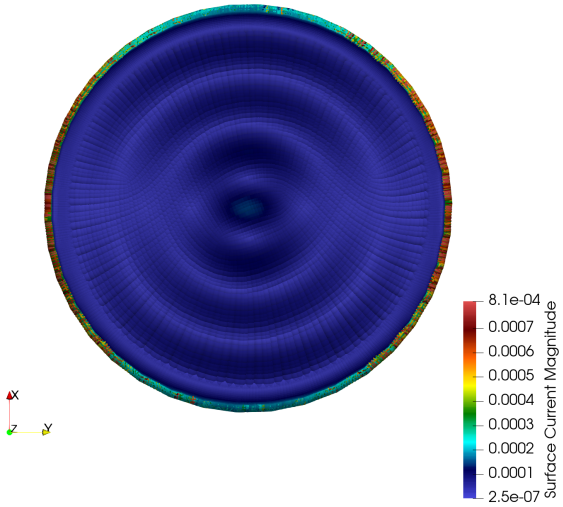
(b) Adaptive *hp* LCN (top): p_s=7, p_t=5, p_e=3



(c) Regular LCN (top): uniform h, p=7



(d) Adaptive *hp* LCN (bottom)



(e) Regular LCN (bottom)

Figure 5.4: Surface current magnitude $|\mathbf{J}|$ on a PEC disk of radius $a = 2\,\mathrm{m}$ at 300 MHz. adaptive *hp* LCN uses p_s=7 on smooth patches, p_e=3 on the edge region with relative h-refinement, and a transition region of p_t=5; Regular LCN uses uniform h and order 7 everywhere. The adaptive configuration better reproduces the concentric modal structure in the shaded annulus and recovers the high-gradient edge layer without overshoot, closely matching the analytic reference. In contrast, the uniform-order case shows visible smoothing in the shaded region and mild ripples at the rim.

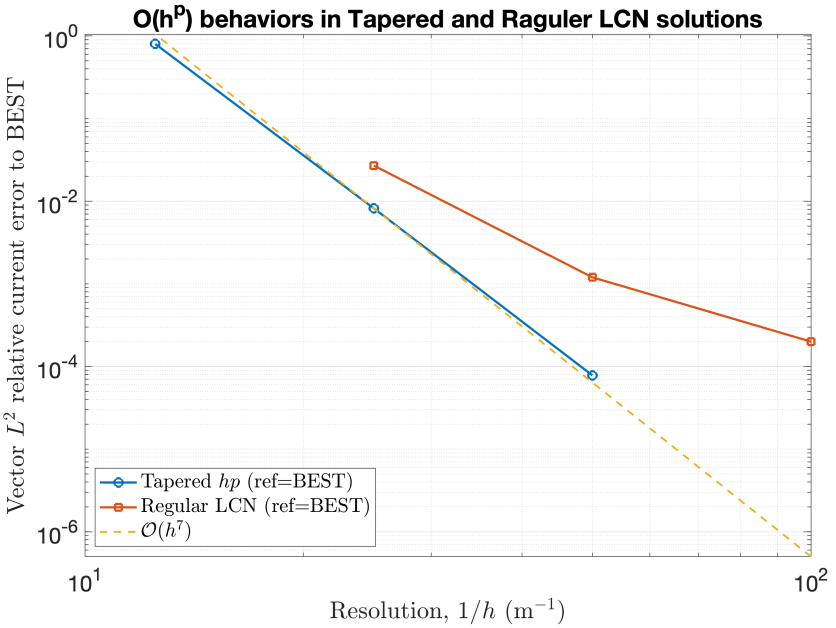


Figure 5.5: Mean surface-current error versus smooth-patch size h for adaptive hp LCN $(p_s, p_t, p_e) = (7, 5, 3)$ with relative h-refinements and for regular LCN (uniform $p = 7$). Errors are measured against the internal best solution $\mathbf{J}_{\text{ref}}^{LCN}$ (575,552 unknowns; tightened near radius, acceleration level, near-block quadrature, $\mathcal{ACA}$ tolerance, $\mathcal{H}$–**LU** arithmetic tolerance, and preconditioner), evaluated on the same quadrature nodes.

5.5.2 PEC Cube (CFIE)

We next consider a canonical closed PEC cube of side length $L = 2\,\text{m}$, corresponding to approximately 4λ at 600 MHz and 8λ at 1.2 GHz. The problem is formulated with the combined-field integral equation (CFIE) using a fixed mixing parameter to suppress interior resonances.

The mesh refinement strategy was applied to the PEC Cube obtained by setting the refinement ratios $(\rho_t,\rho_e) = (2,4)$ for each region. Then, the hp-refinement strategy is adopted with orders $(p_\text{s},p_\text{t},p_\text{e})=(11,8,5)$, corresponding to smooth faces, one–two transition layers, and edge-adjacent elements, respectively. This tapering ensures gradual reduction of both

element size and polynomial order toward singular edges. For comparison, we also consider a *regular* LCN discretization in which order $p = p_s = 11$ is applied uniformly across all patches with a single mesh size. In both cases, $\mathcal{ACA}$ tolerance, admissibility, and $\mathcal{H}$-matrix settings are fixed so that only the discretization strategy changes.

Accuracy is quantified by global L^2 surface-current error, while edge singularity recovery is assessed through line cuts of $|\mathbf{J}|$ normal to representative edges.

Fig. 5.6 shows the classification of cube patches into smooth, transition, and edge regions. At 600 MHz, the adaptive setting (Fig. 5.7a) suppresses oscillations near corners that appear in the regular uniform-order case (Fig. 5.7b). This demonstrates the effectiveness of order tapering in capturing edge singularities properly while reducing cost. Fig. 5.8 illustrates a large-scale adaptive run at a higher frequency of 1.2 GHz with $N \approx 5.24 \times 10^5$ unknowns, confirming that the proposed configuration remains stable and scalable in the multi-wavelength regime.

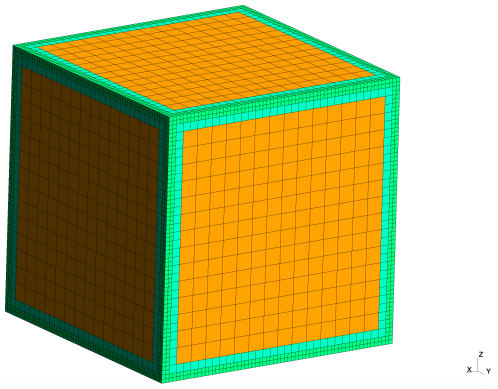
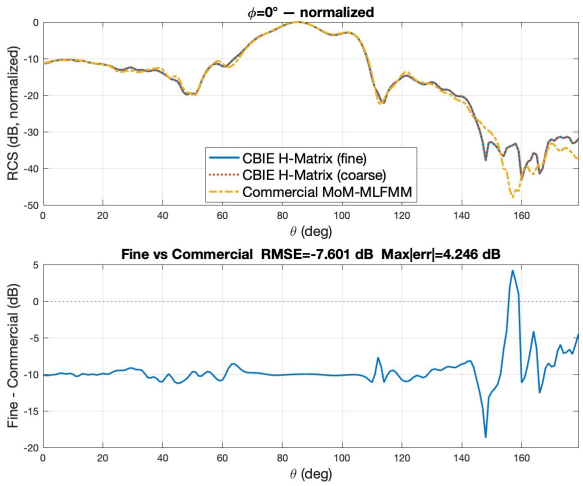
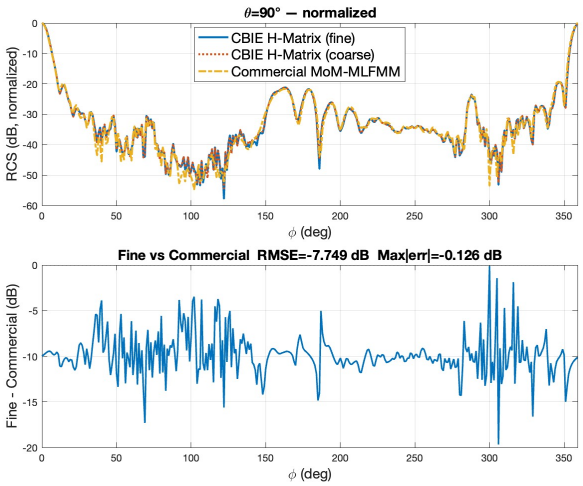


Figure 5.6: Cube mesh classification (2 m edge length): edge-adjacent elements (green), transition layers (blue), and smooth-region elements (orange).



(a) Adaptive LCN at 600 MHz: $(p_s, p_t, p_e) = (11, 8, 5)$.



(b) Regular LCN at 600 MHz: $p = 11$ everywhere.

Figure 5.7: CFIE surface-current magnitude on the 2 m cube at 600 MHz. Lower edge order with transition layers mitigates corner oscillations compared to the uniform-order discretization.

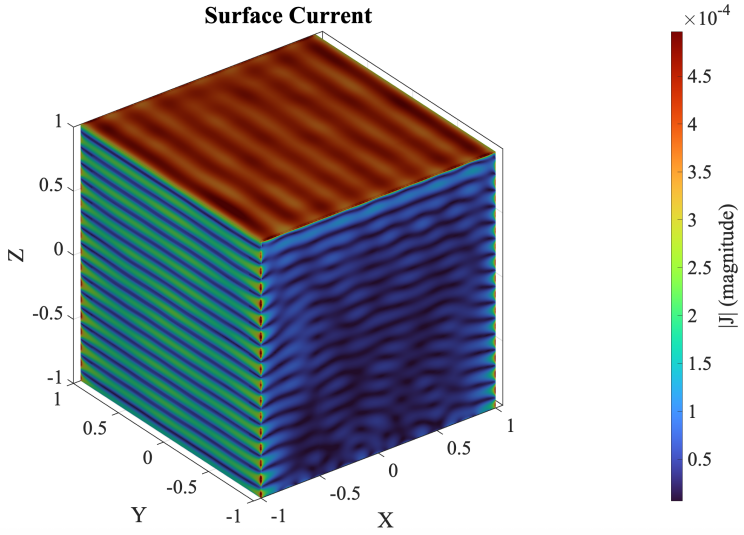


Figure 5.8: Adaptive *hp*-refined LCN at 1.2 GHz ($\approx 8\lambda$) with $N \approx 5.24 \times 10^5$.

5.5.3 36λ B2 Geometry (MFIE)

We next consider a B2-like CAD surface, derived from a NURBS model, characterized by multiple sharp leading/trailing edges and junctions. This geometry provides a practical test for *hp*-refined discretizations due to the combination of large smooth fuselage and wing patchs with sharp edges such as wing tips, chines, and trailing edges. The problem is solved with the MFIE.

Figure 5.9 shows the resulting adaptive *hp*-refined mesh, resulted by setting the refinement ratios (ρ_t,ρ_e) = (4,8) for each region. To illustrate the effect of polynomial adaption, Fig. 5.10 presents the computed MFIE surface current distributions at 600 MHz for two representative configurations: (a) $(p_{\mathrm{s}}, p_{\mathrm{t}}, p_{\mathrm{e}}) = (7, 5, 3)$, and (b) $p = p_s = 7$ in entire grid. The higher edge order in Fig. 5.10(b) yields noticeably sharper resolution of the current near geometric features, while Fig. 5.10(a) demonstrates the economy of using lower edge orders with transition layers still providing smooth blending.

In this example, filling the near-interaction matrix with adaptive *hp*-refinement (N = 257,118 unknowns) required approximately one hour twenty minutes using MPI parallelism on 192 processors, whereas the same operation on a regular mesh (N = 108,535 unknowns) took about 2.5 hours. The longer runtime for the regular mesh arises because the use of higher polynomial orders in edge elements makes it more difficult for the Adaptive Integration Method (AIM) in near interaction calculations to achieve the prescribed error tolerance, requiring additional integration levels to maintain accuracy in solution. The adaptive refinement also affects the condition number of the system matrix and, consequently, the efficiency of the preconditioner, roughly doubling the number of iterations needed to reach the same accuracy.

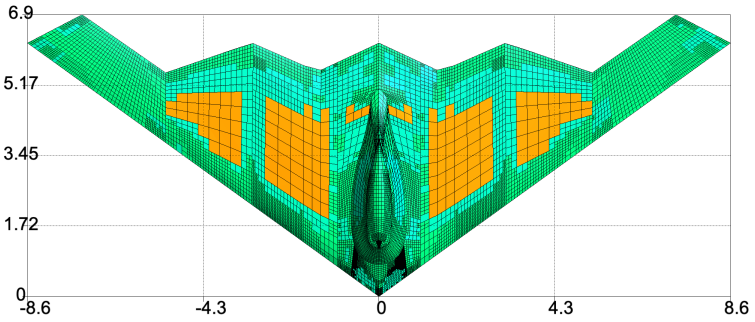
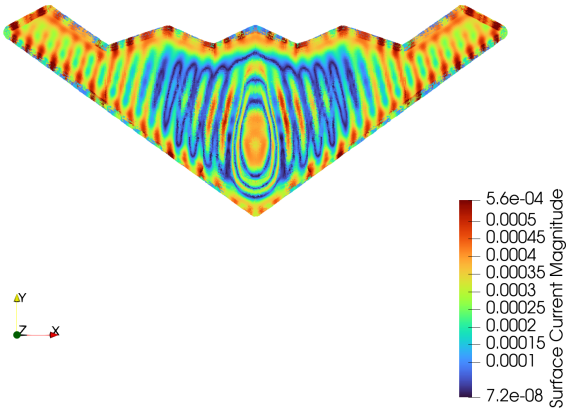
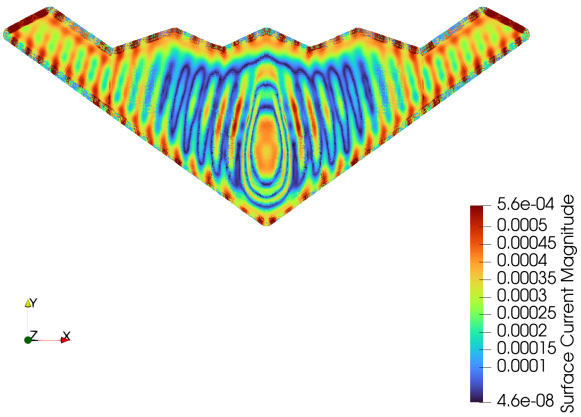


Figure 5.9: B2 adaptive *hp*-refined mesh: edge elements (green), transition patches (blue), and smooth interior elements (orange).



(a) Adaptive *hp*-refinement with $p_{\mathrm{s}} = 7$, $p_{\mathrm{t}} = 5$, $p_{\mathrm{e}} = 3$.



(b) Regular mesh configuration with $p = p_s = 7$.

Figure 5.10: 36λ B2 (MFIE): surface-current magnitude.

5.5.4 Missile Geometry (MFIE)

We next consider a missile-like CAD surface characterized by a long forebody, a blended mid-section, and thin trailing edges. This configuration is well suited to adaptive *hp* refinement because it combines large smooth patches with sharp geometric features that induce localized field singularities. The problem is solved with the MFIE for two electrical sizes: *9*λ and *18*λ.

To capture the singular behavior near the trailing/leading edges, edge neighborhoods are *h*-refined with the refinement ratios $(\rho_t,\rho_e) = (2,4)$ for each region, and assigned to

polynomial orders $(p_s, p_t, p_e) = (5, 3, 2)$. Figure 5.11 shows the resulting adaptive *hp*-refined quadrilateral mesh, with edge elements highlighted in green, transition patches in blue, and smooth interior elements in orange.

Figure 5.12 presents the MFIE surface current magnitude for both sizes on the same quadrilateral mesh topology. The adaptive pattern concentrates degrees of freedom where needed, near tips and trailing edges, while using higher-order basis functions on the smooth body to keep the overall system size moderate. For the 9λ case, the adaptive *hp*-refinement resulted in $N = 400{,}636$ unknowns; for the 18λ case, it results in $N = 626{,}186$.

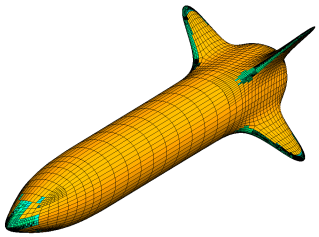
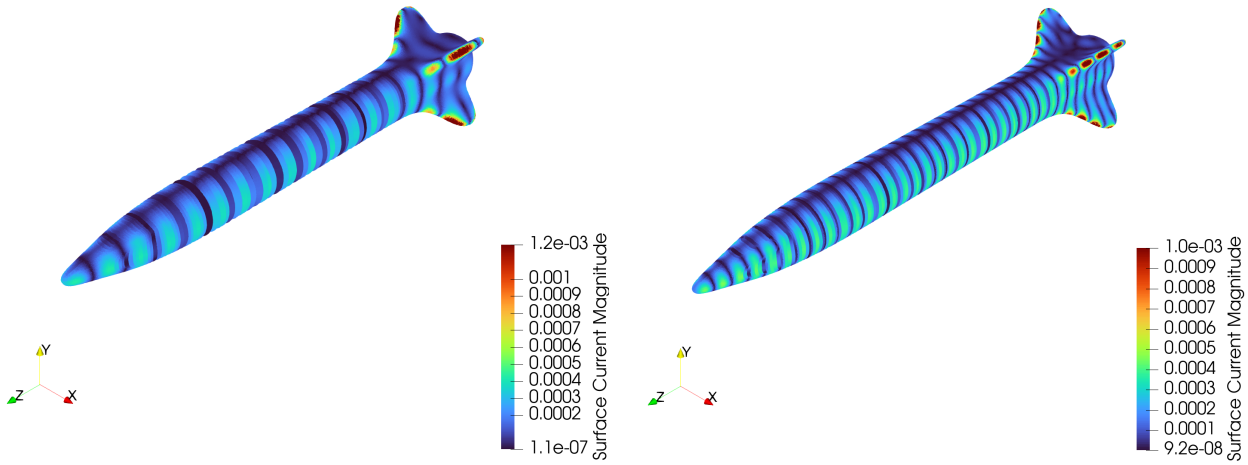


Figure 5.11: Adaptive *hp*-refined quadrilateral mesh for the missile geometry: edge neighborhoods ($p_e = 2$, green), transition layers ($p_t = 3$, blue), and smooth patches ($p_s = 5$, orange).

5.5.5 Performance Summary

Table 5.1 reports memory per degree of freedom and the solver runtime breakdown. The results in Fig. 5.5 (disk), Fig. 6.13 (NASA Almond RCS), and Fig. 6.15 (B2 aircraft RCS) show that the adaptive *hp*-refinement strategy maintains solution accuracy while reducing computational time and memory compared to a regular discretization.



(a) 9λ case ($N = 400{,}636$). (b) 18λ case ($N = 626{,}186$).

Figure 5.12: MFIE surface-current magnitude for Missile in two different electric sizes. The *hp*-refinement strategy maintains high fidelity near sharp features while controlling the global DOF count as electrical size grows from 9λ to 18λ.

Table 5.1: Complexity summary (peak memory and timings).

Case	**Formulation**	**Config.**	N	**Peak Mem.** (GB)	$\mathcal{H}$–**LU** (m)	**Solve** (m)
Disk	MFIE	Adaptive	575,552	149.04	40.76	205.36
		Regular	156,800	44.94	13.00	112.83
Cube	CFIE	Adaptive	524,000	135.69	37.11	186.97
		Regular	92,928	26.64	7.71	66.87
B2	CFIE	Adaptive	494,236	127.98	35.00	176.35
		Regular	217,070	62.22	18.00	156.20
Missile (9λ)	MFIE	Adaptive	400,636	115.08	39.13	188.22
Missile (18λ)	MFIE	Adaptive	626,186	180.55	42.24	235.35

5.6 Conclusion

An *hp*-refinement strategy with edge/transition/smooth patches for point-based high-order integral equation solvers focused on accurate surface-current recovery on sharp-edged PEC geometries has been presented. By combining edge-focused *h*-refinement, reduced edge order,

and one–two regions of transitions, the method tries to restore the nominal $\mathcal{O}(h^p)$ behavior of high-order schemes in smooth regions while avoiding order-mismatch artifacts at refinement interfaces. A heterogeneous $\mathcal{H}$-matrix configuration, with tighter tolerance for operator assembly and looser tolerances on preconditioner leaf/non-leaf blocks reduces preconditioner cost and accelerates $\mathcal{H}$–**LU** without degrading current accuracy.

Results for an infinitesimal disk (MFIE, analytic current), a cube (CFIE), a B2-like aircraft geometry (CFIE), and a missile (MFIE) shows that *hp*-refinement strategy achieves high-order accuracy with substantially reduced runtime/memory relative to uniform high-order meshes. The approach integrates naturally with hierarchical matrix assembly and $\mathcal{H}$–**LU** preconditioning on modern HPC systems (e.g., Trillium with 192 MPI ranks for near interactions and limited OpenMP for $\mathcal{H}$-build/$\mathcal{H}$–**LU**).

Future work includes fully automated choice of p_{e} based on local indicators, and transition placement driven by geometry-aware heuristics, aiming toward accuracy-controlled high-order boundary-integral simulations.

Bibliography

[1] Digital Research Alliance of Canada, "Trillium supercomputer." `https://docs.alliancecan.ca/wiki/Trillium`, 2025. Accessed: 2025-08-24.

[2] A. J. Poggio and E. K. Miller, "Integral equation solutions of three-dimensional scattering problems," in *Computer Techniques for Electromagnetics* (R. Mittra, ed.), pp. 159–264, Oxford, UK: Pergamon Press, 1973.

[3] J. R. Mautz and R. F. Harrington, "H-field, e-field and combined-field solutions for conducting bodies of revolution," *Archiv für Elektronik und Übertragungstechnik (AEÜ)*, vol. 32, pp. 157–164, Apr. 1978.

[4] J. R. Mautz and R. F. Harrington, "A combined-field integral equation for electromagnetic scattering from homogeneous bodies," *Archiv für Elektronik und Übertragungstechnik (AEÜ)*, vol. 33, no. 4, pp. 159–164, 1979.

[5] S. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, vol. 30, pp. 409–418, May 1982.

[6] L. Canino *et al.*, "Numerical solution of the helmholtz equation in 2d and 3d using a high-order nyström discretization," *J. Comput. Phys.*, vol. 146, pp. 627–663, Nov. 1998.

[7] P. Y. Ufimtsev, *Method of Edge Waves in the Physical Theory of Diffraction*. Moscow, USSR: Soviet Radio, 1962.

[8] R. Coifman, V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: A pedestrian prescription," *IEEE Antennas Propag. Mag.*, vol. 35, pp. 7–12, Jun. 1993.

[9] N. Engheta, W. D. Murphy, V. Rokhlin, and M. S. Vassiliou, "The fast multipole method (fmm) for electromagnetic scattering problems," *IEEE Trans. Antennas Propag.*, vol. 40, pp. 634–641, Jun. 1992.

[10] E. Bleszynski, M. Bleszynski, and T. Jaroszewicz, "Aim: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems," *Radio Sci.*, vol. 31, pp. 1225–1251, Sep. 1996.

[11] W. Hackbusch, "A sparse matrix arithmetic based on $\mathcal{H}$-matrices. part i: Introduction to $\mathcal{H}$-matrices," *Computing*, vol. 62, no. 2, pp. 89–108, 1999.

[12] A. F. Peterson and B. Klock, "An improved nyström method for electromagnetic scattering," *IEEE Trans. Antennas Propag.*, vol. 46, pp. 171–175, Feb. 1998.

[13] I. Jeffrey, J. Aronsson, M. Shafieipour, and V. Okhmatovski, "Error controllable solutions of large-scale problems in electromagnetics: Mlfma-accelerated locally corrected nyström solutions of cfie in 3d," *IEEE Antennas Propag. Mag.*, vol. 55, pp. 294–308, Jun. 2013.

[14] J. Hu, E. Sever, O. Babazadeh, *et al.*, "H-matrix accelerated direct matrix solver using chebyshev-based nyström boundary integral equation method," *IEEE Open J. Antennas Propag.*, vol. 6, pp. 171–180, Feb. 2025.

[15] R. D. Graglia, A. F. Peterson, and M. Boella, *Higher-Order Techniques in Computational Electromagnetics*. Raleigh, NC, USA: SciTech Publishing, 2016.

[16] H. Contapanagos, B. Dembart, and E. J. Moravec, "Well-conditioned boundary integral equations for three-dimensional electromagnetics," *IEEE Trans. Antennas Propag.*, vol. 50, pp. 1824–1830, Dec. 2002.

[17] M. Shafieipour, I. Jeffrey, J. Aronsson, and V. I. Okhmatovski, "On the equivalence of rwg method of moments and the locally corrected nyström method for solving the electric field integral equation," *IEEE Transactions on Antennas and Propagation*, vol. 62, pp. 772–782, Feb. 2014.

[18] S. D. Gedney and J. C. Young, "The locally corrected Nyström method for electromagnetics," in *Computational Electromagnetics: Recent Advances and Engineering Applications* (R. Mittra, ed.), pp. 149–198, New York, NY: Springer, 2014.

[19] R. Kriemann, *HLIBpro: C Language Interface*. Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany, 2008.

[20] O. Babazadeh and et al., "Optimal preconditioners for hybrid direct–iterative h-matrix solvers in boundary element methods," *IEEE Journal on Multiscale and Multiphysics Computational Techniques*, 2025. to appear.

[21] J. Meixner, "The behavior of electromagnetic fields at edges," *IEEE Transactions on Antennas and Propagation*, vol. AP-20, pp. 442–446, July 1972.

[22] R. A. Hurd, "The edge condition in electromagnetics," *IEEE Transactions on Antennas and Propagation*, vol. AP-24, pp. 70–73, Jan. 1976.

[23] J. G. V. Bladel, *Electromagnetic Fields*. Hoboken, NJ: IEEE Press / Wiley, 2 ed., 2007. See Sec. 5.2, *Field Behavior at a Conducting Wedge*.

[24] M. Idemen, "Confluent edge conditions for the electromagnetic wave at material edges," *Wave Motion*, vol. 31, no. 1, pp. 57–66, 2000.

[25] W. C. Chew, *Waves and Fields in Inhomogeneous Media*. Piscataway, NJ: IEEE Press, 1990.

[26] R. D. Graglia, D. R. Wilton, and A. F. Peterson, "Higher order interpolatory vector bases for computational electromagnetics," *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 3, pp. 329–342, 1997.

[27] J. Meixner, "The behavior of electromagnetic fields at edges," *IEEE Transactions on Antennas and Propagation*, vol. 20, no. 4, pp. 442–446, 1949. Originally in German: *Z. Angew. Phys.*, vol. 3, 1949.

Chapter 6

A High-Order Fast Boundary Element Method as a Benchmark Computational Framework for RCS Analysis of PEC Targets

This chapter introduces an advanced boundary element benchmark framework tailored for rapid and precise Radar Cross Section (RCS) predictions of electrically large Perfect Electric Conductor (PEC) structures. At its heart lies a Chebyshev-based Boundary Integral Equation (CBIE) discretization applied to the Magnetic Field Integral Equation (MFIE), which achieves spectral-like convergence rates of $\mathcal{O}(h^p)$ even on coarse meshes (Research Question 1). To avoid the high memory and computational cost of dense system matrices, we employ hierarchical $\mathcal{H}$-matrix compression to approximate far-field interactions with low-rank blocks while retaining full precision for near-field terms (Research Question 2). This compressed

representation is coupled with a hybrid MPI/OpenMP parallel implementation, distributing both matrix assembly and solution phases across multicore and multi-node architectures.

Comprehensive numerical studies including canonical spheres, complex aircraft geometry, and multi-scale industrial components, demonstrate that our solver scales to the millions of unknowns with modest memory requirements. Finally, Benchmark comparisons against commercial tools confirm not only superior convergence properties but also reductions in time-to-solution of up to 60% on large problems (Research Question 3).

This work is submitted as a journal paper for publication in the *IEEE Open Journal of Antennas and Propagation (OJAP)* and currently is under review. The remainder of this chapter presents the paper.

Abstract

A fast and accurate boundary element method framework for benchmarking the Radar Cross Section (RCS) computations for various Perfect Electric Conductor (PEC) targets, including sharp-edge models with electrically large sizes, is proposed. The approach employs a high-order Chebyshev-based Boundary Integral Equation (CBIE) formulation of the Magnetic Field Integral Equation (MFIE). Acceleration is achieved using both the $\mathcal{H}$-matrix framework and distributed parallelism for filling inadmissible blocks, substantially improving computational efficiency in both time and memory usage. We refer to the resulting solver as $\mathcal{H}$-CBIE and demonstrate its $O(h^p)$ error convergence, scalability, and accuracy through extensive numerical results, h being the mesh element size and p the order of discretization. Localized hp-refinement is incorporated for sharp-edge problems to maintain accuracy near edges and junctions. We include a comprehensive complexity analysis of the high-order accelerated framework, detailing the error-dependent wall time scaling for each stage of the solution process. The resulting scaling dependencies show that the total solution wall time grows quadratically in N with the target accuracy when high-order discretizations are used, whereas it rises exponentially for traditional low-order methods, underscoring the substantial efficiency gains of the proposed approach and its suitability for benchmarking.

6.1 Introduction

Computing the Radar Cross-Section (RCS) of various targets has been one of the primary application areas for Computational Electromagnetics (CEM) since its inception in the 1950s [1, 2, 3] and 1960s [4, 5, 6]. Integral equations (IEs) discretized with Boundary Element Methods (BEMs) [7, 8] have been the methods of choice as they provide accurate, physics-based evaluations of the RCS while requiring only boundary discretization. These traits made BEMs most suitable for electrically large targets. Approximate asymptotic methods such as Physical Theory of Diffraction (PTD) [9], Geometric Theory of Diffraction (GTD) [10], and the Uniform Theory of Diffraction (UTD) [11] enable RCS solutions for even larger targets but often lack accuracy in capturing low-level RCS side lobes. The development of 3D fast algorithms like the Fast Multipole Method (FMM) [12, 13] and the Adaptive Integral Method (AIM) [14] significantly bridged this gap, enabling rigorous RCS computations for targets spanning thousands of wavelengths [15]. Various BEM discretization schemes have been developed over the decades (e.g. RWG MoM [7], LCN [8], HO MoM [16], etc. [17]), which can be used for discretization of various integral equation formulations suitable for RCS computations (e.g. CFIE, Calderón Preconditioned EFIE [18], potential IEs [19, 20, 21], etc.[22]). As different approaches produce differently conditioned dense matrix equations, requiring custom tailored acceleration and preconditioning strategies, the question arises as to what choices lead to the best computational framework for solving a particular class of RCS problems. Identifying the benefits of various approaches necessitates benchmarking these different computational strategies in terms of accuracy, efficiency, and scalability on problems of interest.

Benchmarking of RCS solvers has a long history. One of the earliest systematic efforts was led by Volakis and collaborators, who established canonical test cases such as the NASA

almond, ogive, and cone–sphere for solver validation [23]. Subsequent works extended this paradigm through hybrid finite element–integral formulations, massively parallel benchmark projects, and detailed performance analyses of fast multipole accelerated RCS computations [24, 25]. These contributions provided a foundation for reproducible comparison of solver accuracy and scalability across the CEM community.

Over the past three decades, this benchmarking tradition has evolved from Volakis' original canonical RCS validation problems of the 1990s into a globally coordinated effort encompassing analytical, numerical, and experimental benchmarks. A comprehensive retrospective of exact analytical validations was recently provided in [26], which reviewed all separable geometries admitting closed-form or series-expansion solutions suitable for code verification. Large-scale, multi-institutional benchmarking frameworks have emerged in parallel: the long-running JINA/ISAE workshop in France [27], the European Defence Agency (EDA) benchmarking series [28], and the U.S. Electromagnetic Code Consortium (EMCC) [29], each contributing decades of reproducible datasets and lessons learned. The Austin RCS Benchmark Suite [31, 30] and its extensions represent the next generation of reproducible, open-access RCS problems enabling quantitative performance evaluation of solvers across algorithms and architectures. Beyond scattering, similar benchmarking philosophies have been extended to multiphysics simulations, including 3D finite-element particle-in-cell (FEMPIC) solvers [32] and to signal/power integrity (SI/PI) modeling of electronic packages [33]. Collectively, these works form the modern prior art in electromagnetic validation, reproducibility, and performance benchmarking, from Volakis' early canonical targets to today's IEEE benchmarking initiatives. The present work builds upon this foundation by introducing a high-order, $\mathcal{H}$-matrix accelerated CBIE/MFIE formulation designed to generate reference-quality RCS benchmarks for complex PEC targets with sharp-edge geometries

and electrically large dimensions.

Formally, the act of producing a benchmark for a particular CEM solver means applying it to solve a problem, comparing its result to a reference solution, and quantifying the computational resources (CPU time and cores, memory consumption) required to produce the resulting level of accuracy. Such a benchmark is used to comparatively assess the performance of innovations in CEM techniques applied to the same problem. We may seek to tune the new technique to achieve the same error level and compare the computational resources required, or we may seek to use the same computational resources, and compare the resulting accuracies. This competition, through benchmarks, drives innovation, pushing computational tools toward greater precision, and efficiency.

As CEM solvers continuously evolve, their capabilities have advanced to the point where benchmarks are measured for complicated geometries that do not admit analytic solutions. A key element in producing benchmarks for such geometries is the ability to produce a reference solution that is accurate to a prescribed error level beyond what is expected of other solvers. A framework for producing such reference solutions has emerged in the form of high-order BEMs such as LCN [8], CBIE [17], and HO-MoM [16], combined with error-controllable acceleration of the solver using fast algorithms such as FMM, AIM or the $\mathcal{H}$-matrix framework [12, 13, 14, 34]. When combined with efficient preconditioner strategies [35] these techniques can deliver large-scale solutions with very low errors (far below the errors induced by manufacturing tolerances) at a reasonable wall time, making them ideal for the purpose of generating reference solutions in benchmarking datasets. Explicitly, this approach of using point-based, high-order methods, combined with fast algorithms and effective preconditioners is developed under the higher-order imperative and has been enshrined as *Rokhlin's Dogma*[36, 37].

The fact that these solvers are high-order in both their current and geometric representations means that they are capable of delivering highly accurate solutions far more efficiently than their low-order counterparts. This not only makes the methods suitable for generating reference solutions, but, for sufficiently high accuracy requirements (or sufficiently large problems), their performance becomes the benchmark to beat.

For example, RCS computations currently rely primarily on low-order RWG-MoM solutions to the Combined Field Integral Equation (CFIE), accelerated via FMM or AIM [12, 13, 14]. These frameworks yield $O(h)$ error in computed currents and $O(h^2)$ error in RCS computations, where h is the characteristic element size[36]. As will be shown herein, such error behavior limits achievable accuracy because computational requirements grow exponentially with required accuracy. Hence, higher-order alternatives are required.

To address the need for robust and wall time-effective approaches for generating error-controllable reference data for RCS computations, this work demonstrates the capabilities of an $\mathcal{H}$-matrix preconditioned high-order iterative solver for PEC scattering problems. The numerical formulation is developed for the Chebyshev boundary formulation, applied to the MFIE.

Our benchmark framework tackles challenges such as matrix conditioning in resonant scenarios, adaptive *hp*-refinement, and large-scale parallel efficiency for smooth and sharp-edge PEC geometries. Issues with iterative convergence are mitigated through an error-controllable $\mathcal{H}$-matrix preconditioner [35], ensuring robust and accurate RCS computations that can serve as benchmark references even in challenging scattering scenarios. Moreover, we demonstrate that for high accuracy requirements, the proposed framework significantly outperforms RWG MoM, particularly when seeking four to five digits of precision, thus establishing a standard for benchmarking.

An important feature of this work is an extensive, stage-by-stage complexity analysis of the high-order accelerated framework, highlighting the error level-dependent wall time associated with near-field integration, $\mathcal{H}$-matrix assembly, and factorization. This detailed analysis enables practitioners to make informed decisions regarding solver configuration and accuracy targets, and further distinguishes the framework as a robust tool both for construction of the reference RCS solutions and setting performance standards.

6.2 A Need for High-Order Solvers

The presented RCS reference solution framework is grounded in the development of a computational electromagnetics solver that: 1) adheres to Rokhlin's dogma requiring "methods must be fast, high-order, well-conditioned and error-controllable" [12, 13], and 2) asserts Wandzura's principle that computational complexity estimates are meaningless without notion of accuracy [36, 37]. Together these ideas drive development of a fast, high-order, error-controllable solver where the wall time's relationship to the resulting error level is known.

To illustrate the importance of a high-order error-controllable framework, we now establish a relationship between the number of unknowns required to achieve a given level of accuracy by varying discretization parameters h and p. We begin by assuming that the desired error ϵ in solution of the pertinent integral equation scales as:

$$\epsilon = \mathcal{O}(h^p), \tag{6.1}$$

where h is the characteristic mesh size and p is the polynomial order of the expansion in each dimension for each mesh element. For simplicity we assume a uniform polynomial order, i.e.,

the same p on every element. Viewed jointly in h and p, the asymptotic estimate in (6.1) is an upper bound whose value decreases whenever p is raised or h is refined, both operations that inevitably increase the total number of unknowns N.

Assuming scattering from a 2-D surface embedded in 3-D space we have

$$N \propto \frac{p^2}{h^2}. \tag{6.2}$$

In order to reduce the error at a given discretization (h_1, p_1) we consider one of two options:

(i) h-refinement: decrease the element size from h_1 to h_2 at the fixed polynomial order p_1;

(ii) p-refinement: increase the polynomial order from p_1 to p_2 at the fixed element size h_1.

We assume an initial error level $\epsilon_1 = 10^{-d_1}$, obtained with discretization (h_1, p_1), and seek to achieve a target error $\epsilon_2 = 10^{-d_2}$.

In what follows, we derive the expressions for the number of unknowns required to achieve the desired error level for the two refinement cases.

Case 1 (*h-refinement*): We set $h_2 = \alpha h_1$ (with $0 < \alpha < 1$). From equation (6.2):

$$\frac{N_2}{N_1} \propto \frac{h_1^2 p_1^2}{h_2^2 p_1^2} = \frac{1}{\alpha^2}. \tag{6.3}$$

Defining $\Delta d = d_2 - d_1$, we have

$$\frac{\epsilon_2}{\epsilon_1} = \frac{10^{-d_2}}{10^{-d_1}} = 10^{-\Delta d} = \frac{\alpha^{p_1} h_1^{p_1}}{h_1^{p_1}} = \alpha^{p_1}. \tag{6.4}$$

It follows that

$$\alpha = 10^{-\Delta d/p_1}, \tag{6.5}$$

which upon substituting into (6.3) gives

$$\frac{N_2}{N_1} = 10^{2\Delta d/p_1}. \tag{6.6}$$

Thus, h-refinement leads to an exponential increase in the number of unknowns versus the increase in digits of precision Δd.

Case 2 (*p-refinement*): Set $p_2 = p_1 + \beta$. The ratio of unknowns is now

$$\frac{N_2}{N_1} \propto \frac{h_1^2(p_1+\beta)^2}{h_1^2 p_1^2} = \frac{(p_1+\beta)^2}{p_1^2}. \tag{6.7}$$

The ratio of errors gives

$$\frac{\epsilon_2}{\epsilon_1} = 10^{-\Delta d} = \frac{h_1^{p_1+\beta}}{h_1^{p_1}} = h_1^{\beta}. \tag{6.8}$$

For mathematical convenience we assume $h_1 = 10^{-\eta}$, such that $10^{-\Delta d} = 10^{-\eta\beta}$, and

$$\beta = \frac{\Delta d}{\eta}. \tag{6.9}$$

Hence

$$\frac{N_2}{N_1} \propto \frac{\left(p_1 + \frac{\Delta d}{\eta}\right)^2}{p_1^2} = 1 + \frac{2\Delta d}{p_1\eta} + \frac{(\Delta d)^2}{p_1^2\eta^2}. \tag{6.10}$$

Comparing (6.6) and (6.10), we see that while h-refinement incurs an exponential growth of unknowns with Δd, the increase in unknowns arising from p-refinement is only quadratic with Δd. This highlights the importance of high-order methods in benchmark studies. Numerical results will emphasize that a growth in problem size arising from h-refinement becomes computationally prohibitive for highly accurate solutions. Numerical results in Section 6.5 will demonstrate that the rapid degree of freedom growth predicted for the h-refinement

model (Eq. (6.6)) translates into a much steeper increase in wall time, whereas the p-refinement model (Eq. (6.10)) achieves higher accuracy with a far milder runtime growth, consistent with the lower number of unknowns required by the high-order solution (e.g. Fig. 6.6).

6.3 A High-Order Framework for Error Controlled RCS Benchmarks

In what follows, we attempt to provide a concise overview of the synthesis of the MFIE formulations using CBIE boundary element method. $\mathcal{H}$-matrix acceleration is also applied to get an error-controllable RCS benchmark framework. Although our aim has been to make this presentation complete for understanding, the reader is referred to [34, 38, 35, 17] for additional details.

6.3.1 MFIE formulation of the scattering problem

The time-harmonic Magnetic Field Integral Equation (MFIE) relates the unknown electric surface current density $\mathbf{J}^e$ on a perfectly electrically conducting (PEC) surface S to the incident magnetic field $\mathbf{H}^{inc}$.

By virtue of vanishing tangential electric field on a PEC, $\hat{\mathbf{n}}(\mathbf{r}) \times \mathbf{E}(\mathbf{r}) = 0$ for $\mathbf{r} \in S$, the equivalent magnetic current $\mathbf{J}^m = \mathbf{E} \times \hat{\mathbf{n}}$ is zero. From the equivalence principle, the magnetic field in the exterior region $\mathbf{r} \notin V$ can be written as

$$\mathbf{H}(\mathbf{r}) = \int_S \nabla \times \big(\mathbf{J}^e(\mathbf{r}')\, G_0(\mathbf{r}, \mathbf{r}')\big)\, \mathrm{d}s' \; + \; \mathbf{H}^{\mathrm{inc}}(\mathbf{r}), \qquad \mathbf{r} \notin V, \tag{6.11}$$

where $G_0(\mathbf{r},\mathbf{r}') = \dfrac{e^{-ik_0|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|}$ and $\mathbf{J}^e$ is the equivalent electric surface current.

Choosing the observation surface S' to coincide with the current-carrying surface S yields the MFIE in second-kind form.

$$-\frac{\mathbf{J}^e(\mathbf{r})}{2} + \hat{\mathbf{n}}(\mathbf{r}) \times P.V. \int_S \nabla \times \Big(\mathbf{J}^e(\mathbf{r}')\, G_0(\mathbf{r},\mathbf{r}')\Big)\, ds' = -\hat{\mathbf{n}}(\mathbf{r}) \times \mathbf{H}^{\text{inc}}(\mathbf{r}), \qquad \mathbf{r} \in S. \quad (6.12)$$

where $\hat{\mathbf{n}}(\mathbf{r})$ denotes the outward unit normal to S at $\mathbf{r}$. Eq. (6.12) can be discretized using point-based methods such as the Chebyshev-based boundary integral equation (CBIE) method and represented as a system of linear algebraic equations (SLAE) [17]. After solving the SLAE for the surface current density on the PEC object, the RCS can be computed as:

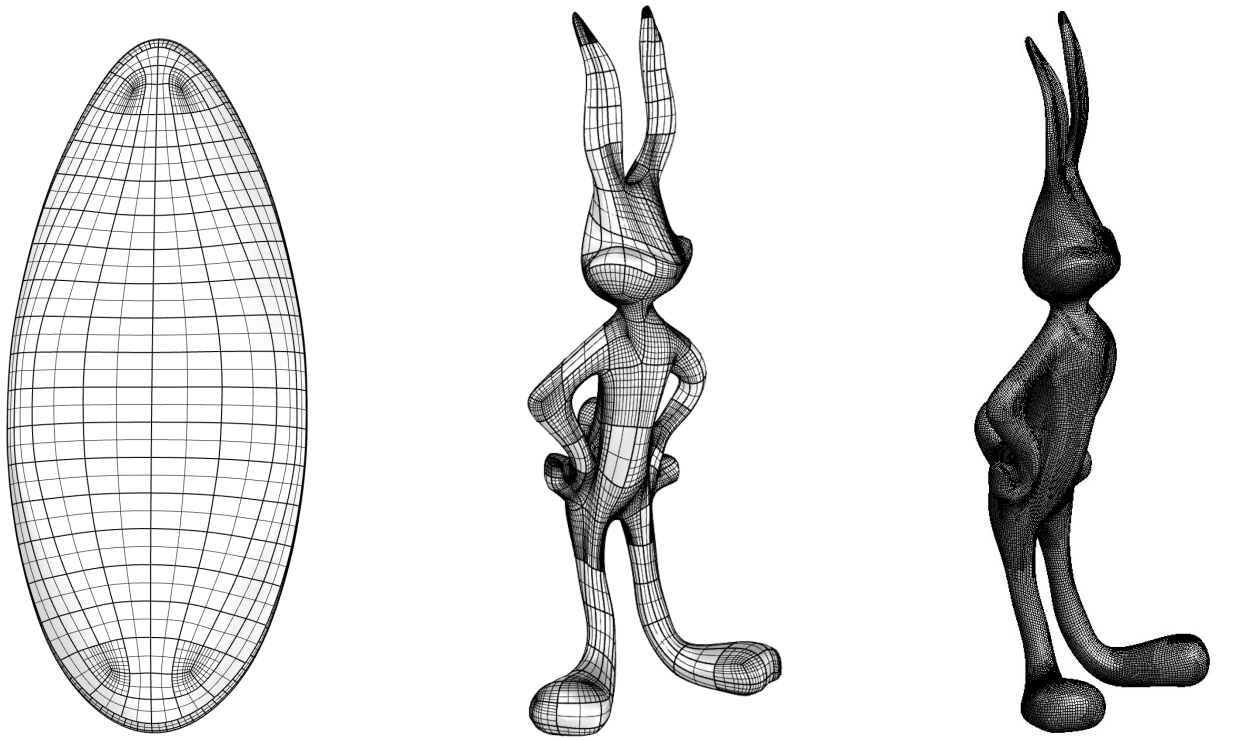
$$\sigma_{\text{RCS}}(\hat{\mathbf{r}}) = \lim_{R\to\infty} 4\pi R^2 \frac{\|\mathbf{E}^{\text{scat}}(R\hat{\mathbf{r}})\|^2}{\|\mathbf{E}^{inc}\|^2}. \quad (6.13)$$

In this expression, σ_{RCS} quantifies the scattered power per unit solid angle normalized by the incident power density, where R is the observation distance from the scatterer and the limit $R \to \infty$ enforces the far-field approximation. In this limit, $\mathbf{E}^{\text{scat}}(R\hat{\mathbf{r}})$ denotes the scattered electric field observed in the direction $\hat{\mathbf{r}}$. $\|\mathbf{E}^{inc}\|$ is a plane-wave incident field. This is the standard definition used to characterize electromagnetic scattering and to compute the RCS of complex PEC geometries.

6.3.2 Geometry and surface current representation with NURBS and Bézier quadrilaterals

A high-order solution framework benefits from high-order geometric models capable of efficiently representing complicated geometries. We use Non-Uniform Rational B-Splines (NURBS) for geometry descriptions. NURBS are mathematical constructs used for describing arbitrary shapes [39]. Their adaptability and precision make them useful tools across a variety of disciplines, including illustration, animation, and manufacturing. In the context of electromagnetic scattering targets, NURBS provide an accurate representation of the model that can be discretized into quadrilateral Bézier patches. While NURBS surfaces offer more general surface descriptions than Bézier surfaces, the latter are preferable for CEM applications due to their size h being tailored to desired electrical mesh size as well as anticipated $O(h^p)$ error according to approximation order p for the current. Thus, general NURBS surfaces are tessellated (broken into smaller surfaces) until the smaller surfaces can be accurately represented by a quadrilateral Bézier surface. This tessellation ultimately produces a mesh of Bézier elements. Quadrilateral Bézier patches are commonly chosen because they offer computational efficiency and a straightforward design. It should be noted that converting NURBS to Bézier patches may cause a loss of geometric accuracy, particularly in regions with more curvature or complex features on the NURBS surface. Hence, careful consideration and refinement might be necessary during the tessellation process to achieve a necessary level of geometric accuracy.

In Fig. 6.1a and Fig. 6.1b, two complex geometries considered for RCS computations are shown. The models, referred to as the Almond and the Bunny, have been modeled using NURBS in Rhino [39]. Bézier quadrilateral patches are ultimately used to apply an adaptive integration strategy and solve the MFIE, as depicted in Fig. 6.1c.



(a) Almond (b) Bunny (c) Bunny Bézier patches

Figure 6.1: NURBS representations and Bézier patching: (a) Almond, (b) Bunny, and (c) Bunny Bézier quadrilateral patches.

6.3.3 The Chebyshev-based Boundary Integral Equation

Once the Bézier patches are defined, they provide a smooth parametric mapping for the unknown surface current density $\mathbf{J}^e$ on PEC geometries. To discretize the MFIE, we use a Chebyshev-based boundary integral equation (CBIE) method [17], which employs Chebyshev polynomials $T_n(x)$ and Gauss–Chebyshev quadrature for high-order accuracy and fast convergence. On each Bézier patch, parameterized by $\xi, \eta \in [-1, 1]$, the current is expanded as

$$\mathbf{J}^e(\mathbf{r}) = J_1(\mathbf{r})\,\frac{\mathbf{a}_1(\mathbf{r})}{\mathcal{J}(\mathbf{r})} + J_2(\mathbf{r})\,\frac{\mathbf{a}_2(\mathbf{r})}{\mathcal{J}(\mathbf{r})}, \tag{6.14}$$

where $\mathbf{a}_1 = \partial\mathbf{r}/\partial\xi$ and $\mathbf{a}_2 = \partial\mathbf{r}/\partial\eta$ are the covariant tangent vectors on the patch, and $\mathcal{J}(\mathbf{r})$ is the Jacobian determinant that corrects the area transformation in numerical integration [40, 17]. Fig. 6.2 illustrates this local mapping.

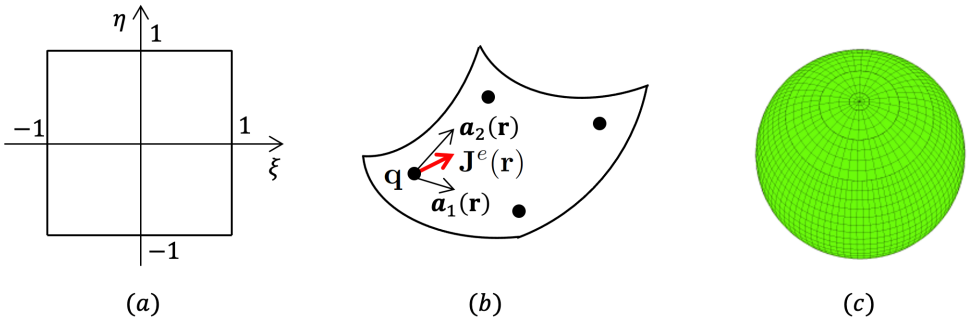


Figure 6.2: (a) Reference square in (ξ, η)–plane; (b) Bézier patch with covariant basis vectors $\mathbf{a}_1$, $\mathbf{a}_2$ at quadrature point $\mathbf{q}$; (c) Bézier mesh on the sphere.

On each patch, we construct a local transformation system that enforces error-controlled integration of the MFIE kernel against all two-dimensional polynomial basis functions up to order p [14]. The right-hand side of this system collects the contributions from singular or nearly singular integrals over the patch surface S_e, which are evaluated accurately via suitable change-of-variables techniques. Solving these local correction systems for near interaction matrix elements Z^{near} yields weights that are then inserted into the global matrix equation

$$\left[\begin{bmatrix} \mathbf{Z}_{1,1}^{\text{far}} & \mathbf{Z}_{1,2}^{\text{far}} \\ \mathbf{Z}_{2,1}^{\text{far}} & \mathbf{Z}_{2,2}^{\text{far}} \end{bmatrix} + \begin{bmatrix} \mathbf{Z}_{1,1}^{\text{near}} & \mathbf{Z}_{1,2}^{\text{near}} \\ \mathbf{Z}_{2,1}^{\text{near}} & \mathbf{Z}_{2,2}^{\text{near}} \end{bmatrix}\right] \cdot \begin{bmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \end{bmatrix}, \tag{6.15}$$

with the impedance matrix $\mathbf{Z}_{\text{CBIE}}$ decomposed into far and near interactions blocks. The near-field blocks represent self- and near interactions including singular and hypersingular kernels that require specialized mitigation (e.g., a global change of variables in the CBIE formulation) to maintain accuracy, whereas the far-field blocks approximate the remaining

interactions with regular quadrature rules.

6.3.4 Near-field interactions

We evaluate each near-interaction integral on successively refined sub-patches until the local interpolation error drops below the target ϵ. If ℓ denotes the refinement level, the patch area shrinks approximately by a factor $4^{-\ell}$ (each refinement step quarters the patch), so the work per element accumulates as a geometric series. With $N_e = N/p^2$ element pairs in the adaptive integration, the integration depth L is:

$$\begin{aligned} L(d,p) &= \min\Big(\max\big(K(d,p),\, L_{\min}\big),\, L_{\max}\Big), \\ K(d,p) &= \left\lceil \frac{d + \log_{10}(C\,h^q)}{q\,\log_{10} 4} \right\rceil + \lceil \log_2 p \rceil - 1, \end{aligned} \tag{6.16}$$

where it is bounded by $L_{\min} = 4$ and $L_{\max} = 10$. Here, q is the quadrature rule order in the adaptive integration. The term $\lceil (d + \log_{10}(C\,h^q))/(q \cdot \log_{10} 4) \rceil$ implies the standard quadtree refinement error bound

$$C\left(\tfrac{h}{4^L}\right)^q \le 10^{-d},$$

so that d decimal digits of accuracy are achieved. The additional $\lceil \log_2 p \rceil - 1$ term captures extra oscillations from p-refinement. Clamping levels of adaptive integration by $L_{\min}$ and $L_{\max}$ prevent premature termination and runaway refinement, respectively.

This model prevents excess refinement for moderate tolerances ($d \le 6$) while still allowing deeper splits under tight tolerances.

Table 6.1: Asymptotic growth of near interactions T_{near} and $\mathcal{H}$-matrix arithmetic $T_{\mathcal{H}}$ and T_{LU} under h- and p-refinement

Wall time contribution	**Relative growth vs. additional accuracy digits Δd**	
	***h*-ref.**	***p*-ref.**
$T_{\text{near}} = \mathcal{O}\big(N_e\, p^4 q^2 \sum_{\ell=0}^{L} 4^\ell\big)$	$\dfrac{N_2}{N_1} \sim 10^{\,2\Delta d/p_1}$	$\dfrac{N_2}{N_1}\left(\dfrac{p_2}{p_1}\right)^2 \sim \big(1 + \mathcal{O}(\Delta d) + \mathcal{O}(\Delta d^2)\big)\left(\dfrac{p_1 + \Delta d/\eta}{p_1}\right)^2$
$T_{\mathcal{H}} = \mathcal{O}(r^2 N \log N)$	exponential	quadratic in N, polynomial in r
$T_{\text{LU}} = \mathcal{O}(r^2 N \log^2 N)$	exponential	quadratic in N, polynomial in r

6.3.5 $\mathcal{H}$-matrix acceleration

To reduce the memory and runtime cost of storing and factoring the dense CBIE coefficient matrix (6.15), we employ a hierarchical-matrix ($\mathcal{H}$-matrix) acceleration. The $\mathcal{H}$-matrix construction and arithmetic (including $\mathcal{H}$-LU and triangular solves) are implemented using the HLibPro library [41] following standard $\mathcal{H}$-matrix techniques [42, 43]; further implementation details can be found in [42, 43, 35]. In this work, we follow the hybrid direct–iterative $\mathcal{H}$-matrix approach of [35] by using a lower-accuracy $\mathcal{H}$-LU factorization as an iterative preconditioner to accelerate convergence while controlling wall time.

6.4 Error-Level-Dependent Wall Time

The wall time of the proposed $\mathcal{H}$-matrix accelerated framework consists of three main components:

1. Near-field fill via adaptive integration,

2. $\mathcal{H}$-matrix assembly,

3. $\mathcal{H}$-LU factorization.

The near-field fill involves $N_e = N/p^2$ element pairs, each requiring an $\mathcal{O}(p^4)$ tensor-product rule per sub-patch. The corresponding wall time is given by

$$T_{\text{near}}(d,p) = \mathcal{O}\left(N_e\, p^4 q^2 \sum_{\ell=0}^{L(d,p)} 4^\ell \right), \tag{6.17}$$

where $L(d,p)$ denotes the refinement depth required to reach a target accuracy $\epsilon = 10^{-d}$ as defined in (6.16), and q is the quadrature order.

The $\mathcal{H}$-matrix assembly and $\mathcal{H}$-LU factorization stages scale as $\mathcal{O}(r^2 N \log N)$ and $\mathcal{O}(r^2 N \log^2 N)$, respectively, where r is the maximum admissible block rank and $N = N_e(p+1)^2$ is the total number of unknowns[44].

Combining these contributions, the overall wall time is bounded by

$$\begin{aligned} \mathrm{T}(N) = \quad & \underbrace{\mathcal{O}\left(r^2 N \log N\right)}_{\mathcal{H}\text{-matrix assembly}} \\ & + \underbrace{\mathcal{O}\left(r^2 N \log^2 N\right)}_{\mathcal{H}\text{-}LU\text{ factorization}} \\ & + \underbrace{\mathcal{O}\left(N_e\, p^4 q^2 \sum_{\ell=0}^{L(d,p)} 4^\ell \right)}_{\text{Near-field fill}} \end{aligned} \tag{6.18}$$

where the first term bounds to the wall time of assembling the hierarchical matrix, while the second term arises from the $\mathcal{H}$-LU factorization [45] and last term encapsulates the detailed scaling of the near-field fill time as given in (6.17).

The quadrature order for kernel evaluations in the adaptive integration is chosen according to the polynomial degree of the solution. Specifically, we set $q = 4$ for solution degree

$p < 4$, and $q = p + 1$ otherwise. This heuristic has been shown to yield $\mathcal{O}(h^p)$ convergence in surface current error for smooth geometries.

In practice, realizing this quasi-optimal scaling, especially in a high-order benchmarking framework depends critically on minimizing the fill time for near-field interactions. Increasing the polynomial order p raises the number of quadrature points per element required to accurately integrate the MFIE kernel, thus increasing the time of each near-field evaluation. As described in Section 6.2, high-order discretizations require exponentially fewer unknowns, compared to h-refinement, to achieve a prescribed error level. Consequently, the balance between near-matrix fill time and total solve time increasingly favors high-order solutions as higher accuracy is targeted.

To illustrate, Fig. 6.3 shows the relative error in the surface current achieved as a function of near-interaction fill time for the problem of computing the RCS for a 2λ PEC sphere (compared to the exact Mie solution). At a target error of $\epsilon = 10^{-2}$, we compare:

- A zeroth-order solution ($p = 0$) on finely refined meshes of 3560, 6753, and 9432 elements.

- First-, second-, third-, and fourth-order solutions ($p = 1, 2, 3, 4$) on coarser meshes of 88, 96, and 388 elements, respectively.

All high-order schemes ($p = 1, 2, 3, 4$) exhibit lower fill times than the zeroth-order ($p = 0$) case with pure h-refinement, and this performance gap widens as the error tolerance is tightened. Thus, despite a larger constant in the near-field work, p-refinement achieves a given accuracy at a lower overall wall time than h-refinement.

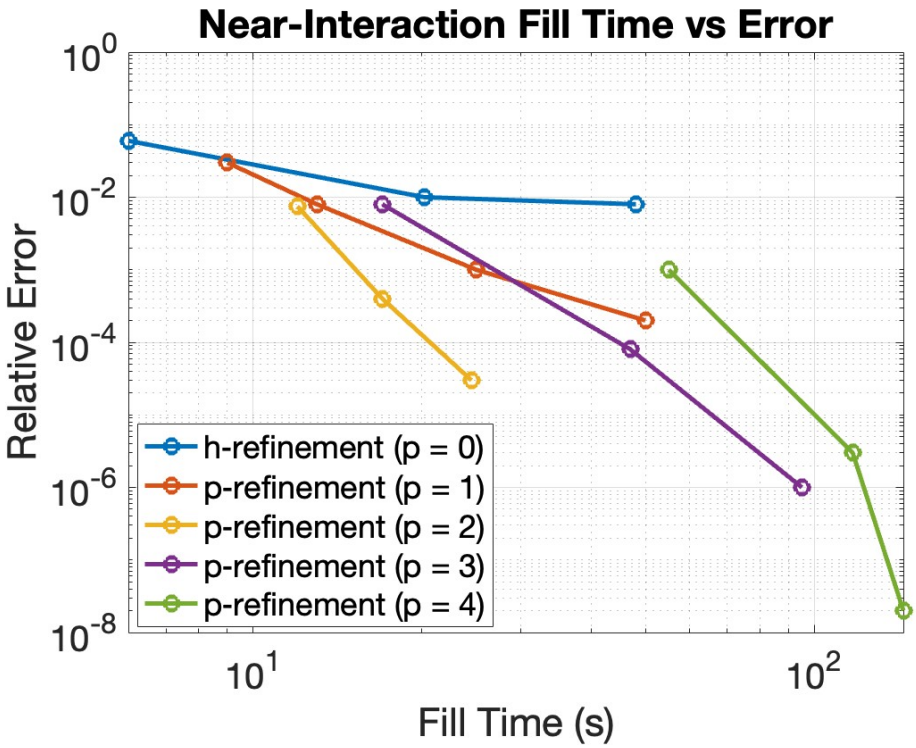


Figure 6.3: Near-interaction fill time versus relative error in surface current for a 2λ PEC sphere (Mie solution) under h- and p-refinement strategies.

6.4.1 Wall time trends for h- versus p-refinement

h-refinement Tightening the tolerance from $\epsilon_1 = 10^{-d_1}$ to $\epsilon_2 = 10^{-d_2}$ ($\Delta d = d_2 - d_1 > 0$) is accomplished by shrinking the element size h. Invoking (6.6) with a fixed polynomial degree $p = p_1$ gives

$$\frac{N_2}{N_1} \sim 10^{2\Delta d/p_1},$$

so the ratio N_2/N_1, and therefore every solver stage whose wall time in (6.18) scales at least linearly with the element count N_e, grows exponentially with the accuracy increment Δd. Among these stages, the $\mathcal{H}$-LU factorization is asymptotically dominant because of its polyloglinear complexity.

p-refinement When the desired accuracy is reached by raising the polynomial order from p_1 to $p_2 = p_1 + \beta = p_1 + \Delta d/\eta$ on a fixed mesh ($h = h_1$) where $h_1 = 10^{-\eta}$, the far-field numerical ranks r that govern $\mathcal{H}$-matrix storage and factorization increase only polynomially with p. Further, the asymptotic error expansion obtained in (6.10) shows that

$$\frac{N_2}{N_1} \sim 1 + \mathcal{O}(\Delta d) + \mathcal{O}\big(\Delta d^2\big),$$

so the ratio N_2/N_1 grows quadratically in Δd. As a result, the wall time of $\mathcal{H}$-fill, and $\mathcal{H}$-LU wall time grow polynomially in r and quadratically in N as a function of Δd. The increase in near fill time can be calculated from the growth of N and p through (6.17). For the near-field term we assume the adaptive integration is capped at its highest grid level ($\ell_{\max} = 10$); because this cap renders the contribution from adaptive integration effectively constant, we omit it from the relative-growth expressions.

Table 6.1 compiles the asymptotic time growth of each refinement strategy. In other words, the listed factors indicate how each wall time component scales relative to its initial value as the accuracy increment Δd grows.

6.5 Results

Our objective is to highlight the effectiveness of high-order methods accelerated by the $\mathcal{H}$-matrix framework for benchmarking RCS computations. To this end, several examples are presented covering geometries of varying complexity.

The $\mathcal{H}$-matrix operations were implemented using HLibPro [41] within the $\mathcal{H}$-matrix preconditioned iterative solver described in [35]. In this solver, two ACA tolerances were employed for the preconditioner (at the leaf and non-leaf levels), along with one global ACA

tolerance for building the coefficient $\mathcal{H}$-matrix. Specifically, these values were empirically set to, $\tau_{\text{ACA}}^{\text{CBIE}} = 10^{-5}$, $\tau_{\text{ACA, leaf}}^{\text{Precond.}} = 10^{-3}$, and $\tau_{\text{ACA, non-leaf}}^{\text{Precond.}} = 10^{-4}$, so as to provide a balanced trade-off between accuracy, computational efficiency, and memory consumption.

To evaluate the effectiveness of the proposed high-order $\mathcal{H}$-CBIE solver, two classes of benchmark examples are considered: smooth and non-smooth geometries. This division reflects the different numerical challenges posed by these two classes of PEC targets. In particular, smooth geometries exhibit non-singular surface currents that are well captured by high-order basis functions, whereas sharp-edged geometries introduce singular surface current behavior at edges and corners, requiring careful balancing to maintain accuracy and solver stability.

6.5.1 Smooth PEC Geometries

In the class of smooth geometries, we study three cases: an analytic (exact) sphere, a NURBS-based arbitrary sphere, and a CAD-derived Bunny. These provide a clean baseline for assessing accuracy, convergence, and memory use in the absence of geometric singularities. The two sphere variants let us probe $\mathcal{H}$-CBIE behavior with and without geometry-approximation error (analytic vs. NURBS-based).

Exact Sphere

As a first example, we consider an exact sphere obtained via an exact cube-to-sphere mapping which doesn't generate any modeling error[46]. This PEC sphere has a radius $r = 1$ m, and it is analyzed at two electrical sizes, illuminated by a plane wave propagating along $+x$. Each test reports the most informative bistatic-RCS cuts; Mie-series results or commercial RWG-MoM-MLFMM data are overlaid where available.

$f = 3$ GHz (20λ) In this example for an exact sphere, order 7 and order 8 polynomials are used on 384 Bézier patches. The incident field is a V-polarized planewave. Fig. 6.4a shows order convergence across the full angular sweep, confirming that the chosen patch layout is adequate at $D/\lambda = 20$ when paired with a suitable polynomial order p.

$f = 4$ GHz (26.7λ) At this electrical size the same patch layout, with a further increase in p, still tracks the Mie solution closely. Fig. 6.4b shows the $\phi = 0$ RCS for the 26.7λ sphere with order 14 polynomials with the tangential electric dipole excitation. As shown in this figure, $\mathcal{H}$-CBIE attains 6–7 digits of accuracy with respect to the Mie solution on the r=1 m sphere at f=4 GHz, consistent with the well-conditioned behavior of high-order formulations. In contrast, matching this accuracy with a low-order RWG-MoM discretization would require an impractical number of unknowns (orders of magnitude larger than the 150,528 used by order-14 $\mathcal{H}$-CBIE), demonstrating the efficiency advantage of high-order bases for integral equations.

Table 6.2 summarizes timing and memory for $\mathcal{H}$-CBIE versus a commercial RWG–MoM MLFMM at $f = 4$ GHz (MLFMM tolerance $\epsilon = 10^{-9}$). $\mathcal{H}$-CBIE ran on Niagara (Digital Research Alliance of Canada) using 32 Intel "Skylake" 2.4 GHz or "Cascade Lake" 2.5 GHz cores [47]; the commercial RWG–MoM MLFMM ran on a 32-core Intel *Xeon* Gold 6240 (2.6 GHz) system in collaboration with ASELSAN, Turkey [48]. Near-interaction matrix fill was parallelized with MPI across 32 cores, and $\mathcal{H}$-matrix assembly/LU used OpenMP (up to 4 threads).

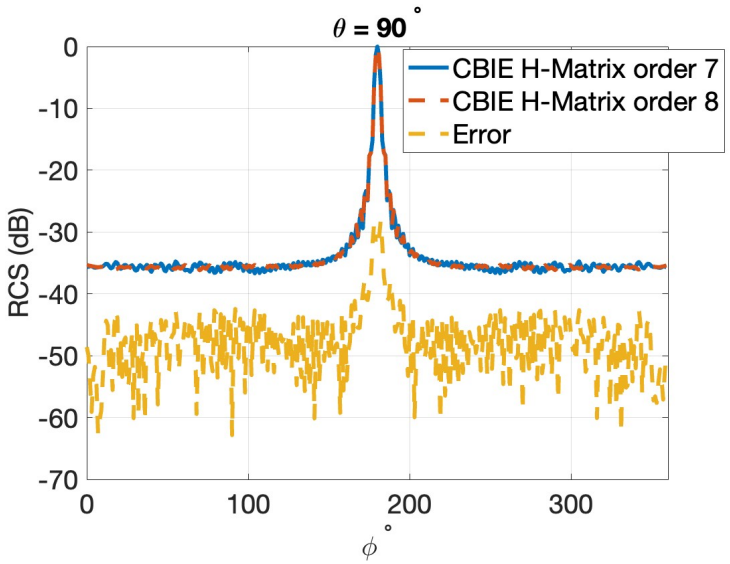
To verify the expected $O(h^p)$ error convergence behavior, CBIE solutions of increasing polynomial order p and decreasing Bézier element size h are compared with the analytic Mie series. The relative errors for a PEC sphere of radius $r = 1$ m at $f = 300$ MHz and

Table 6.2: Time and memory summary for the 1 m radius PEC sphere at $f = 4$ GHz (32 cores).

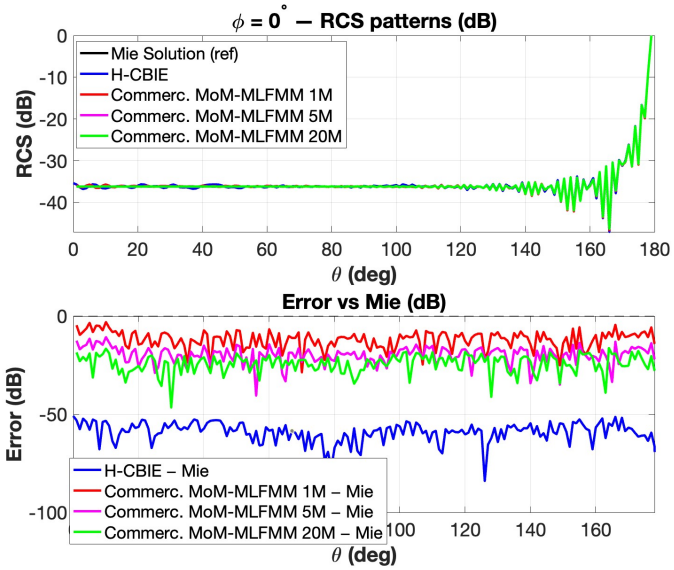
Solver	N **(unknowns)**	**CPU Time**	**Mem.**
$\mathcal{H}$-CBIE	150,000	149 min	48 GB
RWG–MoM–MLFMM	814,000	5 min	94 GB
RWG–MoM–MLFMM	5,000,000	67 min	138 GB
RWG–MoM–MLFMM	20,000,000	738 min	1182.0 GB

$f = 600$ MHz are plotted in Fig. 6.5(a) and Fig. 6.5(b), respectively. As the mesh is refined and the order is raised, the errors fall and the curve slopes steepen, in line with the $O(h^p)$ prediction. Achieving the same accuracy with a low-order discretisation would require a far denser mesh, vastly more unknowns, and substantially longer run-times, whereas the high-order CBIE formulation attains higher precision with only a fraction of the unknowns. This efficiency is critical for resolving the deep RCS nulls that arise at large electrical sizes, features that are effectively unattainable with low-order schemes.

Fig. 6.6 contrasts the dominant-term prediction with normalized runtimes from our $\mathcal{H}$-CBIE and a commercial RWG-MoM MLFMM solver on a PEC sphere. Let $T_{h,1}$ and $T_{p,1}$ denote the first wall-time samples of the h- and p-refinement sequences, respectively (both taken at $d = 3$). All model predictions and simulation runtimes are reported in relative form by normalizing each sequence by its own baseline, i.e., $\tilde{T}_h(d) = T_h(d)/T_{h,1}$ and $\tilde{T}_p(d) = T_p(d)/T_{p,1}$, so each curve reflects the relative wall-time growth. The relative wall time for h-refined meshes grows approximately exponentially with the target digits of accuracy, i.e., $\tilde{T}_h(d) \sim a^d$, whereas the corresponding trend for p-refined solutions is approximately quadratic, i.e., $\tilde{T}_p(d) \sim d^2$, which is consistent with the asymptotic wall-time analysis derived in Section IV for a target with an analytic solution.

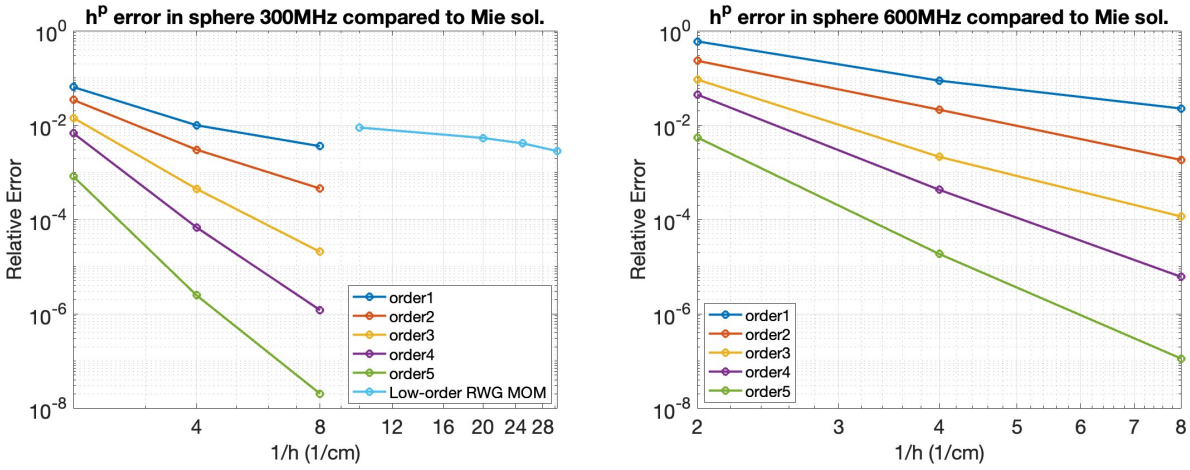


(a)



(b)

Figure 6.4: (a) $\theta = 90°$ normalized RCS of a 1 m radius PEC sphere at $f = 3\,\text{GHz}$ (self-convergence test), (b) $\phi = 0°$ normalized RCS for the TED case at $f = 4\,\text{GHz}$, compared with the Mie solution.



(a) (b)

Figure 6.5: Relative error of surface current versus mesh size h and polynomial order p: (a) sphere ($r = 1$ m) at $f = 300$ MHz; (b) sphere ($r = 1$ m) at $f = 600$ MHz. The error is computed as $\text{Error}(\mathbf{J}) = \|\mathbf{J}^{\text{CBIE}} - \mathbf{J}^{\text{Mie}}\|/\|\mathbf{J}^{\text{Mie}}\|$, with the Mie solution as reference.

Arbitrary NURBS-Based Sphere Up to this point, the Sphere case in our $\mathcal{H}$-CBIE solver used an analytic geometry representation, eliminating geometry-modeling error. As a final sphere test, we use a NURBS surface generated in Rhino (tolerance 10^{-5}) and its quadrilateral Bézier representation obtained with OpenNURBS via knot insertion and reassembly using the same knot data. The Bézier extraction is numerically lossless (we use 10^{-9} as the tolerance), so any geometry-approximation error arises solely from the Rhino NURBS generation and is therefore controlled. This geometry-approximation error is particularly relevant for high-order solvers such as $\mathcal{H}$-CBIE, where we deliberately use a modest number of curvilinear Bézier patches and raise the local polynomial order p to achieve high accuracy with fewer unknowns. In contrast, low-order RWG–MoM typically employs planar (flat) triangular elements and relies on aggressive h-refinement to approximate curved surfaces. While this can substantially reduce geometry-induced error, it does so at the expense of an exponential increase in the number of degrees of freedom and the associated wall time.

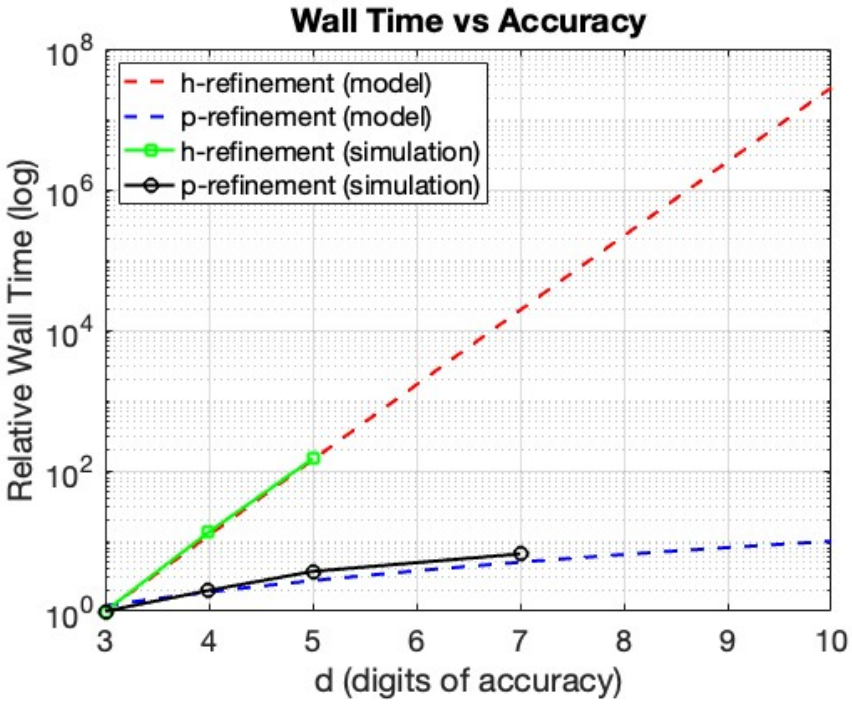
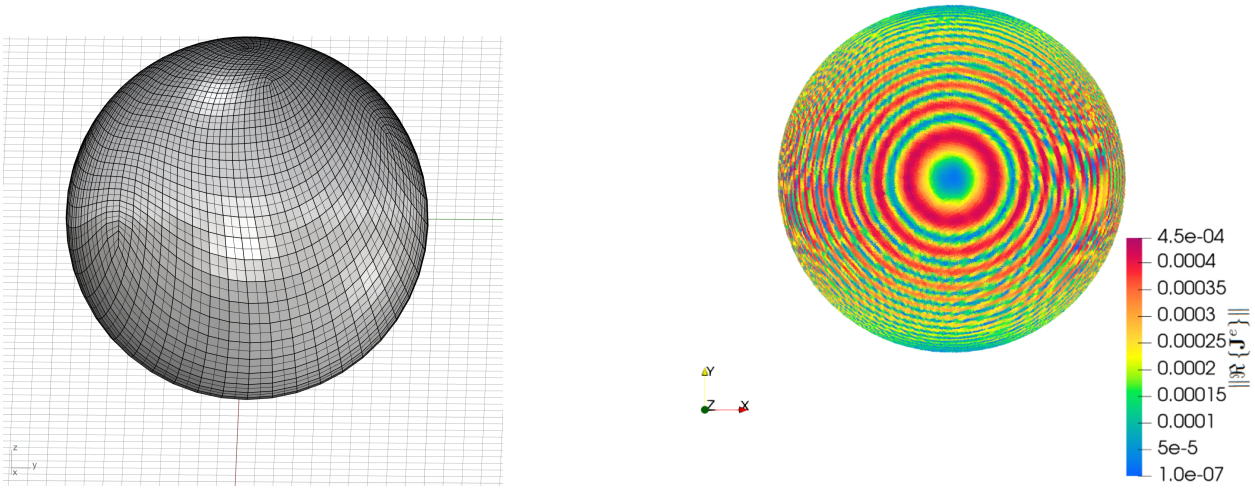


Figure 6.6: Normalized wall time growth: model (solid) versus simulation data (markers) for h- and p-refinement sequences.

To benchmark this NURBS-based sphere, we compare $\mathcal{H}$-CBIE performance against the Austin RCS Benchmark families (Method 1: MoM column-decomposition; Method 2: MoM–AIM; Method 3: MoM–$\mathcal{H}$-matrix/ACA) [31]. The Austin configuration uses a sphere with D=19.2 m at 320 MHz, i.e., $D/\lambda \approx 20.49$ and $k_0 a \approx 64.38$; we adopt the same parameters for our $\mathcal{H}$-CBIE runs.

Fig. 6.7 (a) shows the geometry model generated in Rhino using NURBS, achieving 4–5 digits of geometric modeling accuracy. We solve this case with a fifth-order (p=5) $\mathcal{H}$-CBIE discretization on 4,337 quadrilateral Bézier patches, resulting in 216,850 unknowns (DoF). Fig. 6.7 (b) shows the resulting surface current magnitude for a V-polarized plane wave incident from $+z$.

Fig. 6.8–6.9 show the Austin markers (Efficient, Fast, Serialized). Our $\mathcal{H}$-CBIE data



(a) (b)

Figure 6.7: Arbitrary NURBS-based sphere: (a) Rhino NURBS geometry; (b) surface-current magnitude for a V-polarized plane wave incident from $+z$.

appears as solid and dashed black lines. It was obtained with runs on 32 cores. For purpose of comparison, Fig. 6.8 converts Austin serialized timings to 32 cores assuming ideal scaling (serial/32), and Fig. 6.9 shows both serialized and converts per-process memory (serial/32).

Notably, the $\mathcal{H}$-CBIE curves exhibit a steeper downward slope than the Austin baselines: error decreases by orders of magnitude for relatively modest increases in wall time (Fig. 6.8) and per-process memory (Fig. 6.9). Overall, $\mathcal{H}$-CBIE occupies a favorable accuracy–memory–time region, attaining average error near 10^{-4} with total wall time on the order of 10^3–10^4 s and per-process memory around 2 GB, while remaining competitive at N_{proc}=32.

Based on the sphere tests, the proposed $\mathcal{H}$-CBIE with NURBS-based geometry reproduces the expected Mie/reference trends, exhibits clear *hp*-convergence, and maintains favorable accuracy–resource trade-offs at scale. These results indicate that combining high-order discretization with $\mathcal{H}$-matrix compression delivers reliable accuracy at reduced wall time

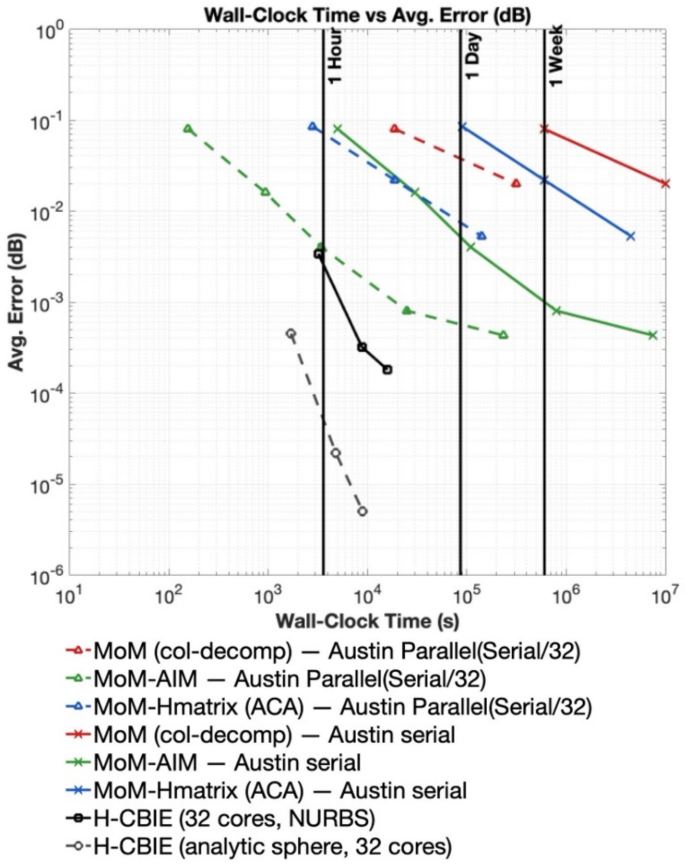


Figure 6.8: Wall-clock time versus average error (dB) for the Austin benchmarks [30] and the $\mathcal{H}$-CBIE solver. Austin timings are converted to 32 cores for the purpose of comparison; $\mathcal{H}$-CBIE timings are measured on 32 cores.

compared to the RWG MoM solvers with commonly used acceleration schemes. The close agreement with expected trends supports extension to larger, more complex smooth targets.

$\|\Re\{\mathbf{J}^e\}\|$

The Bunny

Our next smooth-geometry example evaluates both the scalability and robustness of the proposed algorithm on a large-scale, complex PEC target. The Bunny geometry is discretized

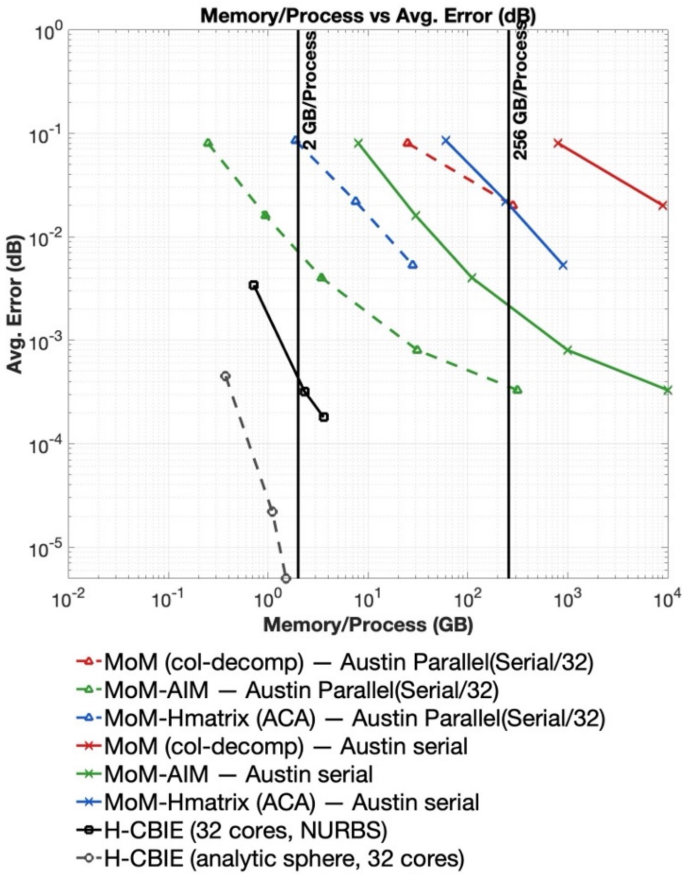


Figure 6.9: Per-process memory versus average error (dB) for the D=19.2 m, f=320 MHz sphere. Austin results are shown as serial and as converted per-process values (serial/32) for purpose of comparison; $\mathcal{H}$-CBIE points are measured per-process on 32 cores.

into N_p = 56,471 Bézier patches and analyzed using second- and third-order current expansions, resulting in N = 451,768 and N = 1,016,478 unknowns for the order-2 (O2) and order-3 (O3) current expansions, respectively. This case is intentionally pushed to $\mathcal{O}(10^6)$ unknowns to demonstrate practical scalability of the $\mathcal{H}$-CBIE framework on a non-canonical, large-scale geometry. Fig. 6.10a shows the induced surface current on the Bunny target (electrical height $\approx 42\lambda$). Fig. 6.10b and 6.10c present the corresponding RCS cuts for the O2 and O3 solutions.

The Bunny example is used to demonstrate the self-convergence and scalability of the algorithm on a complicated, yet smooth, geometry. Achieving a comparable level of accuracy in commercial software would require substantially greater memory and runtime due to the rapid growth in the number of unknowns; for this reason, we do not include commercial-software results for this case. Specifically, the relative O2–O3 discrepancy (comparing to O3 RCS) remains below approximately -60 dB. Note that the O2 and O3 curves are nearly indistinguishable in the plots. These results demonstrate rapid convergence with respect to expansion order while confirming the method's ability to handle million-unknown problems on complex geometries.

6.5.2 Sharp-Edge PEC Geometries

The examples presented above demonstrate that $\mathcal{H}$-CBIE delivers an error-controllable computational framework when applied to smooth geometries. Specifically, it can achieve error-controllable precision making it suitable for delivering highly accurate data for the purposes of benchmarking other CEM solver performance (with lower accuracy requirements). Moreover, when high precision is sought in the solution itself, the framework delivers solutions exponentially faster than lower-order methods, and not only produces benchmark data, but sets benchmark standards for performance. Having established these advantages for smooth targets, we now turn to problem geometries involving sharp edges, where the underlying $O(h^p)$ error convergence breaks down near edges and junctions, necessitating localized refinement to mitigate solution error.

For the sharp-edged geometries that follow, the surface current expansion near sharp edges is treated using a localized hp-refinement strategy. Specifically, sharp features are modeled with fine h and lower orders p_{edge}, while large smooth patches use coarser h with

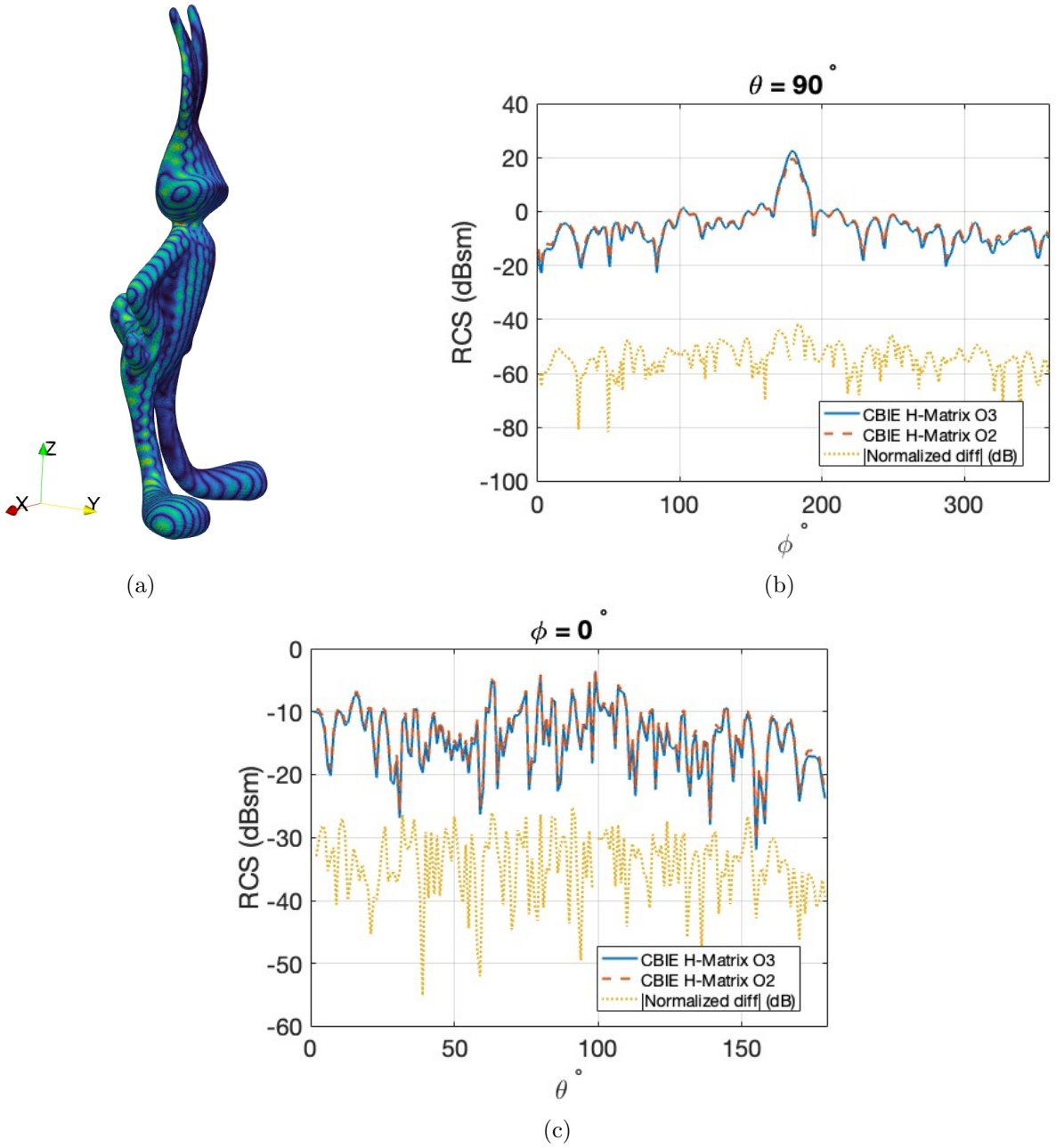


Figure 6.10: The 42λ-tall Bunny corresponding to $N = 1016478$ unknowns: (a) surface current, (b) $\theta = 90°$ RCS cut, (c) $\phi = 0°$ RCS cut. The error curve is computed by comparing the O2 and O3 $\mathcal{H}$-CBIE solutions (self-convergence).

higher orders p_{smooth}. Elements with about $2:1$ size ratio and order $p_{\text{transition}}$ are introduced to provide a smooth transition between the regions to preserve accuracy and conditioning.

It is important to note that, in our workflow, we generate quadrilateral patches from the original CAD surface in Rhino and feed them into our $\mathcal{H}$-CBIE solver. This remeshing step can introduce small differences in the final geometry and thus slightly alter the RCS results. Furthermore, while we attempted to use the same geometry in both our in-house solver and the commercial MoM–MLFMM solver, some commercial workflows may apply additional mesh processing (e.g., automatic remeshing/optimization), which can lead to slight differences in the resulting surface discretization. These discrepancies in meshing and geometry representation will potentially lead to differences in the RCS results, particularly near deep nulls where the scattering response is most sensitive to small geometric changes. As a result, the following results provide self-convergence tests in addition to comparisons between solvers.

NASA Almond

In this example, we consider the NASA almond of length $L = 25.23744$ cm, a widely used RCS benchmark target [49]. Fig. 6.11 shows the CBIE discretization. On the *same* mesh discretization, we consider two mixed-hp orders for the current expansion: $(p_{\text{edge}}, p_{\text{transition}}, p_{\text{smooth}}) = (3,4,6)$ with $N = 256{,}948$ unknowns (fine case) and $(3,4,5)$ with $N = 201{,}534$ unknowns (coarse case). At $f = 3.5$ GHz ($L/\lambda \approx 2.9$), the corresponding surface current visualizations in Figs. 6.12a and 6.12b (V-polarized and H-polarized, incidence from $+\hat{x}$) are shown for the *Fine* case $(3,4,6)$.

In the RCS comparisons in Fig. 6.13, we evaluate both discretizations to demonstrate self-convergence of the $\mathcal{H}$-CBIE solver and to emphasize that any residual discrepancies relative

to the UT Austin benchmark measurements [49] arise primarily from geometry perturbations introduced during quadrilateral-patch generation in Rhino. The same sensitivity to small geometry modifications is likewise observed in the commercial RWG–MoM–MLFMM solution.

It is important to note that the UT Austin almond measurements themselves carry a finite experimental tolerance. The detailed measurement campaign at Lockheed Martin's Rye Canyon facility reports that RCS results are generally within up to ± 2 dB of simulations [50]. This bound is consistent with typical aerospace manufacturing tolerances: for small aerodynamic skin panels, deviations are on the order of ± 0.05–± 0.2 mm, while for large fuselage sections they can reach ± 0.2–± 1.0 mm. At the almond validation frequency ($f = 3.5$ GHz, $\lambda \approx 85.7$ mm), such tolerances correspond to geometric perturbations ranging from $\lambda/1700$ to $\lambda/85$. In addition, when transferring the almond geometry from the original NURBS representation in Rhino to quadrilateral Bézier patches for our solver, we observe local surface deviations on the order of 0.1–0.3 mm. Since conversion from NURBS to Bézier patches is not an exact process, these perturbations are comparable to the cited manufacturing tolerances and therefore provide a plausible explanation for residual discrepancies in RCS results between $\mathcal{H}$-CBIE, the UT Austin measurements, and the commercial RWG–MoM–MLFMM solutions. Accordingly, we include a conservative ± 2 dB uncertainty band around the UT Austin curves in all plots to reflect this benchmark tolerance.

Monostatic $\theta = 90^\circ$ RCS is then computed for both H and V polarizations, and compared against the UT-Austin benchmarks [49] and a commercial RWG–MoM–MLFMM solver.

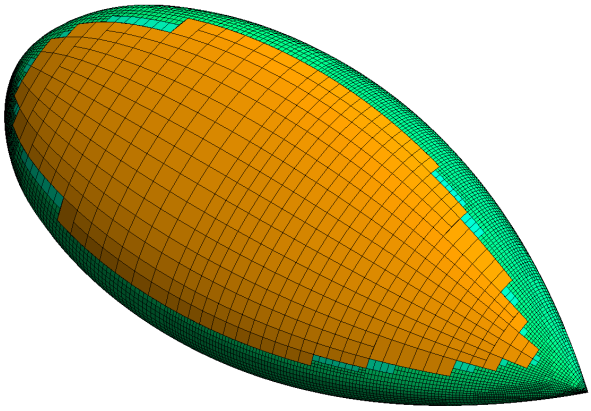
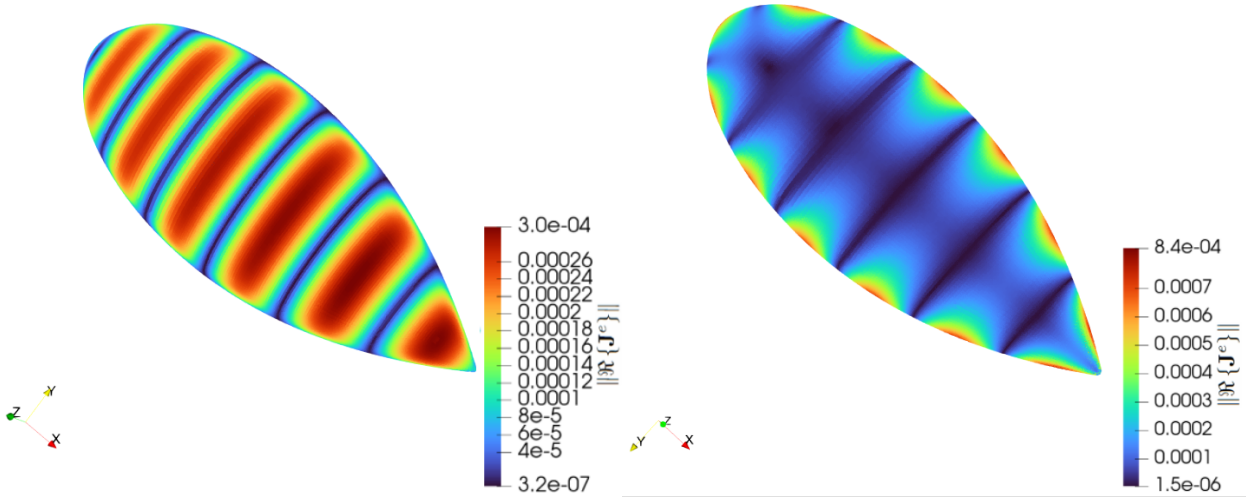


Figure 6.11: NASA almond mesh discretization with adaptive *hp*-refinement.



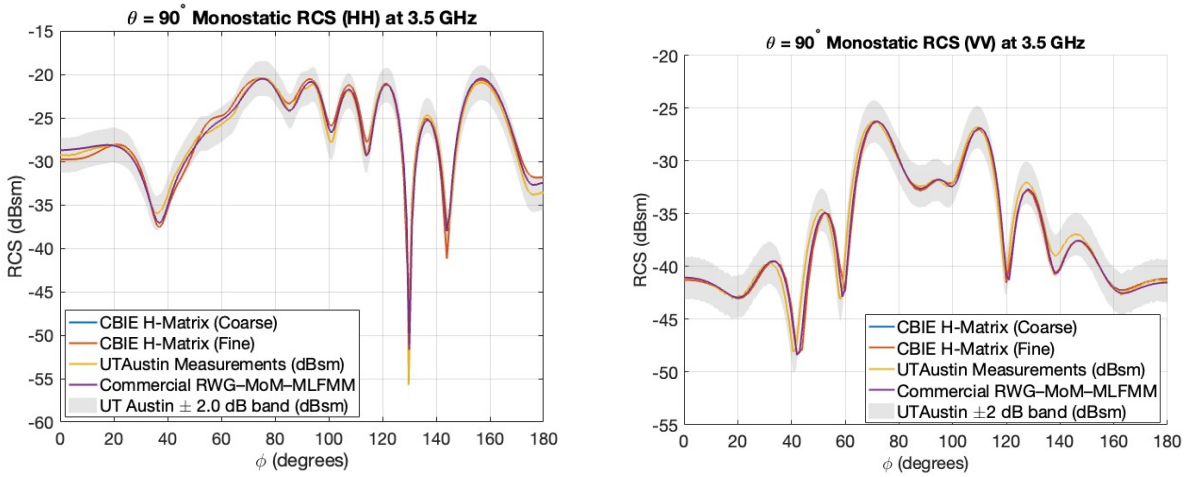
(a) V-polarized, x-directed incidence

(b) H-polarized, x-directed incidence

Figure 6.12: NASA almond surface current magnitude at 3.5 GHz under (a) V- and (b) H-polarized plane-wave excitation.

B2 Aircraft

As a final example, the same adaptive *hp*-refinement strategy is applied to a more complex sharp-edged geometry. Fig. 6.14 shows the adaptive *hp* mesh and surface current of a B2-like CAD surface (NURBS-derived) with multiple sharp edges generated by the adaptive



(a) H-polarized Excitation

(b) V-polarized Excitation

Figure 6.13: NASA almond, $f = 3.5$ GHz: monostatic RCS comparison against UT-Austin and commercial references.

refinement framework within $\mathcal{H}$-CBIE. The geometry is excited by a $f = 600$ MHz x-directed V-polarized plane-wave. At this frequency the geometry has a 34λ wing span.

A fixed mesh was adopted for two different polynomial order expansions. Denoting the expansion order configuration as $(p_{edge}, p_{transition}, p_{smooth})$, the problem was solved using a lower accuracy configuration $(3, 4, 6)$ with $N = 177,344$ unknowns and a higher accuracy configuration $(3, 5, 7)$ with $N = 244{,}536$ unknowns. Results were also collected using a commercial MoM–MLFMM solver ($N = 5.1$M unknowns).

Table 6.3 summarizes solver statistics. The $\mathcal{H}$-CBIE reached a residual of 4.31×10^{-14} in only six iterations for the higher-accuracy solution, with wall time of 4.84 hrs and a memory usage of 144.6 GB. The lower-accuracy solution achieved essentially the same residual accuracy (2.5×10^{-14}) in five iterations, at a reduced wall time of about 3.5 hours and 105 GB of memory. Both cases remain far below the wall time of the 5.1M unknowns RWG-MoM-MLFMM run, which got stuck at a residual of 4.61×10^{-3} after 500 iterations and required

Table 6.3: Performance comparison: $\mathcal{H}$-CBIE vs. commercial MoM–MLFMM.

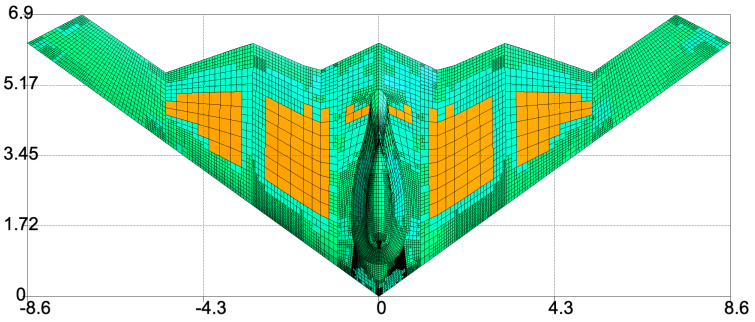
Method	**Conv.**	**Iter.**	**Residual**	**CPU Time**	**Mem.**	N
$\mathcal{H}$-CBIE (fine)	Yes	6	4.31×10^{-14}	4.84 hrs	144.6 GB	244,536
$\mathcal{H}$-CBIE (coarse)	Yes	5	2.5×10^{-14}	3.51 hrs	105 GB	177,344
RWG-MoM-MLFMM	No	500	4.61×10^{-3}	11.41 hrs	2.67 TB	5.1M

Estimated from the fine case using proportional scaling in N (ratio 177,344/244,536 ≈ 0.725); actual wall time and memory typically vary within $\pm$(10–20)% depending on cluster tree, admissibility, and ACA thresholds.

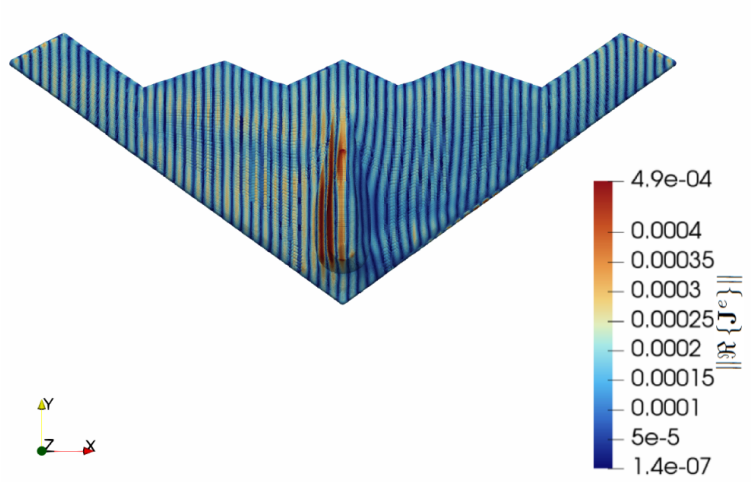
11.41 hours and 2.67 TB of memory.

Fig. 6.15a compares the $\phi = 0^\circ$ RCS obtained with the $\mathcal{H}$-CBIE. The CBIE solution appears to be stable given that the solution at the edges is capped at order 3, while discrepancies exist between our solver and the RWG MoM-MLFMM solver. Notably, at deeper nulls (near $\theta = 155^\circ$), the discrepancy is largest. We attribute this difference to the lack of convergence in the RWG MoM-MLFMM solution below a residual of 4.61×10^{-3}, and also, geometry perturbations in generating mesh for both solvers. Fig. 6.15b shows the $\theta = 90^\circ$ RCS cut, where again the CBIE solver is self- consistent with order refinement but differs from the RWG MoM-MLFMM solution, likely due to a lack of convergence and geometry modifications in the latter.

These B2 results, obtained using a heuristic *hp*-refinement strategy to balance accuracy between smooth regions and along edges, appear to provide $\mathcal{H}$-CBIE with a path to meaningfully accurate solutions to tough scattering problems (more accurate than typically attributed to low-order formulations like RWG-MoM).



(a) Adaptive *hp* mesh (edge, transition, smooth).

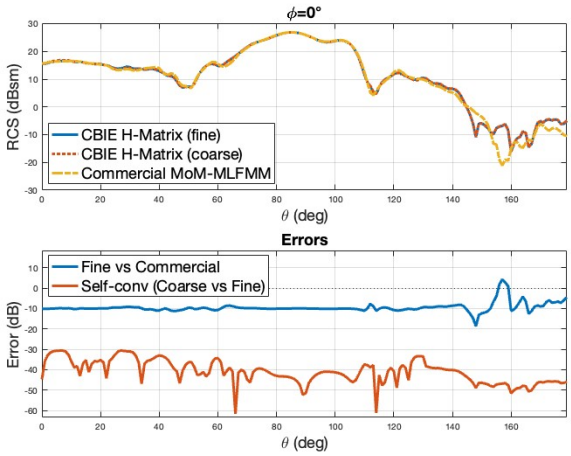


(b) Surface current

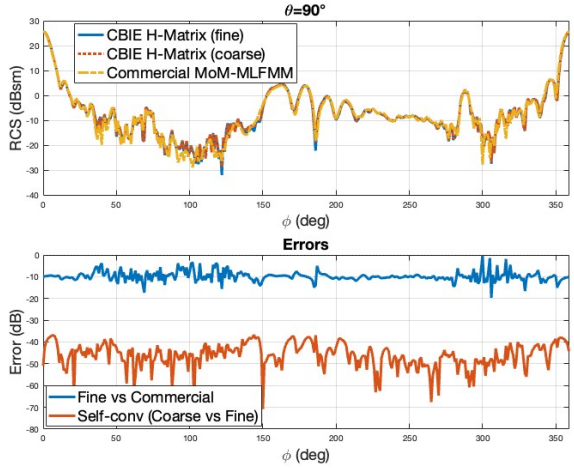
Figure 6.14: B2-like aircraft at 34λ: (a) adaptive *hp* mesh and (b) surface current distribution with V-polarized excitation.

6.6 Conclusions

This work has presented a high–order strategy for RCS benchmarking that couples point–based discretization (CBIE/MFIE) with $\mathcal{H}$-matrix acceleration. The framework begins with an order–adaptive mesh that exploits the $O(h^p)$ error behavior, ensuring a prescribed number of accurate digits d while minimizing the total number of unknowns. Once the system matrix is compressed, CPU time and memory scale as $O(N \log N)$, and the ensuing $\mathcal{H}$–LU factorization raises the overall wall time only quadratically in N as requested digits



(a) $\phi = 0°$ RCS cut: $\mathcal{H}$-CBIE vs. commercial MLFMM.



(b) $\theta = 90°$ RCS cut: $\mathcal{H}$-CBIE vs. commercial MLFMM.

Figure 6.15: B2-like aircraft at 34λ: RCS accuracy comparison. The $\mathcal{H}$-CBIE exhibits stable self-convergence, while the low-order RWG–MLFMM approach fails to converge after 500 iterations.

of accuracy d increase linearly. In contrast, a classical RWG MoM formulation requires an exponential increase in the number of unknowns, and therefore in run–time and memory, to achieve comparable digits of precision d.

As this solver can be precision–controlled (at least for smooth geometries), the resulting RCS solutions can serve as high quality benchmark data provided with an accurate geometric modeling. Furthermore, our heuristic local *hp*-refinement strategy near sharp edges appears to provide meaningfully accurate solutions for RCS computations. Thus, $\mathcal{H}$-CBIE applied to smooth or sharp-edged targets should yield a benchmark against which new and established algorithms can be measured. We have demonstrated this concept on smooth (sphere, bunny) and non-smooth (NASA almond, B2) scatterers. Furthermore, we validated our claims for sphere in two steps: (i) against Mie solution and a commercial RWG–MoM–MLFMM solver using an analytic exact sphere in our $\mathcal{H}$-CBIE (eliminating geometry-modeling error), and

(ii) against the UT–Austin sphere benchmarks using a NURBS-based sphere (introducing a controlled geometry approximation). For the problems and accuracy considered, $\mathcal{H}$-CBIE requires one to two orders of magnitude fewer unknowns, substantially less memory, and shorter runtime in the smooth and sharp-edged target case studies. This confirms that high–order integral equation formulations are capable of delivering reference level accuracy for arbitrary (including sharp-edged geometries) without resorting to excessive mesh densities.

The accompanying complexity study confirms that the proposed framework delivers accuracy through p-refinement at a wall time that grows quadratically, not exponentially, with digits of precision d. This advantage widens rapidly at higher frequencies, making the method uniquely attractive for large-scale target RCS benchmarking where low level side lobes and fine angular details must be resolved. The results produced by this solver, validated on smooth, piecewise–smooth, and complex sharp-edged geometries could serve as benchmark references for future advances in fast, high-fidelity electromagnetic solvers.

Acknowledgements

The authors thank IEEE AP-S Fellowship Program for supporting this work.

REFERENCES

Bibliography

[1] D. Reitan and T. Higgins, "Calculation of the electrical capacitance of a cube," *J. Appl. Phys.*, vol. 22, no. 2, pp. 223–226, 1951.

[2] E. N. Vasil'ev, "Excitation of a smooth perfectly electrically conducting body of revolution," *Izv. Vyssh. Uchebn. Zaved. Radiofiz.*, vol. 2, no. 4, pp. 588–601, 1959 (in Russian).

[3] K. Mei and J. Van Bladel, "Scattering by perfectly conducting rectangular cylinders," *IEEE Trans. Antennas Propag.*, vol. 11, no. 2, pp. 185–192, Mar. 1963.

[4] K. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Trans. Antennas Propag.*, vol. 14, no. 3, pp. 302–307, May 1966.

[5] A. Hrennikoff, "Solution of problems of elasticity by the framework method," *ASME J. Appl. Mech.*, vol. 8, pp. A619–A715, 1941.

[6] J.-M. Jin, *The Finite Element Method in Electromagnetics*, 2nd ed. New York, NY, USA: Wiley, 2002.

[7] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, vol. 30, no. 3, pp. 409–418, May 1982.

[8] L. Canino *et al.*, "Numerical solution of the Helmholtz equation in 2D and 3D using a high-order Nyström discretization," *J. Comput. Phys.*, vol. 146, no. 2, pp. 627–663, Nov. 1998.

[9] P. Ya. Ufimtsev, *Method of Edge Waves in the Physical Theory of Diffraction*. Moscow, USSR: Soviet Radio, 1962.

[10] J. B. Keller, "Geometrical theory of diffraction," *J. Opt. Soc. Am.*, vol. 52, no. 2, pp. 116–130, 1962.

[11] R. G. Kouyoumjian and P. H. Pathak, "A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface," *Proc. IEEE*, vol. 62, no. 11, pp. 1448–1461, Nov. 1974.

[12] N. Engheta, W. D. Murphy, V. Rokhlin, and M. S. Vassiliou, "The fast multipole method (FMM) for electromagnetic scattering problems," *IEEE Trans. Antennas Propag.*, vol. 40, no. 6, pp. 634–641, Jun. 1992, doi: 10.1109/8.144597.

[13] R. R. Coifman, V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: A pedestrian prescription," *IEEE Antennas Propag. Mag.*, vol. 35, no. 3, pp. 7–12, Jun. 1993, doi: 10.1109/74.250128.

[14] E. Bleszynski, M. Bleszynski, and T. Jaroszewicz, "AIM: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems," *Radio Sci.*, vol. 31, no. 5, pp. 1225–1251, Sep. 1996.

[15] W.-J. He, X.-W. Huang, M.-L. Yang, and X.-Q. Sheng, "Massively parallel multilevel fast multipole algorithm for extremely large-scale electromagnetic simulations: A review," *Prog. Electromagn. Res.*, vol. 173, pp. 37–52, 2022.

[16] R. D. Graglia, A. F. Peterson, and M. Boella, *Higher-Order Techniques in Computational Electromagnetics*. Raleigh, NC, USA: SciTech Publishing, 2016.

[17] J. Hu, E. Sever, O. Babazadeh *et al.*, "H-matrix accelerated direct matrix solver using Chebyshev-based Nyström boundary integral equation method," *IEEE Open J. Antennas Propag.*, vol. 6, no. 1, pp. 171–180, Feb. 2025, doi: 10.1109/OJAP.2024.3485817.

[18] Y. P. Chen, S. Sun, L. Jiang, and W. C. Chew, "A Calderón preconditioner for the electric field integral equation with layered-medium Green's function," *IEEE Trans. Antennas Propag.*, vol. 62, no. 4, pp. 2022–2030, Apr. 2014, doi: 10.1109/TAP.2013.2297396.

[19] L. Greengard *et al.*, "Fast direct solvers for integral equations in complex three-dimensional domains," *Acta Numer.*, vol. 18, pp. 243–275, 2009.

[20] N. V. Nair and B. Shanker, "Generalized method of moments: A novel discretization technique for integral equations," *IEEE Trans. Antennas Propag.*, vol. 59, no. 6, pp. 2280–2293, Jun. 2011.

[21] W. C. Chew, M.-S. Tong, and H. U. Bin, *Integral Equation Methods for Electromagnetic and Elastic Waves*. New York, NY, USA: Springer, 2022.

[22] S. Zheng, A. Menshov, and V. I. Okhmatovski, "New single-source surface integral equation for magneto-quasi-static characterization of transmission lines situated in multilayered media," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 12, pp. 4341–4351, Dec. 2016, doi: 10.1109/TMTT.2016.2623625.

[23] J. L. Volakis *et al.*, "Benchmark radar targets for the validation of computational electromagnetics programs," *IEEE Antennas Propag. Mag.*, vol. 35, no. 1, pp. 12–35, 1993.

[24] J. L. Volakis and T. Kempel *et al.*, "The FEMATS Project: A parallel finite element method for large-scale RCS analysis," *NASA Tech. Rep.*, 1995.

[25] S. Bindiganavale and J. L. Volakis, "Error and execution time analysis of the fast multipole method for RCS computations," *Radio Sci.*, vol. 30, no. 5, pp. 1235–1248, 1995.

[26] E. Erricolo, "Validating computational software with exact analytical solutions: A review of the last three decades," *IEEE Antennas Propag. Mag.*, vol. 67, no. 2, pp. 64–75, 2025.

[27] J.-L. Guiraud *et al.*, "Benchmarking computational electromagnetics: Long-lasting collaborative effort animated by the JINA/ISAE Scientific Committee," *IEEE Antennas Propag. Mag.*, vol. 67, no. 2, pp. 32–42, Apr. 2025, doi: 10.1109/MAP.2025.3537411.

[28] F. Weinmann *et al.*, "The European Defence Agency Workshop "Radar Signatures & EM Benchmarks": A scientific forum for comparison and evaluation of electromagnetic simulations," *IEEE Antennas Propag. Mag.*, vol. 67, no. 2, pp. 43–50, Apr. 2025, doi: 10.1109/MAP.2025.3537408.

[29] D. L. Dault and L. Griffiths, "Computational electromagnetics code validation and the Electromagnetic Code Consortium: Case studies, challenges, lessons learned, and recommendations," *IEEE Antennas Propag. Mag.*, vol. 67, no. 2, pp. 10–19, Apr. 2025, doi: 10.1109/MAP.2025.3537413.

[30] A. Maicke, J. T. Kelley, and A. E. Yilmaz, "Performance benchmarking with the Austin RCS Benchmark Suite: Beyond empirical verification and validation," *IEEE Antennas Propag. Mag.*, vol. 67, no. 2, pp. 95–108, Apr. 2025, doi: 10.1109/MAP.2025.3537557.

[31] J. T. Kelley, B. A. MacKie-Mason, D. A. Chamulak, C. C. Courtney, and A. E. Yilmaz, "A review of the Austin RCS Benchmark Suite: Next generation benchmarks to enable CEM performance benchmarking," *IEEE Antennas Propag. Mag.*, vol. 67, no. 2, pp. 20–31, Apr. 2025, doi: 10.1109/MAP.2025.3537404.

[32] O. H. Ramachandran and B. Shanker, "Benchmarking–charge-conserving, unconditionally stable, electromagnetic finite-element particle-in-cell codes: Validating 3D nonlinear multiphysics simulations," *IEEE Antennas Propag. Mag.*, vol. 67, no. 2, pp. 76–86, Apr. 2025, doi: 10.1109/MAP.2025.3537958.

[33] H. Barnes *et al.*, "Benchmarking and reproducibility in computational and experimental characterization of electronic packages for signal/power integrity: Four benchmarks serve as standardized cases," *IEEE Antennas Propag. Mag.*, vol. 67, no. 2, pp. 51–63, Apr. 2025, doi: 10.1109/MAP.2025.3537412.

[34] W. Hackbusch, "A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices," *Computing*, vol. 62, no. 2, pp. 89–108, 1999.

[35] O. Babazadeh *et al.*, "Optimal preconditioners for hybrid direct–iterative $\mathcal{H}$-matrix solvers in boundary element methods," *IEEE J. Multiscale Multiphys. Comput. Techn.*, to be published, 2025, doi: 10.1109/JMMCT.2025.3547827.

[36] S. Wandzura, "FastTalk wandzura fam04," in *Workshop on Fast Methods in Engineering (FAM04)*, Univ. of Maryland, MD, USA, 2004. [Online]. Available: `https://cscamm.umd.edu/programs/fam04/FastTalk_wandzura_fam04.pdf`

[37] I. Jeffrey, J. Aronsson, M. Shafieipour, and V. Okhmatovski, "Error controllable solutions of large-scale problems in electromagnetics: MLFMA-accelerated locally corrected Nyström solutions of CFIE in 3D," *IEEE Antennas Propag. Mag.*, vol. 55, no. 3, pp. 294–308, Jun. 2013.

[38] A. F. Peterson, S. L. Ray, and R. Mittra, *Computational Methods for Electromagnetics*. Piscataway, NJ, USA: IEEE Press, 1998.

[39] L. Piegl and W. Tiller, *The NURBS Book*. New York, NY, USA: Springer, 1997.

[40] J. A. Stratton, *Electromagnetic Theory*. Piscataway, NJ, USA: IEEE Press, 2007.

[41] R. Kriemann, *HLIBpro: C Language Interface*. Leipzig, Germany: Max Planck Institute for Mathematics in the Sciences, 2008.

[42] W. Hackbusch, B. Khoromskij, and R. Kriemann, "Hierarchical matrices based on a weak admissibility criterion," *Computing*, vol. 73, no. 3, pp. 207–243, 2004.

[43] R. Kriemann, "LU factorization on many-core systems," *Comput. Visual Sci.*, vol. 16, pp. 105–117, 2013, doi: 10.1007/s00791-014-0226-7.

[44] R. Gholami, J. Mojolagbe, A. Menshov, F. S. Lori, and V. Okhmatovski, "$\mathcal{H}$-matrix arithmetic for fast direct and iterative MoM solution of surface-volume-surface EFIE for 3-D radiation problems," *Prog. Electromagn. Res. B*, vol. 82, pp. 189–210, 2018.

[45] R. Gholami, A. Menshov, and V. Okhmatovski, "$\mathcal{H}$-matrix accelerated solution of surface-volume-surface EFIE for fast analysis of scattering on composite dielectric objects," *IEEE J. Multiscale Multiphys. Comput. Techn.*, vol. 4, no. 1, pp. 152–162, May 2019.

[46] C. Ronchi, R. Iacono, and P. S. Paolucci, "The 'Cubed Sphere': A New Method for the Solution of Partial Differential Equations in Spherical Geometry," *Journal of Computational Physics*, vol. 124, no. 1, pp. 93–114, 1996, doi: 10.1006/jcph.1996.0047.

[47] SciNet Consortium, "Niagara Quickstart," University of Toronto, Toronto, ON, Canada, 2025. [Online]. Available: `https://docs.scinet.utoronto.ca/`.

[48] ASELSAN A.Ş., "About us," Ankara, Turkey, 2025. [Online]. Available: `https://www.aselsan.com/en/about-us`.

[49] The Univ. of Texas at Austin, Computational Electromagnetics Group, "AustinCEM Benchmarks," Austin, TX, USA, 2025. [Online]. Available: `https://github.com/UTAustinCEMGroup/AustinCEMBenchmarks`.

[50] J. T. Kelley, D. A. Chamulak, C. C. Courtney, and A. E. Yilmaz, "Rye Canyon radar cross-section measurements of benchmark almond targets," *IEEE Antennas Propag. Mag.*, vol. 62, no. 1, pp. 96–106, Feb. 2020, doi: 10.1109/MAP.2019.2955702.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

Computational Electromagnetics (CEM) plays a central role in modern science and engineering, providing the foundation for analyzing and designing complex electromagnetic systems ranging from antennas and radar signatures to microwave circuits and scattering from real-world structures. As the electrical size and geometric complexity of practical problems continue to grow, achieving both accuracy and computational efficiency has become increasingly challenging. This thesis has addressed these challenges by developing a unified, high-order, and hierarchically accelerated framework for solving large-scale integral equation formulations of electromagnetic scattering. In particular, we focused on PEC targets in homogeneous media and used high-order point-based CBIE/LCN discretizations of EFIE/MFIE/CFIE operators, tightly coupled with $\mathcal{H}$-matrix compression and parallelism schemes to obtain error-controlled RCS predictions at practical memory and time costs.

The work presented here delivers a suite of methodological and algorithmic innovations

that significantly advance high-order, integral equation-based CEM:

1. **Hybrid direct–iterative preconditioning strategy:** We combined hierarchical $\mathcal{H}$-**LU** factorization with tailored preconditioned solvers (e.g., GMRES/LIM) to balance robustness and speed for electrically large, high-order systems. This approach delivers fast initial reduction of residuals, reliable late-stage convergence, and predictable memory usage. In practice, we observed that modest $\mathcal{H}$-**LU** accuracies, when paired with carefully chosen restart and stopping criteria, yield end-to-end speedups without sacrificing solution fidelity in the far field.

2. **Heterogeneous $\mathcal{H}$-matrix tolerances:** By applying distinct $\mathcal{ACA}$ thresholds for leaf and non-leaf blocks during $\mathcal{H}$-**LU**, we preserved solution accuracy while substantially reducing computational cost. The rationale is that leaf blocks dominate storage, whereas certain off-diagonal blocks chiefly influence preconditioner quality rather than final accuracy. Calibrating τ_{ACA} across these roles reduces rank growth, limits fill-in during factorization, and better aligns compression error with the discretization error inherent to the *hp*-refined LCN scheme.

3. **Comprehensive RCS benchmarking framework:** We established a suite of RCS benchmarks on both canonical and complex PEC geometries (e.g., spheres, almond-type shapes, and aircraft-/missile-like CAD surfaces). The framework standardizes incident/observation configurations, polarization control, frequency sweeps, and error reporting (monostatic/bistatic cuts and angular grids). Within this unified protocol, the proposed solvers consistently matched reference solutions on canonical targets and maintained stable accuracy as geometry detail and electrical size increased, while offering favorable memory footprints and reduced runtimes relative to baseline dense

approaches.

4. *hp*-**refinement in point-based integral equation discretizations:** Our Locally Corrected Nyström scheme on quadrilateral Bézier-patch meshes pairs high-order coarse patches in smooth regions with low-order fine patches in edge/transition zones. Locally corrected weights reproduce vector polynomials up to the target degree, ensuring that the discrete operator matches the continuous one on that space. This yields high-order accuracy on smooth portions while controlling conditioning near geometric singularities. The *hp*-refinement strategy minimizes DOFs for a given error tolerance, lowers near-set size, and reduces the number of expensive near-corrections without compromising edge-field resolution.

5. **Scalable parallel software suite:** All core kernels including matrix assembly, $\mathcal{H}$-matrix compression ($\mathcal{ACA}$), triangular solves in $\mathcal{H}$-**LU**, and Krylov iterations were parallelized for shared- and distributed-memory systems. Task partitioning respects patch boundaries and block-admissibility to maximize concurrency while limiting communication. The resulting implementation shows robust strong/weak scaling behavior across a range of meshes and orders, enabling practical turnaround times for large RCS campaigns.

6. **Open, extensible CEM toolkit:** The methods have been encapsulated in a modular software library with clean separation of concerns: geometry import and patching, *hp*-aware LCN assembly, $\mathcal{H}$-arithmetic backends, and solver orchestration. Configuration is declarative (mesh/order/tolerances/incidence), facilitating reproducible studies and straightforward integration into external workflows. The design accommodates additional operators (e.g., dielectric formulations), alternative low-rank schemes, and

new hardware backends with minimal disruption.

Together, these developments contribute to the broader goal of high-fidelity electromagnetic modeling by bridging the gap between numerical precision and computational scalability. The proposed $\mathcal{H}$-matrix accelerated, *hp*-refinement LCN framework provides a robust foundation for next-generation solvers capable of addressing geometrically complex and electrically large scattering problems with practical memory and runtime demands. This work not only advances the numerical state of the art in CEM but also provides a practical path toward scalable simulation tools for real-world engineering applications. While the emphasis has been on PEC scatterers in homogeneous backgrounds, the architecture, discretization, and compression choices are all compatible with broader operator sets, suggesting a clear path to multi-physics and multi-material extensions.

7.2 Future Work

Building on the foundation of the solver developed in this work, several directions merit further exploration. Each item below is actionable and aims to extend capability while preserving the favorable accuracy–cost tradeoffs established in this thesis:

- **Adaptive error-driven *hp*-refinement:** Incorporate reliable a posteriori indicators that reflect both discretization error (e.g., residual-based or goal-oriented estimators for RCS) and operator-approximation error from $\mathcal{H}$-compression. A practical controller would (i) refine h near geometric singularities and in regions with large kh or strong field gradients, (ii) raise p on smooth, curvature-limited patches, and (iii) adjust local near-sets to maintain moment-reproduction accuracy of LCN. The objective is a target-

accuracy workflow that terminates automatically when the estimated error in the RCS metric falls below a user-specified tolerance.

- **Extension from PEC Surface Integral Equations to penetrable media via Volume Integral Equations:** Extend the present surface integral equation framework to penetrable and inhomogeneous media by developing *volume integral equation (VIE)* and *volume–surface integral equation (VSIE)* capabilities. This requires (i) introducing volumetric unknowns and suitable high–order volume discretizations, (ii) redesigning near-singular and singular quadrature/local corrections for volume kernels and coupled operators, and (iii) integrating surface–volume coupling while retaining hierarchical compression and scalable solvers. A staged roadmap is: validate a homogeneous dielectric VIE on canonical scatterers; extend to VSIE for coated/composite targets; and finally combine with the adaptive surface *hp* strategy developed in this thesis.

- **Accelerator integration:** Port assembly, $\mathcal{ACA}$ compression, and $\mathcal{H}$-**LU** triangular solves to GPU/accelerators (e.g. CUDA). Emphasis should be placed on (i) tiling and batching strategies for small dense kernels in $\mathcal{ACA}$ updates, (ii) stream-parallel evaluation of admissible blocks, and (iii) mixed-precision variants where compression is performed in lower precision with iterative refinement for the outer solve. The expected outcome is multi× speedup at constant accuracy for very large DOF counts.

- **Advanced preconditioners and hybrid solvers:** Explore rank-adaptive preconditioners that vary $\tau_{\mathcal{ACA}}$ and block sizes based on local spectral diagnostics, as well as hybrid $\mathcal{H}/\mathcal{H}^2$ variants for far-field blocks. For CFIE systems on intricate CAD, assess block-factorizations that isolate EFIE-dominated subspaces while preserving the MFIE

jump structure. The goal is to reduce iteration counts further and make convergence less sensitive to geometry and frequency.

- **EFIE integration and first-kind robustness studies:** Extend the high–order, $\mathcal{H}$–accelerated pipeline to EFIE-dominated formulations to more strongly stress solver robustness in the first-kind regime. This includes implementing EFIE within the Chebyshev-based boundary integral framework, and studying preconditioning strategies tailored to EFIE (e.g., Calderón-type ideas, operator-based preconditioning, or $\mathcal{H}$-factor based hybrids) under the same *hp* refinement settings used for MFIE/CFIE.

- **Larger-scale demonstrations under expanded HPC resources:** Demonstrate the framework on substantially larger problem sizes by leveraging larger memory footprints, higher core counts, and/or multi-node deployments specifically provisioned for $\mathcal{H}$-matrix assembly/compression and hybrid solves. The study should include (i) end-to-end scaling trends with N, (ii) memory/compression ratios versus tolerance, and (iii) time-to-solution breakdowns (assembly, compression, factorization, iterations), to establish practical limits and pathways toward extreme-scale simulations.

- **High-fidelity antenna simulation and installed performance:** Extend the framework from passive scattering to active radiation problems by implementing localized excitation models, such as delta-gap feeds, waveguide ports, or impressed current distributions. This enables the calculation of antenna-specific metrics, including S-parameters, input impedance, and far-field radiation patterns for antennas integrated onto electrically large platforms. The goal is to demonstrate that the $\mathcal{H}$-matrix solver remains robust under the high dynamic range of field values typical of antenna-platform interactions.

- **Multi-scale electromagnetic analysis:** Investigate the performance of the *hp*-adaptive $\mathcal{H}$-matrix solver on problems characterized by extreme geometric scale disparates, such as sub-wavelength structural details (slots, pins, or fine-scale textures) residing on large-scale smooth surfaces. Future work should focus on optimizing the hierarchical clustering logic to maintain tree balance in the presence of severe mesh density variations. This would showcase the framework's unique capability to bridge the gap between microscopic geometric fidelity and macroscopic electromagnetic scattering in a single, unified simulation.

- **Machine learning–driven adaptivity and compression:** Investigate data-driven models that predict (a) local *hp* choices from geometric features (curvature, edge proximity, kh) and (b) admissible block ranks from separation metrics, before $\mathcal{ACA}$ is invoked. These predictors can serve as initial guesses to cut setup time; the final ranks remain validated by physics-based residual checks to maintain determinism and robustness.

- **Integrated GUI and CAD workflow:** Develop a user-oriented interface that links CAD import (e.g., NURBS/Bézier), region tagging, and *hp* allocation directly to solver runs. Visual overlays for near-sets, polynomial order, local residual estimates, and RCS convergence would shorten the iterate–analyze loop. Such tooling would help transition the framework from research to production use in design studies and certification workflows.

Looking ahead, continued advances in high-performance computing, hierarchical linear algebra, and automation offer clear opportunities to extend the present PEC–SIE framework toward penetrable-media formulations, state-of-the-art benchmarking, and larger-scale

demonstrations. Collectively, these developments can bridge rigorous electromagnetic modelling with practical engineering design workflows, enabling faster and more reliable simulation for next-generation platforms.